A ROBUST PHONETIC DIGIT RECOGNITION SYSTEM FOR DIGITS SPOKEN IN AMERICAN ENGLISH AND ARABIC

By

AHMED MAHMOUD MILYANI
II

Certificate
Cambridgeshire College of Arts & Technology
Cambridge, England
1969

Master of Science
Oklahoma State University
Stillwater, Oklahoma
1976

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
DOCTOR OF PHILOSOPHY
July, 1981

A ROBUST PHONETIC DIGIT RECOGNITION SYSTEM FOR DIGITS SPOKEN IN AMERICAN ENGLISH AND ARABIC

Thesis Approved:

Thesis Adviser

Dean of the Graduate College

ACKNOWLEDGMENTS

Praise be to Allah and Peace be upon His messenger, Mohammed. I thank Allah, The Almighty, for His uncountable help and guidance.

I would like foremost to express my sincere gratitude to Dr. Rao Yarlagadda, my thesis adviser and chairman of my doctoral committee, for his inspiration and continued guidance during the entire phase of my graduate study. His dedication and encouragement has contributed significantly to the completion of this dissertation.

My special thanks and appreciation are due to Dr. R. Mullholand, Dr. R. Cummins, and Dr. L. Broemeling, my committee members, for their interest, helpful comments and suggestions throughout this research.

To Dr. R. Schaefer, I would like to express my appreciation for his effort in my behalf in gathering speech data at the OSU Speech Clinic Laboratory. I would like to recognize Dr. L. Ebbesen and Dr. J. Perrault for their aid with the OSU Mechanical Engineering Interdata Computer.

My sincere appreciation are due to the Saudi Arabian Government for their support during the entire phase of my study.

I am grateful to Mrs. Dolores Behrens for her excellent typing of this thesis.

My sincere appreciation also extends to my parents and all the members of my family and my relatives who had their part of sufferings and sacrifices during the entire phase of my study. The great memory of my father who passed to the other world just after the completion of this thesis shall not be forgotten at this moment in life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Recent advances in computer technology allowed for extensive research and development in the area of speech signal processing [1][10]. One area that has been quite popular is speech recognition [2]. Most of the algorithms used in speech recognition deal with only a limited vocabulary of about 250 words. These algorithms require extensive computer storage and time consuming computation, especially when recognition must be speaker independent. A specialized area of speech recognition is digit recognition, which is the main topic of this thesis. It is clear that the recognition of digits requires the recognition of only 10 words for each language. The limited vocabulary of the spoken digits and the limited number of phonemes it uses, gives hope that an efficient algorithm may be found enabling man to communicate easily with machine. However, the great majority of situations with which we identify the concept of recognition will be found to involve ultimate human perception [11][16]. In fact, the element of human perception [19] is difficult to dissassociate, in our thinking, from the concept of recognition. The problem is essentially one of understanding the human anatomy of speech production [19] , and acoustics [34][39] in order to find a simple yet more reliable, and adaptive model which aid in the development of a robust phonetic algorithm for recognizing an uttered digit of any presently known language.

This thesis presents several new approaches to digit recognition schemes and emphasizes some newly defined and normalized parameters based on area functions. These parameters are used in automatic phoneme segmentation and feature detection of digits spoken in American English and Arabic. Algorithms for segmenting speech sounds into vowel, vowel-like and non-vowel segments are discussed. In addition, parameters used for identifying vowels and detecting nasal segments, turbulence noise segments, dip-classification, etc. are described. Furthermore, an algorithm, based on the correlation coefficients of the RMS energy together with the back-to-total cavity area ratio, is introduced.

Review of the Literature on Speech Recognition

This section deals with some of the earlier work on speech recognition, and, in particular, digit recognition. With the widespread application and recent growth in the use of digital computers, there has been an increasing need for man to be able to communicate with machines in a manner more naturally suited to humans. The realization of this need has motivated a great deal of research in automatic recognition of speech by computer [1][10]. Although only a moderate degree of success has been obtained in solving the problems associated with machine recognition of continuous speech consisting of a series of spoken digits, a greater degree of success has been obtained in recognition of isolated digits.

Research in the mechanical recognition of spoken connected digits not only furnished the foundation for significant advances in pattern recognition and artificial intelligence, but also gave a better

understanding and deeper insight of what a speech recognition system consists of till now, what will be the new system in the future, what makes speech recognition a difficult problem, and what aspect and prospects of this problem remain unsolved. The aim of this research is to find a better, more economical and viable automatic speech and digit recognition system.

Automatic recognition of speech is that process by which a machine attempts to identify correctly certain speech sounds produced by either a human vocal mechanism or some process simulating the output from the human vocal mechanism. Ideally, a speech recognizer should not be constrained by either limited vocabulary size, vocal differences between the individual speakers providing the stimulus to the machine, or any special conditions imposed upon the manner in which speech may be put into the machine [2][3]. Some special input conditions would include, for example, isolation of words by delibrate and artificial intervals of silence or other than normal signal-to-noise conditions. With suitable constraints on vocabulary size, number of talkers and input conditions, several recognition devices have been built and tested with rather encouraging results [2][5][31].

One of the common constraints on an algorithm is the vocabulary size [5], which requires a forced choice from a group of recognizable words. Word recognition is suitable for certain applications (for example, voice dialing of a telephone or programming a computer where only a limited number of commands is required). However, it also has certain limitations. One inherent limitation in word recognition is the trade-off which exists between the size of the acceptable vocabulary and the time required for recognition. This trade-off is a

result of the fact that word recognition generally involves matching some pattern of a spoken word with a stored library of patterns of recognizable words [3]. As the library of acceptable words is increased the storage limit on the machine must likewise increase. The effective storage limit of the machine may be increased by the use of external storage devices, such as disks, tapes, etc. However, the time required to compare a new word to each of the library words may become so large as to make the method impractical [2].

It is common knowledge that the same word spoken by the same person will have some pattern differences. These pattern differences may make it difficult for recognition. Obviously, speaker independent speech recognition is even more difficult since the design of any successful recognition algorithm demands a quantitative knowledge of the words to be recognized in terms of certain features that may be useful. Therefore, a large amount of data is required from different speakers.

Digit Recognition

In recent years, several schemes for digit recognition have been described [1][3][4][12-15]. Most of these have used decision trees based upon empirically derived rules or parameter measurements such as zero crossings, log energy, LPC coefficients, and LPC error. Also, the use of normalized error and pole frequency, followed by phoneme classification of the beginning and ending of the digit has not fulfilled the requirements for an efficient and accurate digit recognition system.

Digit recognition systems or algorithms can be developed using techniques adopted for general speech recognition, such as word and pattern matching with minor modifications. However it is more practical, efficient and accurate, to adopt a new digit algorithm scheme especially at the final recognition stage. One of the advantages is that digit algorithm requires a much smaller vocabulary than one designed for arbitrary word or phoneme recognition. This may lead to the development of a speaker independent digit recognizer. A speaker-adaptive system that can use comparatively simple pattern-matching algorithms to recognize the input digit is simpler and more accurate than any other known system.

For small vocabulary systems, such as digit recognizers with a large number of potential users, it is not feasible to store training data for every possible user. Also, most systems cannot train themselves on new speakers very rapidly. Thus, the turn-around time of new users is often a major factor limiting the use of speaker dependent systems. Consequently it is worthwhile to build up a speaker independent recognition system. In addition, the variation with time of a speaker's voice characteristics may necessitate frequent updating of his reference patterns. Finally, the design of a speaker dependent digit-recognition algorithm is dependent on the uniqueness of each talker's characteristics; whereas, the design of a speaker independent digit-recognition algorithm requires identification of a set of characteristics based on the uniqueness of the phonemic word features obtained from a large number of speakers of different dialect, accent and nationality. The scheme proposed here is dependent upon some

newly defined parameters for quantifying such common characteristics. Also the uniqueness of these patterns and its special features will contribute to an understanding of the acoustic attributes of speech that reliably distinguish the various sounds.

Segmentation

Speech sounds are usually represented by a finite number of distinguishable mutually exclusive, linguistic elements called phonemes. This raises the possibility of segmentation of the acoustical continuum of speech or digit sounds into discrete parts which can then be associated with specific phonemes.

A discrete representation of the time domain requires a segmentation of the continuous speech or connected-digit waveform into some sort of units or "segments". Using a rectangular window, which is described in Chapter III, the spoken digit or digits data can be segmented into n frames. Customarily, each frame consists of 256 data points, which has a duration of 32 ms [1][7] corresponding to an 8 KHz sampling rate. However, it has been found during this research that a frame of 64 data points gives a more reliable and accurate value for our purpose. In addition, using 128 data points per frame of duration 16 ms maintain almost the same accuracy and is more efficient, because it reduces the computational time.

The most often used definition of the term "segmentation" is phoneme segmentation, in which a given continuous speech sentence is phoneme segmented. The number of frames per segment varies according to the duration of the uttered phoneme.

Connected digit, two or more digits uttered successively at normal speech rate, are characterized by a near-continuous (transition) motion of the vocal apparatus from sound to sound. This motion involves continuous changes in the vocal-tract configuration and its modes of excitation. The set of such transitions is wider and more varied in normal vocabulary than in the limited vocabulary of the ten integers.

Humans are able to perform the function of segmentation in a natural way, although, depending on their culture, they may segment phoneme differently. One individual might segment a certain word into four distinct phonemes, whereas another would insist that there are only three phonemes. This is because some phonemes are indistinguishable variants of the same phoneme depending on their position in a given word or digit. Thus, it is unfair to expect a machine to segment a continuous speech or connected digit waveform into discrete segments so that for every segment there is one and only one phoneme. However, for a general phoneme-digit recognition system, it is desirable to have a procedure which first segments the connected digits into isolated digits, using end point detection used by Rabiner [9][12].

Speech Input Rate and Pauses

All automatic speech or digit recognition systems can be considered as belonging to one or two categories: systems designed for the analysis of continuous (connected) speech and those for analysis of isolated (discrete) speech. The two systems have many features in common, which tend to obscure some of the differences. However, isolated speech systems are defined as those systems that require a short pause before and after utterances that are to be recognized as entities [11][23-27].

The minimum duration of a pause that separates independent utterances is on the order of 100 ms [10]. Anything shorter than 100 ms can be confused with the closure of stop constants in the midst of continuous speech that can produce stop gaps approaching 100 ms in duration. In actuality, a stop gap can exceed a 100 ms duration [2][3][10]. For example, the word "seven" can be spoken with a relatively long silence interval "se-". For a trained speaker, however, a 100 ms minimum separation between digits is a reasonable compromise value.

The speaking rate that can be achieved with isolated speech recognition systems is naturally much less than for connected speech. Speaking rates over 300 words per minute can be achieved quite easily for short intervals of connected speech. The upper bound for an isolated word speaking rate has been measured informally for trained speakers reading digits in random order [6][7][47]. A rate of 120-125 digits/min. was achieved with the best speakers. Each of the digits was classified correctly by a machine capable of recognizing isolated utterances. Measuring these rates is not possible without some objective measure that the words are not connected. The human ear is a fairly good judge of whether a brief pause actually exists between rapidly spoken words [5][19][37]. The speech recognition system may have difficulty locating brief pauses.

Average speaking rates between 30 and 70 isolated words (or phrases) have been achieved in factory environments by individuals using voice input systems during their entire 8-hour working day [6]. These average rates will include peak rates close to 120 words/min. and lower than average rates during light workload requirements.

Another important factor concerns the duration of the utterances to be classified. A limited vocabulary system could be hypothesized in which each of the utterances in the limited vocabulary was of considerable duration. However, the previously used limited vocabulary system accepts isolated words as short phrases 2-4 sec. in duration [6].

Human Factors

Clearly, many factors must be considered in order to choose a suitable, well-defined set of parameters for digit recognition. These parameters must allow for pre-emphasis before they are computed. Consequently, the acoustic and segmental aspects of speech have to be considered too, and so doing, the importance of "transitional patterns" of speech might be discovered. Furthermore, despite individual differences in voice quality or speech acoustics and sex differences on speech frequency, the same speech sounds can be recognized. Moreover, humans have the ability to perceive speech in the presence or absence of noise and among competing messages which result in loss of information [11][37]. Given the same information, machines engineered to recognize digits have limited capabilities as compared to human listeners. Surely other factors must be considered, such as accent, educational background, dialects, pronunciation, colds, any upper respiratory and hearing abnormalities.

Last but not least, contextual clues permit the listener to anticipate what might be or might have been said. When combined with prior knowledge about situations or various conversational subjects, one may be able to follow conversations with less than complete perception of all the acoustic information transmitted. Whereas in a digit

recognition system, the perceptual ability of humans depend not only on the quality of the system to transmit the uttered digits but on the position of the digit with a set of uttered connected digits.

Obviously, knowledge of the spoken languages used is fundamental to speech perception more than digit perception. Speech code must be comprehended or its sound patterns, in order to understand the digit order and expand the possibility of compressed digit transmission and recognition algorithm.

Linear Prediction Analysis

The linear prediction method has become important in the area of speech analysis and synthesis because it gives considerable insight into modeling speech production processes [1]. Also it has been shown previously that the linear prediction digital filter represents a non-uniform acoustic tube model of the vocal tract [11]. The computational aspects of linear prediction analysis and spectral modeling will be discussed in Chapter III. This research is intended to use newly defined parameters based on linear prediction analysis and RMS peak ratio. The new parameters are related to the ratio of front, central and back cavity volume to total cavity volume. The new parameters will be used to develop a pattern recognition scheme for each digit.

Previous Systems

One of the first works on the automatic recognition of spoken digits is based on a circuit analyser and quantizer, followed by a pattern matching network [12]. The circuitry is designed such that the frequency band is divided into two bands, upper and lower, with

the separating frequency of 900 Hz. Note that the first two formants are generally separated by this. The frequency of the maximum syllable rate energy within each band is determined using zero crossing. The plots of formants f_2 versus f_1 for the digits 0, 1, to 9 are stored as reference patterns and the pattern of the input digit is compared statistically with the data in store. Unfortunately the statistical analysis is time consuming. Furthermore, the variability of formants creates problems in regard to the pattern matching. In this early system, accuracy range of 97 to 99 percent may be obtained if the same speaker repeats uttering a random series of digits, with 350 ms pauses between digits. The accuracy may fall to 50 percent when different speakers utter a series of random digits.

In the second method, computer simulations are used to compute the power spectra at 10 ms intervals, and segment the words into vowels and consonants [13]. Vowels are classified into one of 11 categories by a multivariate statistical decision, while an emperically derived decision tree is used to classify consonants into one of three categories. A simulated filter bank is used to transform the waveform of the spoken digits into their power spectrum. Each of the 40 filter banks have a half-power bandwidth of 200 Hz. The convolution interval of the time function and the filter response is evaluated to find the power output filter. The ratio of the power output per filter to the total power is computed. A threshold level is adjusted for each band. Then a recognition procedure based on the power band ratio for vowel-consonant, voiced-unvoiced fricative, etc. is developed. This system has disadvantages that the speaker has to be trained, must speak

clearly and has to pause between digits. Important features are lost due to the use of multiple band-pass filters. Also, large variations in pronunciation cause difficulties in preliminary and final recognition.

The third digit recognition system is based on simple segmentation rules according to articulatory features [14]. The digits are segmented according to manner of articulation. Formants f_1 and f_2 are automatically located and evaluated for vowel-like segments. Each spoken digit is represented by a sequence of segments and the values of the first and second formants at the 1st, 2nd, 3rd, 4th and 5th section of the segment is computed. Also, the maximum value of the first formant within that segment is found. An algorithm based on reference patterns is used to match the input digit against the stored established patterns. Error ratio varies from 1.2 percent to 5 percent if the system is trained, and 20 percent and more if the system is untrained.

The fourth recognition system shown in Figure 1 was implemented by Rabiner and Sambur [1][2] for an isolated speaker independent digit recognition system. The system uses end-point detection, four parameter measurements, segmentation of the utterance into intervals, a preliminary decision tree, and a final class decision digit recognition scheme. End-point detection is based on the algorithm developed by the same authors [1][9][12] and uses self normalized measures of the energy and zero-crossing rate (ZCR) of the speech waveform. The four parameters that are of interest are the rate of zero crossings, the energy, and two pole frequencies obtained from LPC analysis and the LPC residual error. These measurements are made for each frame,

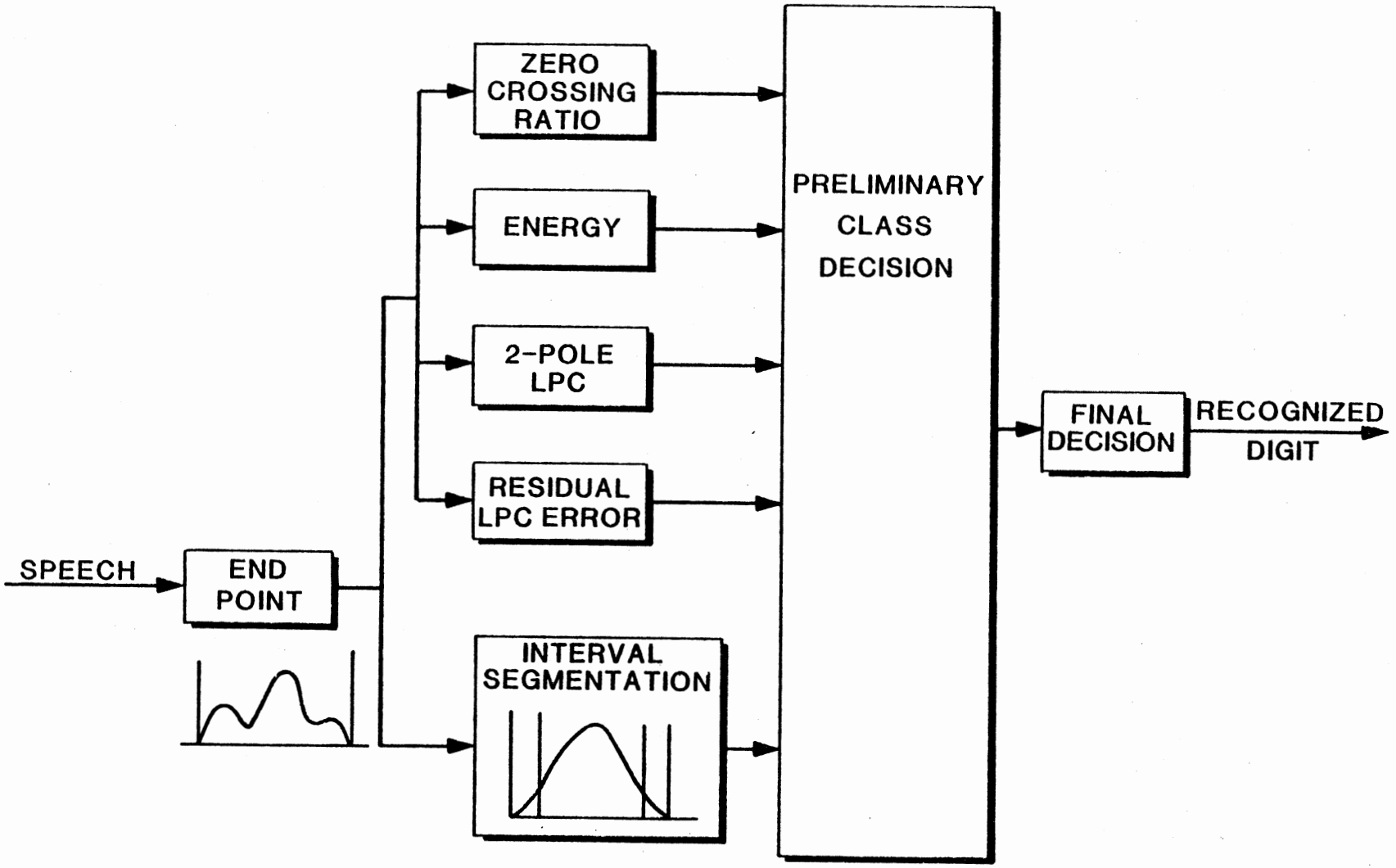


Figure 1. The Over-All Isolated Digit Recognition System Block Diagram
After Sambur and Rabiner [4]

where the frame has 256 data points with a sampling rate of 8 KHz. One of the problems with this method is the location of the word boundaries and the dependence of the zero-crossing rate.

The fifth recognition scheme is a modification of the last method [1][4]. A digit segmentation algorithm for continuous digits is added, then a recognition procedure is developed. The accuracy of recognition depends on the preciseness of locating digit boundaries and phoneme segmentation. However, this scheme is less accurate than the one developed previously for isolated digits.

There are other digit recognition systems considered in terms of English, French and other languages. However, the systems given above are the most prominent at the present time.

Thesis Outline

Chapter II deals with anatomy of speech production, and discusses vocal pitch and loudness, articulations, classification of sounds, vowels and consonants. Formant frequencies and its importance in spectrum analysis is discussed briefly. Effect of intensity of sound and noise, frequency and segmental analysis and coarticulation are explained according to their importance to phonemic digit segmentation and recognition.

Chapter III deals with linear prediction analysis (LPA) and its application to the digit recognition scheme. Newly defined parameters are introduced based on area functions derived from LPA. Identification of boundary locations of connected digits using dip-classification is explained, along with phonemic segmentation based on acoustic-phonetic analysis system.

Chapter IV discusses digit recognition scheme for English. The importance of primary segmentation, primary recognition, and final recognition is emphasized. Acoustic-phonetic segmentation and digit recognition flow chart is explained.

Chapter V deals with the Arabic digit phonemes and its relevence to proposed scheme. A comparison table of Arabic and English phonemes is introduced. The modified Arabic digit recognition procedure based on the decision tree is verified. The final digit recognition algorithm, recognition results, accuracy and the correlation matrix are given.

Finally, Chapter VI includes conclusions, suggestions for further research and possibly avenues into speaker verification.

CHAPTER II

SPEECH MECHANISM

Introduction

Acoustical speech waveform results in an acoustic pressure wave which originates from voluntary physiological movements of the human speech mechanism structures as shown in Figure 2. Speech is usually characterized as language that is spoken and heard, and the term is referred to the sounds made by the human vocal apparatus. The generation of sounds of any kind is dependent on the movements in this apparatus. This chapter describes the anatomy and physiology of speech production, its key components as related to the phonetic English and Arabic digits.

The speech waveform is generated due to the variation of pressure above and below the vocal folds. Air is expelled from the lungs into the trachea and then between the vocal folds. The position of the vocal folds across the breath stream allows them to act as a valve to control air flow. Air from the lungs builds up pressure below the vocal folds during glottis closure, i.e. during exhalation or expiration. The vocal folds will be forced open as soon as a sufficient pressure level is reached. The subglottal air pressure will drop slightly, as soon as a puff of air is passed through the glottis,

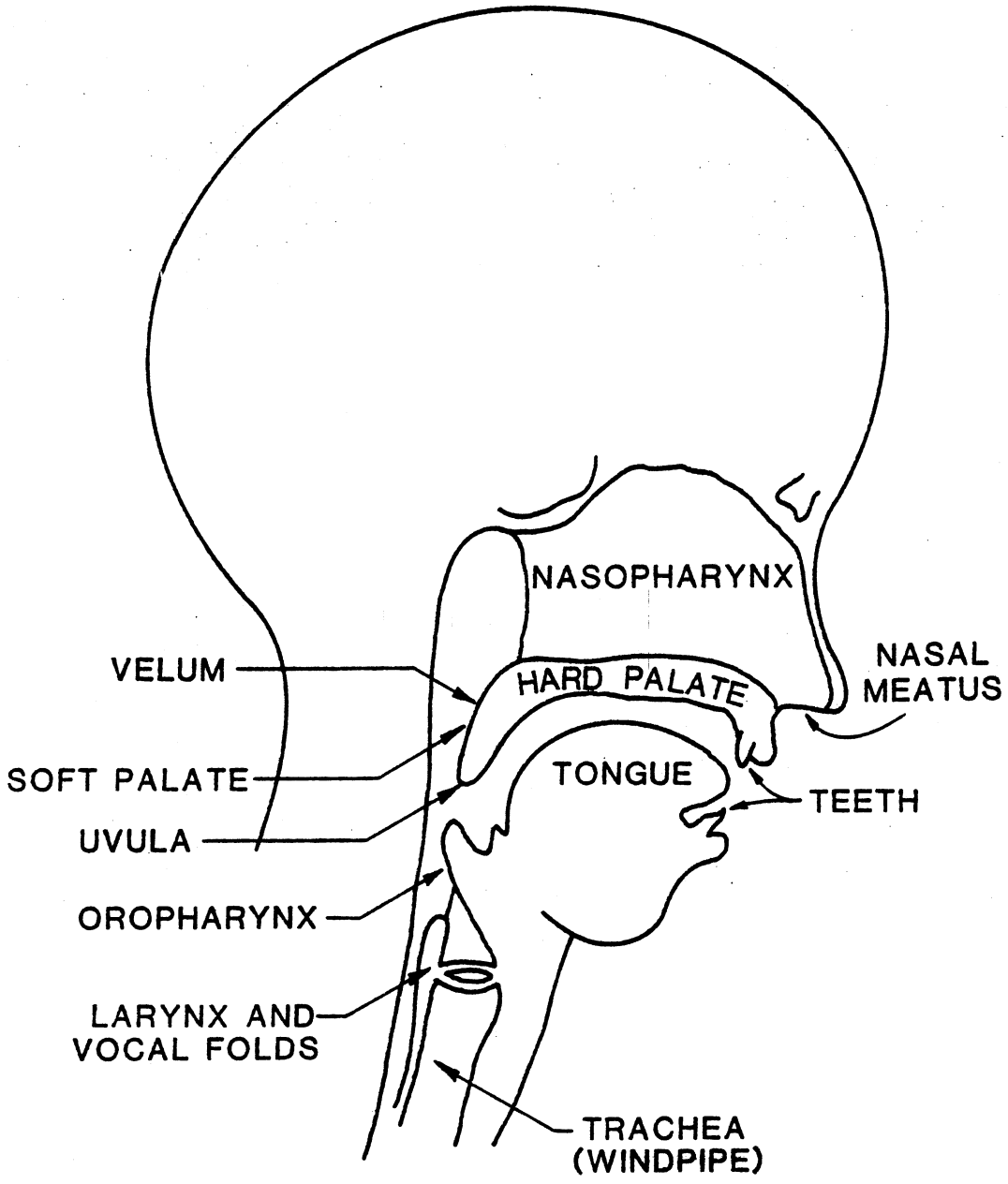


Figure 2. X-Ray Tracing of the Speech Mechanism of a Normal Speaker, Showing the Palate at Rest

then, the vocal folds return to their closed position due to their tension and elasticity. The passage of air through a partially closed glottis causes negative pressure, or suction, which contributes to the complete closure of the vocal folds. This phenomenon is known as the Bernoulli effect [18][19]. The subglottal air pressure will rebuild until it has sufficient force to overcome the forces due to the vocal folds tension and elasticity, and the cycle is repeated. During the generation of voiced sounds the air flowing upward from the lungs causes the vocal folds to open and close at a rate dependent upon the air pressure in the trachea and the physiological adjustment of the vocal folds. This adjustment includes changes in the length, thickness, mucus, and elasticity of the vocal folds. The greater the tension, the higher the perceived pitch of the voice.

The opening between the vocal folds is defined as the glottis. The subglottal air pressure and the time variations in glottal area determine the volume velocity of glottal air flow expelled into the vocal tract. When the movement of the glottis for a complete cycle is repeated about 125 times per second, a tone is generated that has the fundamental frequency of the average adult male voice. The rate at which the glottis opens and closes can be approximately measured acoustically as the inverse of the time interval between observed pitch periods of the acoustic wave. The acoustic energy input to the vocal tract can be determined from the glottal volume velocity wave.

Voice Production

When the vocal cords are relaxed, the air flow is unrestricted through the glottis. When the vocal cords are tensed, their spacing is restricted and the flow of air causes the vocal cords to vibrate in such a way as to modulate the flow of air from the lungs. When the air flow is modulated in this way, the pressure variation of the flow of air into the vocal tract is quasi-periodic and the sound so produced is defined as a voiced sound.

After passing through the larynx, the acoustic pressure and velocity variations of the air are modified by the vocal tract and nasal cavity. The vocal tract is an acoustical tube of non-uniform cross-section, which has its beginning at the vocal cords and ends at the lips [18-20]. The nasal cavity has its beginning at the velum and termination at the nasal meatus. Within the vocal tract and nasal cavity, the pressure and velocity variations of the air are modified by changing the position of the lips, teeth, jaw, tongue, velum and others. These organs are usually referred as the articulators shown in Figure 3. In the production of speech, the articulators are often placed in such a way as to produce a constriction within the vocal tract. This constriction may be made to occur anywhere from the vocal cords (for /h/, as in /ωâhid/ for one in Arabic) to the lips (for /f/, as in five) [19][20]. If the constriction is sufficiently narrow, turbulence results, and the vocal tract is said to be frictionally excited and the sound so produced is defined as a fricative. Friction may be produced at a constriction within the vocal tract with or without

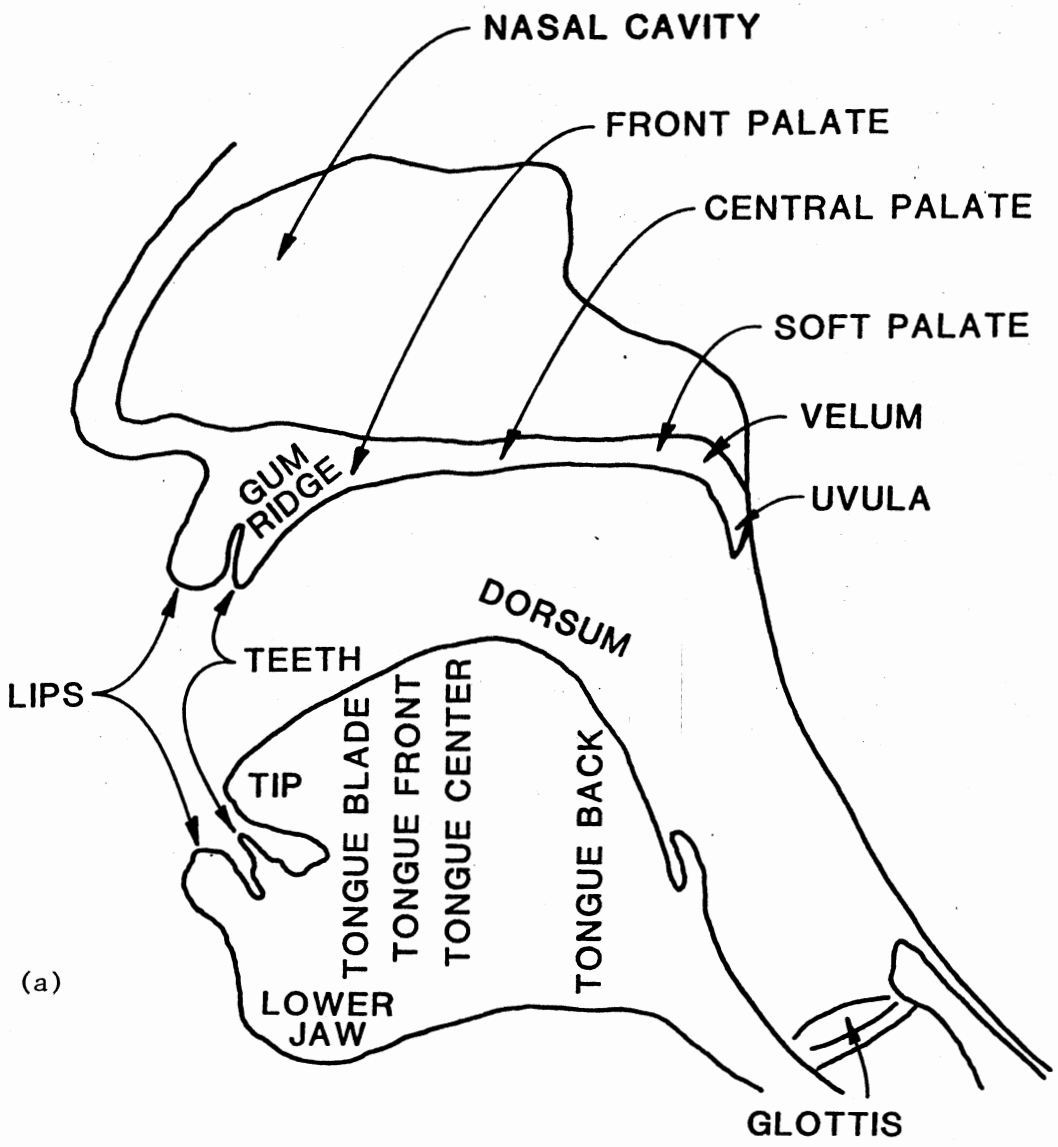


Figure 3. A Schematic View of the Articulators and Places of Articulation Showing the Partioning Assumed for Classifying the Place and Degree of Constriction of the Vocal Tract During Production of Vowel Sounds

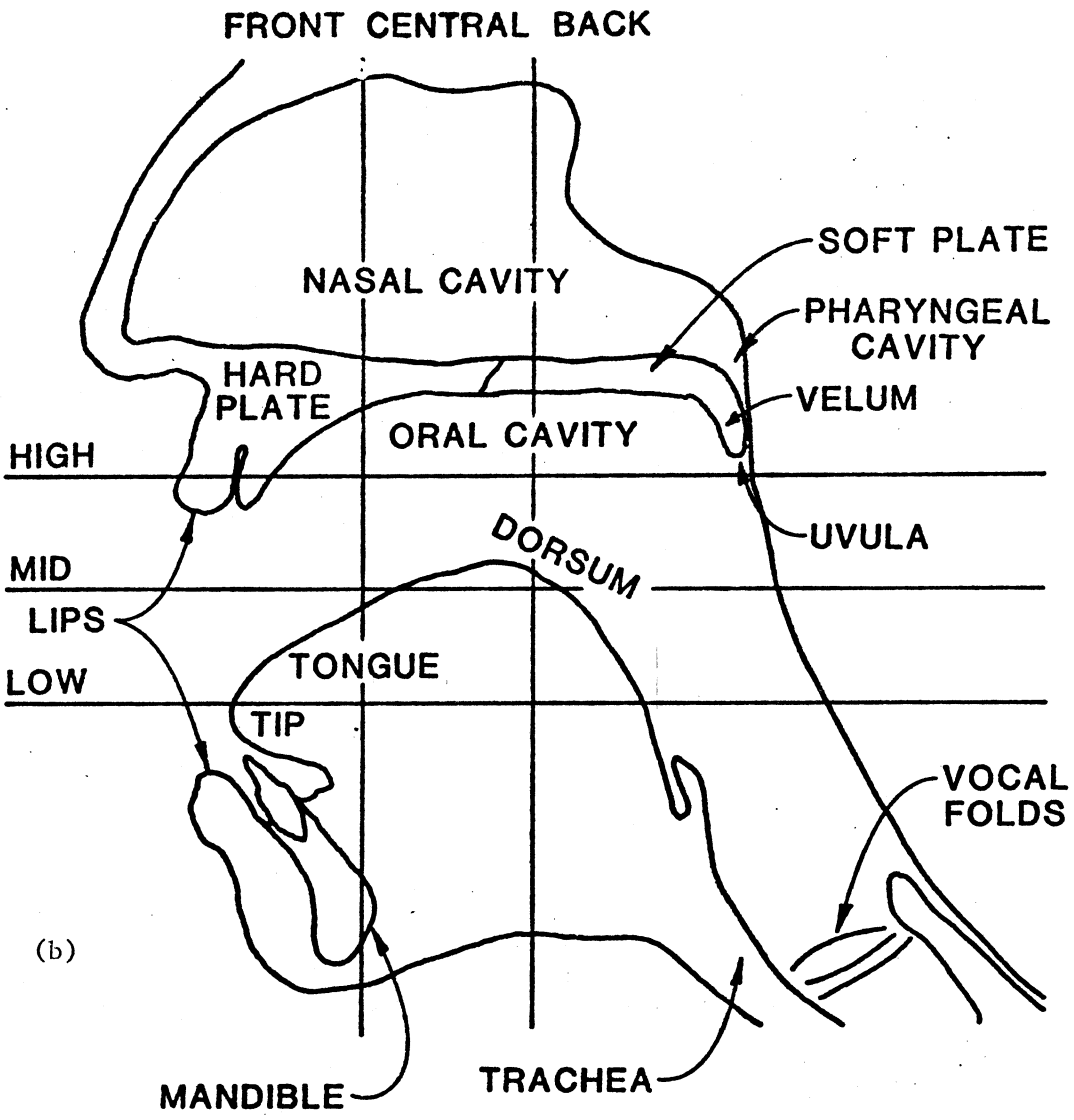
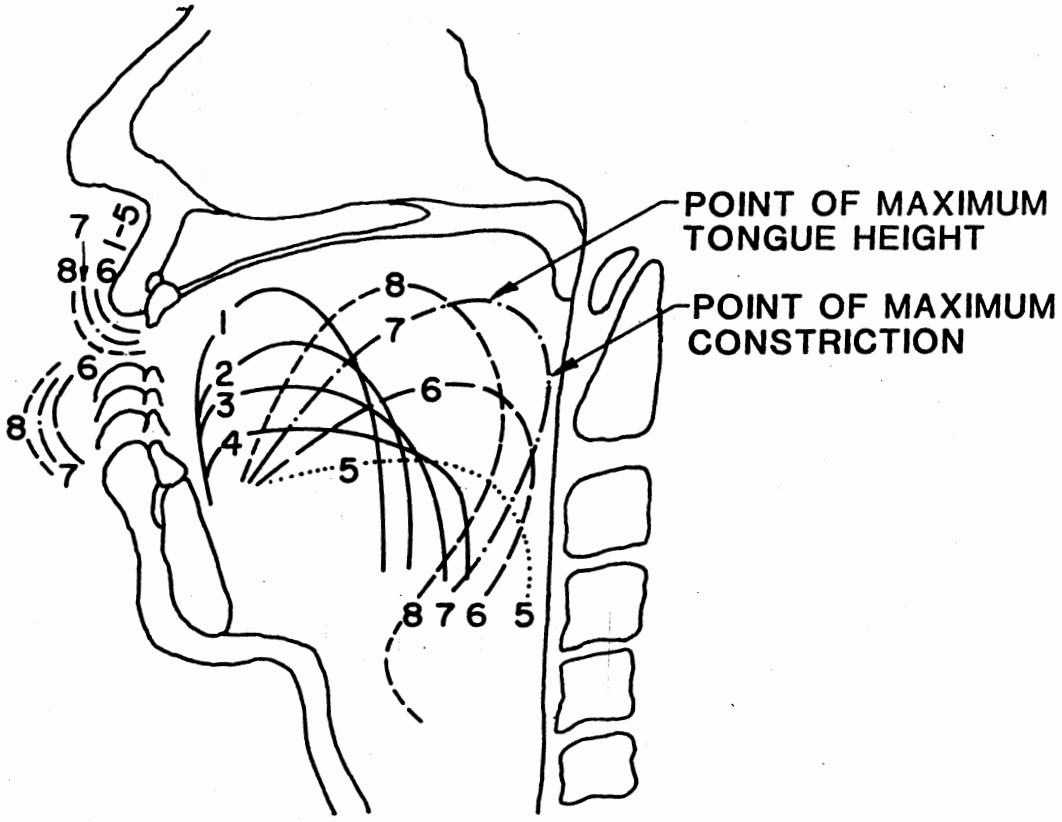


Figure 3. (Continued)

the presence of voicing. When friction and voicing are both present, the sound produced is called a voiced fricative. When friction is present and the vocal cords are relaxed, the sound produced is called an unvoiced fricative. When the vocal cords are used and when there is no friction in the vocal tract, the sound produced is called a voiced non-fricative.

As pointed out earlier, the vocal tract is a non-uniform acoustical tube which is time varying in shape. The major anatomical factors causing this time varying change are the articulators. These articulators cause the cross-sectional area of the lip opening to vary over a range of 0 cm^2 with the lips closed to about 20 cm^2 with the jaw and lips open [10][22][24]. X-ray data show that the cross-sectional area of the vocal tract is controlled primarily by the position and shape of the tongue, as shown in Figures 3-5 [18][24]. The tongue usually forms a constriction or region of minimum cross-sectional area, during the articulation of vowels. The cross-sectional area can vary from 0.3 cm^2 to 10 cm^2 at the lips, with a variation of the restriction radius from d_o = .4 cm to 1.2 cm. The distance from the glottis to the restriction at which the smallest radius is measured varies from d_o = 5 cm to 12 cm, whereas the vocal tract average length is 17 cm, from the glottis to the lips. In general, x-ray results show that during the articulation of vowels, the dimensions of the vocal tract along the length of the tongue are controlled primarily by the position of the tongue constriction and by the degree of tongue constriction. Whereas in the region beyond about 15 cm from the glottis, the mandible and the lips determine the cross-sectional area.



TONGUE AND LIP POSITIONS FOR THE VOWELS

(1) [i]
(2) [e]
(3) [ɛ]
(4) [æ]
(5) [ɑ]
(6) [ɔ]
(7) [o]
(8) [u]

Figure 4. Normal Articulation Process

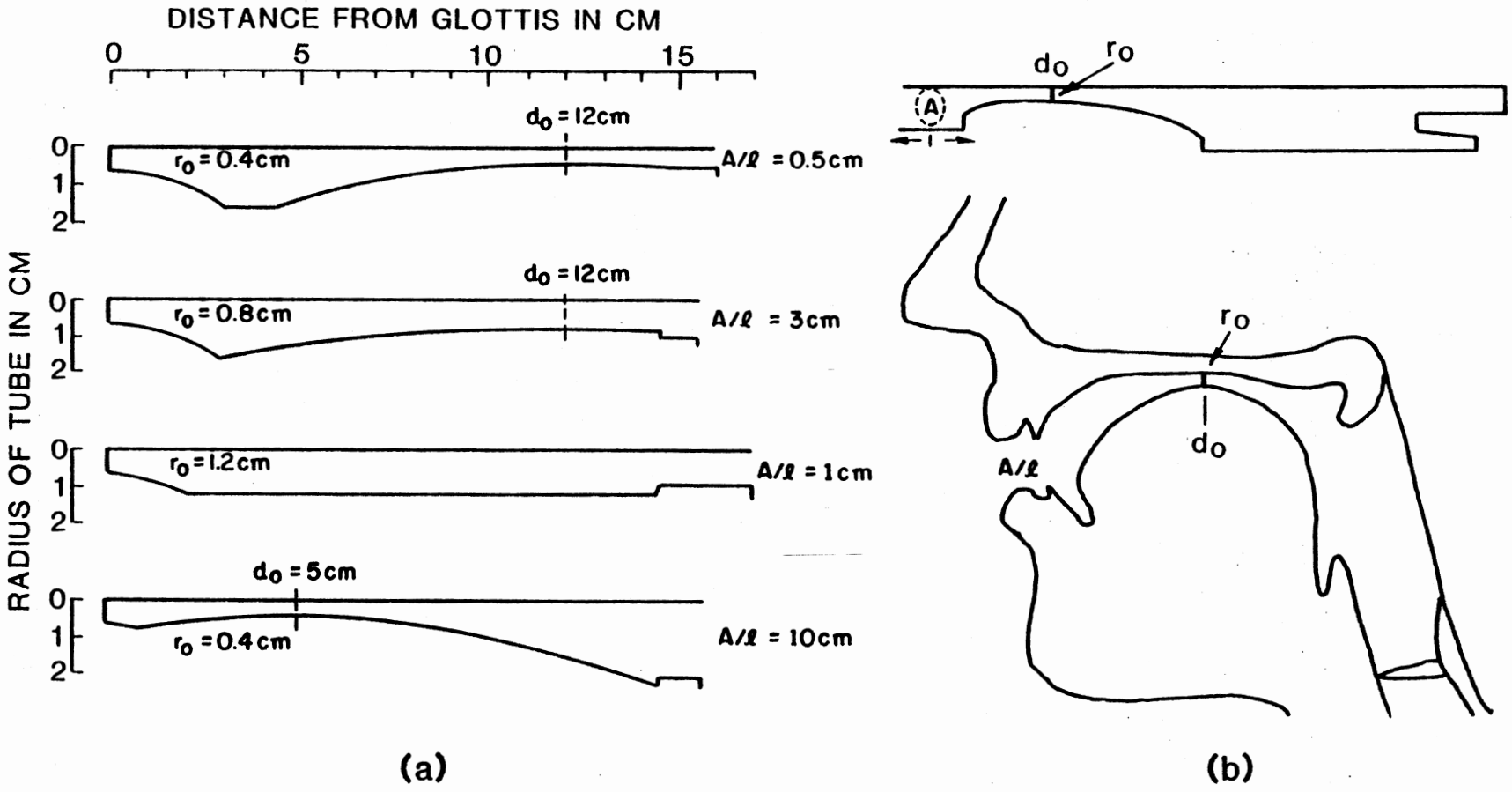
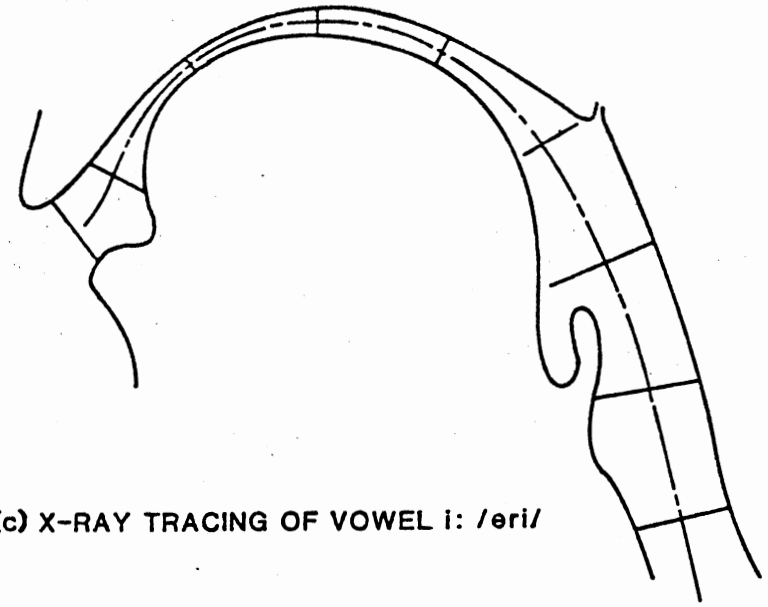
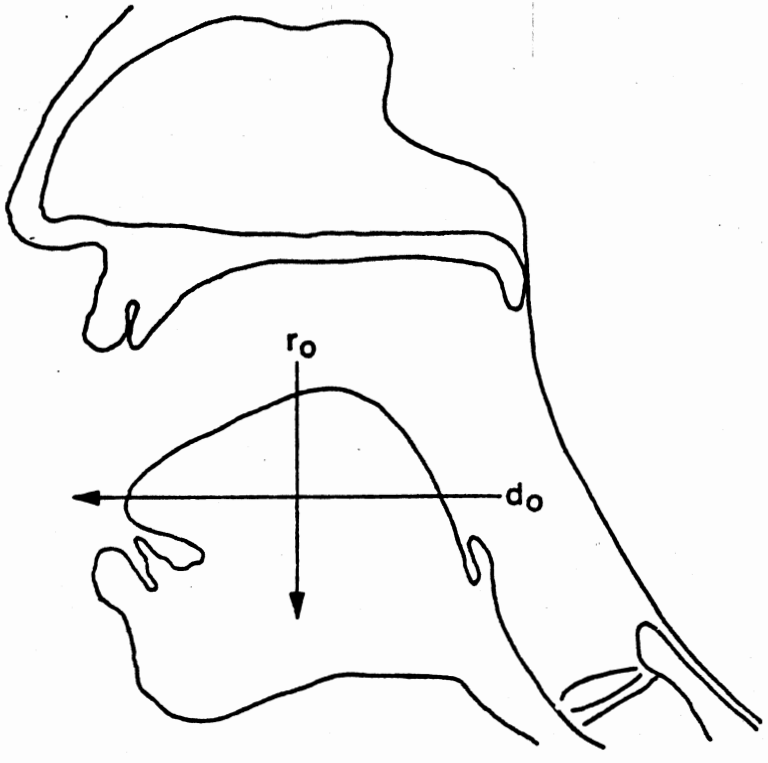


Figure 5. X-Ray of Cross-Section Area for Different Constrictions



(c) X-RAY TRACING OF VOWEL i: /eri/



(d)

Figure 5. (Continued)

The above discussion is for non-nasal sounds. For nasal sounds, the velum closes the vocal tract from the nasal cavity during the production of these sounds. The nasal sounds /n/ (wʌn/ and /naɪn/ in English digits, for example, /m/ (θamâny∂h, for eight in Arabic) and /ŋ/, (/nG/ English and Arabic digits do not have this sound) uses the nasal tract.

Vocal Pitch and Loudness

Vocal-fold tension is increased by contraction of the vocal-fold muscle. Releasing more puffs of air in a given period of time increases the frequency and pitch of the tone produced. In other words, the pitch of the human voice changes in accordance with changes in the mass, tension, and length of the vocal folds. Adult male's voices have lower pitch than female's voices because the male larynx is larger and has longer folds than the female larynx [18][19].

Adjustment of subglottal air pressure changes the intensity of the voice; that is, the greater the subglottal air pressure, the more intense the voice. Pitch is primarily a function of the laryngeal system. Loudness is related to intensity, which is primarily a function of the respiratory system. However, these two systems do not work independently of each other, and to maintain good voice control, the speaker must use the larynx and the air stream in a skillful balance. This coordination requires the participation of all the articulator and many components of the nervous system [23] [25].

Figure 6 is a schematic representation of changes in glottal area during a vibratory cycle. The area is determined by movement of the

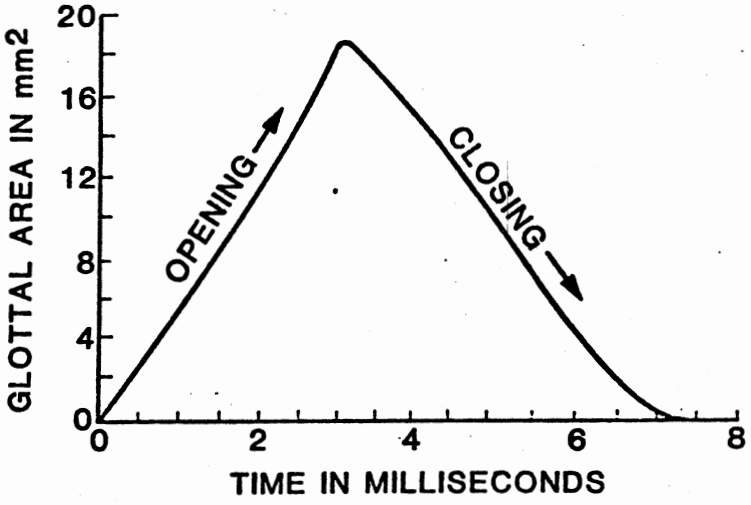


Figure 6. Schematic Representation of Changes in Glottal Area During a Vibratory Cycle

vocal folds, and the steepness of the slope of any segment of the curve represents the velocity with which the vocal folds move [22].

Articulation

X-ray observations of the upper respiratory system show that the larynx opens into a passageway called the pharynx, which in turn opens into two nasal and vocal tracts. The walls of the pharynx, the soft palate (i.e. the velum), tongue, mandible, and lips are somewhat free to move. Their movements change the configuration of the pharyngeal and oral airways or tubes. These structures shown by Figures 3 and 4 are referenced before as articulators because they are involved in the production of speech sounds [22][24]. The soft palate is a muscular continuation of the hard palate. It is very mobile and can move rapidly to close or open the pharyngeal air passage between the oral and nasal cavities. Movement of the pharyngeal walls toward the palate often contributes to the closure of the nasal cavity. The velum is opened during the production of the nasal consonants /m/,/n/ and /ŋ/ (i.e., nG). The velum is open during the production of these three consonants sounds, allowing sound from the larynx to be modified by the nasal cavities as well as by other articulators.

The front teeth contribute to articulation. This can be approximated by the movements and positions of the tongue and the lips. The facial muscles allow the lips to have the proper shape for the production of both vowels and consonants. The lips supplement the tongue fairly well in shaping the vocal tract for the production of vowel and

consonant sounds. The lip sounds are among the first speech sounds that a baby acquires, and these sounds can be found in all oral languages.

Classification of Speech Sounds

It is appropriate to introduce the sounds of speech in terms of the articulators and movements that produce them. The concept of phonemes in continuous speech and connected digit utterances can now be easily appreciated. Therefore sounds belonging to different phonemes is classified according to the movements and positions of the articulators that produce them [1][6][11][24][26].

It is convenient to divide sounds into vowels, dipthongs, semivowels, and consonants for spoken digits as shown in Figures 7-10 for digits spoken in English and Arabic, respectively. Tables I-IV present the classification of phonemes for digits spoken in English and Arabic according to the tongue positions and classification of vowel sounds using international phonetic alphabet.

Consonants

Consonants are differentiated by place of constriction, manner of constriction, or the presence or absence of voicing or laryngeal tone. Unvoiced consonants are produced due to the flow of air through the constriction between maximum tongue hump and the palate, while the vocal folds are not vibrating. Any two consonants will differ from each other in terms of one or more of these three features. Place of articulation encompasses the structures from the front to the back of the mouth, and two articulators are necessary to establish a place of constriction [18][22]. For example the sounds produced by closure of

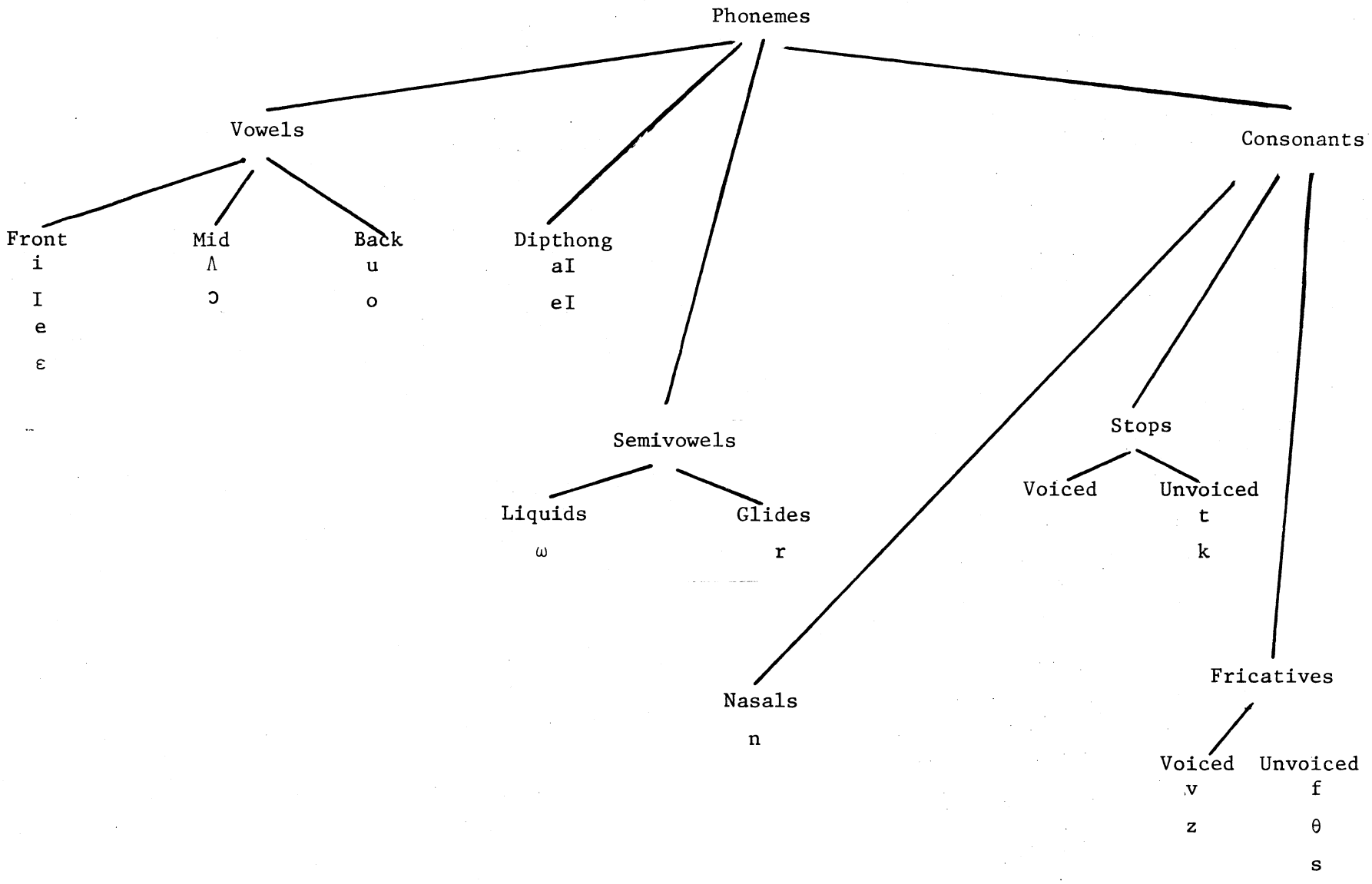
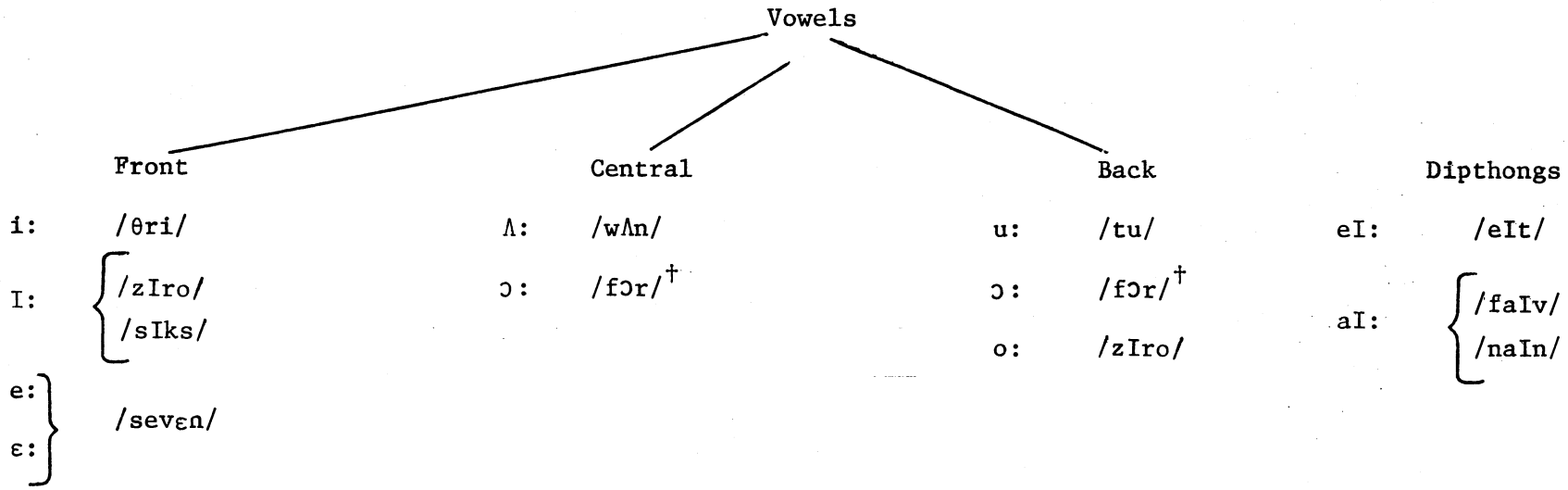
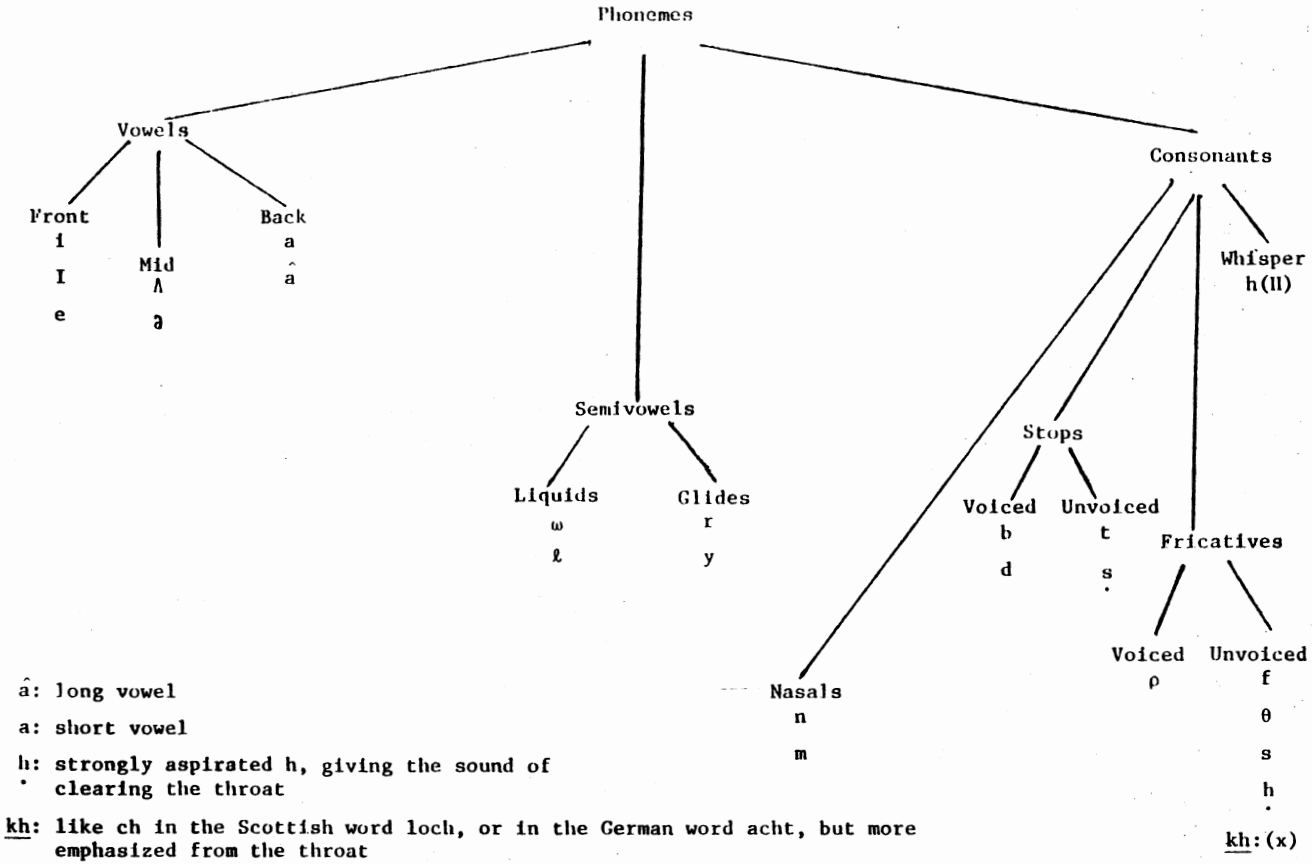


Figure 7. Phonemes in American English Digits



†Rabiner considered /ɔ/ as mid-vowel, whereas Flanagan considered /ɔ/ as back-vowel.

Figure 8. Vowels Used in Digits 0 - 9 Spoken in American English



â: long vowel

a: short vowel

ḥ: strongly aspirated h, giving the sound of clearing the throat

kh: like ch in the Scottish word loch, or in the German word acht, but more emphasized from the throat

ṣ: an s, produced more from the throat setting a larger part of the tongue against the palate; the s in sword or sauce approaches this sound

ρ: Ayn, a strong guttural compressing the throat and exploding the breath

Figure 9. Phonemes of Digits Spoken in Arabic

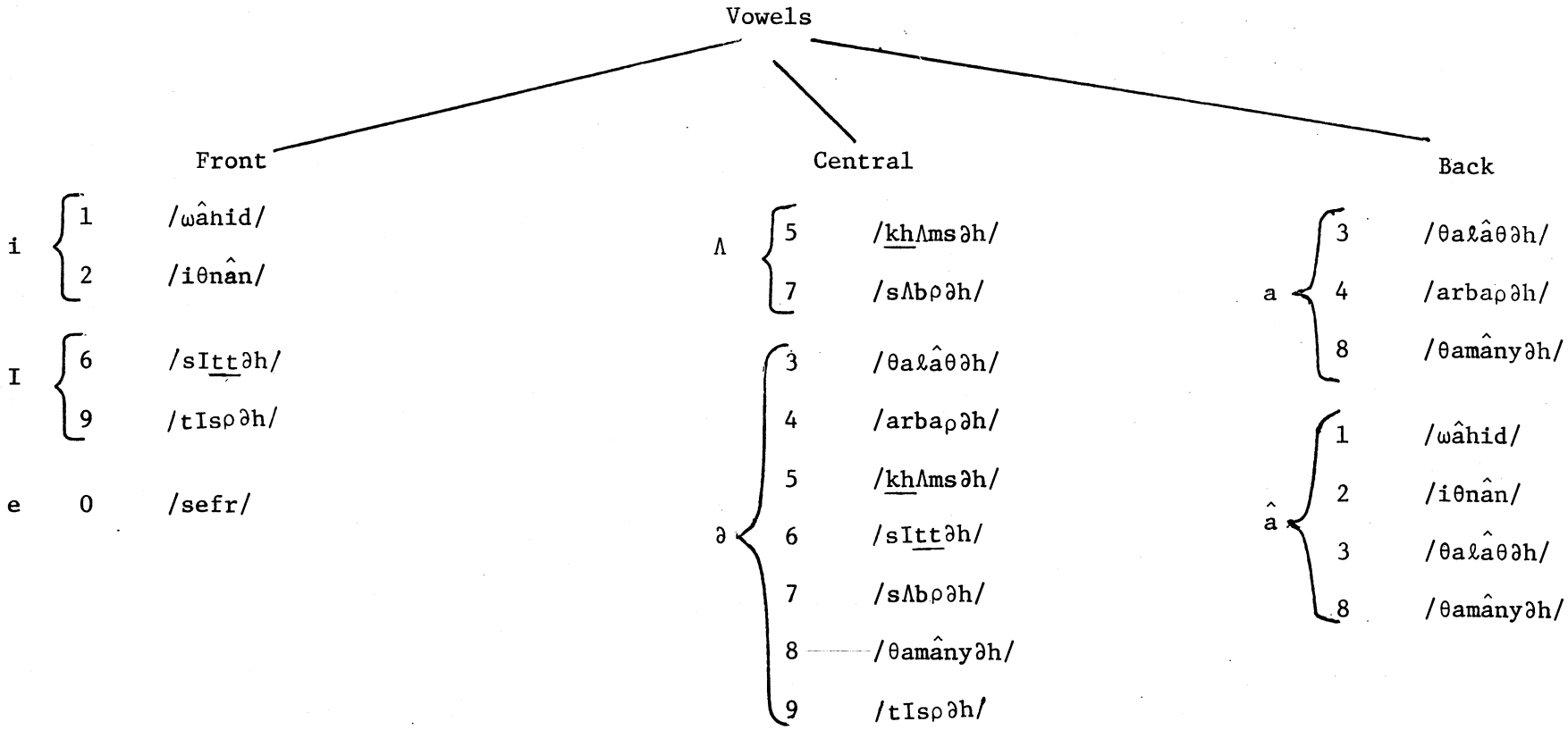


Figure 10. Vowels Used in Digits 0 - 9 Spoken in Arabic

TABLE I

VOWELS USED IN DIGITS SPOKEN IN AMERICAN ENGLISH, ACCORDING TO THE DEGREE OF CONSTRICTION AND TONGUE HUMP POSITION

Degree of Constriction	Tongue Hump Position		
	Front	Central	Back
High	/i/ : /θri/		/u/ : /tu/
	/ɪ/ : /sɪks/, /zɪro/		
Medium	/e/, /ɛ/ : /sevɛn/	/ʌ/ : wʌn	/o/ : /zɪro/, /ɔ/ : /fɔr/
Low			

Dipthongs

eI: /eIt/

aI: /faIv/, /naIn/

TABLE II

SEQUENCE OF SOUND CLASSES OF DIGITS SPOKEN IN AMERICAN ENGLISH

Digit	Phonemes	Sequences of Sound Classes					Notes
0	/zIro/	Voiced fNLC	FV	VLC	BV		
1	/wʌn/	VLC	MV	VLC			
2	/tu/	Unvoiced SNLC	FV	BV			
3	/θri/	Unvoiced fNLC	VLC	FV			
4	/fɔr/	Unvoiced fNLC	BV	MV	(VLC)		
5	/faIv/	Unvoiced fNLC	MV	FV	Voiced fNLC		(Dipthong)
6	/sIks/	Unvoiced fNLC	FV	Unvoiced SNLC	Unvoiced fNLC		
7	/sevɛn/	Unvoiced fNLC	FV	Voiced fNLC	FV	VLC	
8	/eit/	FV	Unvoiced sNLC				(Dipthong)
9	/naIn/	VLC	MV	FV	VLC		(Dipthong)

Voiced, fNLC: voiced fricative noise-like consonant
Unvoiced, fNLC: unvoiced fricative noise-like consonant
Unvoiced, sNLC: unvoiced stop noise-like consonant
VLC: vowel-like consonant
FV: front vowel
MV: middle vowel
BV: back vowel

TABLE III

VOWELS USED IN DIGITS SPOKEN IN ARABIC, ACCORDING TO THE DEGREE OF CONSTRICTION AND TONGUE HUMP POSITION

Degree of Constriction	Tongue Hump Position		
	Front	Central	Back
High	/i/ : /iθnân/,'2'; /ωâhid/		
	/I/ : /sIttəh/,'6'; /tIsρəh/,'9'		
Medium	/e/ : /sefr/,'0'	/ʌ/ : /khʌmsəh/,'5'; /sʌbρəh/,'7'	
		/ə/ : /θaℓâθəh/,'3' ⋮ /tIsρəh/, '9'	
Low			/a/ : /arbaρəh/,'4'
			/â/ : /ωâhid/,'1'; /iθnân/,'2'; /θaℓâθəh/,'3'; /θamânyəh/,'8'

TABLE IV

SEQUENCE OF SOUND CLASSES OF DIGITS SPOKEN IN ARABIC

Digits	Phonemes	Sequences of Sound Classes						
0	/sefr/	Voiced sNLC	FV	Unvoiced fNLC	VLC			
1	/ωâhid/	VLC	Long BV	Unvoiced fNLC	FV	Voiced sNLC		
2	/iθnân/	FV	Unvoiced fNLC	VLC	Long BV	VLC		
3	/θaɫâθəh/	Unvoiced fNLC	MV	VLC	Long BV	Unvoiced fNLC	MV	Whisper
4	/arbaɸəh/	MV	VLC	Voiced sNLC	MV	Voiced fNLC	MV	Whisper
5	/khʌmsəh/	Unvoiced fNLC	MV	VLC	Unvoiced fNLC	MV	Whisper	
6	/sIttəh/	Unvoiced fNLC	FV	Unvoiced sNLC	MV	Whisper		
7	/sʌbɸəh/	Unvoiced fNLC	MV	Voiced sNLC	Voiced fNLC	MV	Whisper	
8	/θamânyəh/	Unvoiced fNLC	MV	VLC	Long BV	VLC	MV	Whisper
9	/tIsɸəh/	Unvoiced sNLC	FV	Unvoiced fNLC	Voiced fNLC	MV	Whisper	

Voiced, sNLC: voiced stop noise-like consonant
Voiced, fNLC: voiced fricative noise-like consonant
Unvoiced, sNLC: unvoiced stop noise-like consonant
Unvoiced, fNLC: unvoiced fricative noise-like consonant
VLC: vowel-like consonant (semi-vowels), Liquids and Glides
FV: front vowel
MV: middle vowel
BV: back vowel
Note: All vowels are short vowel, except /â/, which is a long vowel.

lips (/p, b, m/) as in /arbaɒðh/ (four in Arabic) and /θamâ̂nyðh/ (eight in Arabic). Constrictions produced by lips and teeth results in the production of sounds /f, v/, as in four, five, and seven. During the production of a stop sound, articulators momentarily occlude, or stop, the oral air passage. Air pressure is built behind the occlusion and releasing this results in a stop sound, such as /t/ as in two, and /d/ as in /ωâ̂hid/ (one in Arabic). The affricates are much like the stops, except that the pulse of air is sustained a bit longer as in /t/ and /dz/ [24][25]. Fricatives are caused by the approximation of two articulators, thus directing exhaled air through a narrow opening, causing a relatively continuous stream of noise. The nasals are sounds that are made by lowering the velum, thus directing the sound stream through the nose rather than the mouth. The semi-vowel glides /w, r, j/, as in /wʌn/, /fɔr/ and /ωâ̂hid/ are made with more constriction than vowels, but not enough to cause turbulent air flow. The third type of consonant however is differentiated by the presence or absence of voice, since some consonants are voiceless and some are voiced. The onset of vocal-fold vibration occurs earlier in voiced than in voiceless consonants. That is, vocal-fold vibration is present during portion of the consonants that are perceived as voiceless [18-20]. On studying the phonemes that are used only in spoken digits, the terms vowel-like and non-vowel-like is applied.

Vowels

Classification of vowels require a different analysis from that used for consonants. All vowels are voiced. Although lip configuration is important, differences among vowels are determined primarily by the

position of the tongue tip, tongue edges, and tongue body. Denes and Pinson [21] classify vowels in terms of the position of the highest part of the tongue body. For example, they state that when the tongue body is positioned as high and as far forward as possible without causing turbulence when the lips are spread and when voicing is produced, the vowel /I/ (as in /sIks/ results.

Vowels are classified as front, center, and back and as high, middle, and low relative to the position of the tongue body in the oral cavity as shown in Figures 3 and 4 and Tables I and III. The approximate positions of American English vowels relative to one another can be clearly indicated. Whereas, dipthong sounds are produced whenever a shift in tongue location from that associated with one vowel to that of another vowel results (as in /eIt/ and /naIn/).

Speech Acoustics

An understanding of certain principles of acoustics must be acquired in order to comprehend the production and transmission of speech. Speech is an acoustical phenomenon, and it is a special case of sound production [34]. Although vibration is not sound unless it is heard, but for convenience, it is feasible to accept simply that sound occurs as a result of vibration , as shown in Figure 11.

A source of energy is needed to set a vibrator in motion, for sound generation. Exhalation of air from the lungs may be considered to be the source of energy, and the vocal folds to be the vibrator, for the production of speech. The air in the cavities of the throat and mouth provides the elastic medium for the transmission of these

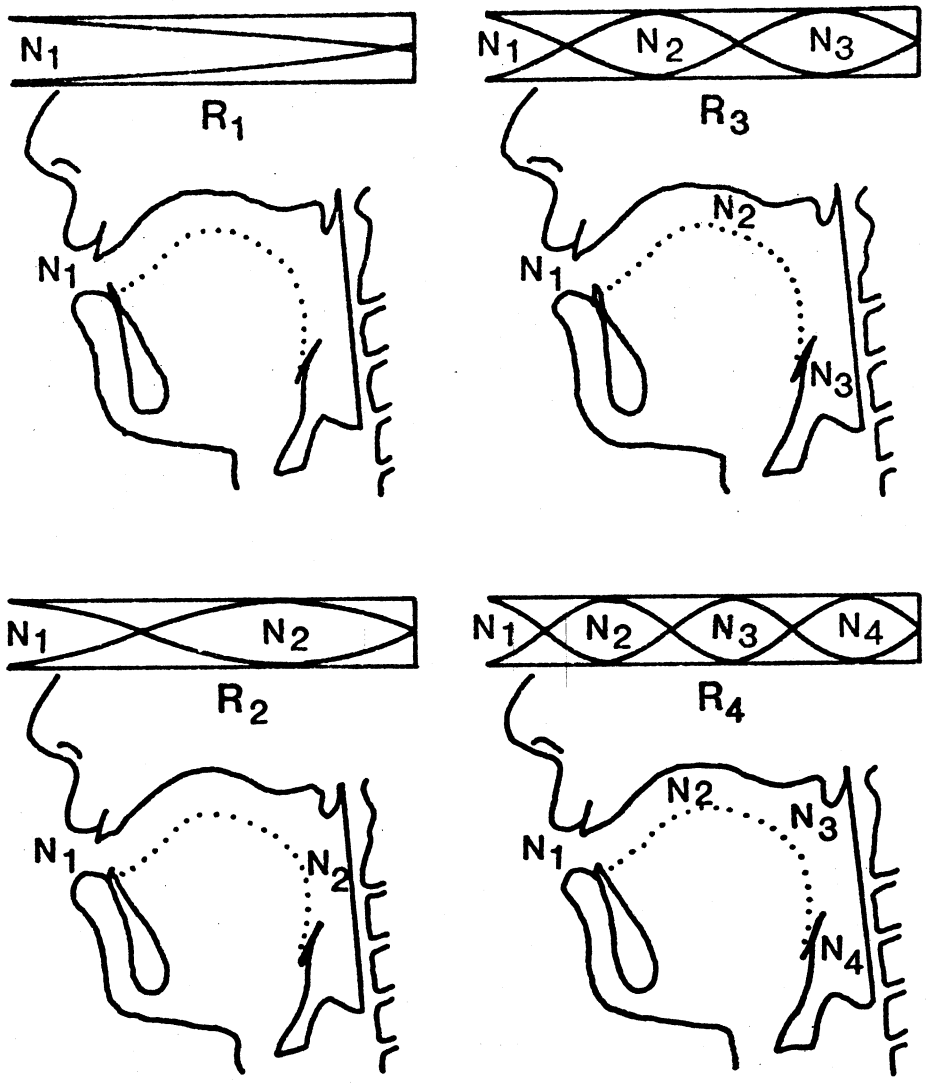


Figure 11. Distribution of Volume Velocity at the Frequencies of Each of the First Four Resonances of an Ideal Neutral Articulation in Which the Vocal Tract Simulates a Tube of Constant Cross-Sectional Area

vibrations or speech sounds. It is feasible to assume that the vibration of the vocal folds causes a series of compression waves or pressure changes in the surrounding air. Since reception and perception of speech by the ear depends on air conduction, special attention must be given to the mode of sound transmission in air.

Sound Propagation

Sound waves in air are described as longitudinal waves, because the molecular movement in air is horizontal, or parrallel to the driving force. Sound waves travel as a result of the patterns of molecular displacement in the same direction as the applied force, and the intensity fade away as the distance from the source increases. The density of the material which is defined as "the number of molecules per unit volume", determines the velocity at which sound may be propagated. It is well known that sound travels faster in denser materials. For example, sound travels faster through steel (about 15,000 feet/second) than through air (about 1100 feet/second).

The theory of speech as wave motion and how speech waves are produced and heard is usually included and covered by the acoustic of speech, which is the field of study, that inspired researchers of various specialties during the last decades. Speech sounds defined from their production within the vocal tract have been till now, the most interesting and popular field of research for classical phonetics, well known as articulatory phonetics. The speech wave, which is defined by the sound pressure variations at a point in front of the speaker

has not only been of great concern to speech research of communication engineers, but also of more concern to speech pathology researchers. Complete specifications of the speech wave can be obtained with the aid of modern sound recording and analysis techniques.

Resonance

The vibration of the vocal folds produce only a complex buzzing sound in isolation. However, in normal speech production, the vocal folds cause variation of pressure, which resonate the vocal tract. The quality of voiced sounds produced by the vocal tract resonance, is quite different from that which would result from vocal-fold vibration alone. Hence, it is clear to state the fact that the spectrum of an acoustic signal is influenced considerably by the acoustic environment in which the signal is produced and propagated [35][39].

By examining the concept of resonance, it can be depicted that all objects or volumes of air in open and closed tubes and cavities vibrate more readily at certain frequencies than at others. That is, the cavity will act as a selective filter, passing some frequencies and rejecting other frequencies. In other words, all objects or volumes of air in open or closed tubes, and cavities, have certain natural frequencies of vibration and thus are more responsive to those frequencies. Consequently, when sound with a complex spectrum are produced, the frequencies in that complex waves inherent resonances frequencies, with greater amplitude if these frequencies are the natural frequencies of a given object or volume of air. This is due to the fact that an object will resonate in sympathy of the vibrating

source. Thus, when sounds with a complex spectrum are produced, the cavity resonates more freely and resonance frequencies with greater amplitude are generated if these frequencies are the natural frequencies of a given object, cavity or volume of air.

In the acoustics of speech, the concept of resonance is of great importance and of particular significance. The vocal-fold vibration contains a fundamental frequency with certain harmonics or overtones, since the vibration of the vocal folds is quasi-periodic. The vocal tract will respond to certain frequencies generated by the vocal folds and reject the others since the cavities of the vocal tract act as a resonator with certain natural frequencies. The vocal tract may assume different resonant frequencies, because, the cavities of the vocal tract are modified continuously during the production of speech.

Formant Frequencies

The resonances of the vocal tract are usually called formants, and the frequencies to which they respond more are known as the formant frequencies. The spectrum of vowel sounds have very apparent formant frequencies, and are revealed as resonance peaks, that is, the peaks of maximum energy or amplitude at given frequencies. Different formant frequencies also occur since the configuration of the vocal tract must be changed to produce different speech sounds. The fundamental frequency and resultant harmonics are produced by the vibration of the vocal folds. The resultant resonant frequencies are not the same as the harmonics produced by the vocal folds unless by coincidence

because the formant frequencies are produced from the changes in the time-varying vocal tract [18]. Figure 12 depicts a periodic signal generated by the vocal folds and the resultant harmonic spectrum, which reflects the fundamental frequency of the vocal-fold vibration and the respective harmonics. Also the vocal-tract configuration and the resonant or formant frequencies that occur as a result of vocal-tract resonance due to the vocal-fold output are clearly indicated in Figure 12.

As mentioned earlier, vowels are characterized by quasi-periodic signals and formant frequencies. Consonants are characterized by random signals which may or may not include periodic information as a result of voicing. The voiceless consonant contains only aperiodic vibration, which results from air turbulence produced by changing and constricting the size of the orifice of the mouth and friction or interference in air flow by the articulators such as the tongue, teeth, and lips. Voiced consonants include the quasi-periodic vibration of the vocal folds and aperiodic vibration depicted in voiceless consonants.

Acoustic Characteristics and Perception

The two speech parameters that are widely used in speech analysis are intensity and frequency. The effects of intensity is first considered since intensity changes are the least complicated subject to study. Speech must be sufficiently loud or intense to be clearly understood. Among the various speech sounds of English, the intensity range is 680 to 1, or 28 dB, from the weakest to the strongest speech sound [11][16][18]. The energy or sound pressure of normal speech measured

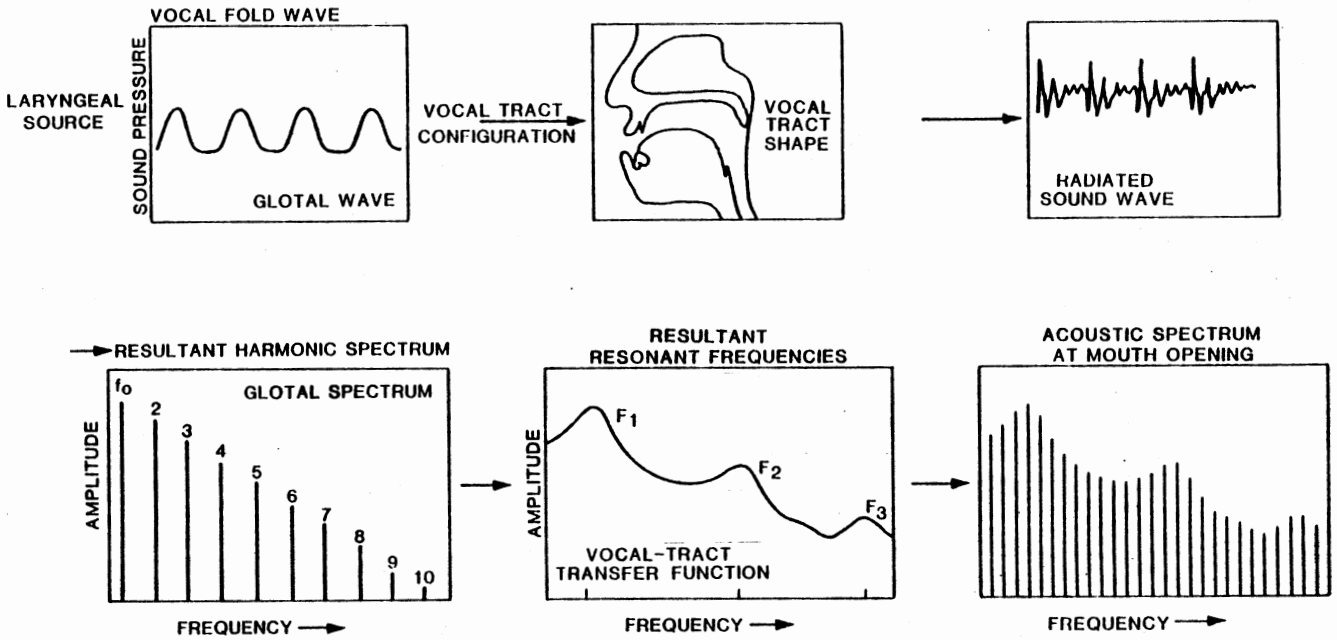


Figure 12. Schematic Diagram of the Physiological and Acoustical Characteristics of Speech Sound Production

at one meter from the lips of the speaker is about 62 dB. Speech signal might reach a sound-pressure level of about 85 dB, if a person talks as loudly as possible, while the softest speech is about 45 dB [16][21].

Vocal-folds vibration supply the internal energy to drive the vowel sounds. The fundamental frequency of the vocal folds is about 125 Hz for males and 250 Hz for females which is an octave higher. The primary energy in vowel sounds is of low frequency, since vowel sounds are quasi-periodic signals composed of the fundamental and related harmonics of the vocal-fold vibration. Also, most of the energy or power of speech are known to be carried (acquired) by vowels and in addition have longer duration than consonants. In fact 60 percent of the energy or power of spoken speech utterance is contained in the frequency range below 500 Hz, which contribute only 5 percent to intelligibility. Also, 60 percent of the intelligibility is contributed by the frequencies above 1000 Hz with only 5 percent of power. Thus most of the power is carried by the vowels, or low-frequency sounds, while the consonants, or higher-frequency speech sounds, which carry lower power, are much more important for speech intelligibility [21].

Effects of Noise

The intelligibility of speech in quiet listening environment is not generally a difficult task. Frequently speech is perceived in the presence of noise, in real-life situations. However, noise does not always interfere seriously in understanding speech unless the noise

and the speech occur in the same frequency range. When this situation occurs, the relationship between the signal-to-noise is important. Speech intelligibility, however, is not affected provided that speech is 100 times more intense than the noise [43]. Speech intelligibility for digits might be reduced by 50 percent if speech and noise are of equal intensities [16]. Speech can be perceived and understood even when it is of lower intensity than noise if the speech and the noise come from two different directions.

Frequency and Intelligibility

Frequency content is obviously important for speech intelligibility. Experiments in which filters are used, depicted important findings, such as the intelligibility of speech for one-syllable words is slightly reduced when frequencies above 1600 Hz are eliminated. But intelligibility is reduced by 25 percent when filtering is extended to 800 Hz. Whereas intelligibility is not significantly affected when frequencies below 1600 Hz are eliminated. In general, the intelligibility of speech is reduced further by about 25 percent when all frequencies below 3200 Hz are filtered out [6][8][9][50].

Segmental Analysis

The perception of vowels depends on the respective formant frequencies of the sound which is revealed by a segmental analysis of vowel sounds. Men, women, and children all use about the same vocal-tract configurations to produce the same vowel sounds, even though different individuals have varying sizes of vocal mechanism and vocal tracts. The relationship among formant frequencies is about the

same even though the formant frequencies may be higher or lower for different people.

Consonant sounds are produced by the constriction of the vocal tract and many are classified as either fricatives (i.e. "s" in /sIks/ or plosive or stops (i.e. "t" as in /tu/). Spectral or frequency differences make it possible to distinguish among fricative sounds. This can be seen from the phoneme in the digit /sIks/, i.e. "s" which has little energy below 4000 KHz, with a resonance peak between 4000 and 7000 Hz [20]. When the turbulent air stream is stopped by closure of the vocal tract and then released, stops or plosives are produced. The initial plosive "t" in the digit /tu/ is more forcefully exploded than the final plosive "t" in the digit /eIt/. Similarly, initial plosive is forcefully exploded in Arabic digits as in /θaɫâθəh/ and /ωâhid/. Thus plosives occurring in the initial position of a digit, are exploded more strongly than those occurring at the final position of a digit. Spectral differences are very essential to distinguish plosives among one another and from consonants, such as the voiced plosive "b" in the Arabic digit /arbaρəh/, which has most energy between 500 and 1500 Hz. In contrast, "t" and "d" have a higher spectrum, with energy up to about 400 KHz. Classification of plosives, however, can be influenced significantly by its use in a spoken digit. Phoneme recognition is possible by segmenting speech and using the above analysis.

Methods of segmentation of isolated words into phonemic units have typically utilized information pertaining to rapid changes in the energy concentration in the frequency spectrum of the speech waveform [7][47][48]. The task of performing segmentation, i.e. a subdivision

of the uttered word into discrete consecutive phoneme sections, is a very important basic step towards machine phonemic recognition. In previous research, primary segmentation is used to group together similar acoustic adjacent minimal segments [52]. If the difference between corresponding parameters is less than a minimum, then two parameters should be considered as identical. Transitional segments where the acoustic characteristics vary with time considerably, is very difficult to locate, therefore secondary segmentation procedure is used to correct possible errors of the primary segmentation. This technique involves much computational work, hence it is not efficient.

Several segmentation techniques have been developed for continuous speech, apart from modifying the segmentation techniques used for isolated words, to perform segmentation of connected digits. This leads to the problem of words or digit boundary locations.

The problem of accurately locating the end points of an utterance is actually a specific case of the more general problem of labeling an interval of a signal as silence, unvoiced, or voiced. If one had a perfect technique for this three-level decision, the end point-location problem would be trivially solved. However, such an ideal algorithm does not exist as yet. Therefore, partial solutions to this more specific problem of isolating speech from a noisy background have been examined.

Coarticulation

The recognition of specific sounds, or phonemes, might be influenced in connected rather than in isolated digits. Speech, however,

has been considered to be composed of a sequence of distinctive, separate sounds; by analogy, "beads on a string". This is essentially the case in analysis of sounds in isolated digits, but actually is not the case in an analysis of connected digits or continuing speech. Coarticulation is a term given whenever there is an interaction of associated sounds. Due to the configuration of the vocal tract for any given sound being influenced by the shape required for the previous sound and respiratory movement for production of a following sound, and because the vocal tract is continuously in transition in the production of connected digits or continuous speech, coarticulation occurs [10].

Vowels

The position for a vowel sound can be assumed essentially to mean shaping the resonators in such a way as to produce the desired acoustic effect. It is clear that the vowel sounds are continuants. In other word, the speech mechanism assumes the position for the vowel and holds it with relatively little movement for a measurable fraction of a second while the sound is produced. Although the periods of time involved are small, continuants are characterized by these brief periods of holding. By contrast, the glide sounds are produced while the mechanism is in movement and their identifying characteristics are the result of this movement. In general, a pure vowel is defined as one in which the mechanism is held relatively stable in contrast to the glides in which the movement is the essence of the sound [39][50]. Noting that the term continuants is applied or referred to vowels, fricative consonants and nasals, while stops refers to plosive consonants and glides to inter vowel.

It is the function of the articulatory mechanism to break up and modify the laryngeal tone and to create new sounds within the mechanism itself. In fact, the speech mechanism as a whole not only articulates, i.e. join together, but also separates and molds the sounds delivered to it by the vibrator and resonator mechanisms. In addition it creates new sounds within itself by utilizing the energy supplied by the power mechanism in such a way as to produce within the oral cavity frictional noises that are independent of the laryngeal tone. Because of this, the articulatory mechanism assumes considerable importance to speech researchers.

Speech mechanisms can be divided into four units, distinct functionally but overlapping structurally. These units are: the power mechanism, the vibratory mechanism, the resonator mechanism, and the articulatory mechanism. These various functions and structures are coordinated through the activity of the voluntary nervous system. This coordination is made possible by four types of activity carried on by the nervous system: (1) motor activity that provides the stimuli that causes muscles to contract; (2) sensory reporting that gives information as to how the movements were produced; (3) auditory monitoring that makes possible the setting up of, and conformance to, speech standards; and (4) the associative function that ties up the auditory symbol with its meaning and with the motor pattern necessary to produce it.

CHAPTER III

CONCEPT AND COMPUTATIONAL TOOLS

Introduction

Due to the complex nature of speech process, it is suitable to have a parametric representation of the acoustic waveform which can be used to extract certain desired speech characteristics. Such parameters, used to describe the acoustic waveform over a specified time interval, might include Fourier coefficients, RMS energy, rate of zero crossing or the locations and values of predominant spectral peaks. The Fourier analysis is the most generally used technique for obtaining quantitative information about the speech waveform. Speech is, in general, non-stationary, and can be considered as stationary on a short-time basis. The short-time analysis is discussed in a later section. The segmented speech needs to be described by a set of well-defined parameters, so these can be used in a speech recognition scheme. These parameters must be simple, yet convey qualitative and quantitative information and characteristics of speech signals.

Due to the fact that linear prediction model and the acoustical tube model are equivalent, the reflection coefficients can be obtained from the area functions, and visa versa [17][41]. Figure 13 shows that the vocal tract is considered as a set of interconnected sections of equal length and varying cross-sectional areas. The linear prediction

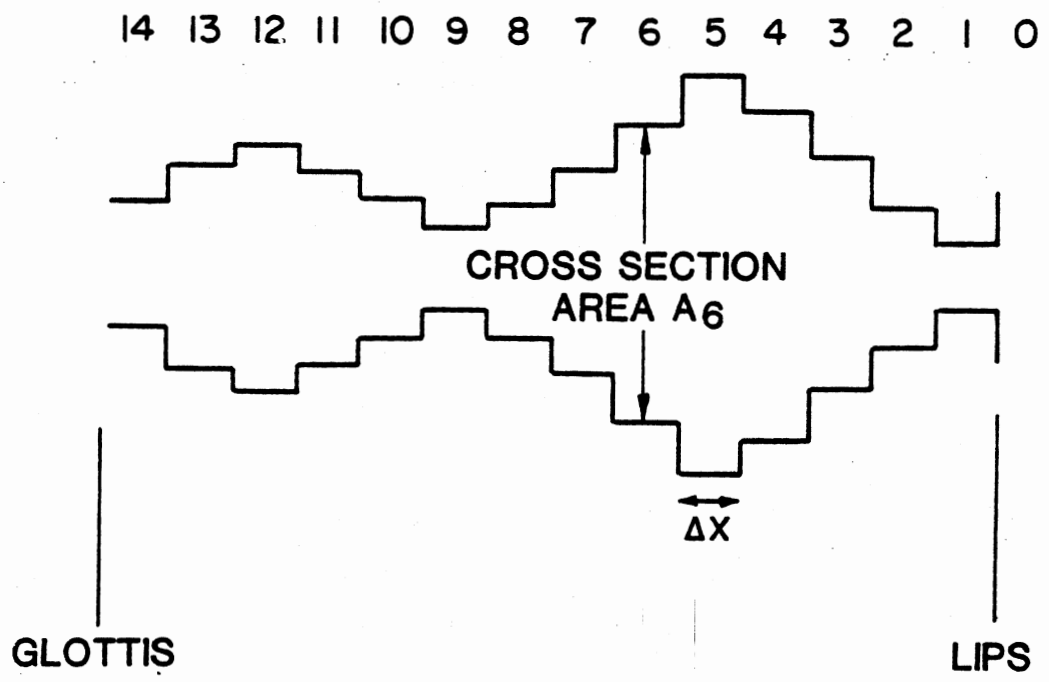


Figure 13. Concatenation of (N=14) Lossless Tubes of Equal Length

analysis model, to be discussed in the next section, is used because it is related to the acoustical tube model. The LPA not only filters out all voiced speech and formants, giving out the pitch period and random noise, but also allows for extraction of some useful parameters needed in the speech algorithm such as area functions. The estimation of the equivalent area function from the reflection coefficients provide for computing either part or the total equivalent area of the vocal tract cavity.

A mathematical discussion of LPA, autocorrelation method, windowing and pre-emphasis is presented. Parameters in terms of the vocal tract cavity ratios are defined, based on the facts obtained from x-ray pictures, that three cavities are observed. The ideal way to compute the RMS energy of the speech waveform is stated. Algorithms used for smoothing the RMS energy is discussed briefly with the aid of a block diagram. Parameters used for dip-classification algorithm are clearly defined. Finally, short-time analysis is briefly introduced. Also, window method is utilized for segmentation scheme and an overlap method is used in the end-point detection algorithm, for detecting digit boundaries accurately.

Linear Prediction Analysis

Linear prediction analysis applies to a class of problems in speech analysis and synthesis in which the present sample is predicted by a linear combination of past samples. The solution is obtained by solving a set of linear simultaneous equations based upon the least-square error criterion of optimality. From the solution, an all-pole digital filter can be derived, which has a discrete frequency response that closely

matches the smoothed spectral characteristics of the analyzed signal [29][44][63]. The significant advantage is that accurate spectral representation is obtained with only a few parameters over a 10 to 32 msec. signal.

An important form of linear prediction has been developed which is referred to as the autocorrelation method [31]. The autocorrelation method is very popular in speech processing, as it allows for a simple implementation when compared to the covariance method [1]. Furthermore, the parameters derived from an acoustic tube model can be related to a set of parameters in the autocorrelation method. Therefore, the autocorrelation method is used in this thesis. A brief review of the autocorrelation method is given below.

The Autocorrelation Method

Let $S(m)$ be the speech signal and $S_n(m)$ be the windowed speech signal. That is,

$$S_n(m) = S(m+n)\omega(m) \tag{1}$$

where $\omega(m)$ is a window (for example, a Hamming window) of length N. It is clear that $S_n(m)$ is non-zero only for $0 \leq m \leq N-1$. Let the predicted signal $\hat{S}_n(m)$ is expressed by

$$\hat{S}_n(m) = \sum_{k=1}^{p} a_k S_n(m-k) \tag{2}$$

where a_k's are some constants that are to be determined and p is the order of the prediction. The prediction error E_n is defined by

$$E_n = \sum_{m=0}^{N=p-1} e_n^2(m) \tag{3}$$

where

$$e_n(m) = S_n(m) - \hat{S}_n(m). \tag{4}$$

The coefficients a_k are obtained by minimizing E_n in (3). This minimization results in a set of normal equations

$$R_n(i) = \sum_{k=1}^{p} a_k R_n(|i-k|) \ , \ 1 \leq i \leq p \tag{5}$$

where $R_n(k)$ corresponds to the kth autocorrelation coefficient and is given by

$$R_n(k) = \sum_{m=0}^{N-1-k} S_n(m) \ S_n(m+k) \ . \tag{6}$$

It is well known that if (5) is expressed in a matrix form, the coefficient matrix is a symmetric Toeplitz. There are several efficient algorithms (Levinson's, Durbins's and Trench's algorithms) [1] available to solve (5).

The Durbin's algorithm (27) is considered to be most efficient and the a_k's in (5) can be computed using this method. This method is used in this thesis. For completeness, the algorithm is given below in terms of $R_n(k)$. For simplicity, the subscript of R is omitted.

$$E^{(o)} = R^{(o)} \tag{7}$$

$$k_i = \left[R(i) - \sum_{j=1}^{i-1} a_j^{(i-1)} R(i-j) \right] \Big/ E^{(i-1)} \ , \ 1 \leq i \leq p \tag{8}$$

$$a_i^{(i)} = k_i \tag{9}$$

$$a_j^{(i)} = a_j^{(i-1)} - k_i a_{i-j}^{(i-1)} \ , \ 1 \leq j \leq i-1 \tag{10}$$

$$E^{(i)} = (1-k_i^2) E^{(i-1)} \ . \tag{11}$$

Equations (7)-(11) are solved recursively for i = 1, ..., p and the solution for (5) is given by

$$a_j = a_j^{(p)} \quad , \ 1 \leq j \leq p \tag{12}$$

Transfer Function Relation

The Z-transform or the error sequence $e_n(m)$, E(Z), can be expressed in terms of the Z-transform or the windowed speech signal $S_n(m)$, S(Z), by

$$E(Z) = [1-F(Z)]\, S(Z) \tag{13}$$

where

$$F(Z) = - \sum_{i=1}^{p} a_i Z^{-i} \ . \tag{14}$$

The block diagram representation of Equation (15) is shown in Figure 14.

Vocal Tract Division

In a later section some new parameters are defined based upon fourteen section representation of the vocal tract [8], as shown in Figure 15. In a recent paper, the vocal tract has been considered in terms of two major sections, called the front and back sections as shown by the dotted lines in Figure 15 [8]. X-ray analysis shows that the vocal tract is actually divided into three cavities [18], namely front, central and back, as shown by the solid lines in Figure 15.

Reflection Coefficients

Corresponding to the fourteen sections, a set of reflection

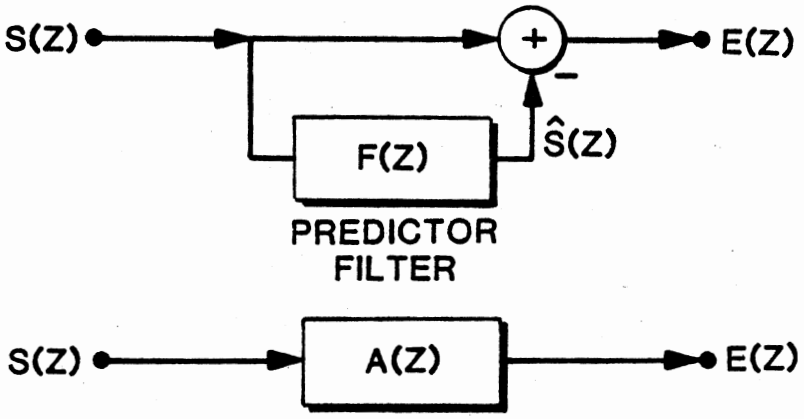


Figure 14. Equivalent Representation of Predictor Filter F(Z) and Inverse Filter A(Z)

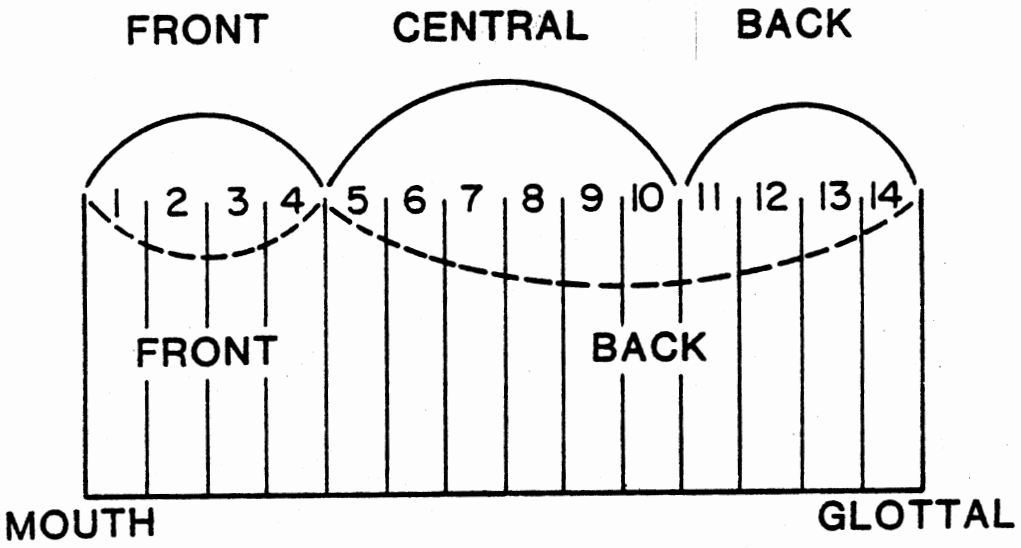


Figure 15. Equivalent Diagram Representing the Vocal Tract as an Acoustical Tube of Equal Lengths

coefficients can be defined. These are

$$k_i = \frac{\frac{A_i}{A_{i+1}} - 1}{\frac{A_i}{A_j+1} + 1} \quad , \; 1 \leq i \leq p = 14 \tag{15}$$

where A_i corresponds to the area of crosssection of the ith section. It is clear that k_i is bounded between -1 and +1. Also, it follows from (15) that

$$\frac{A_i}{A_{i+1}} = \frac{1+k_i}{1-k_i} \quad , \quad 1 \leq i \leq p - 1 \tag{16}$$

and

$$A_i = \frac{1+k_p}{1-k_p} A_{p+1} \tag{17}$$

Since k_i in (15) is of interest, and k_i is a function of a ratio of area function, it can be assumed that $A_{p+1} = 1$ without losing any generality.

The area's A_i's are functions of the reflection coefficient as shown in (17). The reflection coefficients are related to the $a_i^{(i)}$ in the Durbin's algorithm. This is shown in Equation (9). The transfer function of a lossless tube model consisting of p sections has the same form as the transfer function derived from the linear prediction analysis. The reflection coefficients r_i's obtained from the acoustic tube model are related to k_i's by the equation

$$r_i = -k_i \; . \tag{18}$$

Using this relationship, Equation (18) can be expressed as

$$A_i = \frac{1-r_i}{1+r_i} A_{i+1} \tag{19}$$

Parameters in Terms of Vocal Tract Cavity Ratios

Several parameters are defined below for future use in the digit recognition scheme. The front-to-total cavity ratio (FTR) is defined in terms of A_i, the area of cross section of the ith section, by

$$FTR = \sum_{i=1}^{p-10} A_i \Big/ \sum_{i=1}^{p} A_i \tag{20}$$

where p corresponds to the total number of sections (14 here) in the vocal tract. Note that A_1, A_2, A_3, and A_4 corresponds to the areas of cross-section in the front of the vocal tract (see Figure 15). Also, $\sum_{i=1}^{p} A_i$ corresponds to the total sum of the areas of cross-section. The central-to-total cavity ratio (CTR) is defined by

$$CTR = \sum_{i=5}^{p-4} A_i \Big/ \sum_{i=1}^{p} A_i . \tag{21}$$

The back-to-total cavity ratio (BTR) is defined by

$$BTR = \sum_{p-3}^{p} A_i \Big/ \sum_{i=1}^{p} A_i . \tag{22}$$

The front-to-back cavity ratio (FBR) is defined by [8].

$$FBR = \max (A_1, \ldots, A_4) / \max (A_5, \ldots, A_p) \tag{23}$$

Finally, the signed front-to-back cavity ratio (SFBR) is defined by

$$SFBR = Sgn(k_1) \cdot FBR \tag{24}$$

where $sgn(k_1)$ corresponds to the sign of the first reflection coefficient.

The digit recognition uses several other parameters. Before these can be defined, short-time analysis in terms of energy is discussed below.

Short-Time Energy

The amplitude of the speech signal varies appreciably with time, and there is a significant difference between the amplitudes of voiced segments and unvoiced segments. It is convenient to apply the short-time energy of the speech signal to extract the variations in amplitudes. The short-time energy (E_n) is defined [1]

$$E_n = \sum_{m=-\infty}^{\infty} [x(m)\omega(n-m)]^2 \tag{25}$$

where $\omega(n)$ is a window function and $x(n)$ corresponds to the speech signal. Equation (25) can be written as

$$E_n = \sum_{m=-\infty}^{\infty} x^2(m) \cdot h(n-m) \tag{26}$$

where

$$h(n) = \omega^2(n) \quad . \tag{27}$$

Equation (28) implies that the signal $x^2(n)$ is filtered by a linear filter with impulse response h(n).

In this thesis the energy is computed using a rectangular window.

$$\omega(n) = 1 \qquad 0 \leq n \leq N - 1 \qquad (28)$$
$$= 0 \qquad \text{otherwise}$$

Therefore the short-time energy of a discrete-time signal is defined as

$$E_n = \sum_{m=n-N+1}^{n} x^2(m) \qquad (29)$$

This implies that, the short-time energy at sample n is simply the sum of squares of the N samples n-N+1 through n.

It is appropriate to point out that other windows, such as Hamming window, have been used in short-time analysis. For the particular application, retangular window is used for simplicity. For linear prediction analysis, Hamming window is used. For future use, the Hamming window function is given below.

$$h(n) = 0.54 - .46 \cos (2\pi n/(N-1)), \quad 0 \leq n \leq N - 1 \qquad (30)$$
$$= 0, \qquad \text{otherwise.}$$

In the following the pre-emphasis aspects are discussed.

Pre-emphasis

For the purpose of distinguishing voiced and unvoiced speech segments, the speech signal is passed through a system that emphasizes the

frequency range that is desired. For example, low frequency emphasis can be used if the voiced segment is of concern. Similarly, the high frequency emphasis can be used if the unvoiced sound is of concern. The low and high-frequency pre-emphasis implementations are described by Figures 16 (a and b), respectively.

The low and high frequency pre-emphasized signals can respectively be given by

$$x'_{\ell}(n) = x(n) - \mu \cdot x'_{\ell}(n-1) \tag{31}$$

$$x'_{h}(n) = x(n) - \mu \cdot x(n-1) \tag{32}$$

where μ is usually referred to as the pre-emphasis factor.

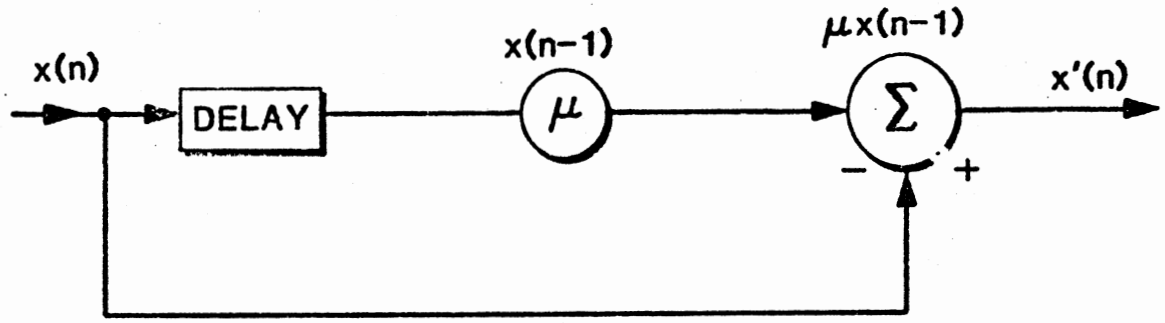
Pre-emphasis values of $\mu = -0.5, -.7$ and -1.0 have been used, and it has been found that $\mu = -1$ gives decent results and, therefore, $\mu = -1.0$ is used in this thesis.

RMS Analysis in Speech Processing

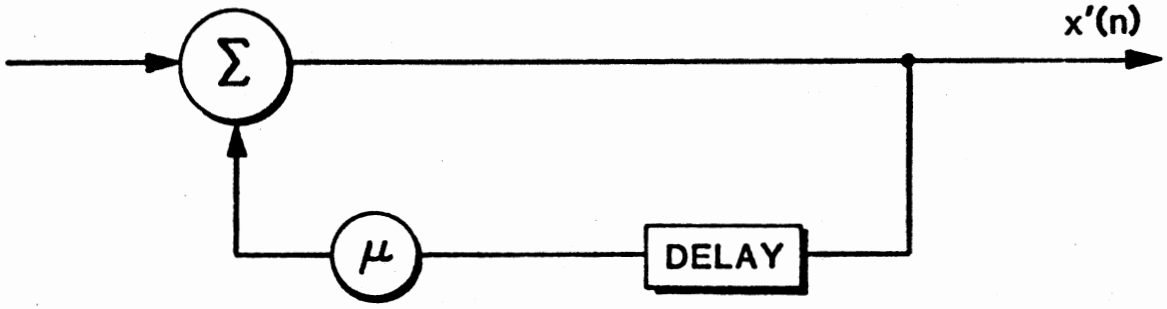
As mentioned before, the short-time energy reflects the amplitude variations of the speech signal. In fact, significant variations of energy is noticed during production of speech sounds especially vowels, semi-vowels, and voiced consonants. Useful phonetic features can be extracted from the speech signal by using the short time energy analysis. Equation (29) will be used here and is redefined here as

$$E_n = \sum_{i=1}^{NPS} x_n^2 \tag{33}$$

where NPS corresponds to the number of data points. In speech processing literature, $(RMS)_n$ is defined by



(a) HIGH-FREQUENCY BLOCK DIAGRAM



(b) LOW-FREQUENCY BLOCK DIAGRAM

Figure 16. High and Low Frequency Pre-Emphasis Block Diagram

$$(RMS)_n = \sum_{i=1}^{NPS} x_n^2/NPS \tag{34}$$

Note that (34) is strictly not root mean squared value, but it gives the energy per frame, which is a measure of RMS.

In order to extract qualitative useful phonetic features from the RMS energy, smoothing is applied. Smoothing out the undesirable ripples in the RMS energy effectively discriminates voiced segments from unvoiced segments. In order to distinguish significantly between vowels and voiced consonants and detect voiced and unvoiced segments, the RMS speech energy of the utterances must be quantized. Quantization aspects are discussed in a later section.

Double Smoothing Algorithms

In speech processing applications, measurements and processing errors can occur in the data. The data, therefore, will exhibit single- or double- point sharp discontinuities of short duration. The data could also have sharp, isolated discontinuities of very short duration due to imperfect analysis procedures. In order to eliminate all the undesirable roughness and discontinuities in the uttered speech, a suitable and appropriate smoothing algorithm must be utilized. A practical smoothing algorithm has been proposed for speech processing [32], which is a combination of median smoothing and linear filtering. Median smoothing preserves signal discontinuities if the signal has no other discontinuity within (N/2) samples.

The basic concept of a linear smoother is the separation of the signals based on their approximately non-overlapping frequency content. For non-linear smoothers it is more appropriate to consider separating signals based on whether they can be considered smooth or rough (noise-like).

Thus a signal x(n) can be considered as $x(n) = S[x(n)] + R[x(n)]$ [1] where S(x) is the smooth part of the signal x and R(x) is the rough part of the signal x. A non-linearity which is capable of separating S[x(n)] from R[x(n)] is the running median of x(n). The output of the running median smoother, $M_L[x(n)]$, is simply the median of the L numbers, x(n), ..., x(n - L + 1). Running medians of length L have desirable properties for a good smoother [32].

An ideal compromise is to use a smoothing algorithm based on a combination of running medians and linear smoothings. The running medians provide some smoothing, and the linear smoother can be of a low order system. A 3-point Hanning filter with an impulse response

$$\begin{aligned} h(n) &= 1/4 \qquad n = 0 \\ &= 1/2 \qquad = 1 \\ &= 1/4 \qquad = 2 \end{aligned} \tag{35}$$

is usually adequate, so that delays can be exactly compensated, due to the symmetry of the linear filter.

Linear smoothers are usually used in speech digital signal processing because they obey a superposition principle and they are time or shift invariant. Figure 17 shows two examples of data sequences which are to be smoothed. For case 1, a slowly varying waveform has been corrupted by a high frequency noise component. For this case a

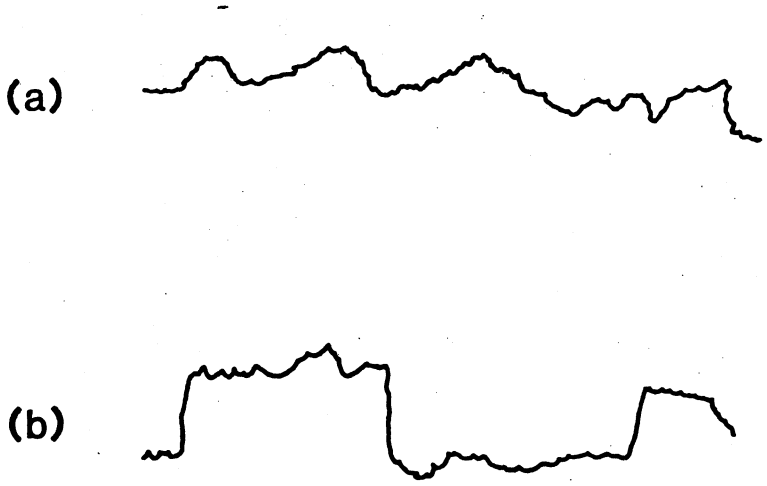


Figure 17. Two Examples of Noisy Signals to be Smoothed

linear smoother (or low-pass filter) is entirely adequate for filtering out the noise. In case 2, there is noise-like component superimposed on the signal, and the signal displays noticeable sharp discontinuities. The discontinuities here contain much high-frequency energy, and are indistinguishable from the noisy component, as far as their spectral content is concerned. A linear smoother shown in Figure 18 would therefore smear out sharp changes in the data as well as filter out the noise. For cases like the data shown in Figure 17b, a nonlinear smoother is desired, which is capable of preserving sharp discontinuities and filtering out the superimposed noise. The algorithm based on 3-point running medians is illustrated in Figure 18. The input $x(n)$ exhibits sharp discontinuities at $n = 6$ and $n = 11$. The output $y(n)$ is defined as the 3-point-median of $x(n - 1)$, $x(n)$, and $x(n + 1)$, i.e. middle value when these three inputs are ordered in value. If a median greater than 9 is used, the discontinuity would be smoothed out and $y(n)$ would be flat.

An important property of median smoothers is their ability to follow low-order polynomial trends in the data as seen in Figure 19. It is seen in the figure that a 3-point-median follows low polynomial trends, whereas a 7-point-median has smoothed out the quartic polynomial considerably.

As mentioned before, median smoothing preserves sharp discontinuities in the data, but it fails to provide sufficient smoothing of the undesirable noise-like component. An ideal solution is a smoothing algorithm based on a combination of running median and a linear smoother as shown in Figure 20. The output $y(n)$ of the simple smoother

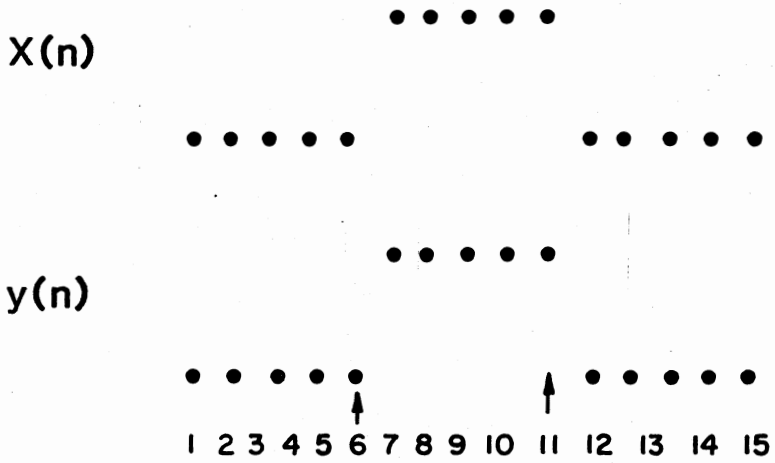


Figure 18. Median Smoothing of a Sequence with a Discontinuity

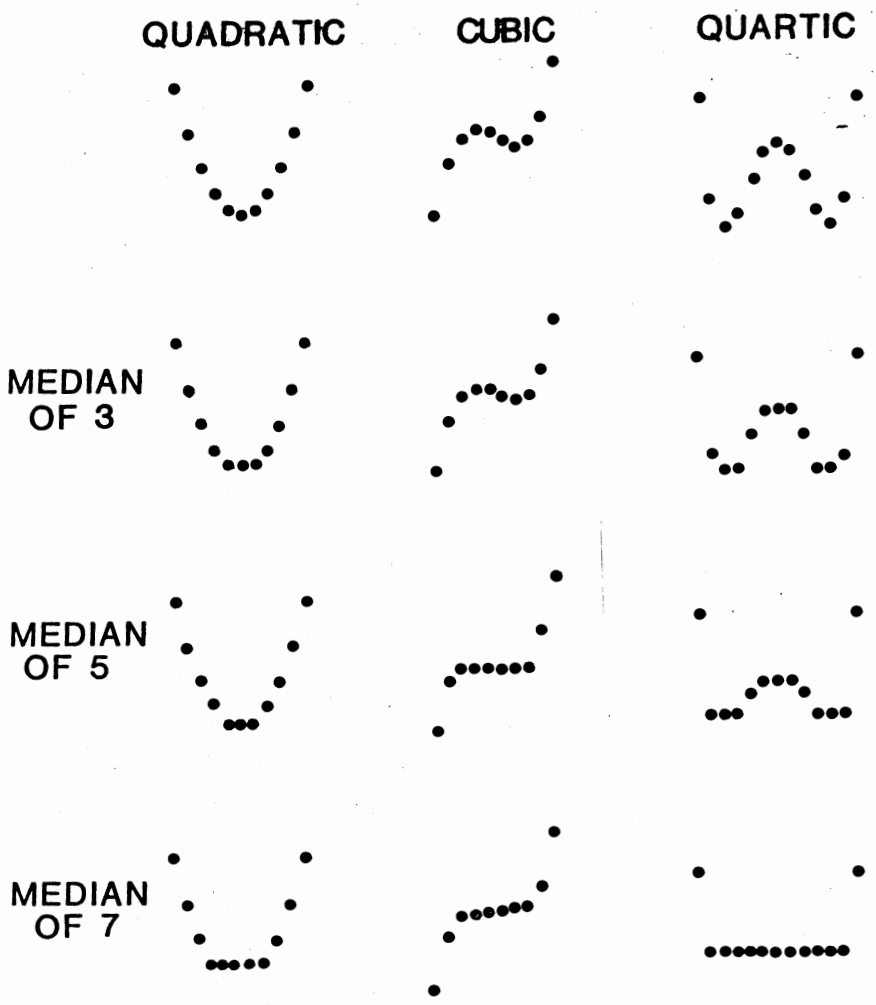


Figure 19. Effects of Various Median Smoothers on Low-Order Polynomials

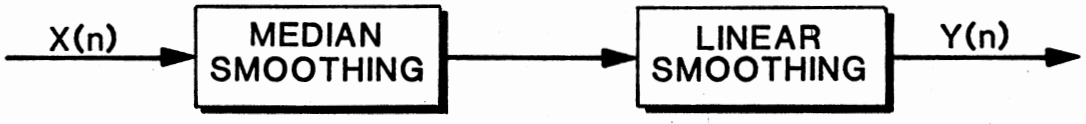


Figure 20. Simple Smoothing Algorithm

in Figure 20 is a smoothed version of x(n), say S[x(n)]. If this simple smoother is not adequate to fulfill the requirements, a double smoothing algorithm can be used as shown in Figure 21.

$$\text{since} \quad y(n) \simeq S[x(n)] \tag{36}$$

$$\text{then} \quad z(n) = x(n) - y(n) \simeq R[x(n)] \tag{37}$$

Smoothing of z(n) yields a correction term, which is added back to y(n) to give $\omega(n)$, the second approximation to S[x(n)]. Hence $\omega(n)$ satisfies the relation

$$\omega(n) \simeq S[x(n)] + S\Big[R[x(n)]\Big] \tag{38}$$

If Z(n) = R[x(n)], i.e. the smoother is ideal, then V(n), the output of the second smoother, would be identically zero, and the second-order correction would be unnecessary.

In order to implement the system shown in Figure 21, one must account for the delays in each path of the smoother and should be compensated. Median smoother has a delay of (L - 1)/2 samples, and each linear smoother has a delay proportional to the number of coefficients in the finite impulse response (FIR) filter.

For the proposed digit recognition algorithm to be discussed in a later chapter, the non-linear smoother shown in Figure 22 is used with a 3-point-median smoother in the front portion and a 5-point-median smoother in the later portion. Different sizes on median smoothers are found to give good results, as discussed below.

For example, 5-point-median has a delay of 2 samples, and a 3-point Hanning window has a delay of 1 sample. Thus the total delay of

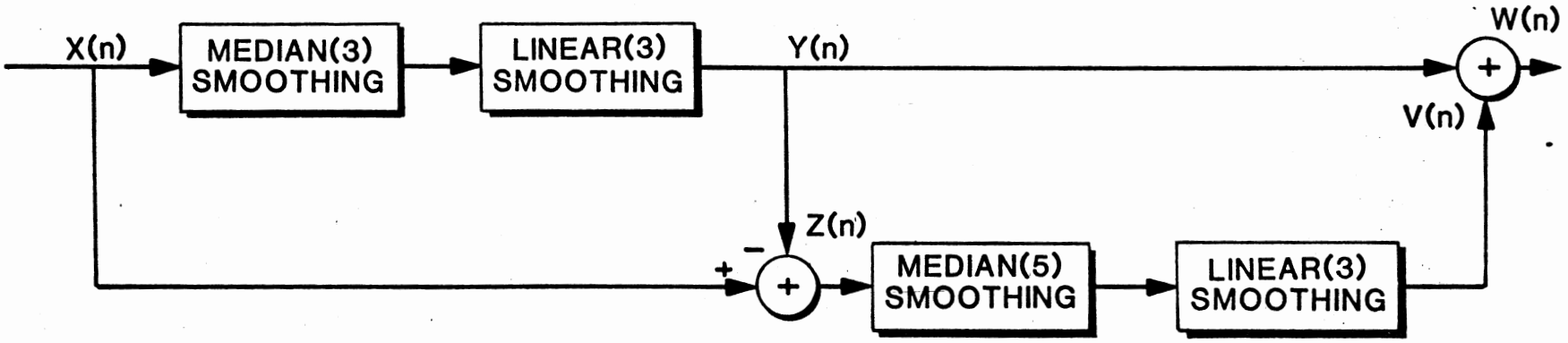


Figure 21. Double Smoothing Algorithm

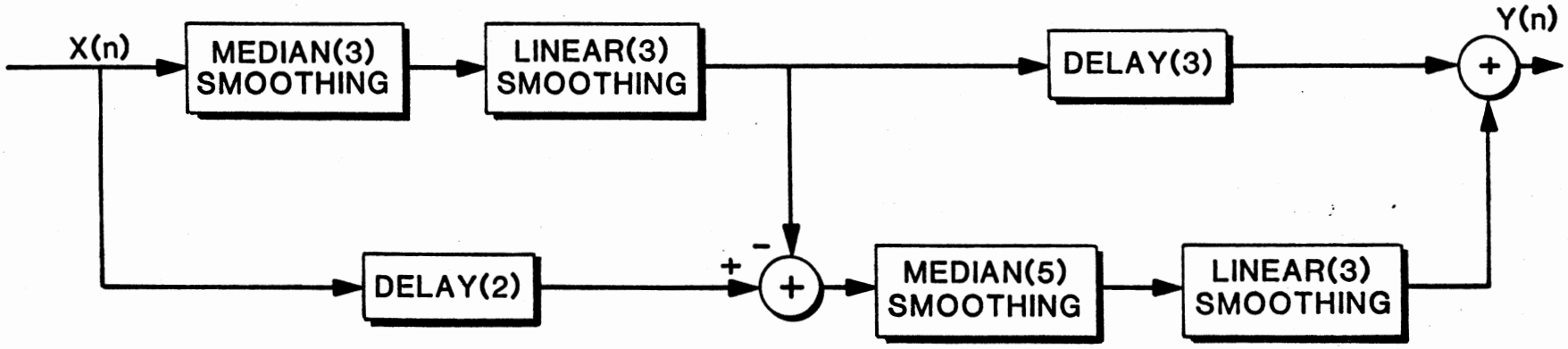


Figure 22. Nonlinear Double Smoothing Algorithm

the first smoother is 3 samples. The remaining important requirement is to implement the system of Figures 20 and 21 so that it provides an algorithm for handling the end points of the data. The effects of several versions of the smoothing algorithm on a speech intensity contour is in Figure 23. The effect of the additional smoothing obtained using higher order medians is clearly seen. Furhter, the differences between using median smoothing alone and the combination with linear smoothing are significant.

Zero Crossings

The zero-crossing rate is used in the proposed algorithm, and the end point detection algorithm. A brief review of ZCR is discussed below. The rate at which zero crossings occur is a simple measure of the frequency content of the signal. For example, a sine wave signal of frequency F_o, sampled at a rate F_s, has F_s/F_o samples per cycle of the sine wave. The long-time average rate of zero crossings is $Z = 2F_o/F_s$ samples, because each cycle has two crossings. But since speech signals are broadband signals then, rough estimates of spectral properties can be obtained using a representation based on the short time average magnitude difference function zero crossing rate. This is [1]

$$Z_n = \sum_{m=-\infty}^{\infty} \left|\text{sgn}[x(m)] - \text{sgn}[x(m-1)]\right| \omega(n-m) \qquad (39)$$

where $\text{sgn}[x(n)] = 1 \quad x(n) \geq 0$

$= -1 \quad x(n) < 0$

and $\omega(n) = 1/2N \quad 0 \leq n \leq N-1$

$= 0, \quad$ otherwise

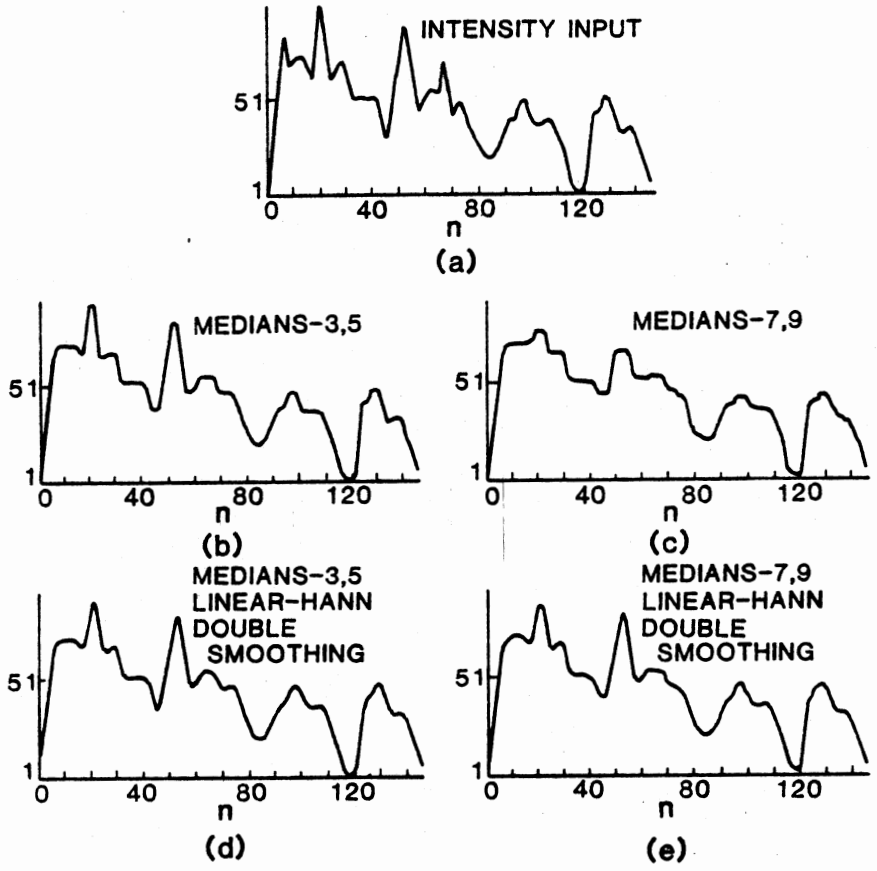


Figure 24. Effects of Several Versions of the Smoothing Algorithm on a Speech Intensity Contour

The computation of Z_n given in (39) appears to be complex. All that is required is to check samples in pairs to determine where the zero crossings occur, and then the average is computed over N consecutive samples. Since a rectangular window of finite length is used, the delay can be exactly compensated. Equation (39) can be simplified as [1].

$$Z_n = \frac{1}{2N} \sum_{m=n-N+1}^{n} |\text{sgn}[x(m)] - \text{sgn}[x(m-1)]| \qquad (40)$$

which can be computed recursively by using

$$Z_n = \left\{ Z_{n-1} + \frac{1}{2N} |\text{sgn}[x(n)] - \text{sgn}[x(n-1)]| - |\text{sgn}[x(n-N)] - \text{sgn}[x(n-N-1)]| \right\} \qquad (41)$$

Equation (41) is used in the proposed algorithm.

Cross-Correlation Function

The uniqueness of the parameters used for spoken digits is of importance in the proposed recognition scheme. The BTR, CTR, and RMS values per digit are compared respectively, using the statistical cross-correlation method discussed below [36].

When a set of independent variables are related to or are dependent upon each other, multicollinearity is said to exist among the variables. In the following, the correlation for digits i and j, are considered, where $0 \leq i, j \leq 9$. Furthermore, let t be the frame number and let there be n frames. For a frame t, x_{ti} and x_{tj} represent the RMS value for digit i and j respectively. The same analysis can be

used for the parameters BTR and CTR. Also, features of several sets of digits can be obtained and stored.

The correlation coefficients are denoted by the symbol $R_{x_{ti},x_{tj}}$ and is given by

$$R_{x_{ti},x_{tj}} = \frac{\sum_{t=1}^{n} (x_{ti}-\bar{x}_i)(x_{tj}-\bar{x}_j)}{\left[\sum_{t=1}^{n} (x_{ti}-\bar{x}_i)^2 \sum_{t=1}^{n} (x_{tj}-\bar{x}_j)^2\right]^{1/2}} \tag{42}$$

where n = max (number of frames for digit i, number of frames for digit j), and where the mean for digit i is:

$$\bar{x}_i = \frac{\sum_{t=1}^{n} x_{ti}}{n} \tag{43}$$

and the mean for digit j is:

$$\bar{x}_j = \frac{\sum_{t=1}^{n} x_{tj}}{n} \tag{44}$$

Note that the parameters will be padded by zero for the digit that has fewer number of frames.

If n = minimum (number of frames for digit i, number of frames for digit j) in (43), then the padding will not be necessary. The results will not be as good for other cases of n. These aspects will be discussed in the next chapter.

It can be shown that $R_{x_{ti},x_{tj}}$ is always between -1 and 1. A value of $R_{x_{ti},x_{tj}}$ close to 1 indicates that the independent variables x_{ti} and x_{tj} are highly related or correlated. In other words a value of $R_{x_{ti},x_{tj}}$ close to 1 denotes that x_{ti} and x_{tj} have similar patterns; that is, their first derivatives are almost the same in RMS, BTR, or CTR. Similarly, a value of $R_{x_{ti},x_{tj}}$ close to -1 indicates that x_{ti} and x_{tj} have opposite patterns; that is, their first derivatives are reversely related. A value of $R_{x_{ti},x_{tj}}$ close to 0 indicates that x_{ti} and x_{tj} are not correlated; that is, the independent variables x_{ti} and x_{tj} have no similarity in their RMS, BTR or CTR patterns.

Window Applications

Speech is a continuously time varying process as mentioned before. There must be a finite number of points that be used at any time, which requires segmentation. Since speech signals are stationary on a short-time basis, it is appropriate to consider Equation (25), which can be redefined as

$$E(n) = \sum_{m=0}^{N-1} [\omega(m)x(n-m)]^2 \tag{45}$$

where $\omega(m)$ is a weighting sequence or window which selects a segment of $x(n)$, and N is the number of samples in the window. For the simple case of $\omega(m) = 1$, $E(n)$ is the sum of the squares of the N most recent values of $x(n)$.

It is to be expected that the function $E(n)$ would display the time varying amplitude properties of the speech signal. Equation (45)

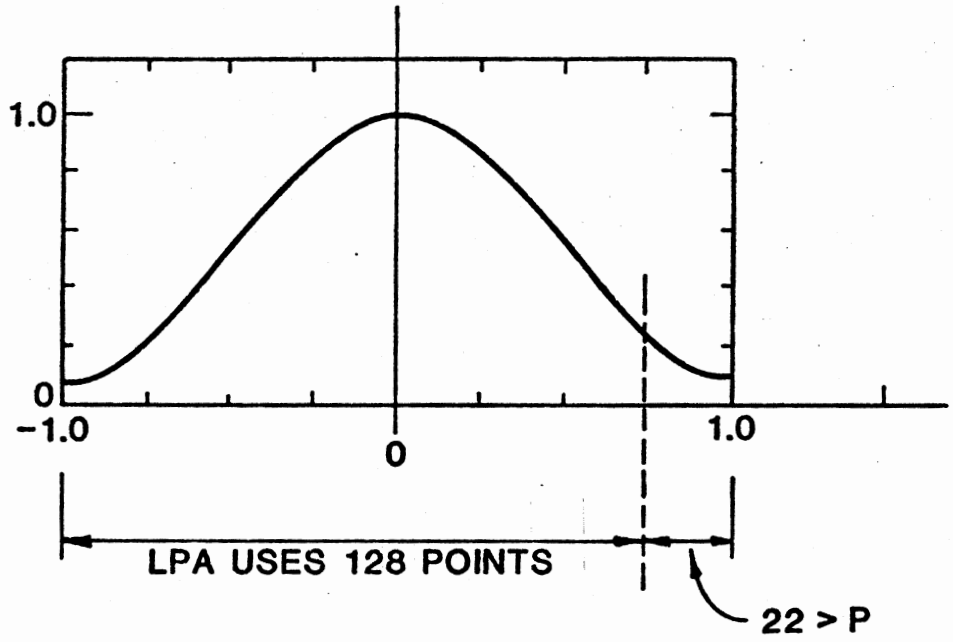
requires careful interpretation. First there is a choice of a window.

The effect of the window on the time dependent energy representation is explained by the properties of the rectangular window given by (28), and the Hamming window by (30). The rectangular window described by (29), corresponds to applying equal weight to samples in the interval (n-N+1) to n. It has been found [1], that the bandwidth of Hamming window is about twice the bandwidth of a rectangular window for the same window length. The Hamming window has lower side lobes than the comparable rectangular window.

It has already been discussed previously that the LPA using autocorrelation method computes the short-time autocorrelation function $R_n(0)$, ..., $R_n(p)$ where $R_n(k)$ is given by Equation (6) and p is the order of the filter with the limit $0 \leq k \leq p$. If LPA is applied for m data points where $0 \leq m \leq N - 1 - k$, then N becomes greater than $m + 1 + k$ which indicates that for m points, LPA window be applied for $m + 1 + k$ points in order to avoid taper effect. For example, 128 points LPA with 14th order needs about 150 points windowing. Also the sketch of Figure 24 given below, justifies the application of the autocorrelation method every 128 points, with a 150 points Hamming window. In other words, the window length N should be greater than the number of data points plus the order of the filter, for example $N \geq m + 1 + p$ and $N \geq 128 + 1 + 14 \geq 143$.

Parameters in Dip-Classification

For segmentation and end point detection, the dip-classification of an RMS energy contour is utilized. A brief review along with some



$$R_n(K) = \sum_{m=0}^{N-1-K} S_n(m)\, S_n(m+K)$$

Where $R_n(K)$ Corresponds to the Kth Autocorrelation Coefficients

Figure 24. Hamming Window

of the parameters used in the proposed algorithm are discussed below. The RMS energy is first smoothed and then normalized to maximum level of 100, to emphasize weakly pronounced voiced sounds. A plot of RMS energy versus time is obtained for given uttered digits as shown in Figure 25.

Let V_i^+ represent the first positive, V_j^- denote the 1st negative peak, or dip, and V_{i+1}^+ be the second positive peak, etc. It can be seen that there is always a minimum or dip between successive peaks.

Let V_j^- be the RMS dip value and

$$\text{Let} \quad R_1 = V_j^-/V_i^+ \tag{46}$$

$$R_2 = V_j^-/V_{i+1}^+ \tag{47}$$

Where

$$R_{min} = \text{Min } \{R_1, R_2\} \tag{48}$$

$$\text{Also Let} \quad X_1 = \log_{10} V_j^- \tag{49}$$

$$\text{and} \quad X_2 = \log_{10} R_{min} \tag{50}$$

Then the functions used for the segmentation algorithm were obtained as follows [8].

$$Z_1 = a_1 \log_{10} V_j^- + a_2 \log_{10} R_{min} \tag{51}$$

$$Z_2 = a_3 \log_{10} V_j^- + a_4 \log_{10} R_{min} \, . \tag{52}$$

The coefficients a_1, a_2, a_3, and a_4 are computed from the design samples on the basis of Fisher's method [38]. Z_1 and Z_2 are computed

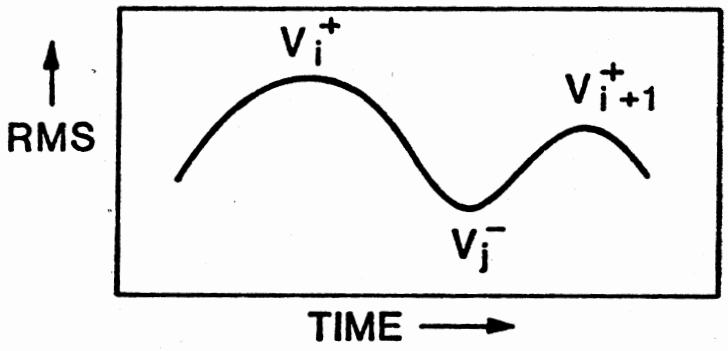


Figure 25. RMS Plot Used for the Definitions of the Extrema V_i^+, V_j^- and Their Ratios, R_1, R_2 and R_{min}

from a set of measurements of X_1 and X_2. Then,

$$Z_1 = 0.63\ X_1 + 0.77\ X_2 \tag{53}$$

$$Z_2 = 0.87\ X_1 + 0.50\ X_2 \tag{54}$$

A plot of the two parameters $\log_{10} V_j^-$ versus $\log_{10} R_{min}$ can be obtained in order to determine the values of two linearly discriminant functions, which are used to minimize the misclassifications of vowels and voiced consonants, as shown below.

$$Z_1 = 0.63\ X_1 + 0.77\ X_2 - 1.03 \quad (LDF1) \tag{55}$$

$$Z_2 = 0.87\ X_1 + 0.50\ X_2 - 0.83 \quad (LDF2) \tag{56}$$

Two other important functions are SL1 and SL2, referred to as the slicing functions, which are used to determine the level of the dip, by intersecting the slope at that point, as shown below.

$$SL1 = V_j^- + C_1\ (V_i^+ - V_j^-) \tag{57}$$

$$SL2 = V_j^- + C_2\ (V_{i+1}^+ - V_j^-) \tag{58}$$

The values of the constants C_1 and C_2 are the means of the statistical distributions of $(V_i^+ - V_j^-)$ and $(V_{i+1}^+ - V_j^-)$ as shown below.

$$SL1 = \text{Sum of } (V_i^+ - V_j^-)/N_{max}. \tag{59}$$

$$SL2 = \text{Sum of } (V_{i+1}^+ - V_j^-)/\ (N_{max}-1). \tag{60}$$

Then $$SL1 = V_j^- + 0.3\ (V_i^+ - V_j^-) \tag{61}$$

$$SL2 = V_j^- + 0.17\ (V_{i+1}^+ - V_j^-). \tag{62}$$

The two constants (-1.03) and (-0.83) in (55) and (56) are the values of the two linear discriminant functions LDF1 and LDF2 respectively as shown in Figure 26 [8]. The LDF1 is defined by taking the vowels as one class and the sonorants (nasals, liquids, and semi-vowels) as the other class. The LDF2 is defined by taking the vowels as one class and the obstruents (all the consonants except the sonorants) as the other class.

The constants (or threshold) of the LDF1 of -1.03 was determined for the design samples so as to minimize the misclassification of vowels as sonorants and to eliminate the misclassification of sonorants as vowels [8]. Likewise, the constant of the LDF2 of -0.86 was determined so as to minimize the misclassification of obstruents as vowels and to eliminate the misclassification of vowels as obstruents.

The type of dips D_1, D_2, D_3 shown in Figure 27, are classified by the sign of Z_1 and Z_2 given by Equations (55) and (56) and the slicing functions SL1 and SL2, given by (61) and (62). The identification for vowel and non-vowel procedure will be discussed in the next paragraph.

Dip-Classification

For segmentation and digit boundary detection, the dip-classification of the smoothed RMS contour is utilized. As a first step, the RMS dips and peaks are extracted from the smoothed RMS function and stored. If the logarithm of a peak value ($\log_{10} V_i^+$) is smaller than

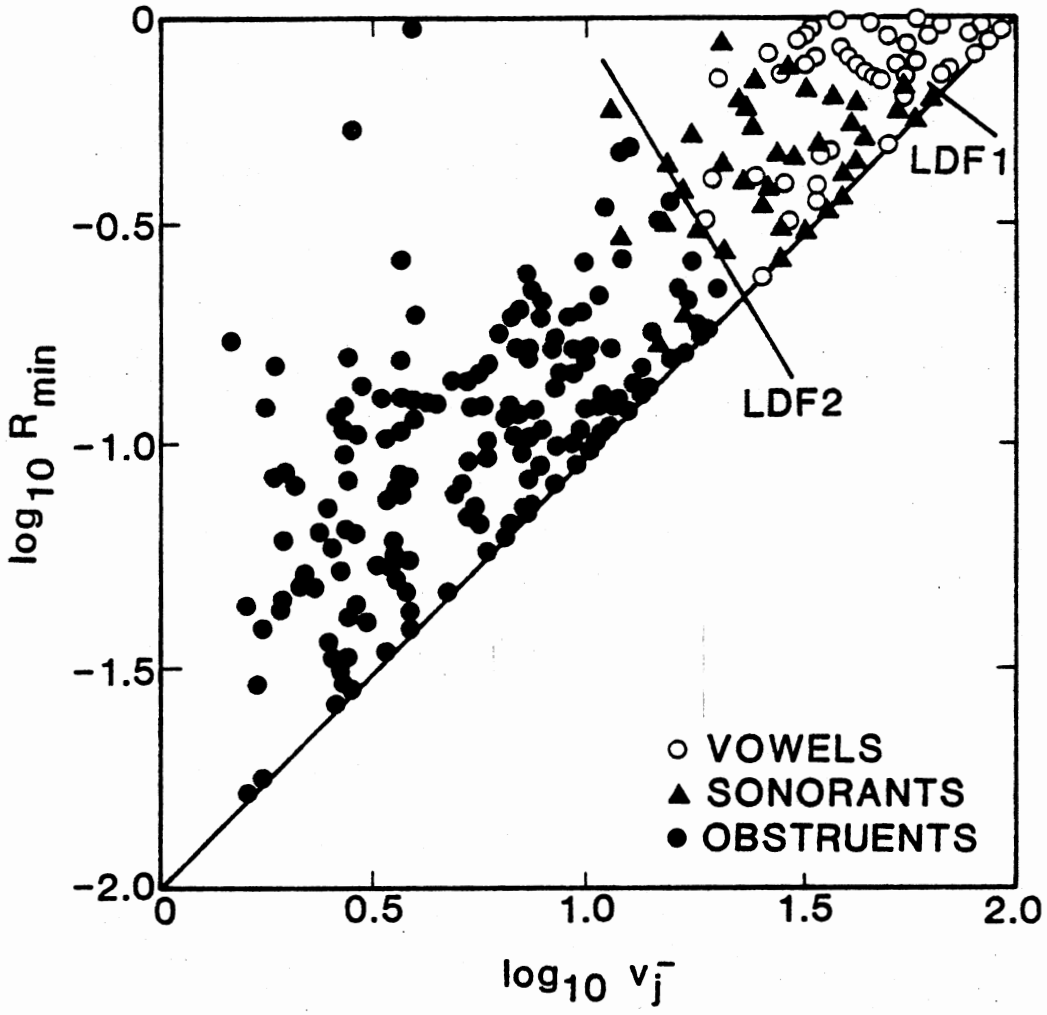


Figure 26. Plot of $Log_{10} R_{min}$ vs $Log_{10}\bar{V}_j$ for Estimating the Values of Linear Discriminant Functions LDF1 and LDF2

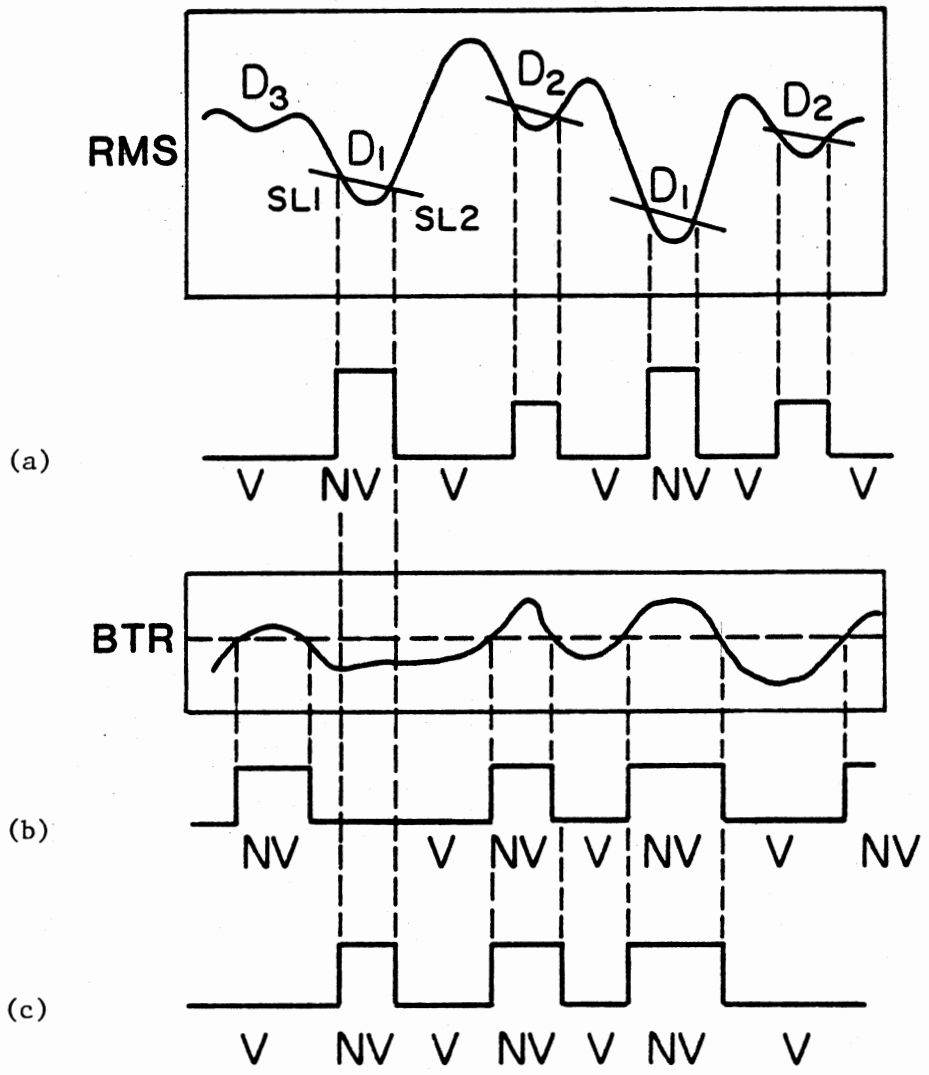


Figure 27. Schematic Diagram Showing the Types of Dips, D_1, D_2 and D_3, RMS, and BTR, Vowel, Non-Vowel Primary Decision and Combines Decision

1.0 or if the SFBR value is larger than a threshold of 4.0 around the peak, that peak and the following dip are not considered, since logarithmic RMS peak values less than 1.0 rarely occur for vowels, and the two functions Z_1 and Z_2 are not computed. Accordingly if the important constraint is met, i.e. if $\log_{10} V_i^+$ is greater than 1.0 and the SFBR value is smaller than a threshold of 4.0, then the functions Z_1 and Z_2 are computed. Hence whenever a dip is detected as a significant dip, that dip is classified into one of the three types of dips, dip 1 (D_1), dip 2 (D_2), or dip 3 (D_3) according to the sign of Z_1 and Z_2. Therefore the dip is an element of dip 1 (D_1) if both Z_1 and $Z_2 \leq 0$. The dip is an element of dip 2 (D_2) if $Z_1 \leq 0$ and $Z_2 > 0$. Finally the dip is an element of dip 3 (D_3) if both Z_1 and Z_2 are greater than zero. In summary,

$$\text{dip} \in D_1 \quad \text{if} \quad Z_1 \leq 0 \text{ and } Z_2 \leq 0$$

$$\text{dip} \in D_2 \quad \text{if} \quad Z_1 \leq 0 \text{ and } Z_2 > 0$$

$$\text{dip} \in D_3 \quad \text{if} \quad Z_1 > 0 \text{ and } Z_2 > 0$$

Figure 27 illustrates the three types of dips, D_1, D_2 and D_3. Dip 1 indicates non-vowel-like dip, dip 2 is an ambiguous dip, and dip 3 indicates a vowel-like dip.

A vowel, vowel-like and non-vowel decision is made from the RMS energy dips as shown in Figure 27(a). Another decision of vowel and non-vowel only is based on the BTR plot threshold, as shown in Figure 27(b). A final "OR" decision is obtained based on the decision obtained from the RMS and BTR contour, as shown in Figure 27(c).

Scaling and Normalization

The parameters of digit signal segments can be more robust and less variable by normalization or scaling techniques. Applying normalization techniques requires certain rules to represent a digit signal sound faithfully. Improved signal representation through appropriate scaling may speed up the overall recognition process. These techniques include, energy, amplitude and time normalization.

Amplitude Normalization

A practical form of amplitude normalization is achieved by linearly quantizing the RMS energy into 100 levels, assigning 100 to the maximum value within each utterance.

Maximum of Frames Normalization

The aim of this method is to align the time of occurence of the unknown digit utterance events to see how they match. In order to find best similarities between the unknown pattern and the referenced pattern, the maximum number of frames of a given set of digits is used. Digits having lower number of frames are padded with zero.

Scaling (Energy Normalization)

Scaling is a form of energy normalization in which each point of the incoming data is multiplied by a factor to fix the mean. This factor is derived from the inverse of the mean of the signal as shown below.

Let M_α denote the mean value of the signal, then the new value of the signal is given by

$$\text{New value} = (\text{signal level}) \left(\frac{\text{constant}}{M_\alpha}\right)$$

$$M_\alpha = \frac{\text{sum of square values}}{\text{no. of points}}$$

This technique tends to increase the peak value considerably whenever wrong end point detection results in a greater number of points than the number of points within the correct digit boundaries. Therefore, amplitude normalization is adopted.

CHAPTER IV

ACOUSTIC PHONEMIC DIGIT RECOGNITION SCHEME FOR DIGITS SPOKEN IN AMERICAN ENGLISH

Introduction

Phonemic digit recognition is one of the steps towards simplifying communications between man and machine. It is the process whereby an operator can use spoken digit commands that can be recognized by a phonemic digit recognition system. Generally man's communication with machines has been structured according to the operational requirements of the machine. Learning the "language" of the machine and manipulation of special dials or keys in the proper sequence and format is required to communicate with machines. Any deviation from this unnatural machine language can produce errors which are not easily detectable because of the complexities of the rules for proper communication between man and machine.

The main objective is to develop a phonemic digit recognition system which makes it simple for humans to "talk" directly to a machine, without any intermediate keying or handwritten steps. The operator would provide instructions in his natural language, using digit commands to control mechanical systems such as entering and

leaving restricted areas, postal zip codes, banking, inventory, etc. If connected digit is to be recognized, a suitable method of segmentation into recognizable units is required. Since phonemic digit recognition is the goal of this study, a highly accurate method of segmentation, using acoustical and phonetic feature parameters is required. The machine should be able to correlate and recognize the discrete acoustical waveform by segmenting the waveform into a sequence of elements so that the spoken phonemes can be localized and the end points are detected and located. In the following, a brief discussion on the segmentation is given.

Segmentation

Methods for segmentation of isolated and connected digits into phonemic units have typically utilized information pertaining to rapid changes in the energy contour of the given utterance. Rapid changes in speech parameters, such as energy and pole frequency derived from two-pole LPA model is used for detecting voiced segments, because these parameters have high value for vowels and develop a dip for vowel-like consonants. Since the ZCR rate and the normalized error show high values for unvoiced phonemes, and low values for voiced phonemes, they are utilized together to detect unvoiced segments. The ZCR rate and the energy signal are used for end point detection of the spoken digit. In fact the problem of accurately locating the beginning and end of an utterance is actually a special case of the more general problem of labeling an interval of a signal as silence, unvoiced, or voiced. If there is a perfect technique for this three level decision, the end

point-location problem would be solved. However, such an ideal algorithm does not exist yet. Therefore, it is very important to develop an appropriate algorithm for end-point detection of connected digits spoken in any environment. Consequently, it is considered worthwhile to consider dip-classification algorithm for locating digit boundaries. The dip-classification scheme was discussed in the previous chapter.

Recognition Scheme

Most digit recognition systems, known up till now, lack the usage of the natural phonetic features of the spoken digit. As discussed in Chapter I, some recognition procedures are based on finding the spectrum energy for both vowels and voiced consonants. The application of more than one band-pass filter in the above mentioned system causes permanent loss of useful phonetic and acoustic features and some useful information relating to the transition regions and formant peak locations. The above system is not efficient as the computational requirements are severe. Rabiner and Sambur [4][12] improved the efficiency of this system by using silence, voiced and unvoiced segment detection based on energy and pole frequency, ZCR and normalized error measurements. But this scheme lacks the accuracy of detecting digit boundaries and has the disadvantage of training the system. In addition the system must know the number of spoken digits in order to estimate the number of boundaries to be located. Also, the speaker must be trained, and the best uttered digit is used for the recognition scheme. Furthermore, a 2-pole LPA model can't extract the normalized error that eliminates the higher order formants, and a model is needed to account for f_2, f_3, f_4, etc. Another disadvantage is the

artificial introduction of silence at the beginning and end of the connected digits, which limits the usefulness of the system in cases where the machine cannot locate digit boundaries, due to the miscalculation of the ZCR threshold. Hence misclassification and wrong recognition may result.

In order to solve some of the previous mentioned digit recognition system problems, and minimize the computational time, a suitable method of segmenting the spoken digits into recognizable phonetic units based on area functions and RMS energy contour is utilized. Since the RMS energy depends on the amplitude of the signal, an appropriate scaling or normalization is needed prior to parameter measurements. Scaling and normalization is discussed in Chapter II.

Digit Recognition Flow Chart

The digit recognition scheme for digits in English is shown in Figure 28. The analog data is first low-pass filtered with a cut-off frequency of 4 KHz and sampled at 8000 Hz/sec. The first step in the digit recognition scheme is the end-point detection [9][12]. Following the flow chart, one can see that two different ideas are used. The first one is based on the RMS energy peak ratio and threshold to detect the uttered digit. The second one depends on the BTR and SFBR parameters derived from area function of the vocal tract, via LPA. Before implementing the first idea, the incoming digit signal is segmented into 128 points per frame using a rectangular window, because it is feasible to assume that the vocal tract shape will remain unchanged within this short frame or segment.

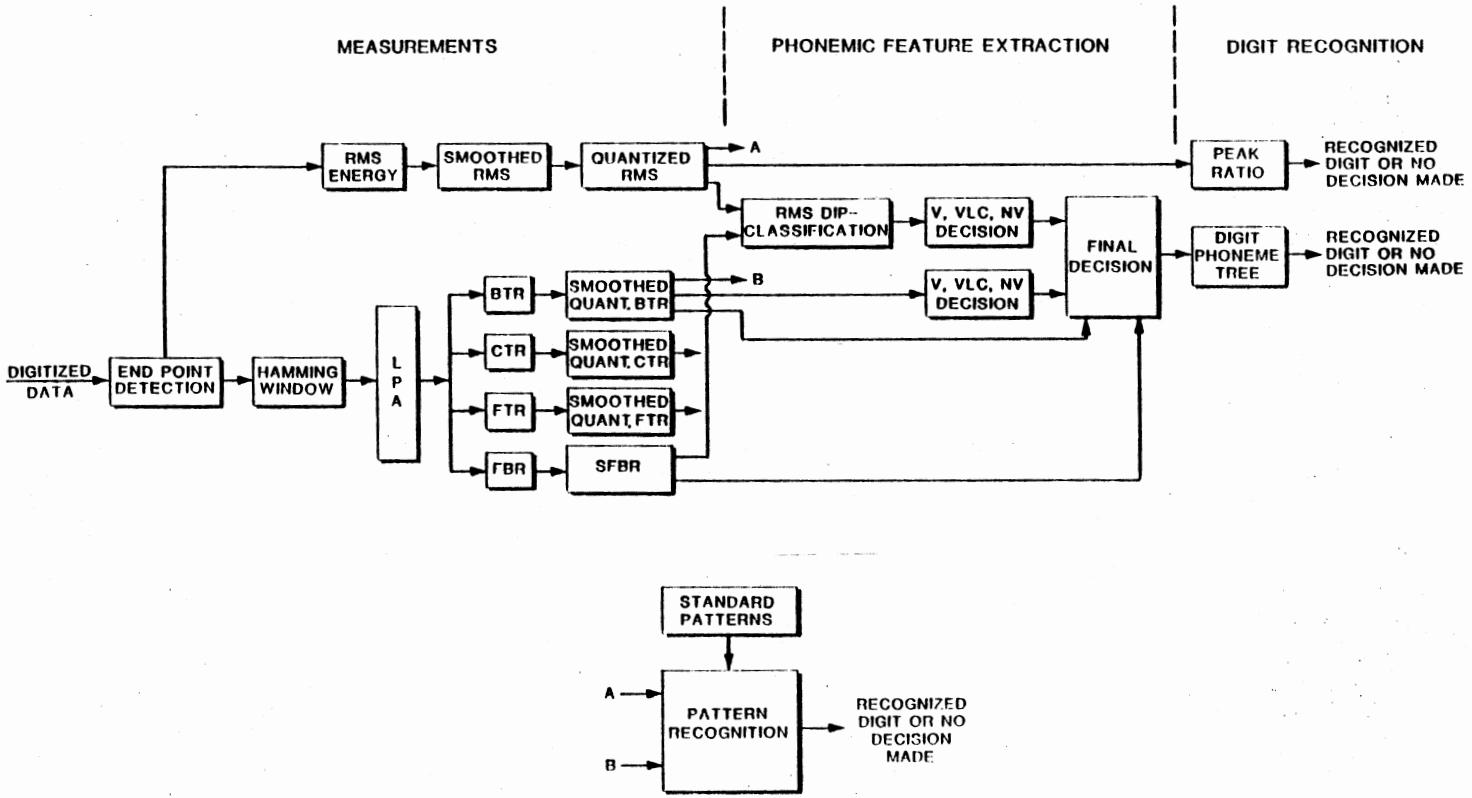


Figure 28. Digit Recognition Flow Chart Based on a Phonemic Feature Detection

From Figure 28, it can be seen that there are three main parts in the overall scheme of digit recognition. These are measurements, phonetic feature detection and, finally, the digit recognition itself. In the first part corresponding to the first idea, the RMS energy, smoothed RMS energy and quantized RMS energy are computed per frame. With a frame length of 128 points, rectangular window is used here for frame segmentation. Two sets of representative smoothed and quantized RMS plots for digits zero through nine in English are given in Figures 29-48 respectively. From these plots, the two largest peaks are obtained and are given in Tables V and VI for ten digits. Also, peak-to-peak ratios of two largest peaks are computed. It is assumed that this ratio is always less than one. From Table V, it can be seen that for digits four, six and eight, there is only one peak and therefore, the ratios are not given. However, from Table VI, there are two peaks for the digits four, indicating an inconsistency.

The range for the threshold values for the largest peak (P_1,P_2 in Tables V and VI) and the ratio for the two largest peaks have been established from previous measurements. These ranges are given in Tables V and VI. First, the measured ratio is compared with the established range. If it does not match with any given range, it will compare with the largest peak range. If it cannot match this range either, then the first idea based on RMS did not work. From the measurements, it has been found that RMS method gives the actual digit about half the time. The other half of the time, it indicates that no decision can be made using RMS plots. This completes the first idea in the digit recognition scheme.

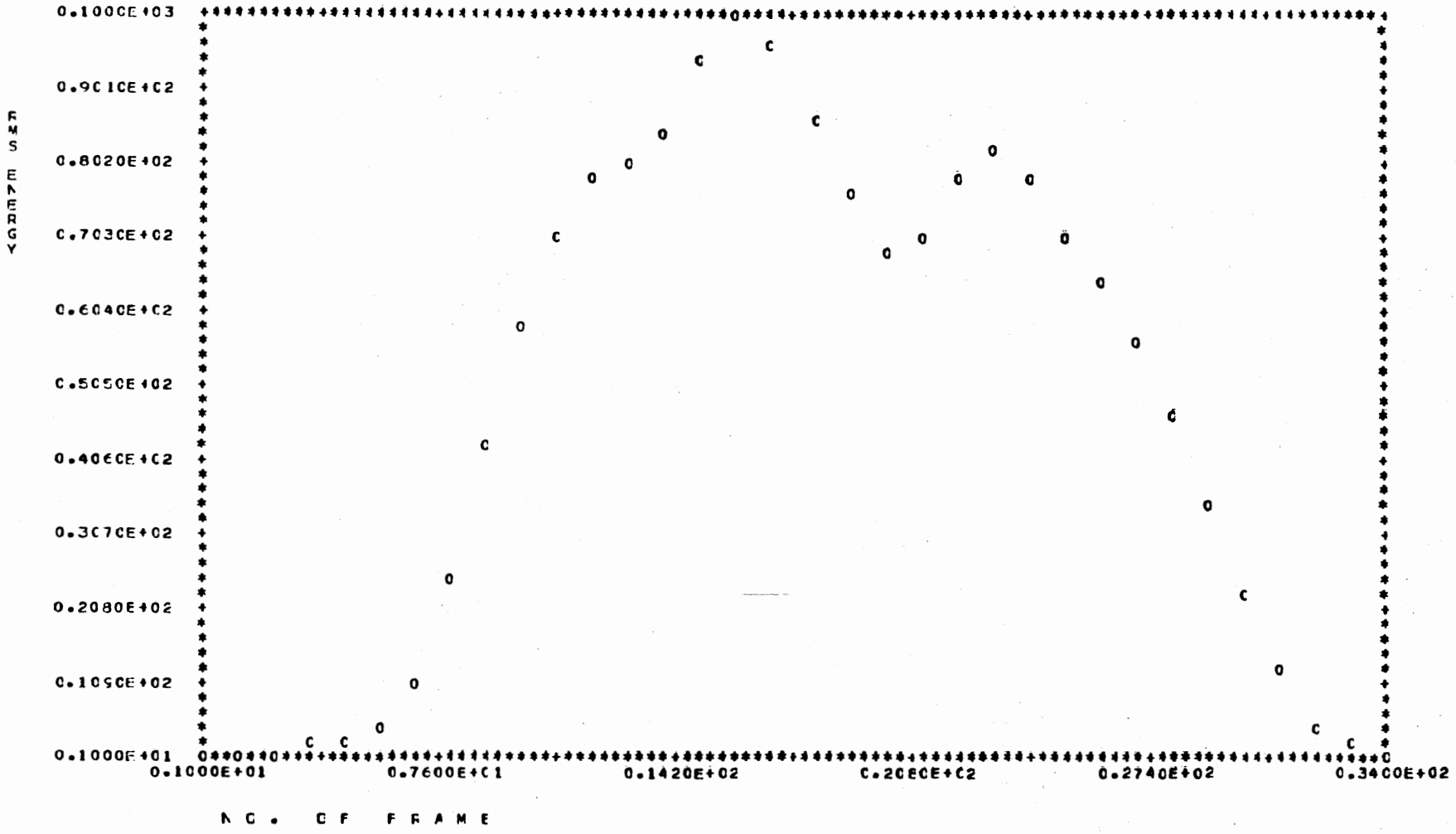


Figure 29. Smoothed RMS Energy Contour for Digit Zero, i.e. /zIro/ Spoken in American English

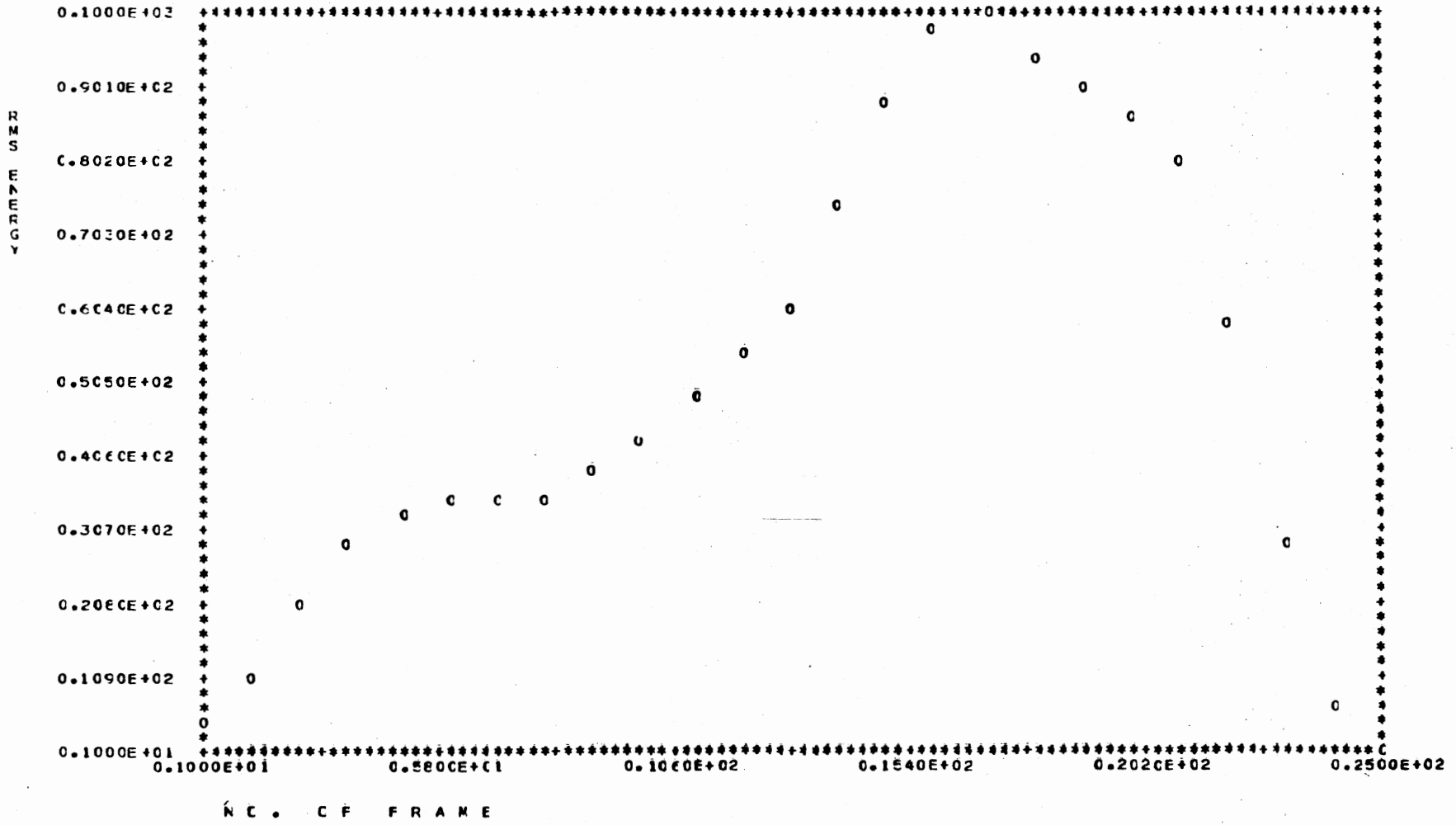


Figure 30. Smoothed RMS Energy Contour for Digit One, i.e. /wʌn/ Spoken in American English

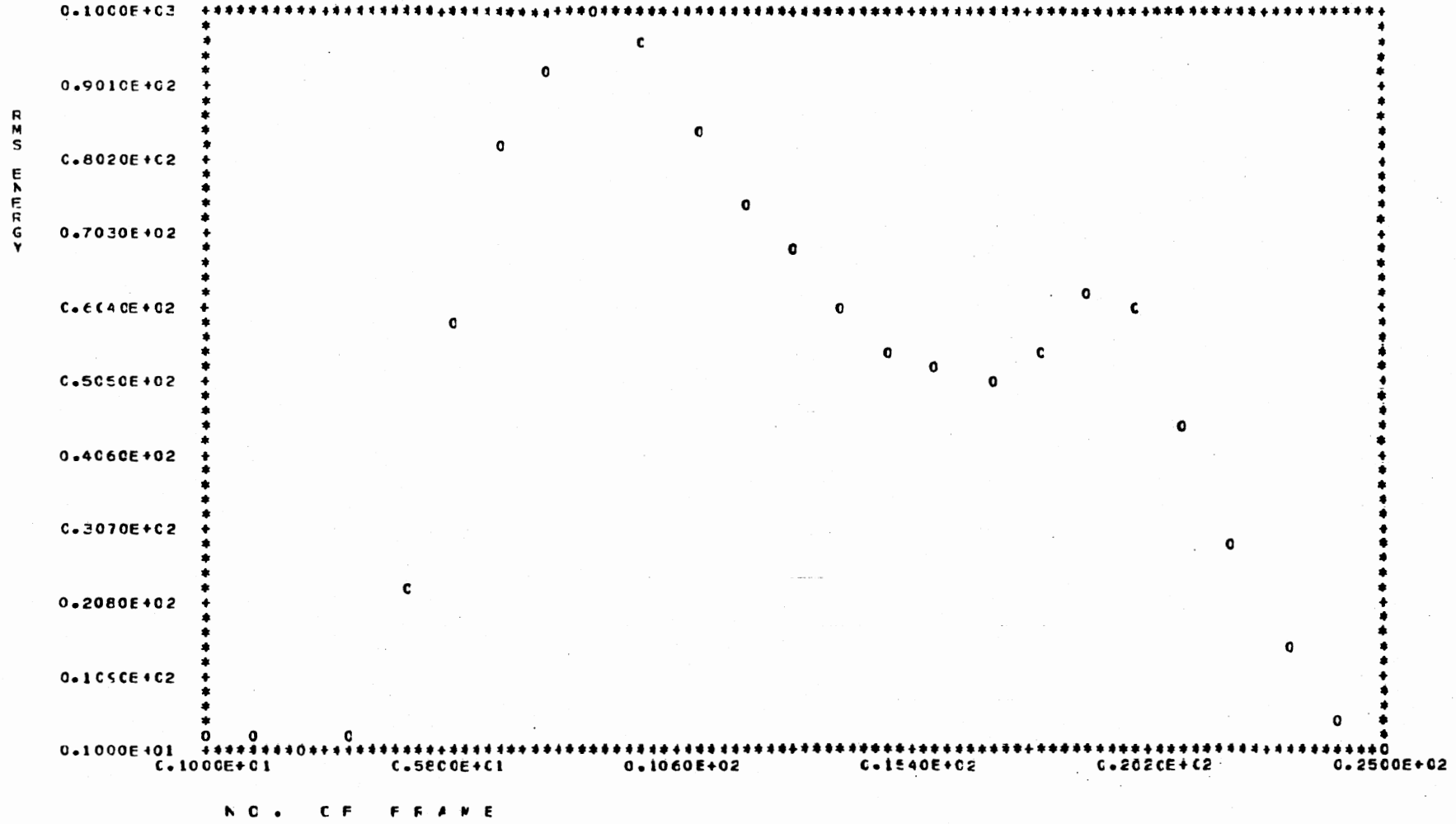


Figure 31. Smoothed RMS Energy Contour for Digit Two, i.e. /tu/ Spoken in American English

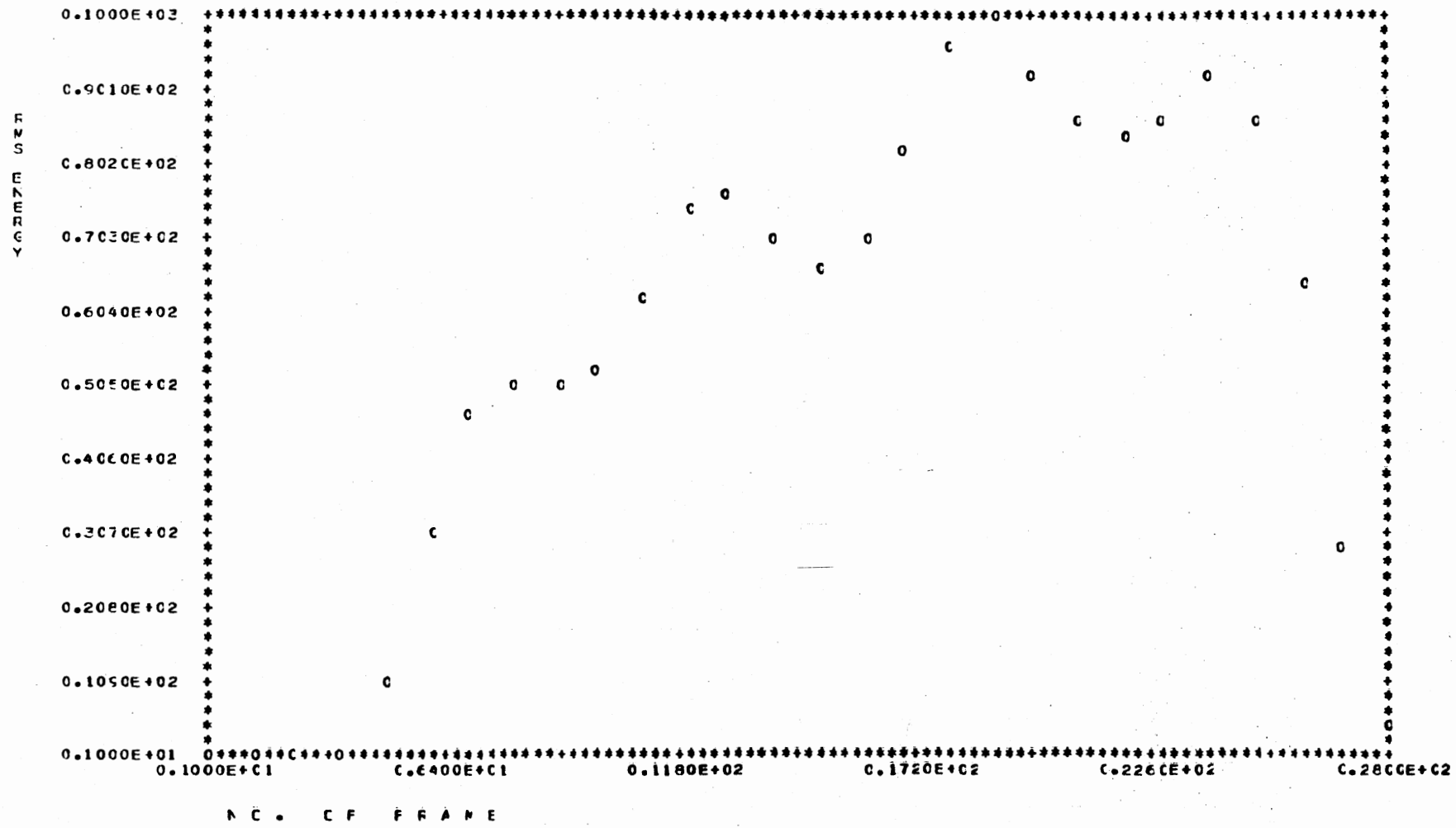


Figure 32. Smoothed RMS Energy Contour for Digit Three, i.e. /θri/ Spoken in American English

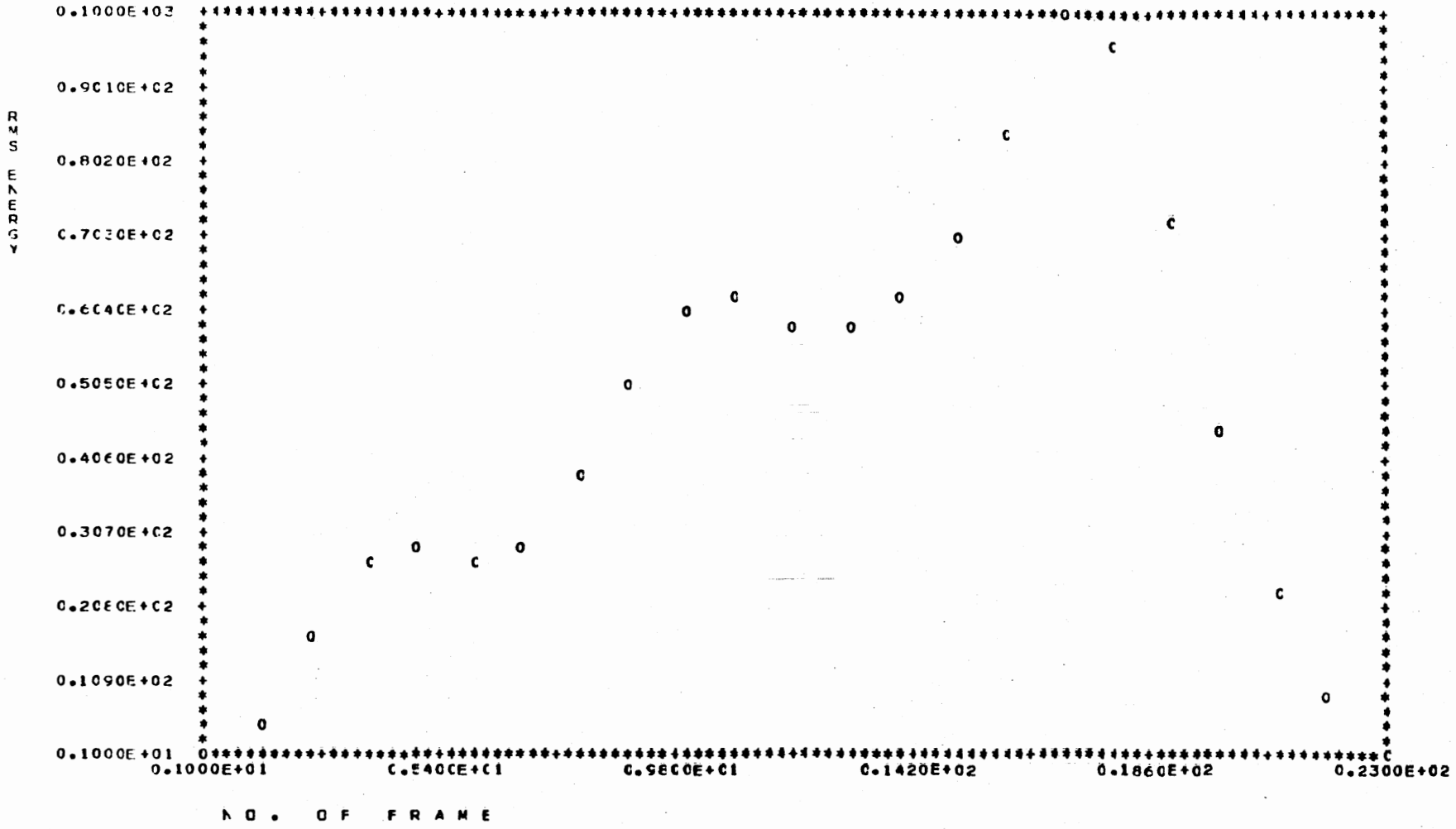


Figure 33. Smoothed RMS Energy Contour for Digit Four, i.e. /fɔr/ Spoken in American English

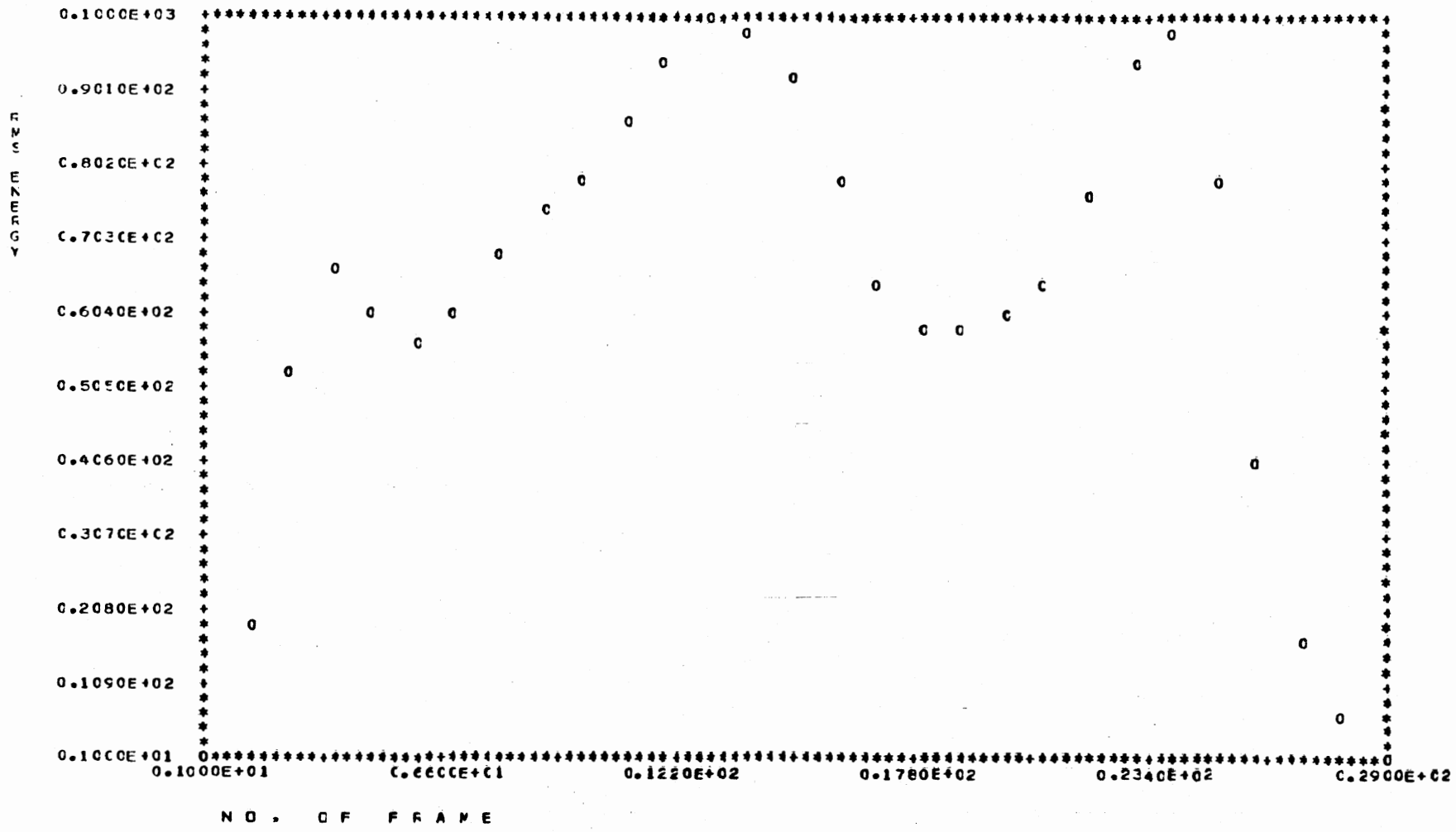


Figure 34. Smoothed RMS Energy Contour for Digit Five, i.e. /faIv/ Spoken in American English

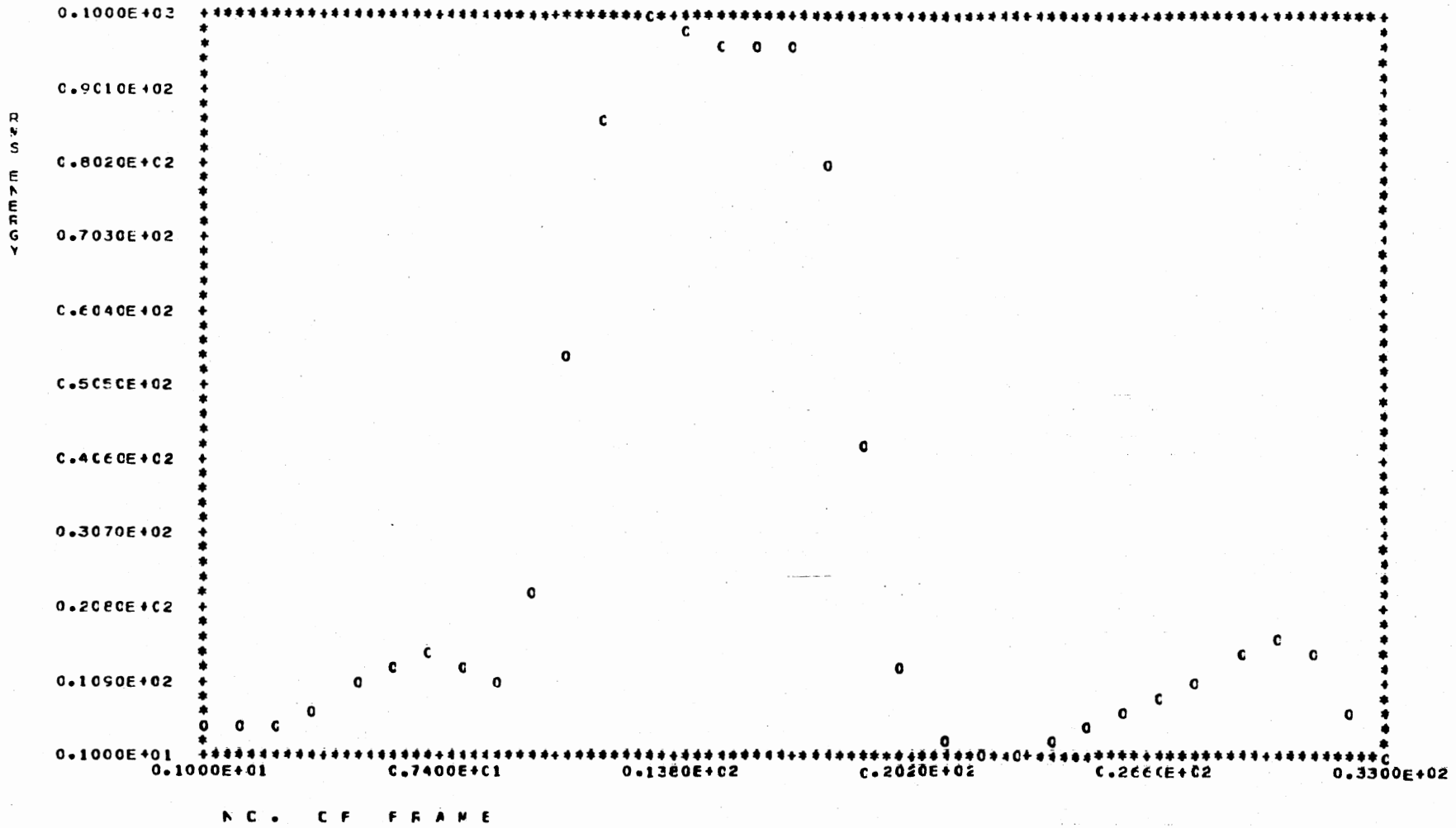


Figure 35. Smoothed RMS Energy Contour for Digit Six, i.e. /sIks/ Spoken in American English

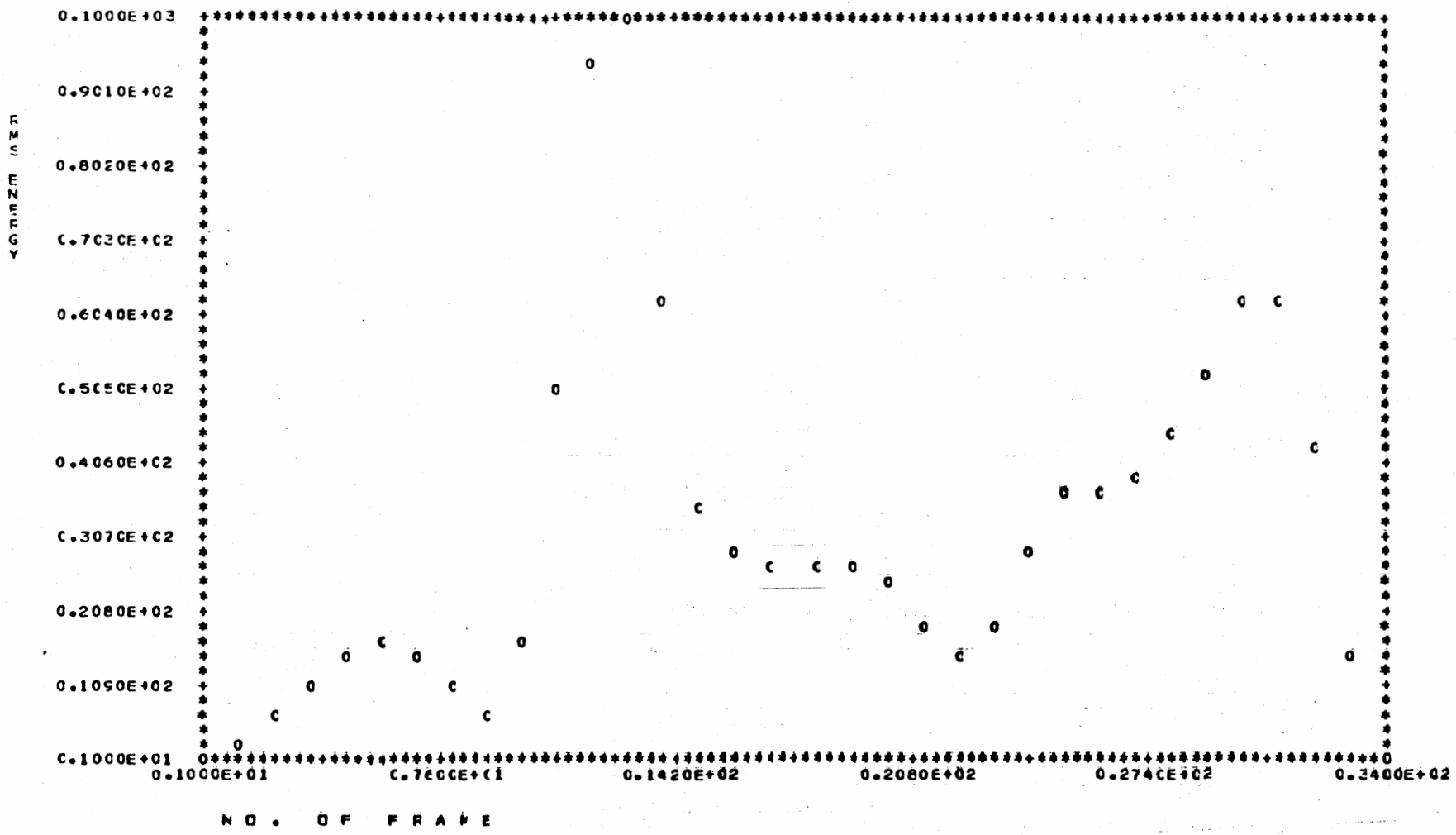


Figure 36. Smoothed RMS Energy Contour for Digit Seven, i.e. /seven/ Spoken in American English

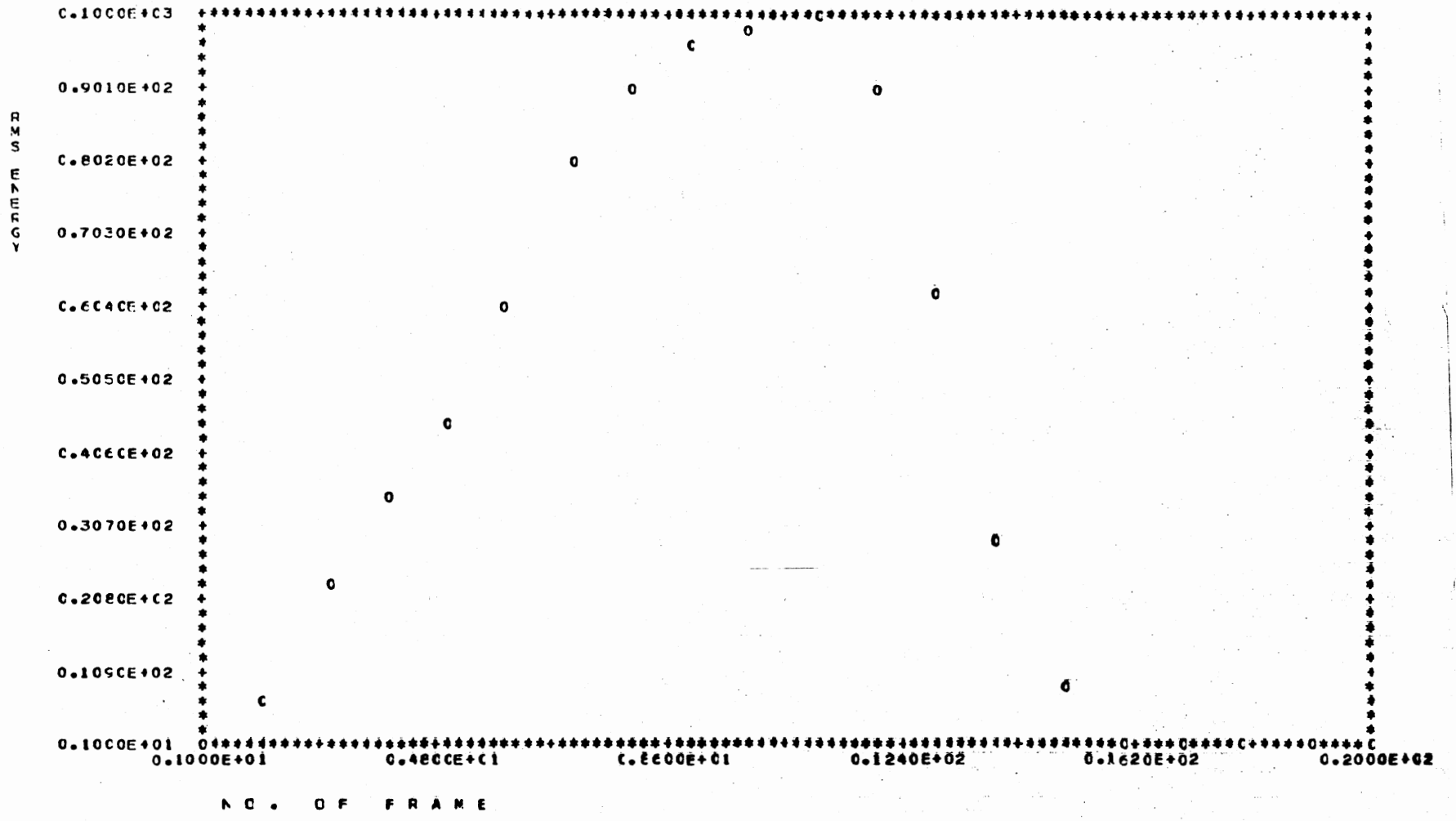


Figure 37. Smoothed RMS Energy Contour for Digit Eight, i.e. /eIt/ Spoken in American English

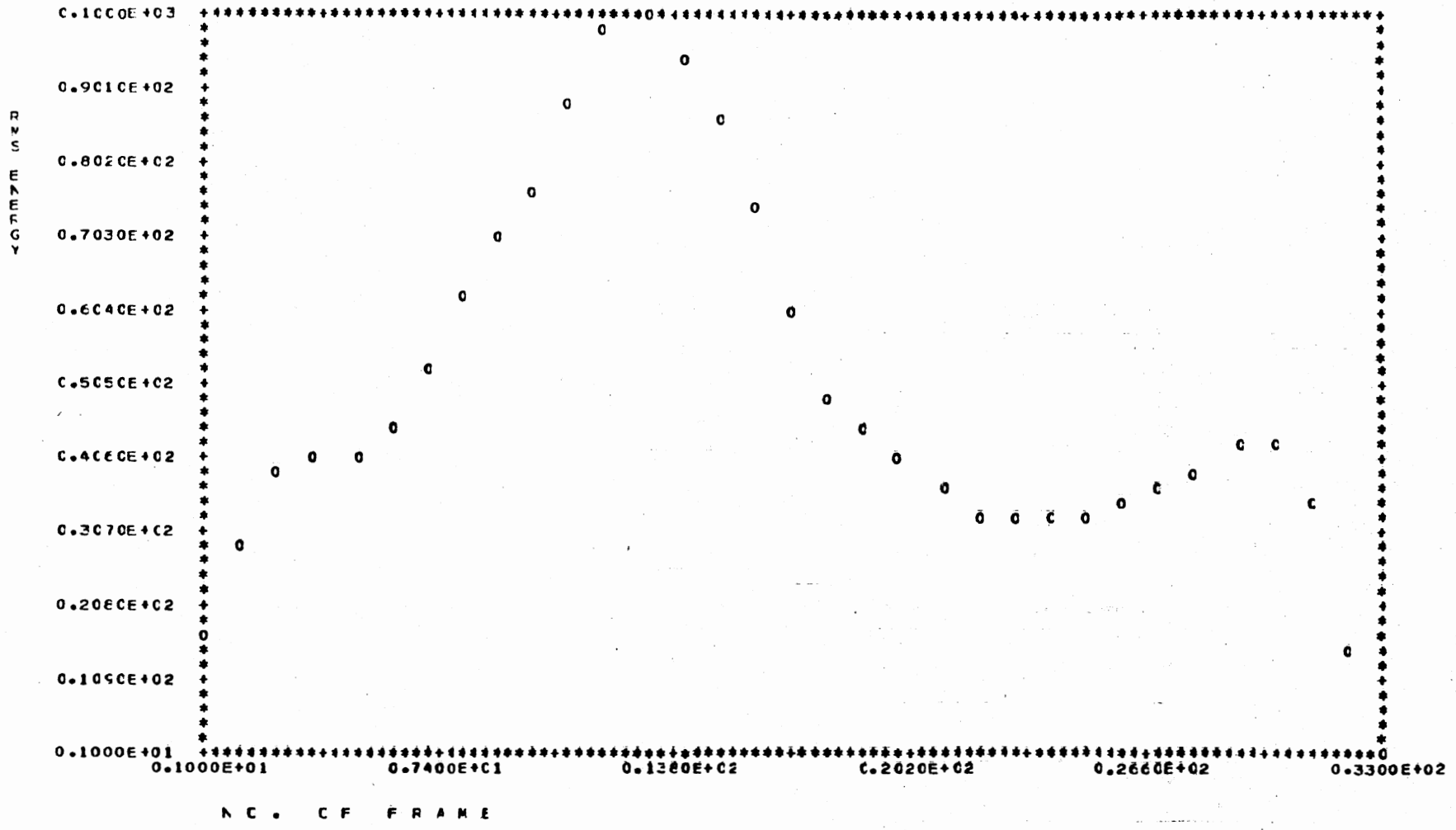


Figure 38. Smoothed RMS Energy Contour for Digit Nine, i.e. /naIn/ Spoken in American English

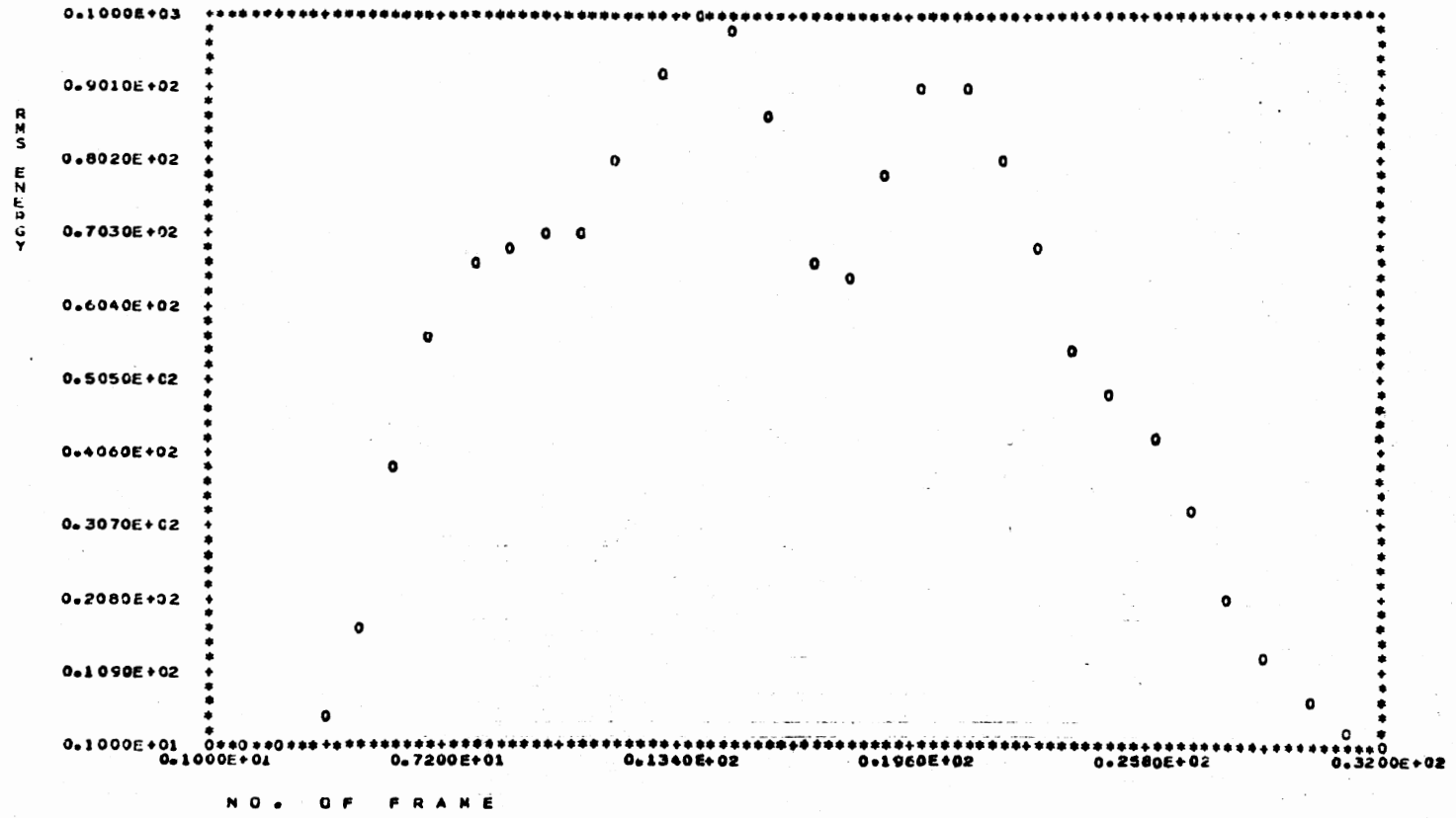


Figure 39. Smoothed and Quantized RMS Energy Contour for Digit Zero, i.e. /zIro/ Spoken by International Speaker

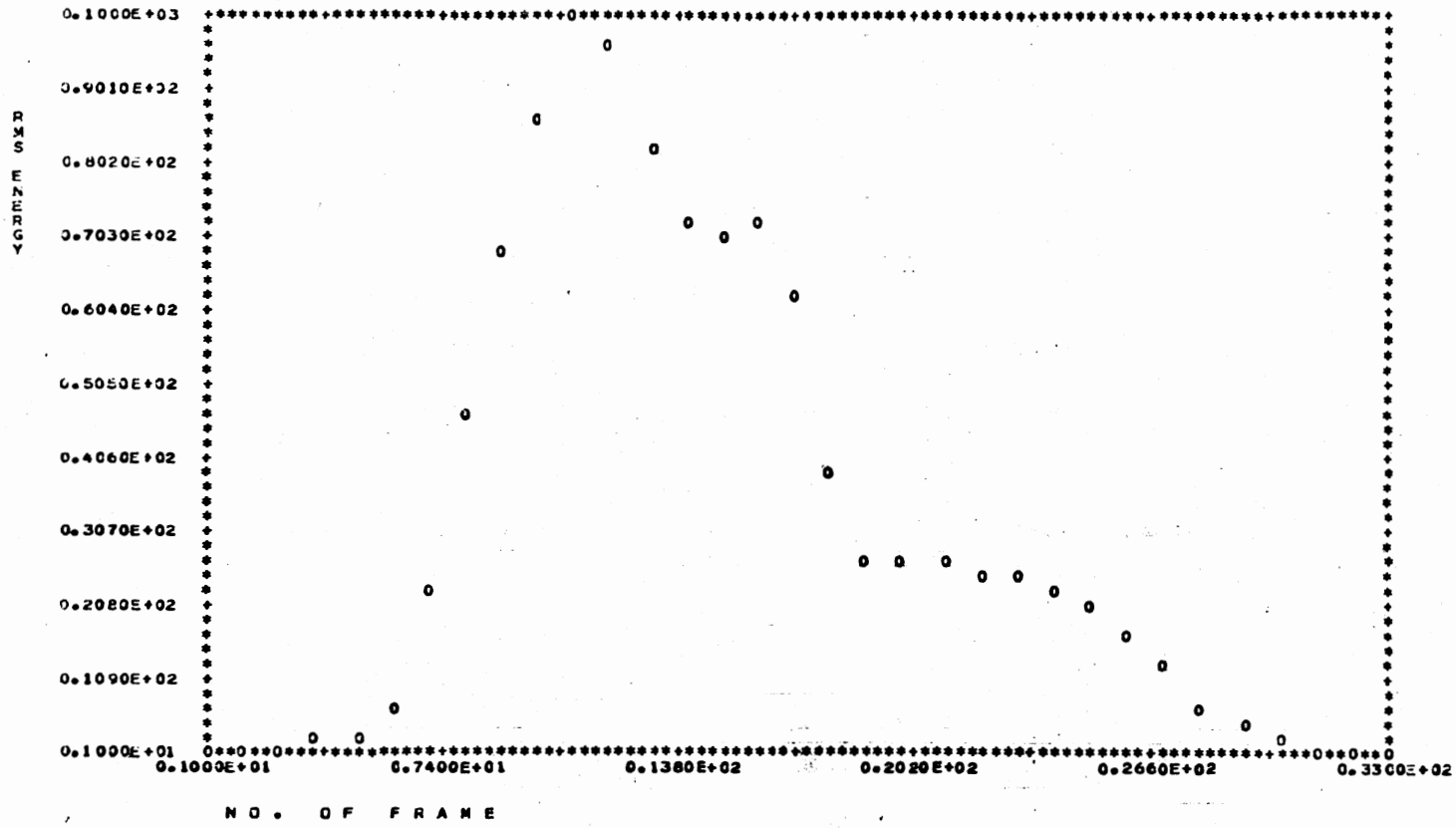


Figure 40. Smoothed and Quantized RMS Energy Contour for Digit One, i.e. /wʌn/ Spoken by International Speaker

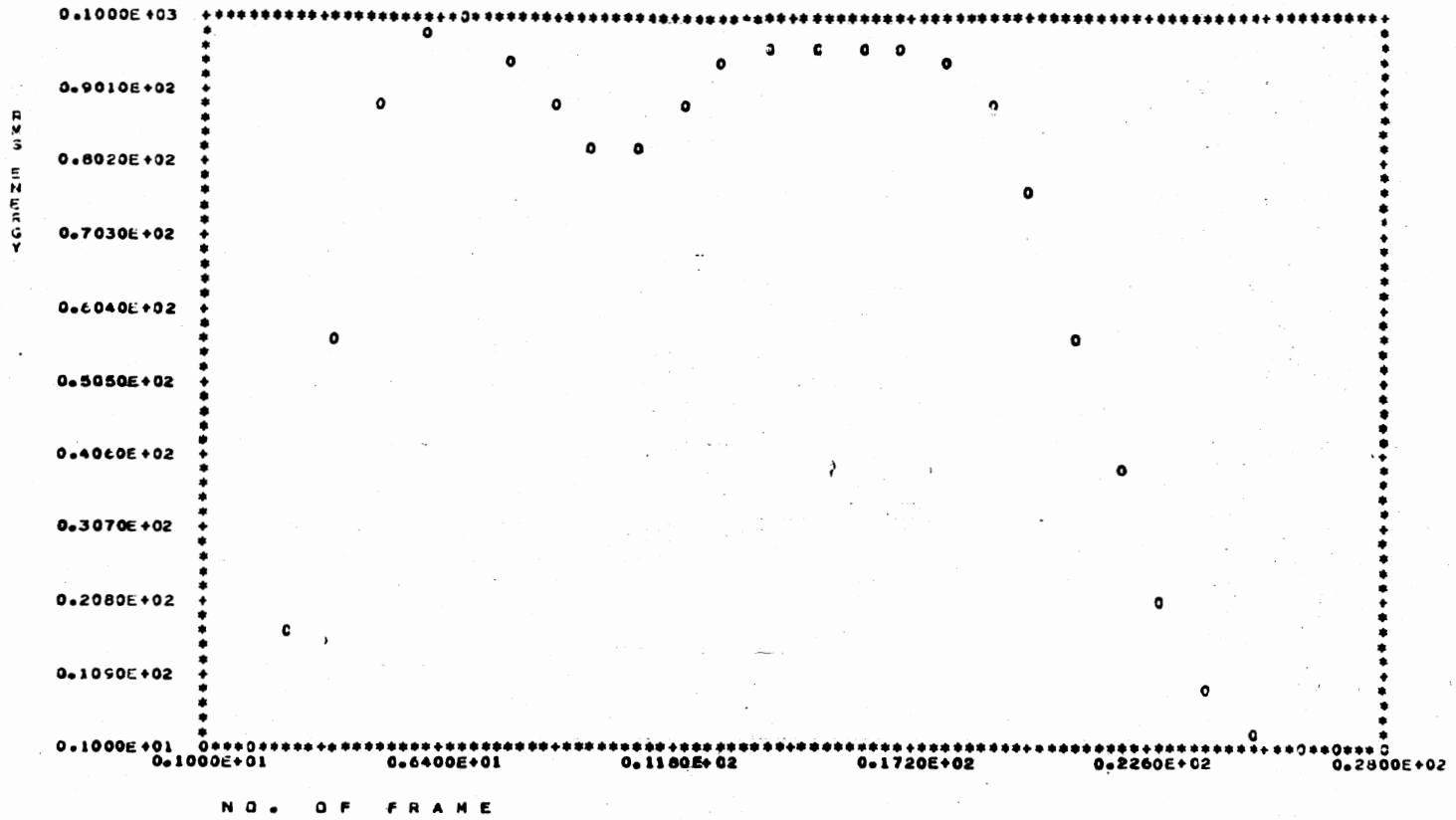


Figure 41. Smoothed and Quantized RMS Energy Contour for Digit Two, i.e. /tu/ Spoken by International Speaker

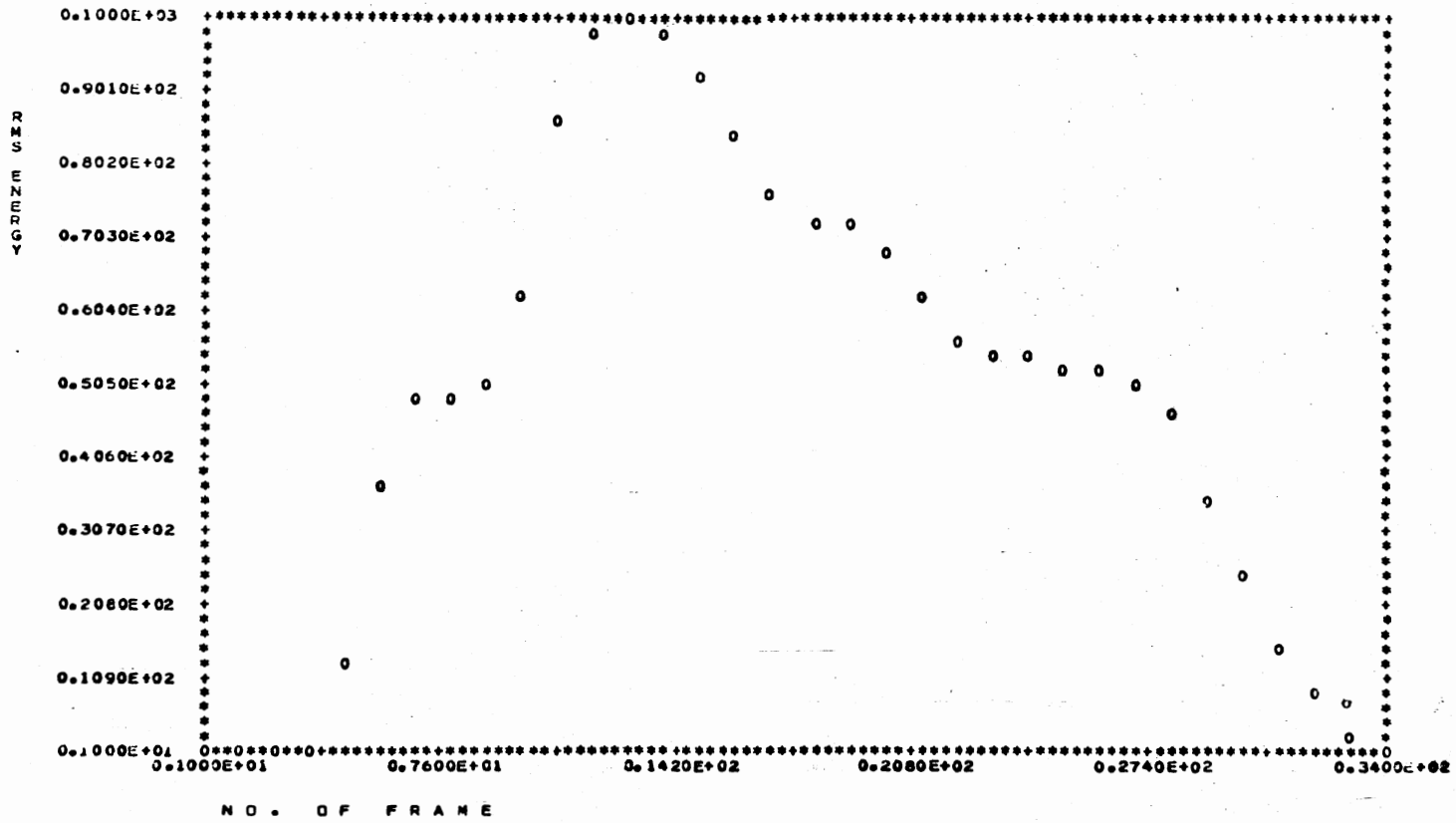


Figure 42. Smoothed and Quantized RMS Energy Contour for Digit Three, i.e. /θri/ Spoken by International Speaker

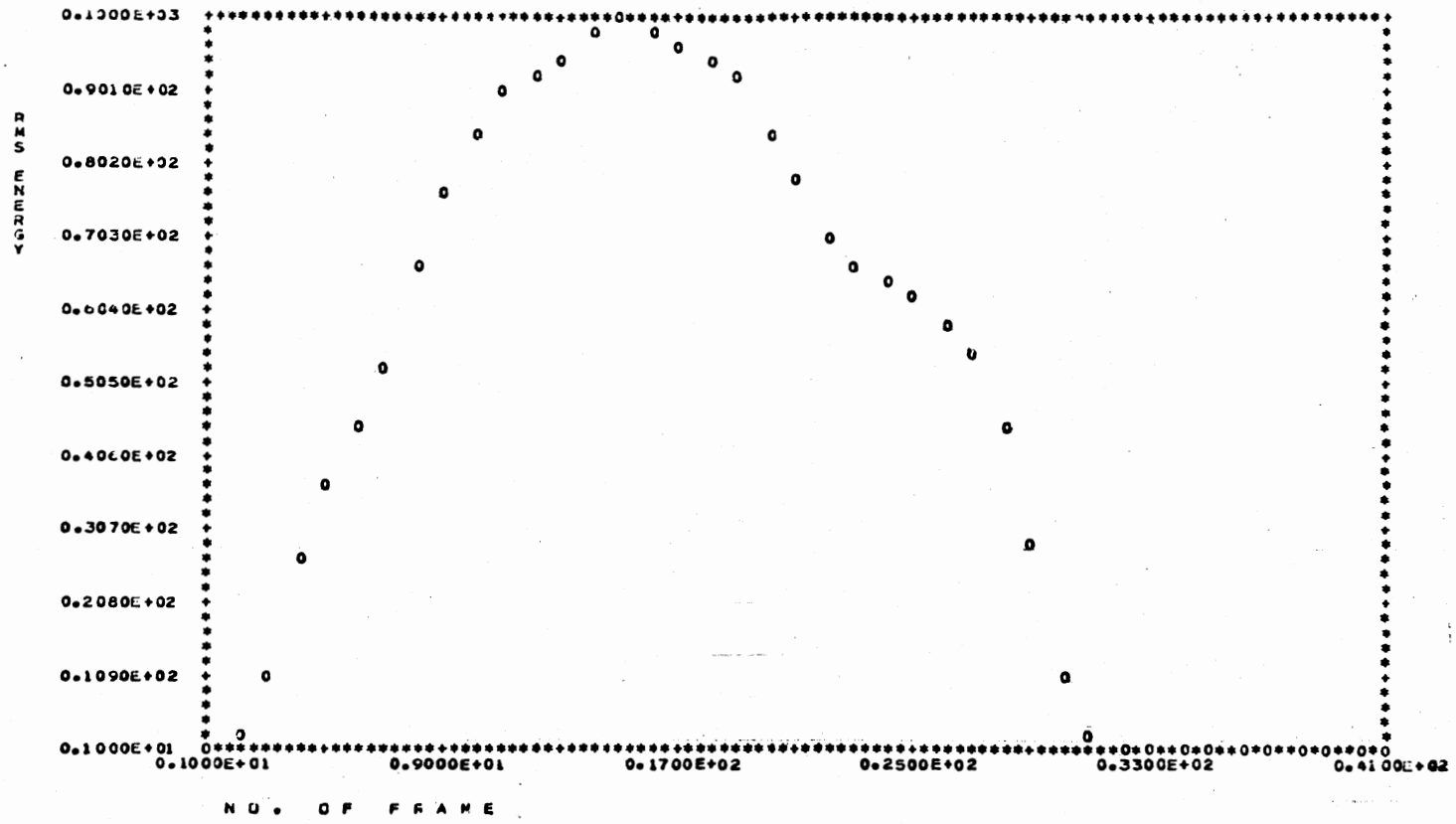


Figure 43. Smoothed and Quantized RMS Energy Contour for Digit Four, i.e. /fɔr/ Spoken by International Speaker

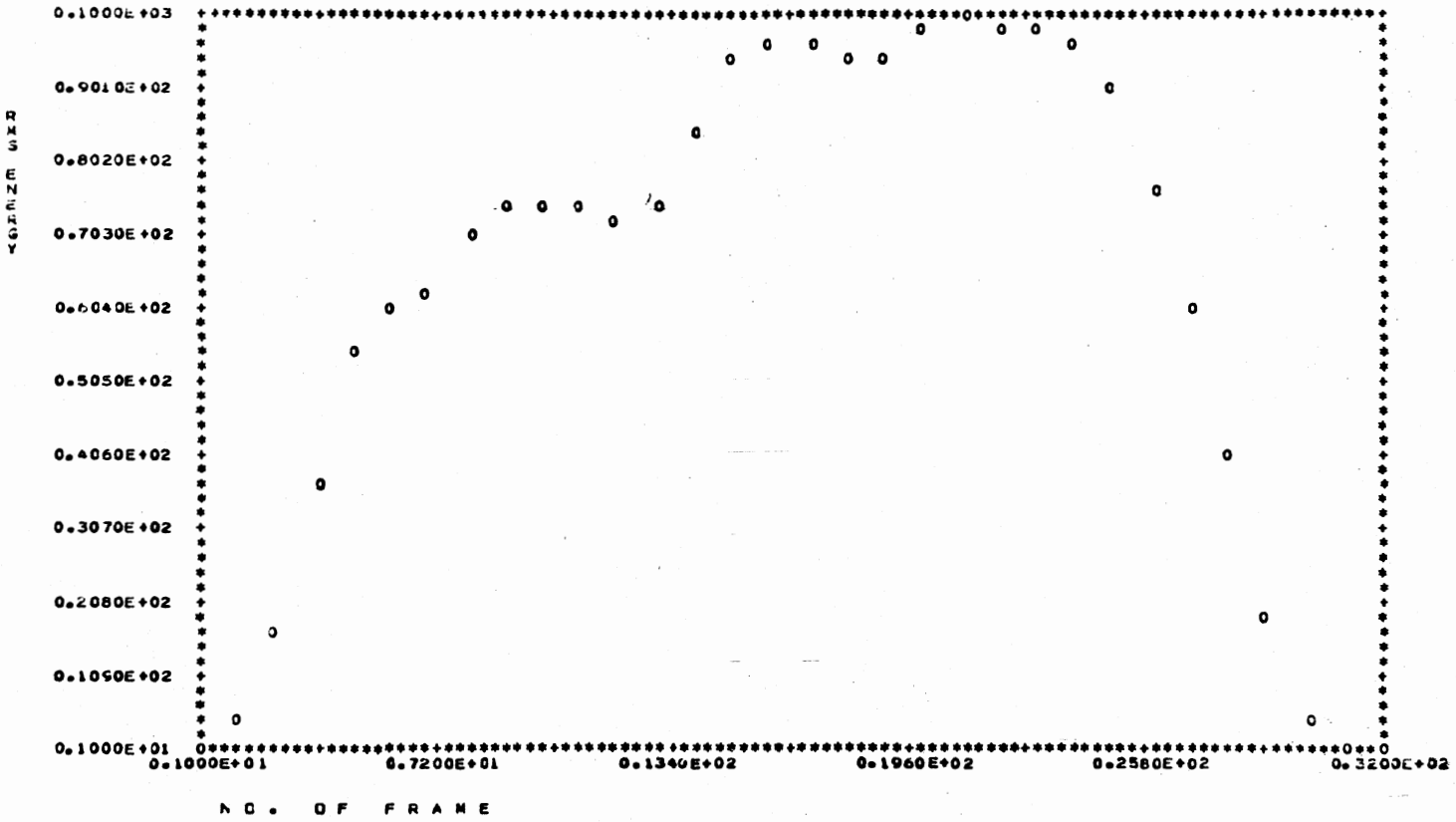


Figure 44. Smoothed and Quantized RMS Energy Contour for Digit Five, i.e. /faIv/ Spoken by International Speaker

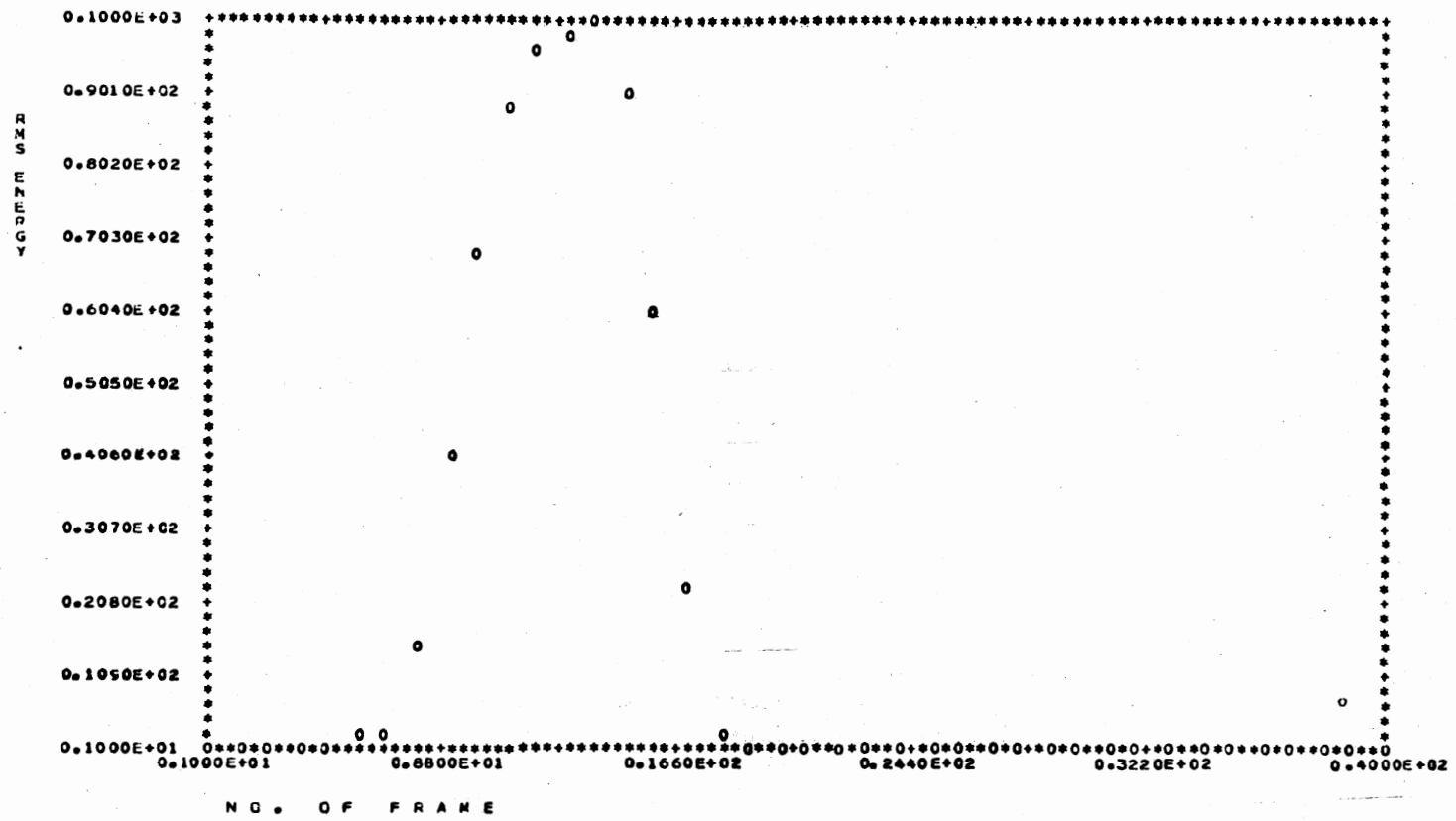


Figure 45. Smoothed and Quantized RMS Energy Contour for Digit Six, i.e. /sIks/ Spoken by International Speaker

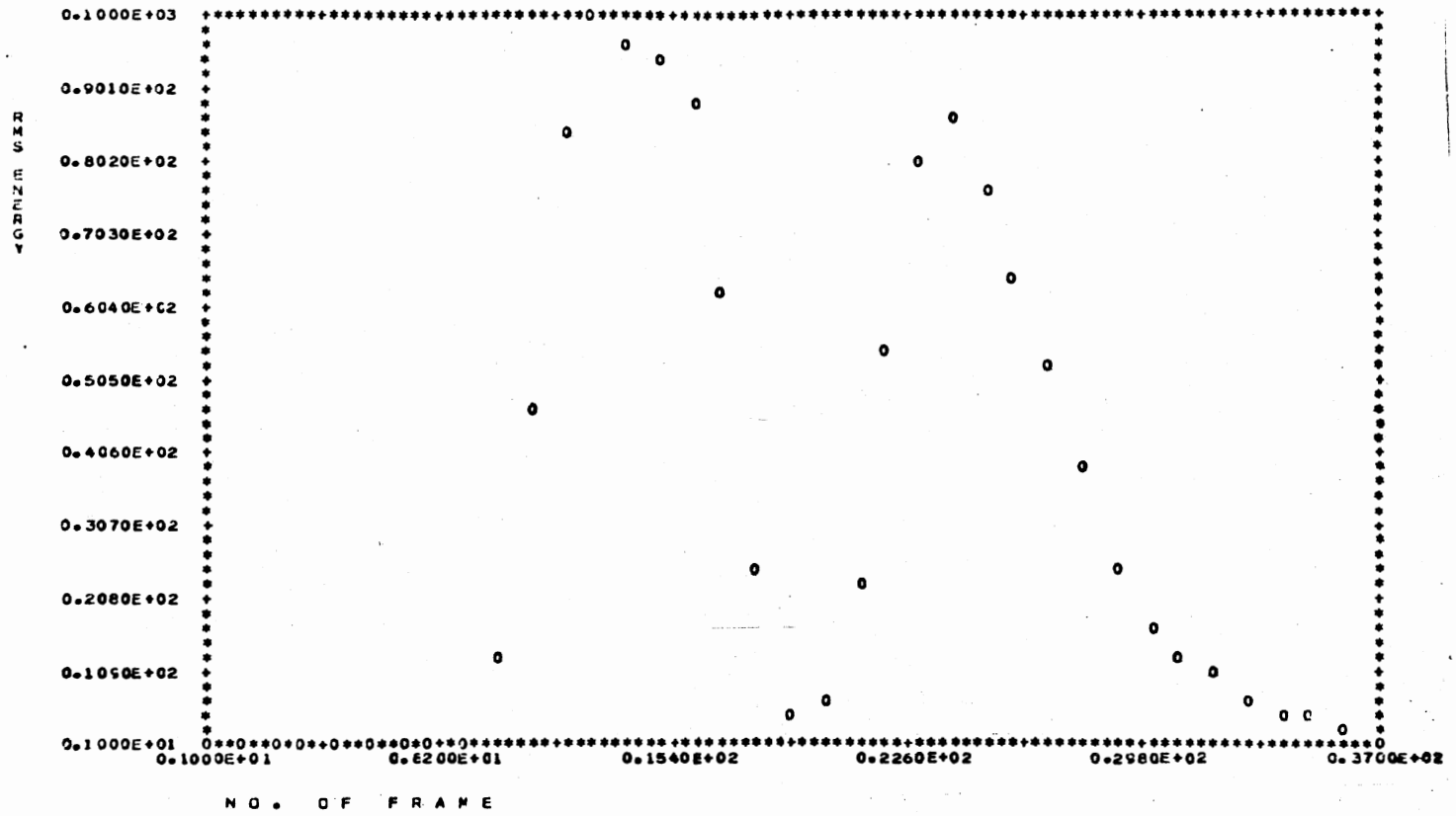


Figure 46. Smoothed and Quantized RMS Energy Contour for Digit Seven, i.e. /seven/ Spoken by International Speaker

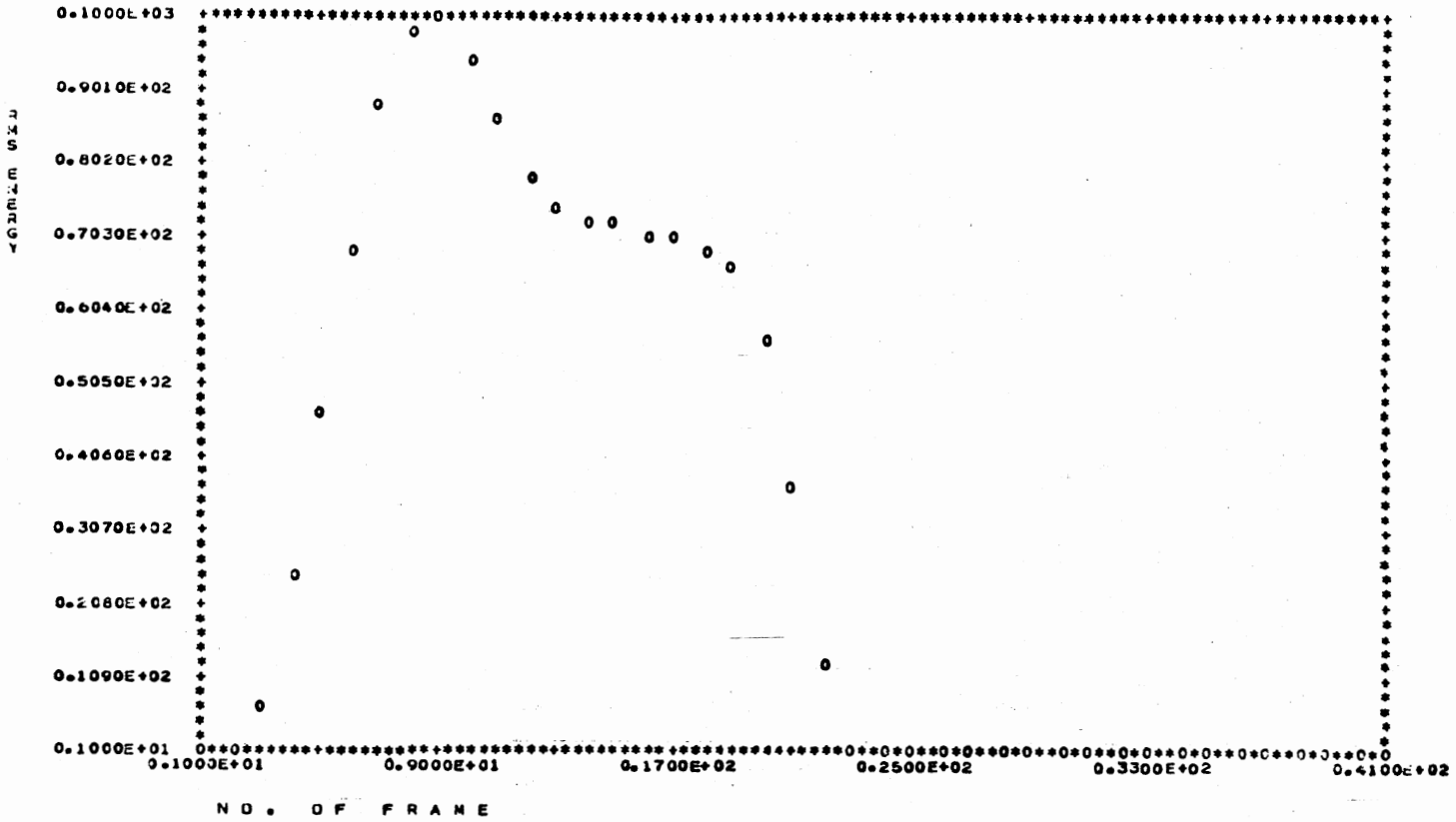


Figure 47. Smoothed and Quantized RMS Energy Contour for Digit Eight, i.e. /eIt/ Spoken by International Speaker

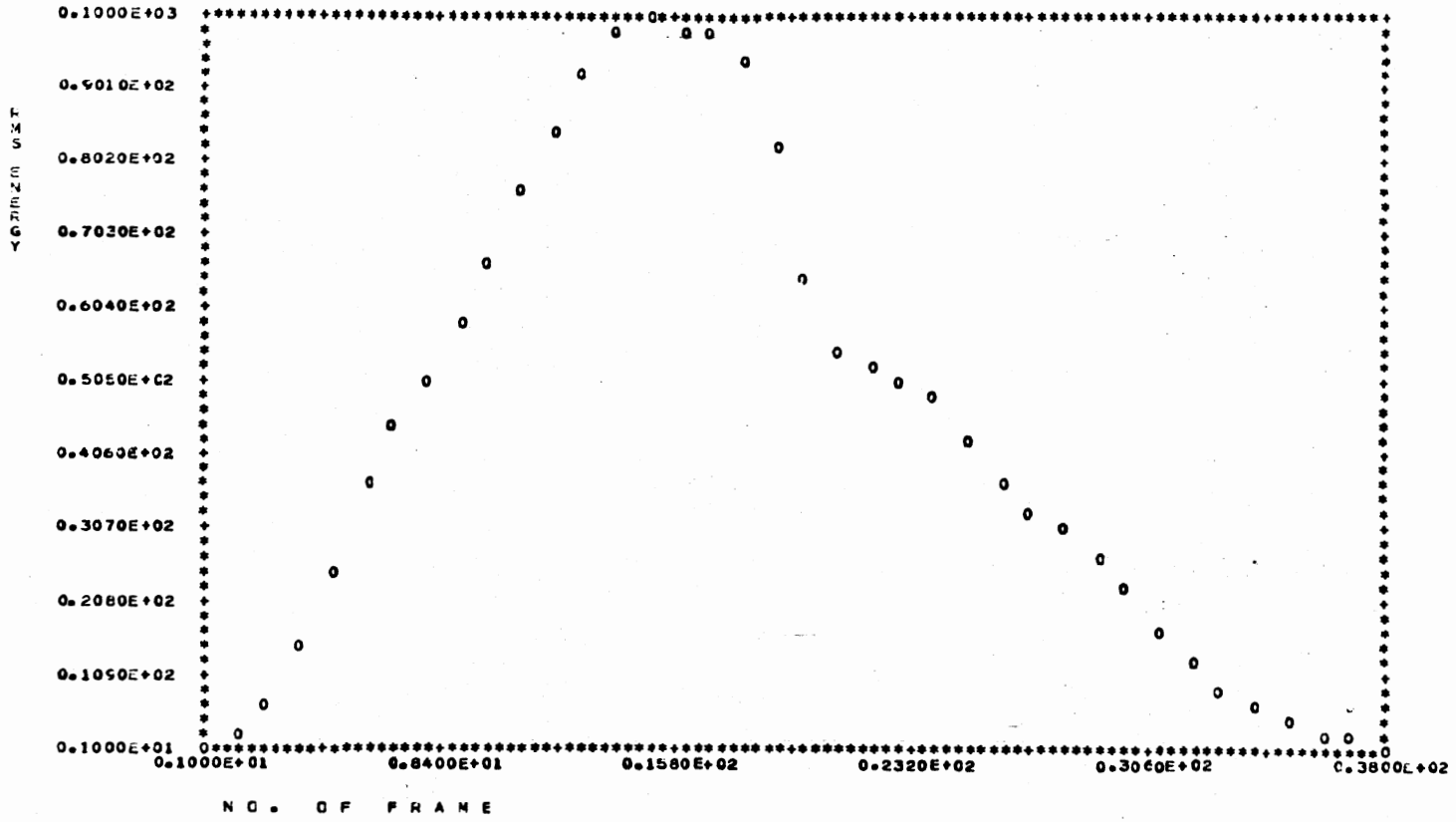


Figure 48. Smoothed and Quantized RMS Energy Contour for Digit Nine, i.e. /naIn/ Spoken by International Speaker

TABLE V

SMOOTHED RMS ENERGY PEAKS AND RATIOS FOR DIGITS SPOKEN IN AMERICAN ENGLISH FOR FIXED MEAN AND VARYING PEAKS

Digit	P_1	P_2	Ratio
/zIro/	15.05	12.33	0.82
/wʌn/	4.19	11.71	0.36
/tu/	8.64	5.14	0.595
/θri/	10.86	14.46	0.75
/fɔr/	12.80		
/faIv/	14.28	14.24	1.0
/sIks/	56.79		
/sevɛn/	36.04	22.84	0.63
/eIt/	22.72		
/naIn/	23.76	10.28	0.43

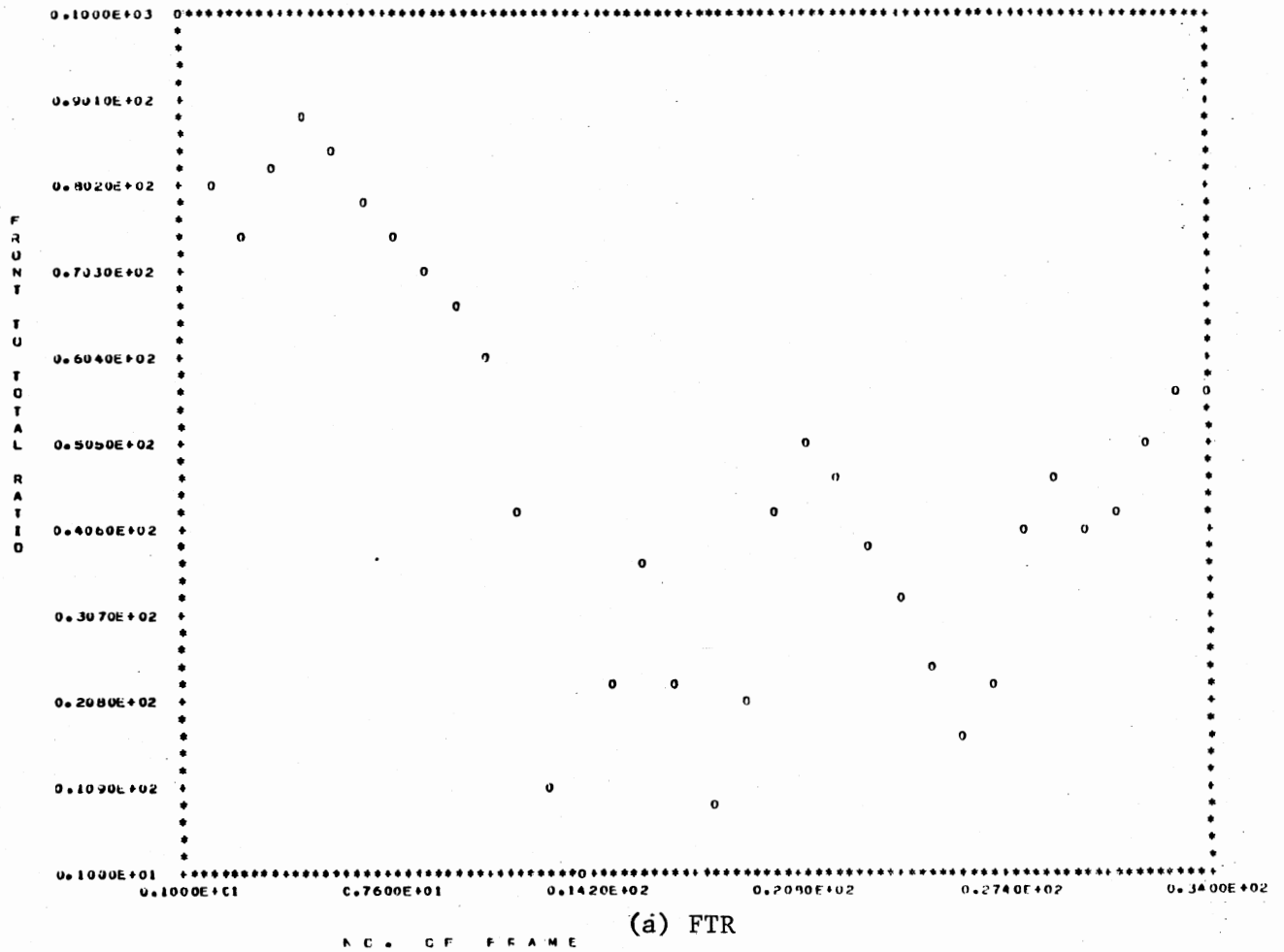
TABLE VI

SMOOTHED AND QUANTIZED RMS ENERGY PEAKS AND THEIR RATIOS FOR DIGITS SPOKEN IN AMERICAN ENGLISH

Digit	P_1	P_2	Ratio
/zIro/	100	91	0.91
/wʌn/	100	72	0.72
/tu/	100	97	0.97
/θri/	48	100	0.48
/fɔr/	100	63	0.63
/faIV/	96	100	0.96
/sIks/	100		
/sevɛn/	100	86	0.86
/eIt/	100		
/naIn/	100		

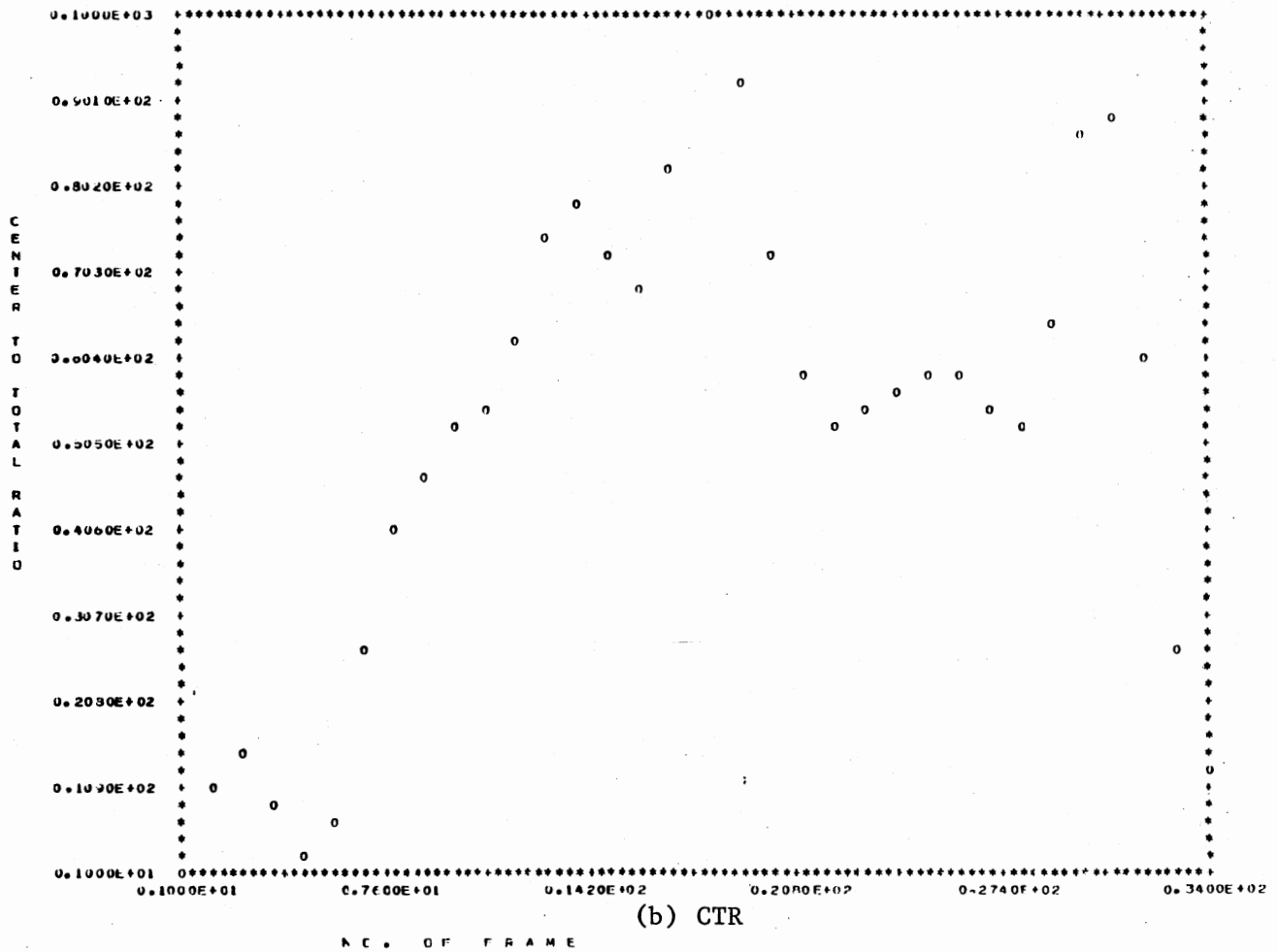
Following the block diagram in Figure 28, the second path after end-point detection is through Hamming window. As discussed in Chapter III, the window length is taken as 150 points. From the windowed data, a 14th order LPA model is computed. From this model, the BTR, CTR, FTR, SFBR, smoothed BTR, smoothed CTR, and smoothed FTR are computed for each frame. These plots are given in Figures 49-58. Plots of smoothed BTR, smoothed CTR, and smoothed FTR was obtained, in order to determine which parameter would give the best phonemic feature, to separate vowels from consonants. A threshold level is set to distinguish between vowel, vowel-like, and non-vowel segments. Using the plots in Figures 49-58, it can be seen that all these parameters have some useful phonetic and acoustical features. In addition, the BTR and CTR indicate better acoustical configuration than the FTR. Furthermore, the BTR is found to give moderately better results than the CTR. However, some modification of the CTR range is required prior to future applications.

The parameters derived are used in the phonetic feature detection stage. For phoneme segmentation and other classifications, the RMS dip-classification algorithm is used, which was discussed in Chapter III. The algorithm detects three types of dips or valleys in the smoothed RMS energy contour according to the signs of the functions Z_1 and Z_2 as a vowel, vowel-like, and non-vowel. Based upon the discussion in Chapter III, the vowel, vowel-like and non-vowel decision is made for the entire RMS energy contour. Also the vowel, vowel-like and non-vowel decision is made using the smoothed BTR contour. Then an "OR" decision is made using the above two classification



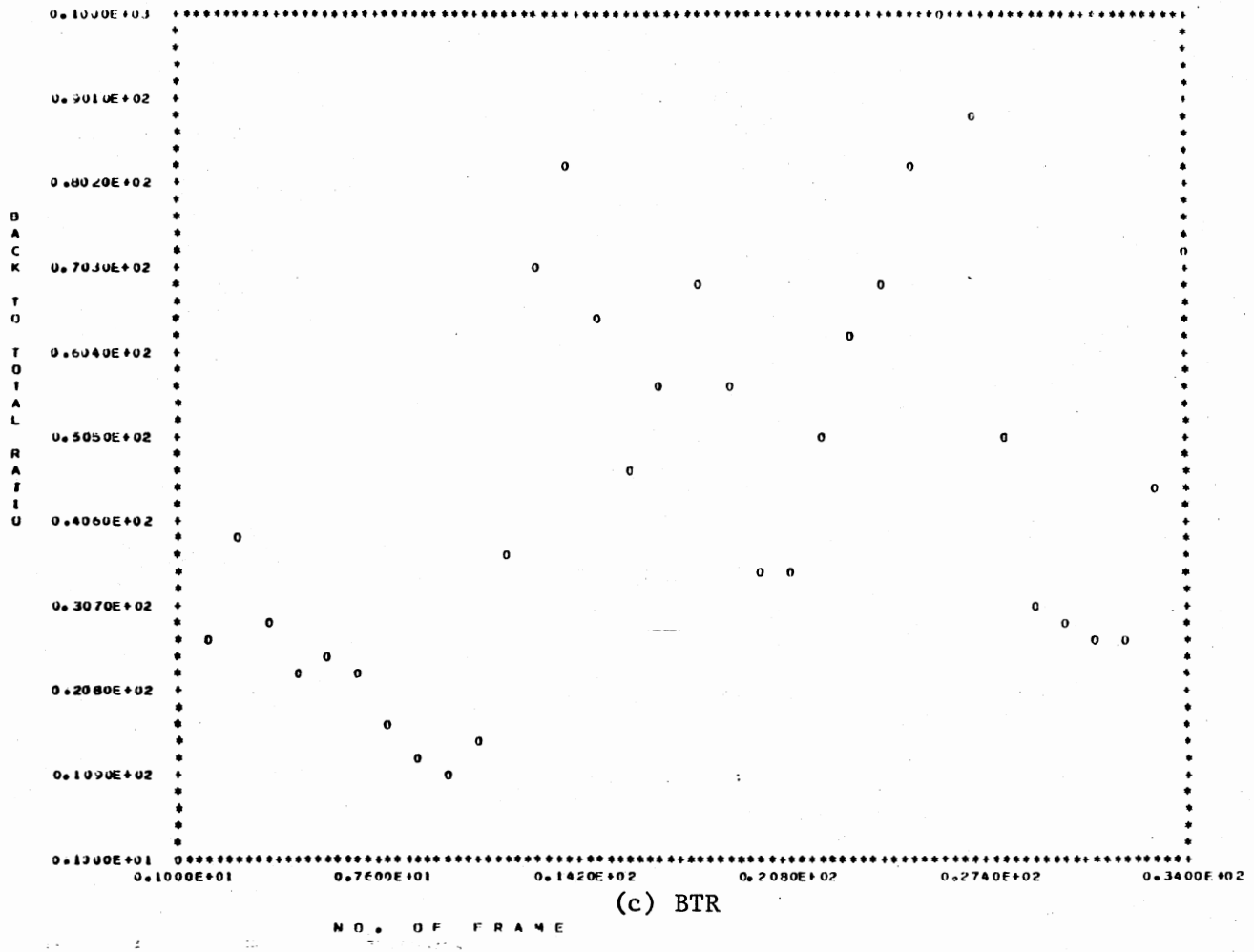
(a) FTR

Figure 49. Smoothed and Quantized Feature Parameters for Digit Zero Spoken in American English



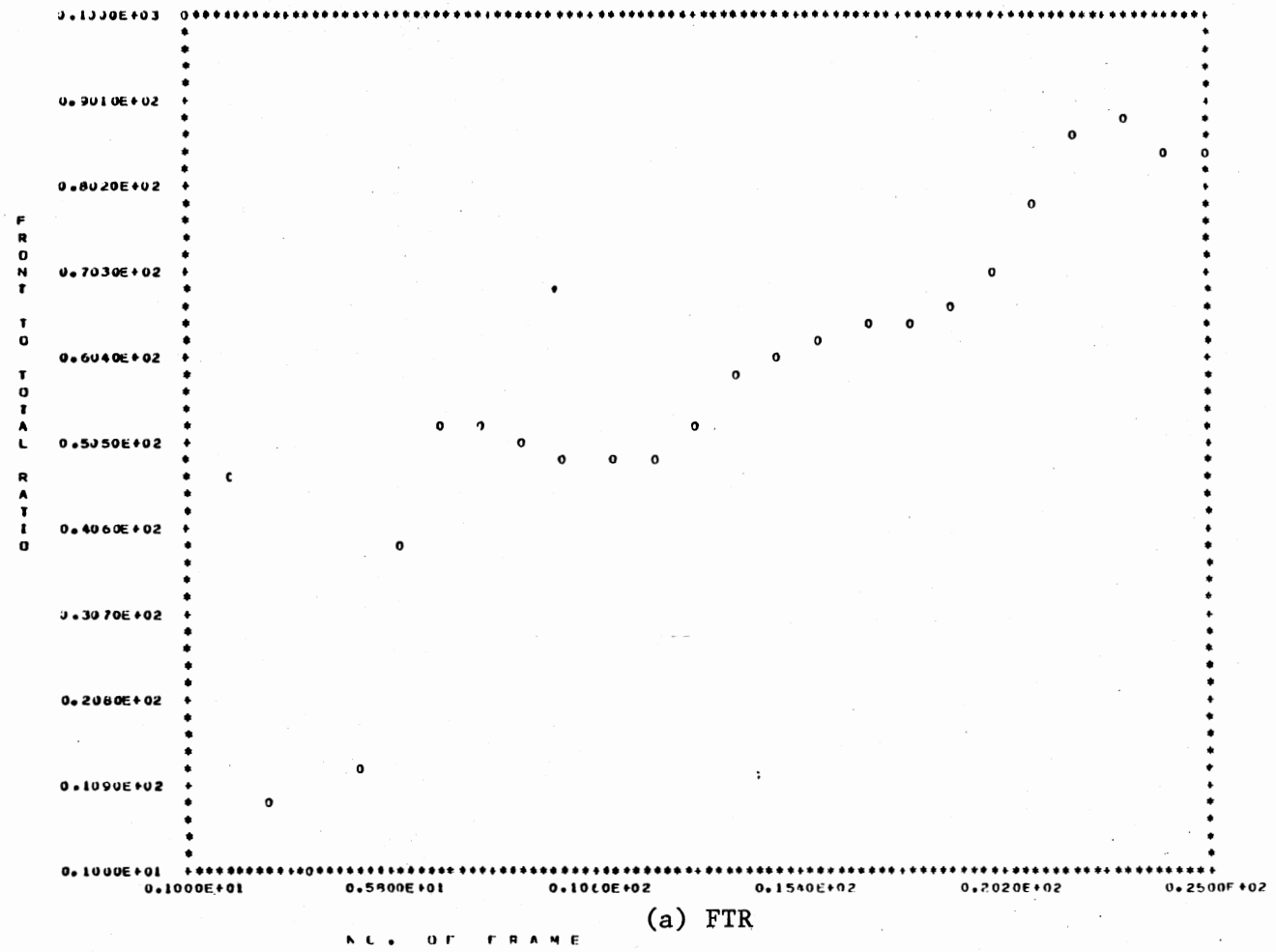
(b) CTR

Figure 49. (Continued)



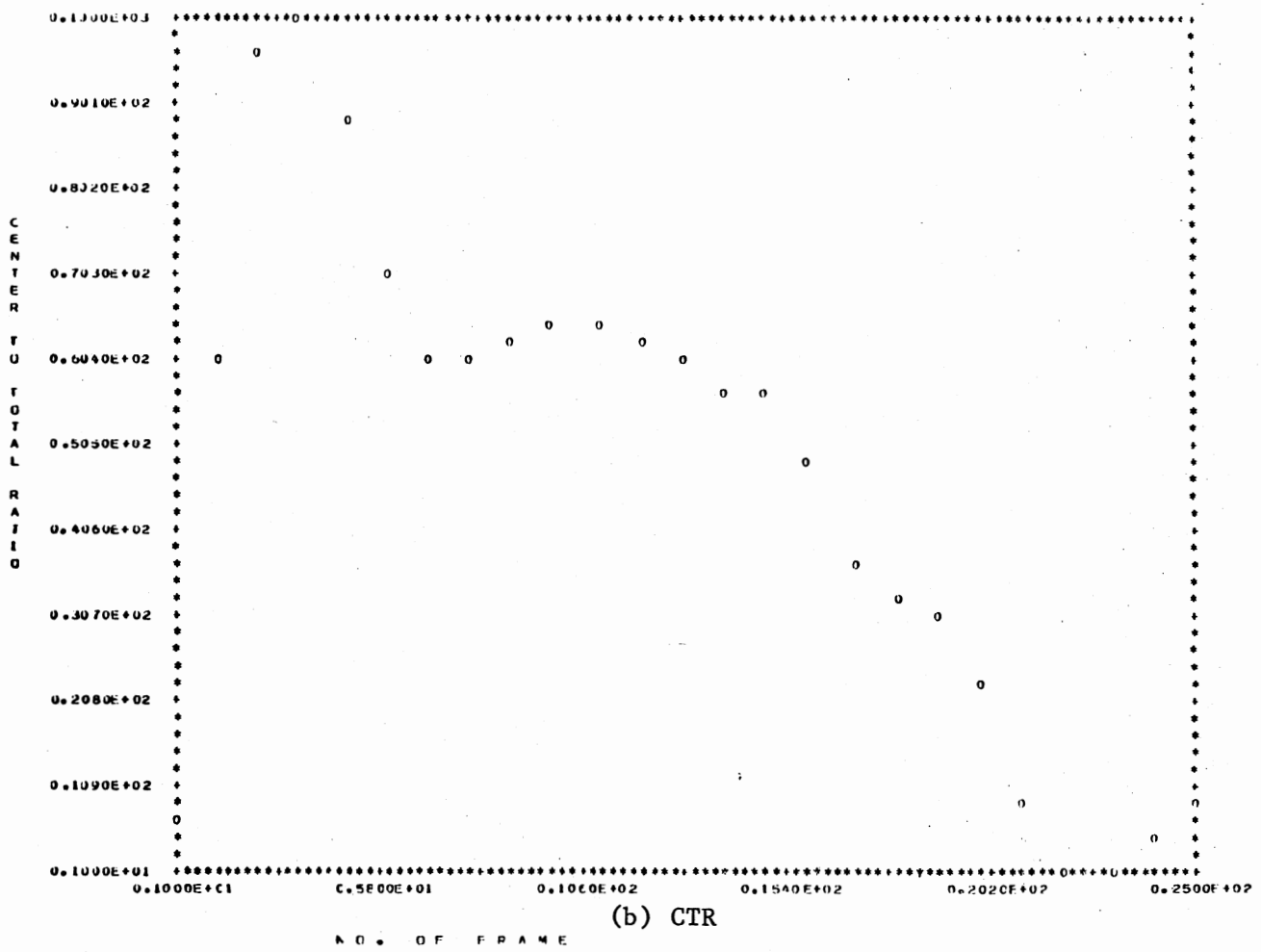
(c) BTR

Figure 49. (Continued)



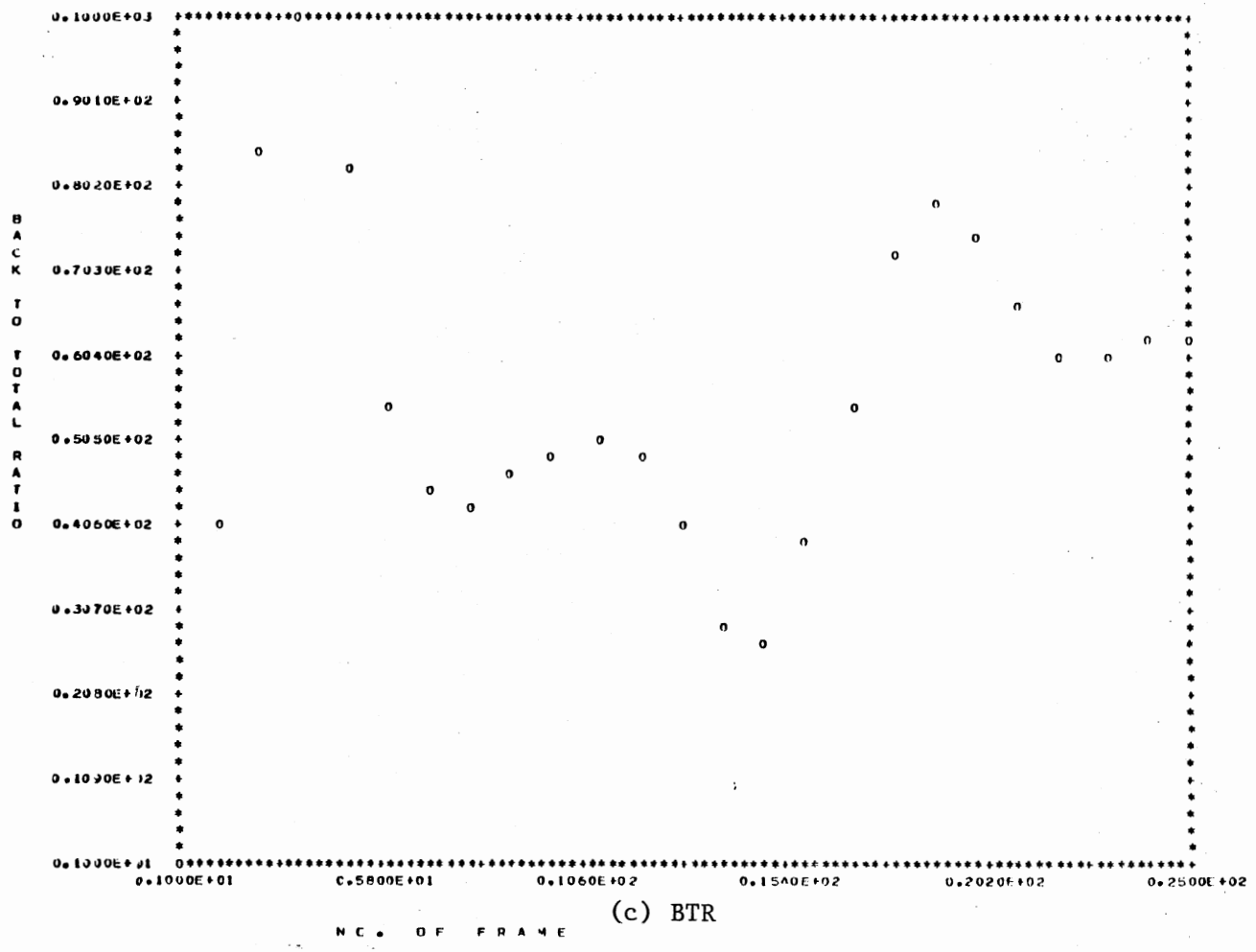
(a) FTR

Figure 50. Smoothed and Quantized Feature Parameters for Digit One Spoken in American English



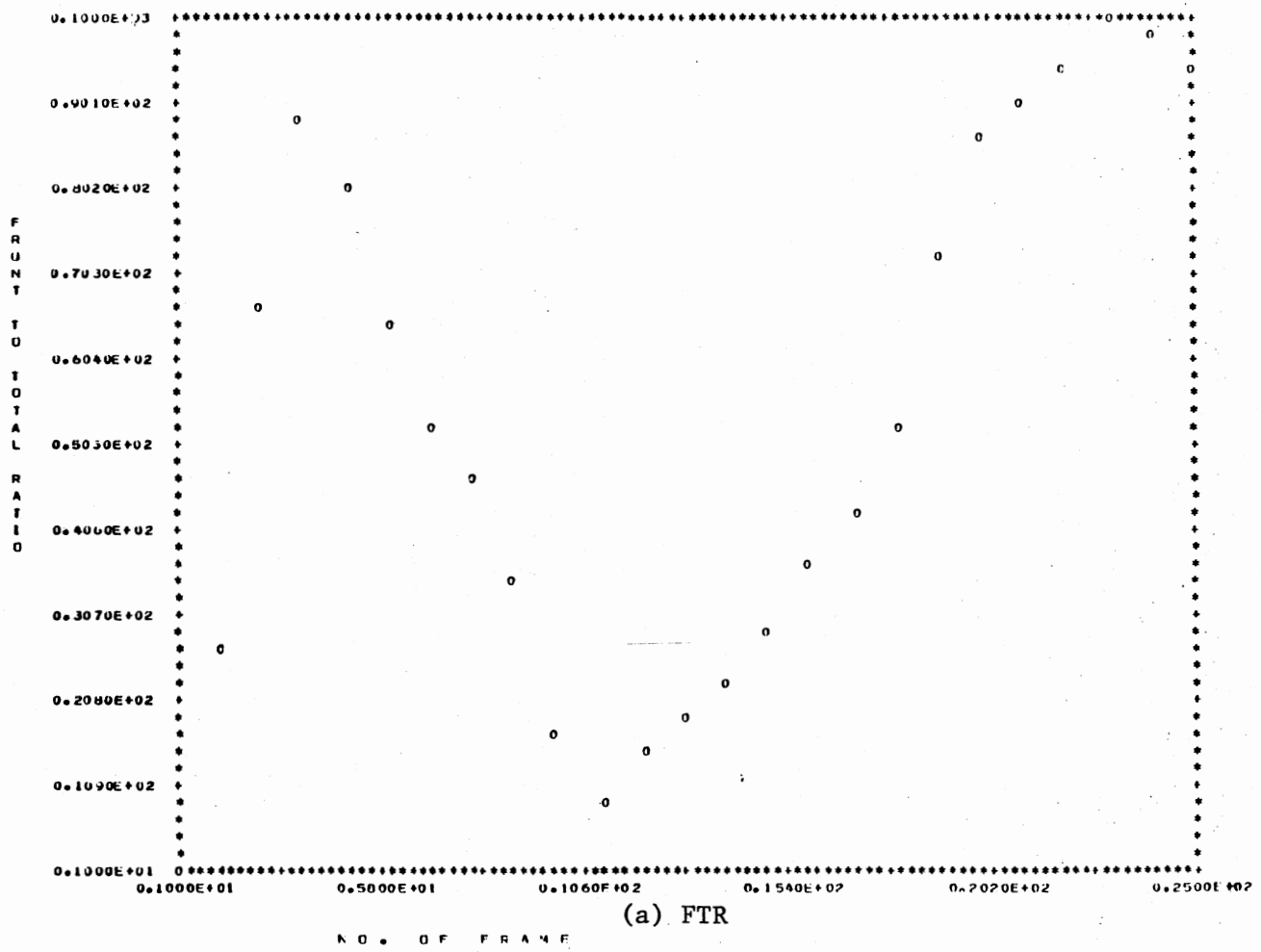
(b) CTR

Figure 50. (Continued)



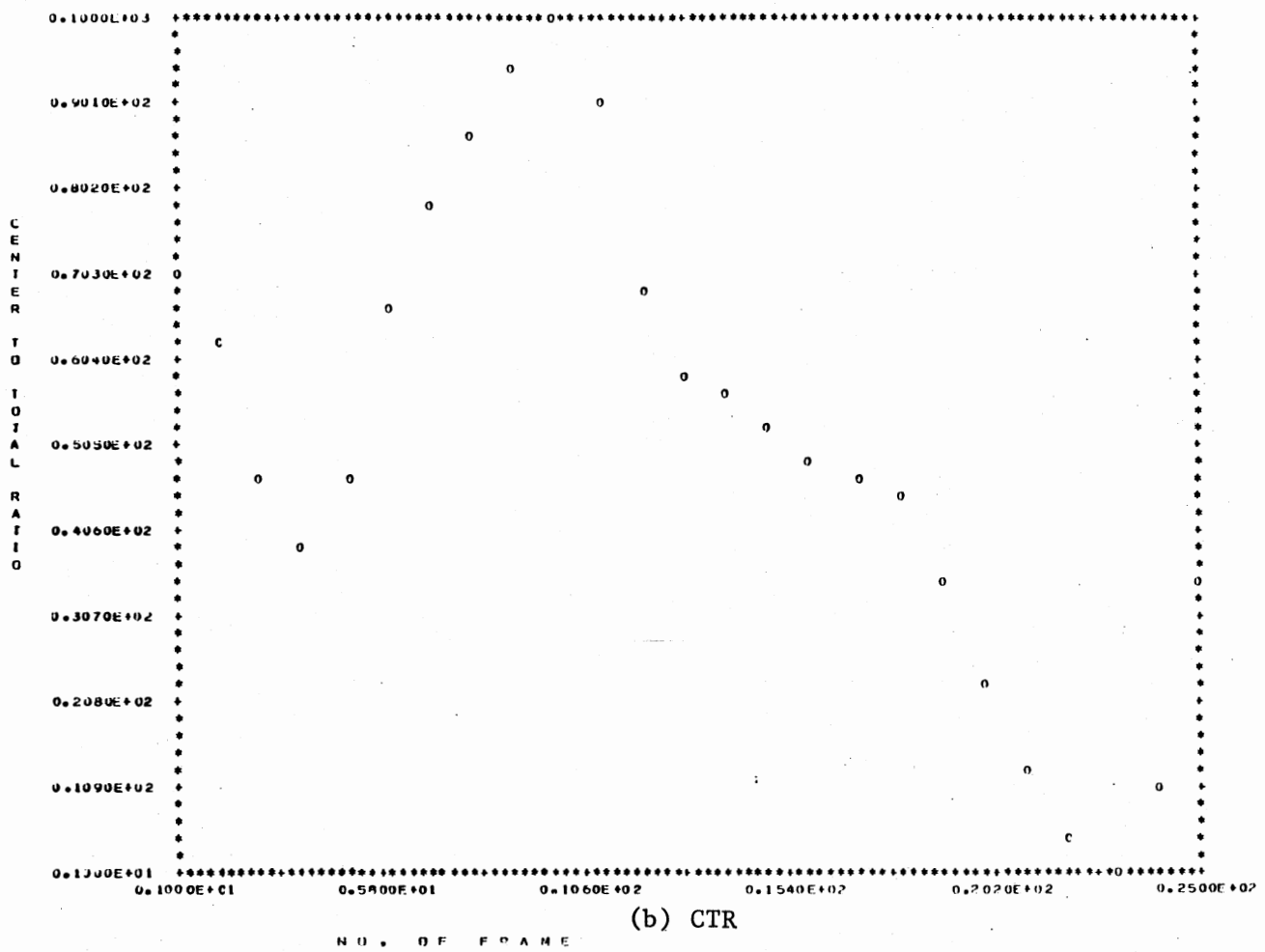
(c) BTR

Figure 50. (Continued)



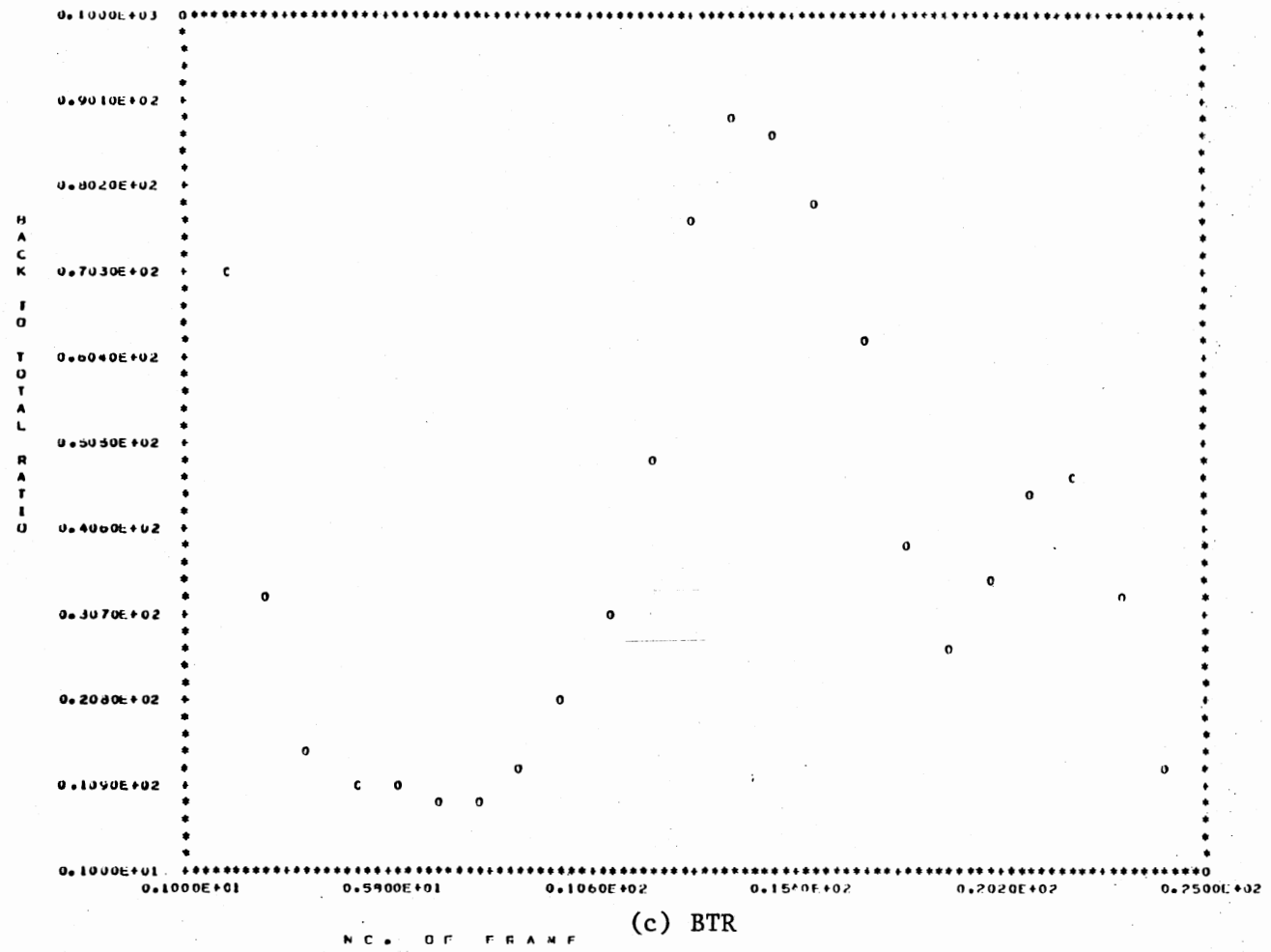
(a) FTR

Figure 51. Smoothed and Quantized Feature Parameters for Digit Two Spoken in American English



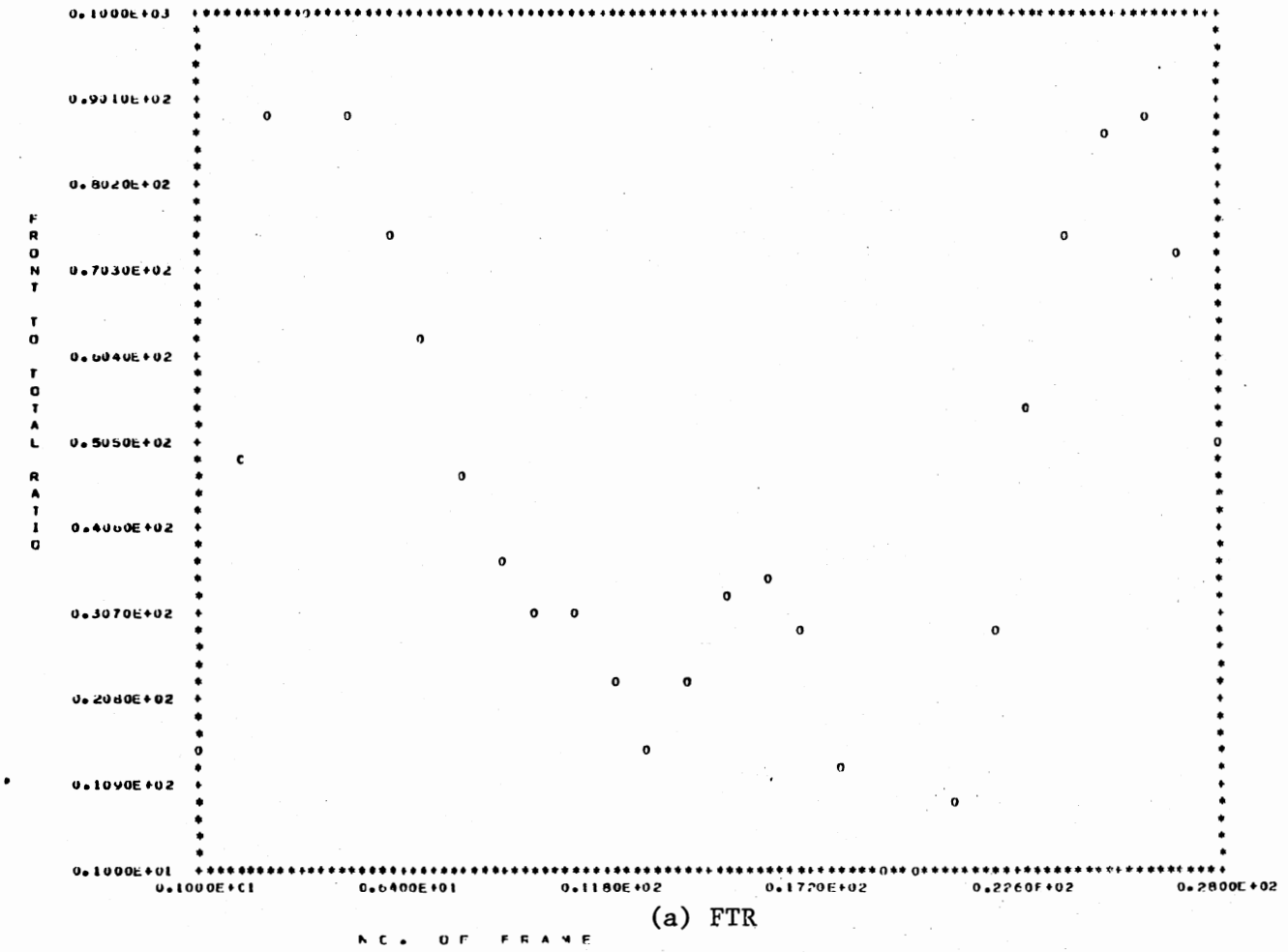
(b) CTR

Figure 51. (Continued)



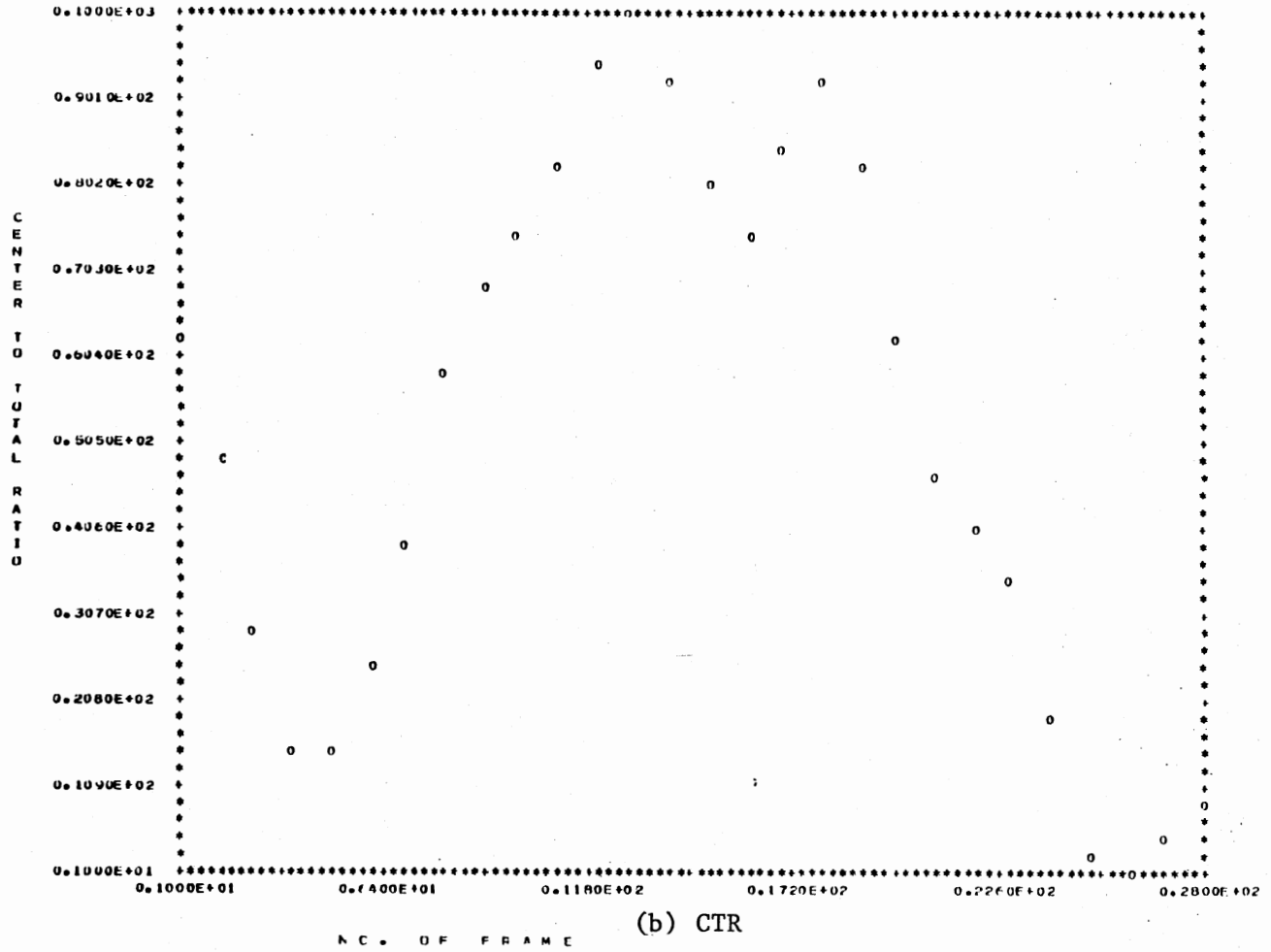
(c) BTR

Figure 51. (Continued)



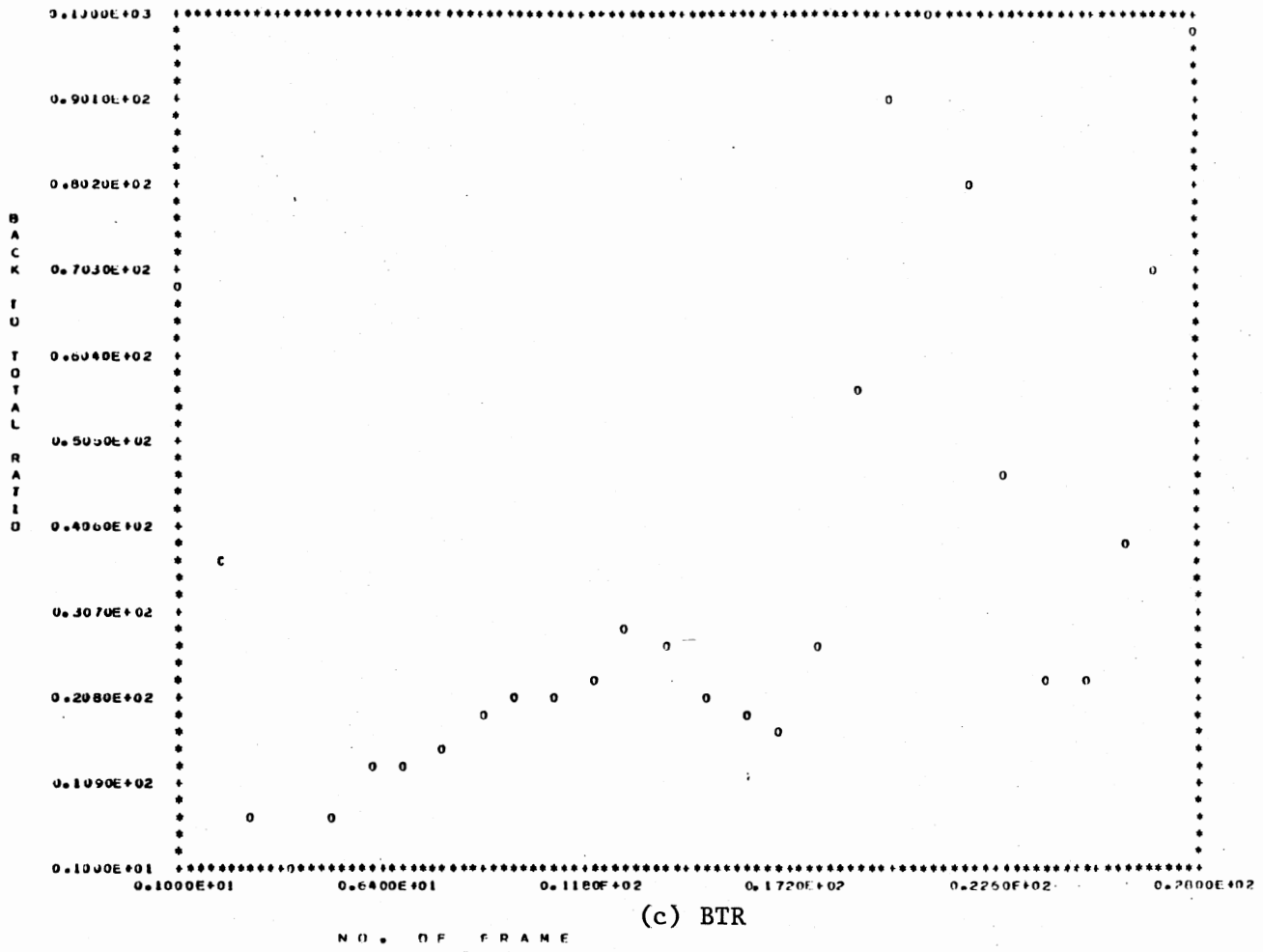
(a) FTR

Figure 52. Smoothed and Quantized Feature Parameters for Digit Three Spoken in American English



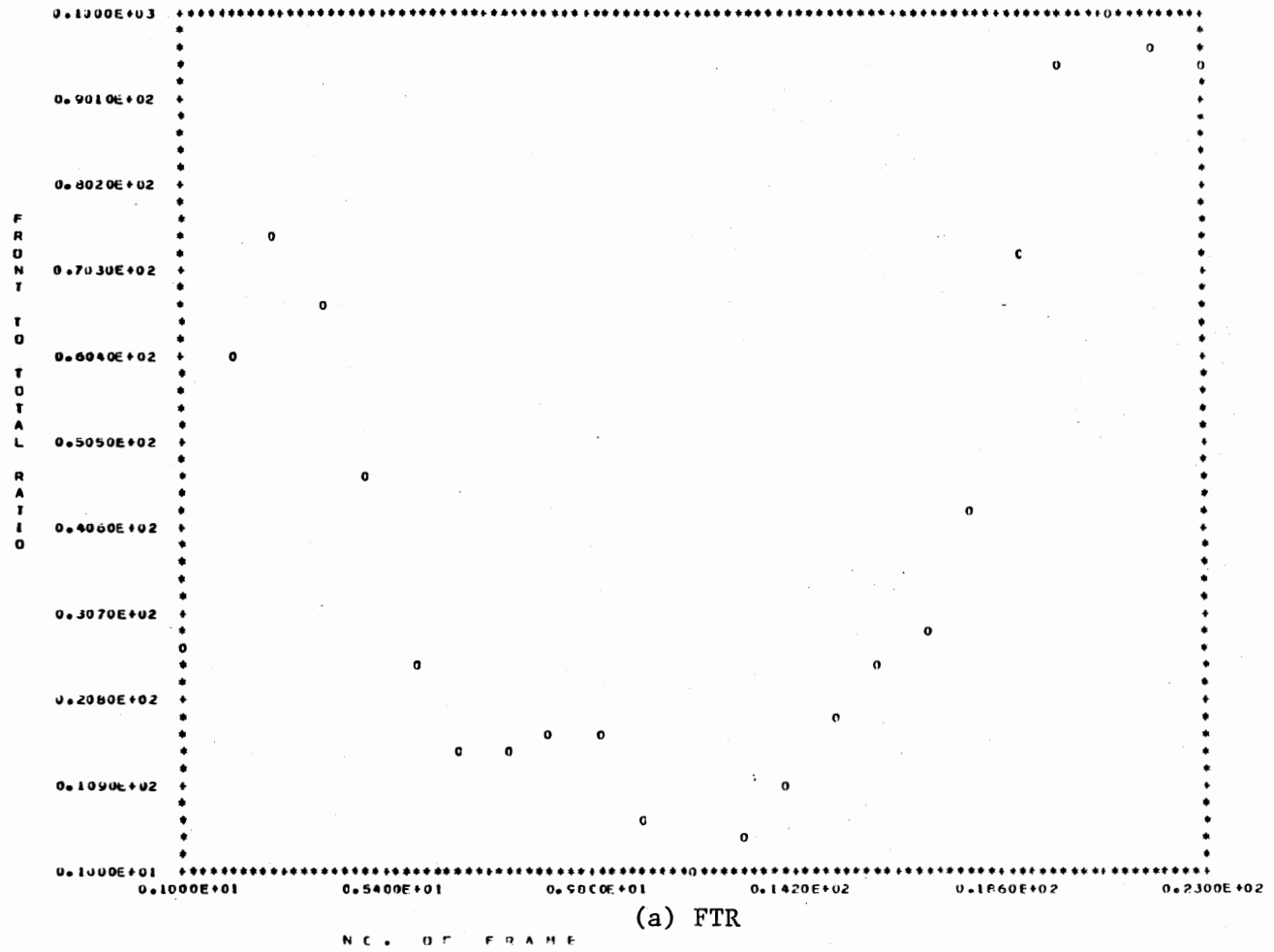
(b) CTR

Figure 52. (Continued)



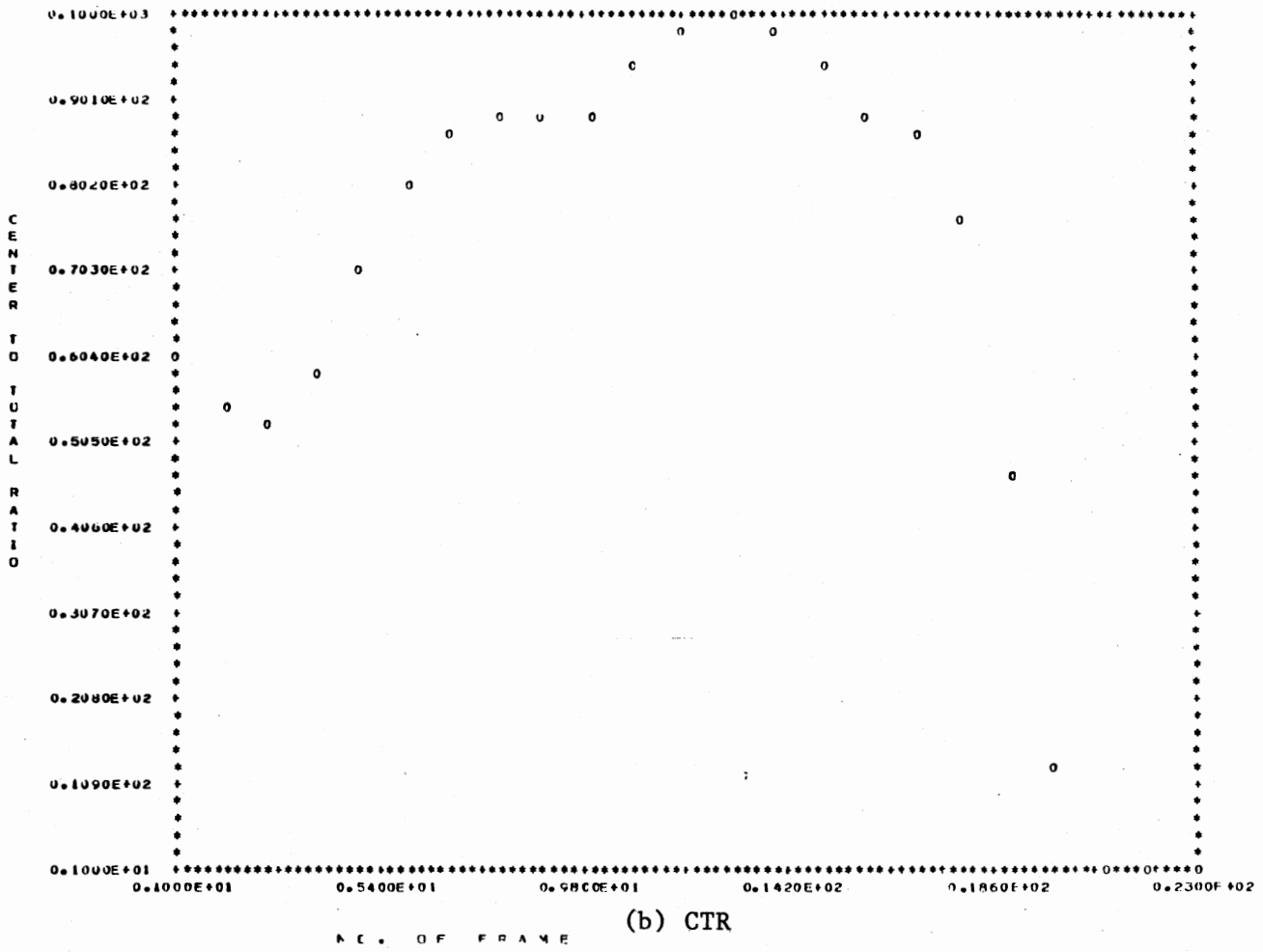
(c) BTR

Figure 52. (Continued)



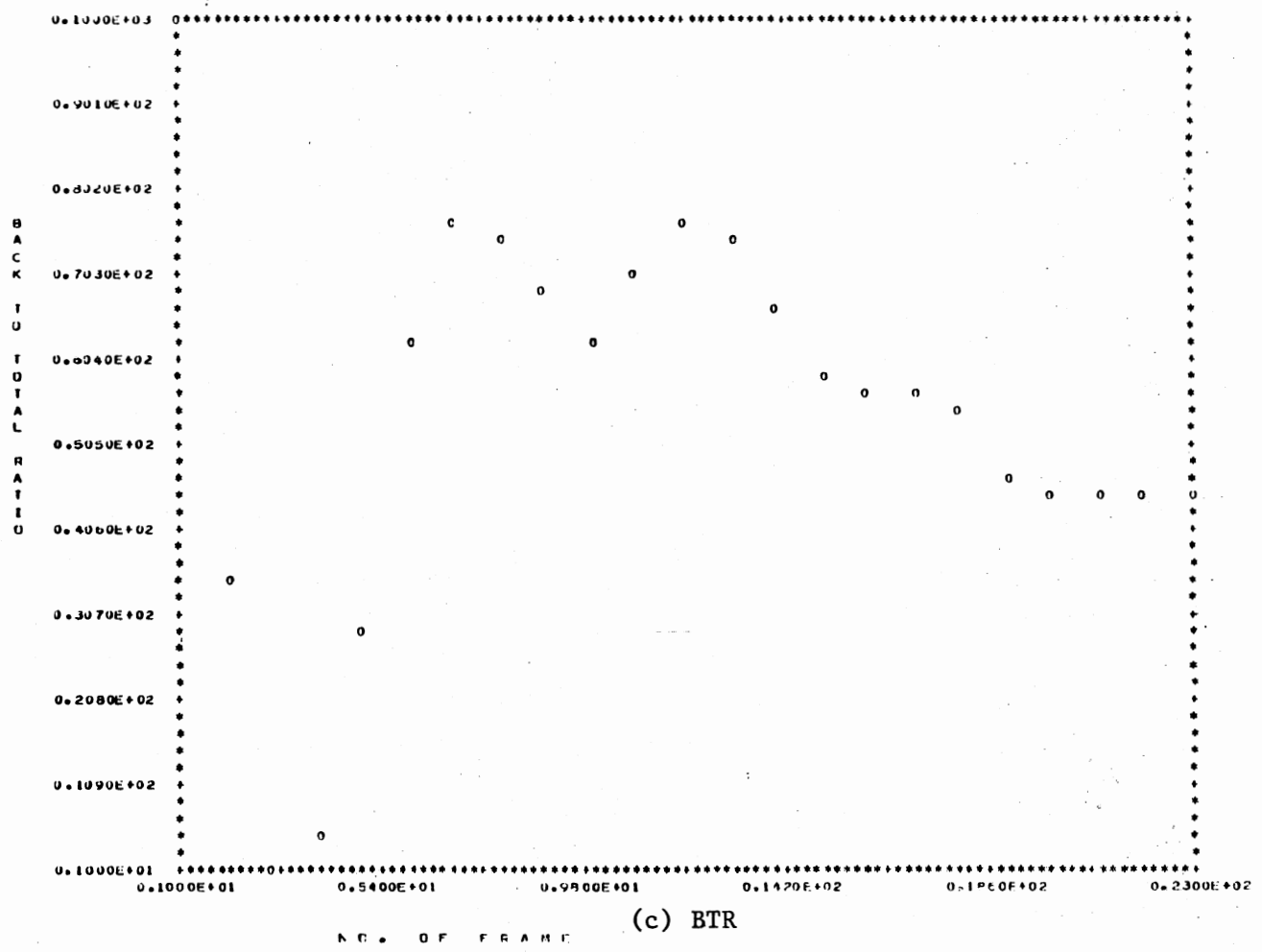
(a) FTR

Figure 53. Smoothed and Quantized Feature Parameters for Digit Four Spoken in American English



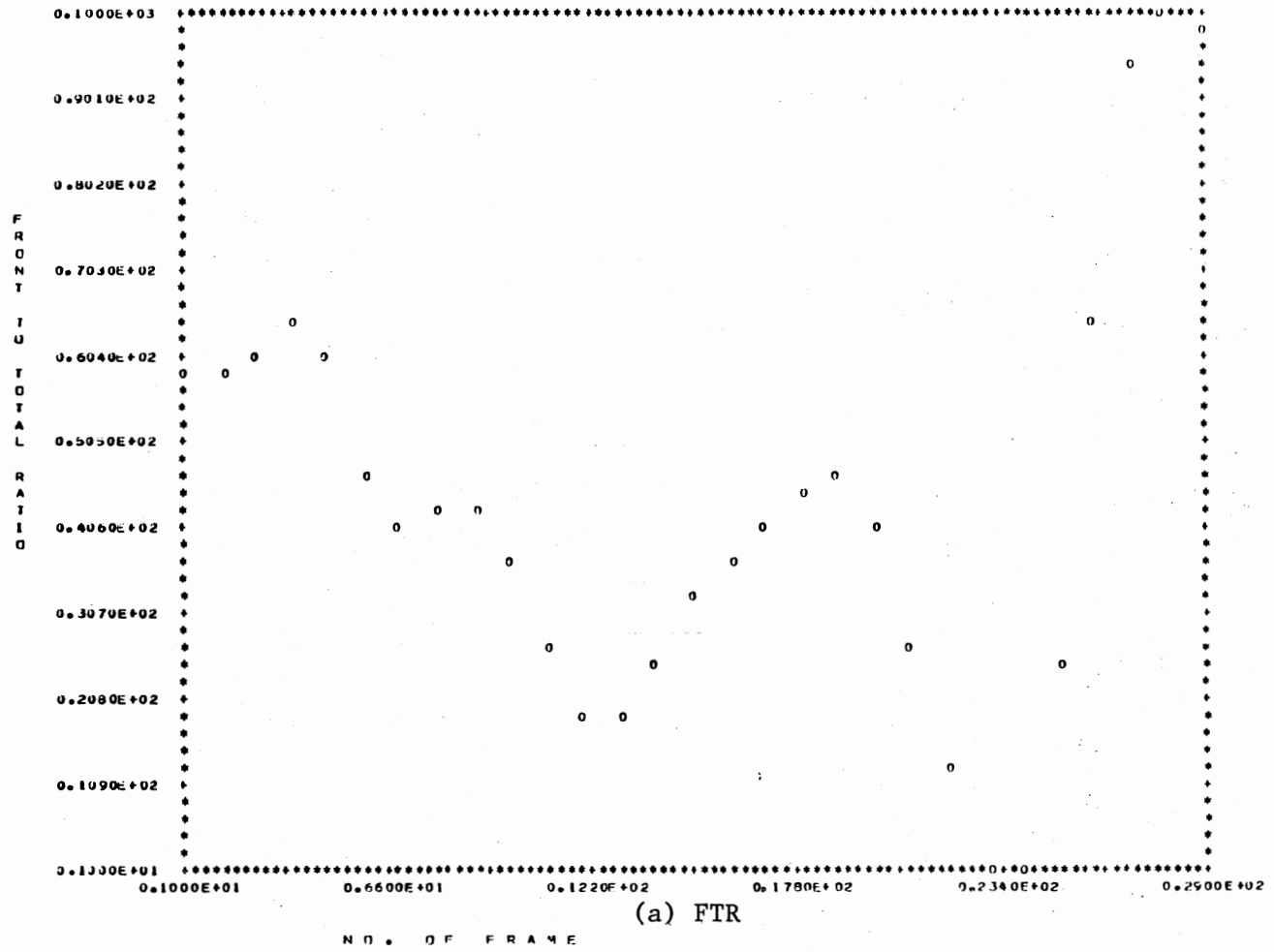
(b) CTR

Figure 53. (Continued)



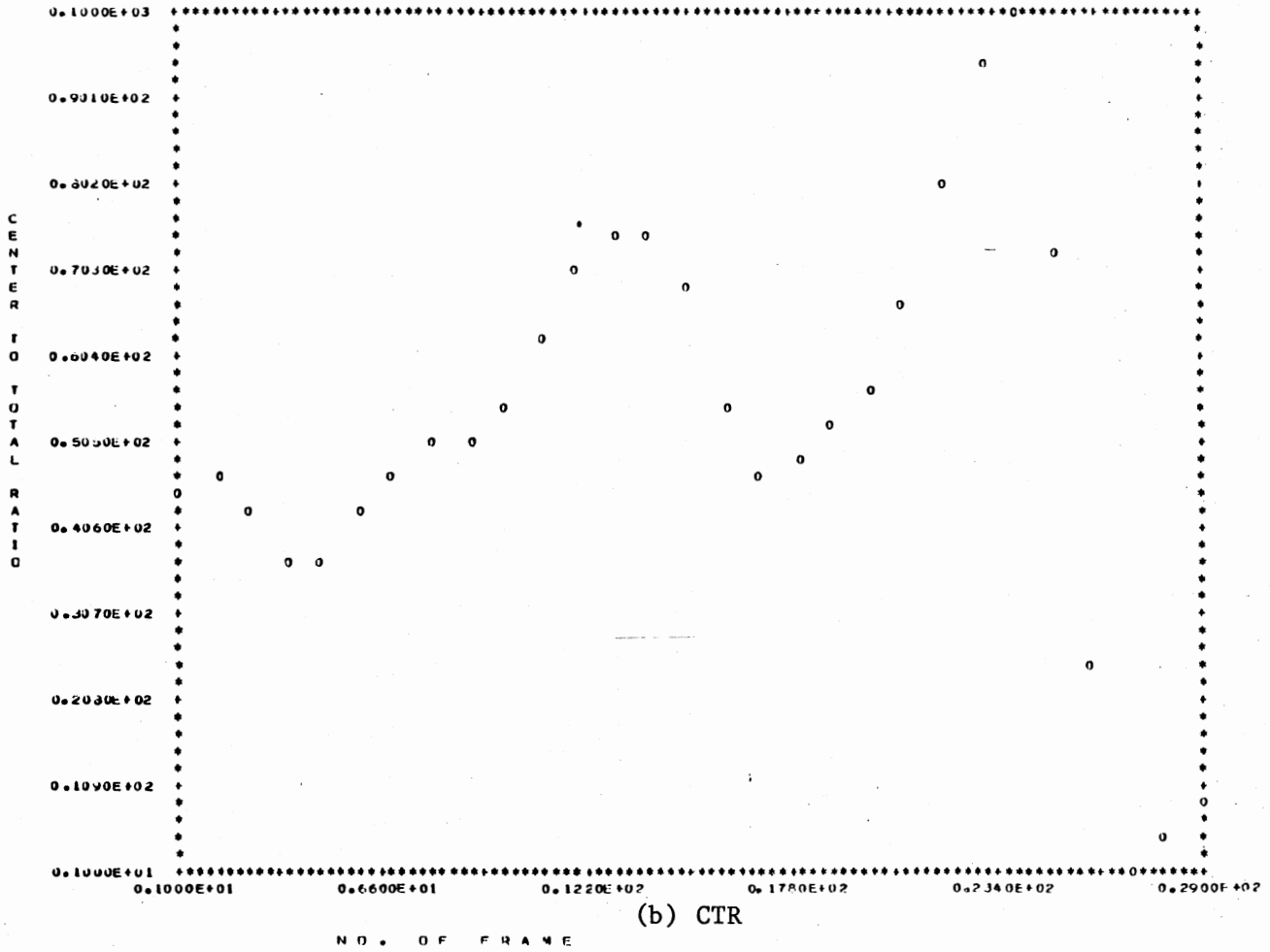
(c) BTR

Figure 53. (Continued)



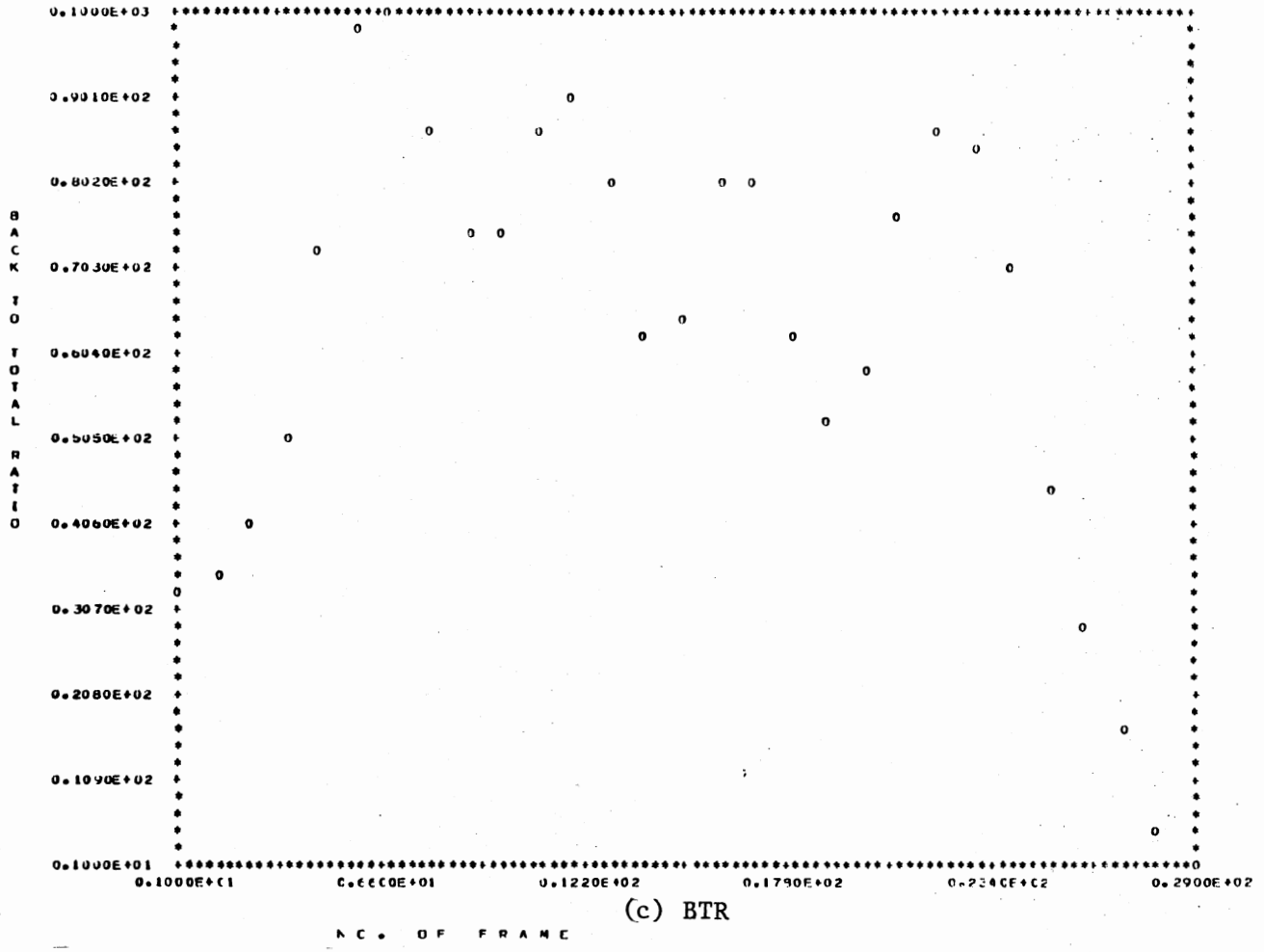
(a) FTR

Figure 54. Smoothed and Quantized Feature Parameters for Digit Five Spoken in American English



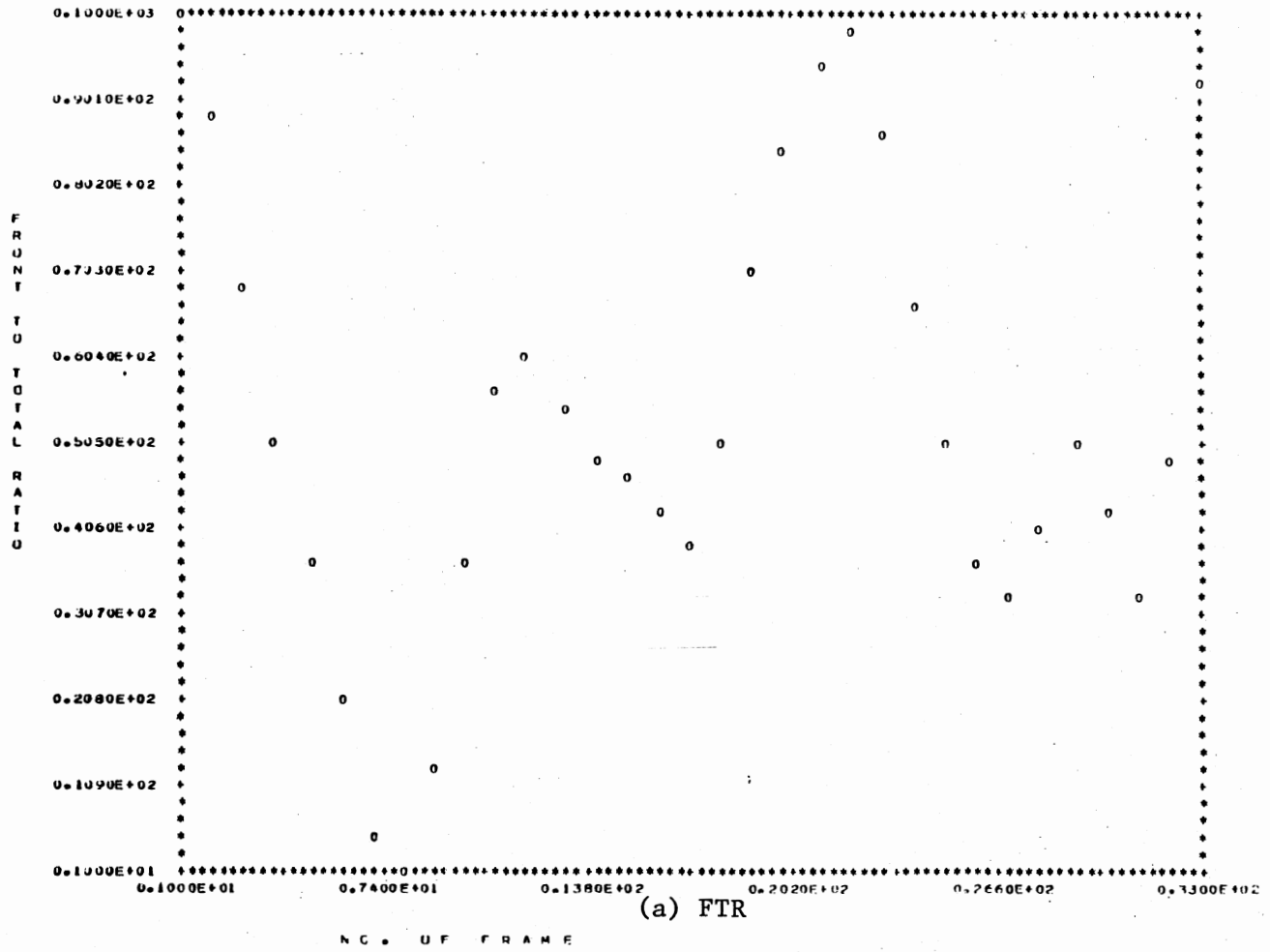
(b) CTR

Figure 54. (Continued)



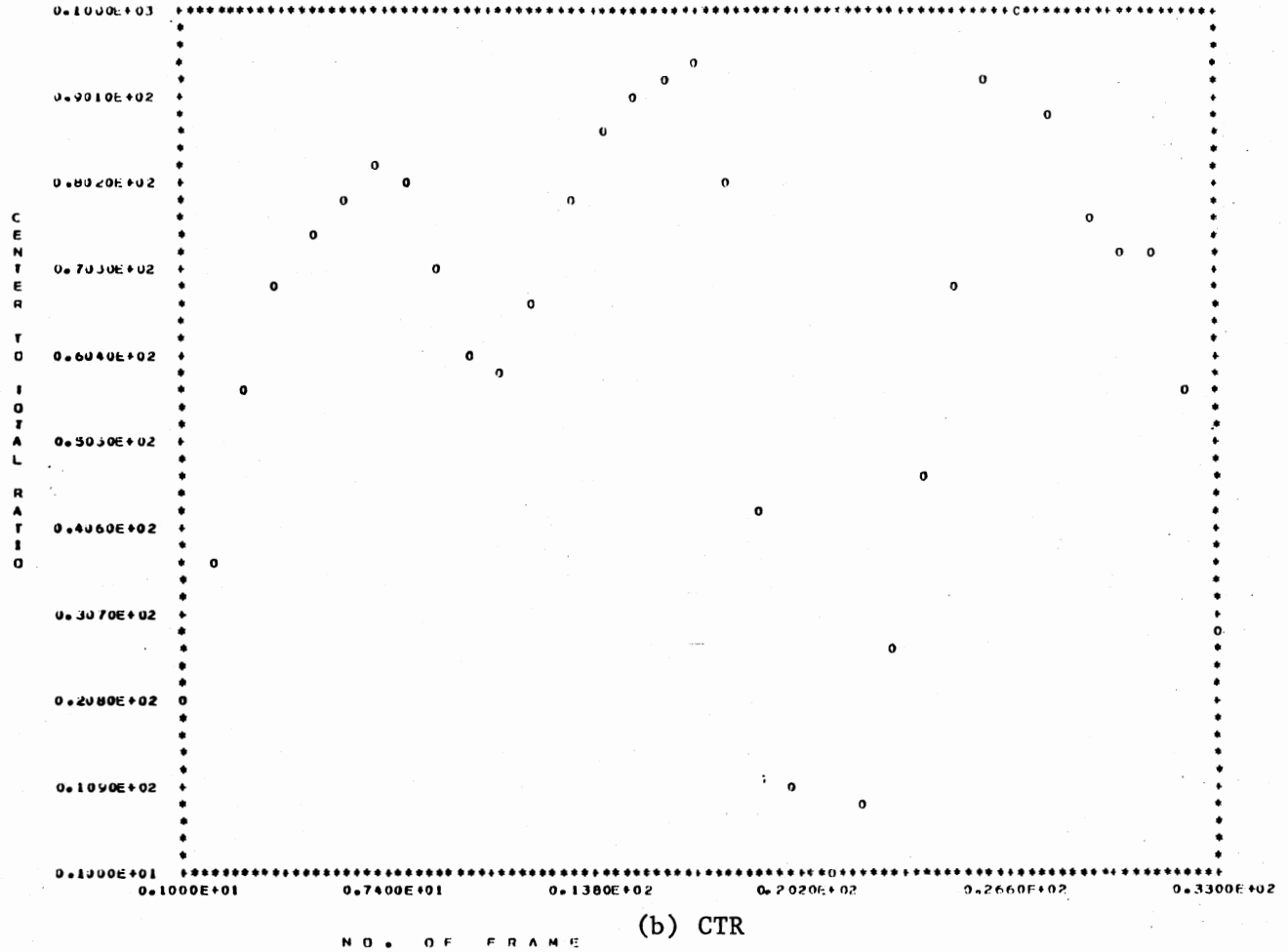
(c) BTR

Figure 54. (Continued)



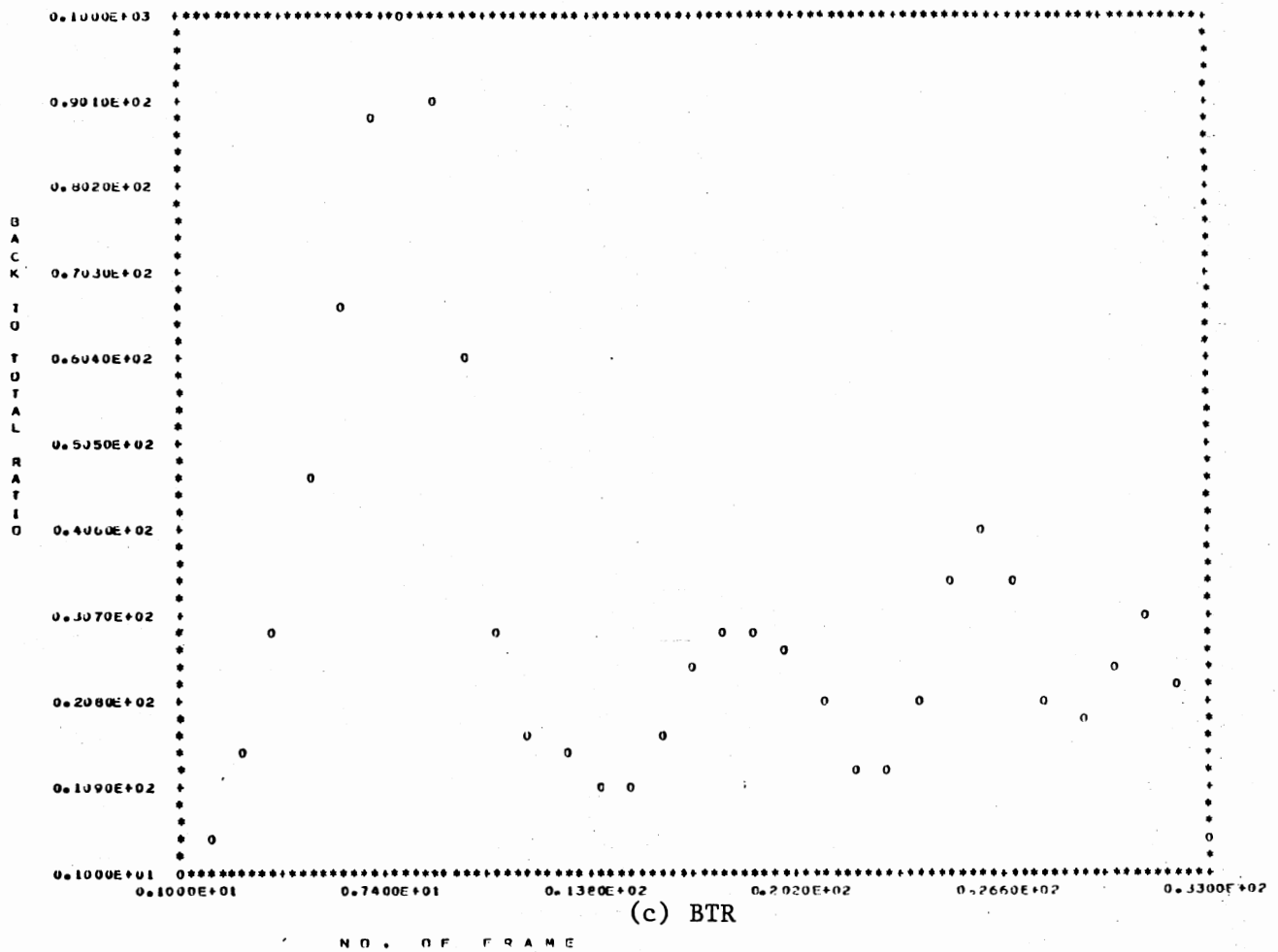
(a) FTR

Figure 55. Smoothed and Quantized Feature Parameters for Digit Six Spoken in American English



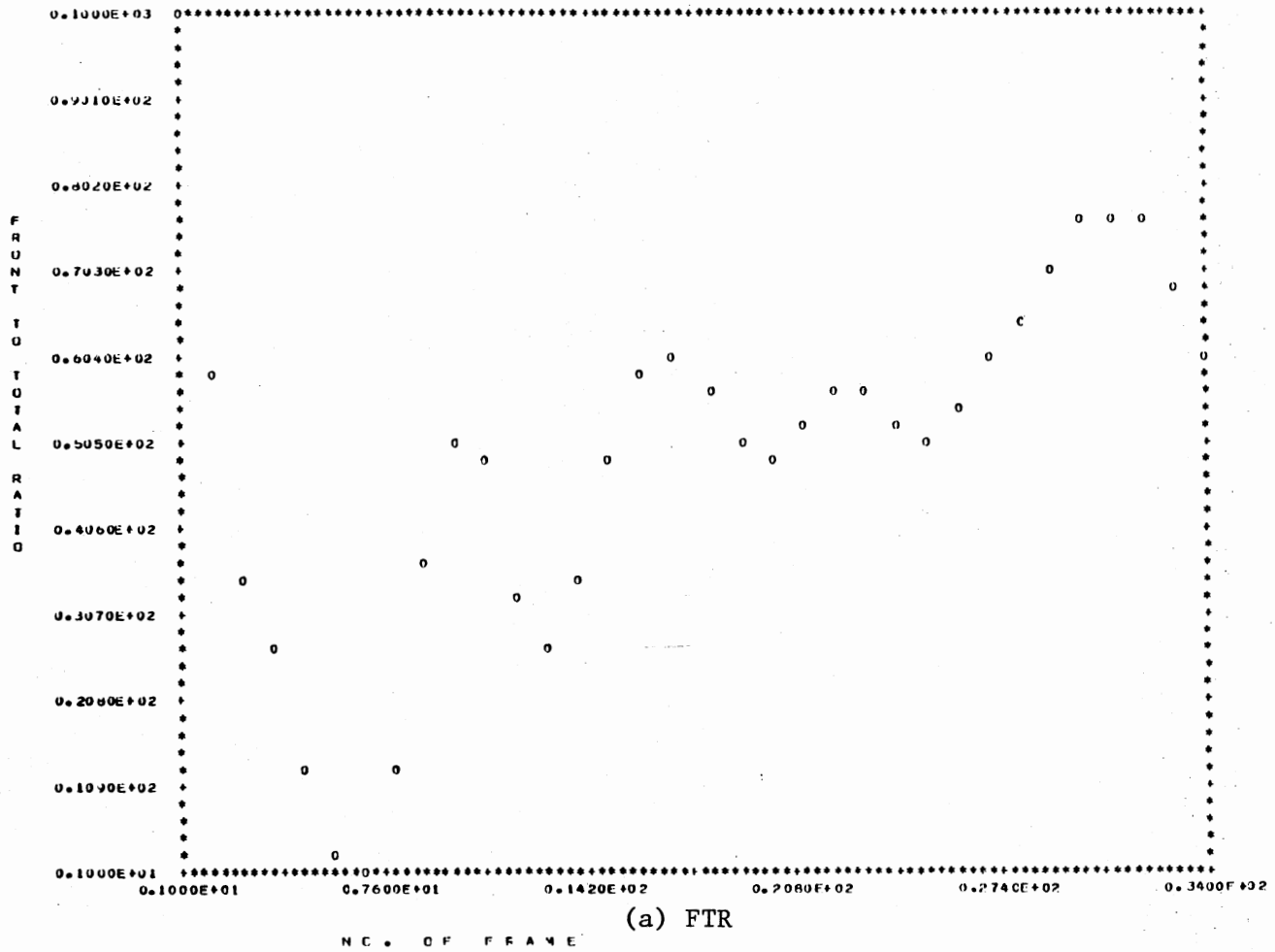
(b) CTR

Figure 55. (Continued)



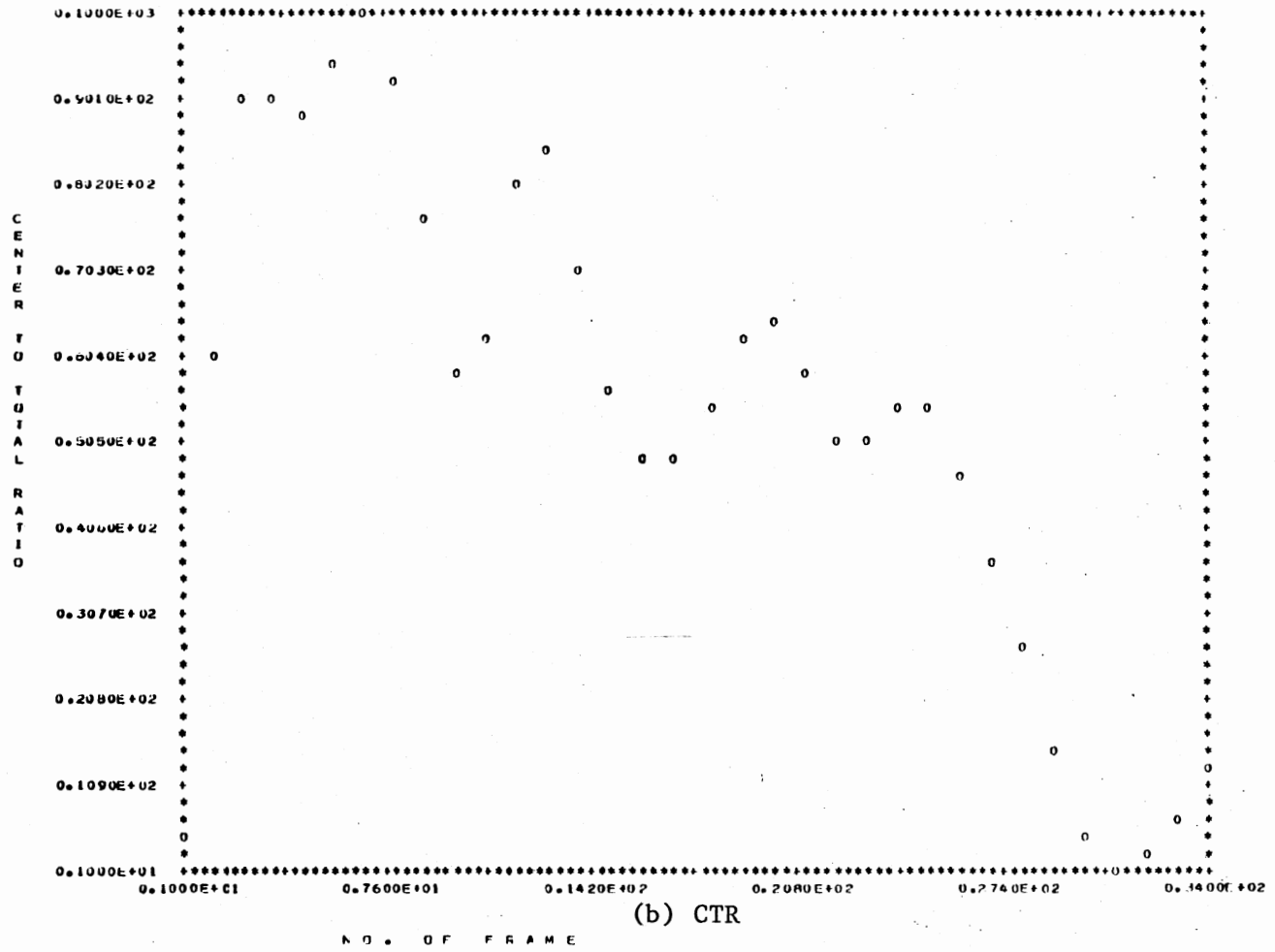
(c) BTR

Figure 55. (Continued)



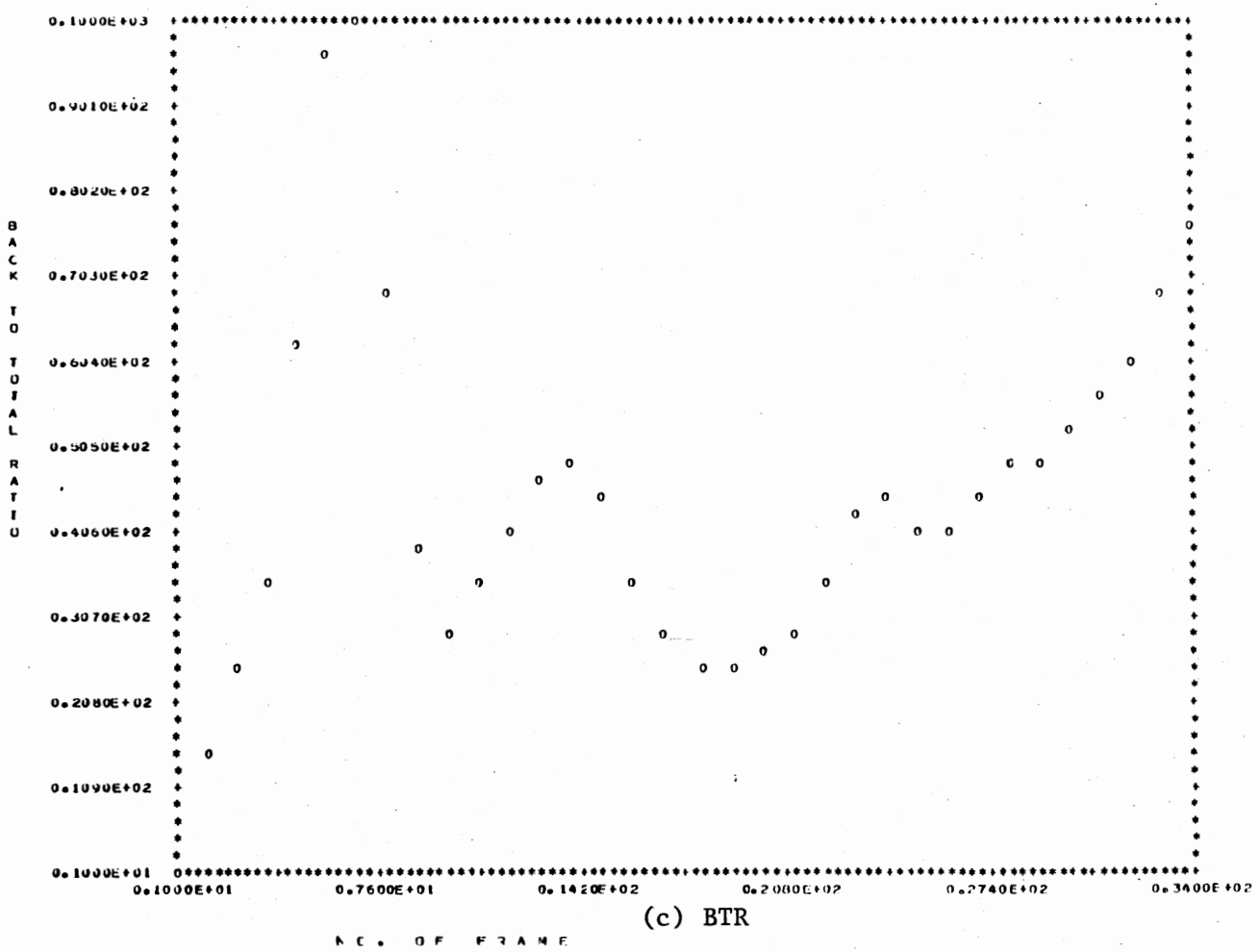
(a) FTR

Figure 56. Smoothed and Quantized Feature Parameters for Digit Seven Spoken in American English



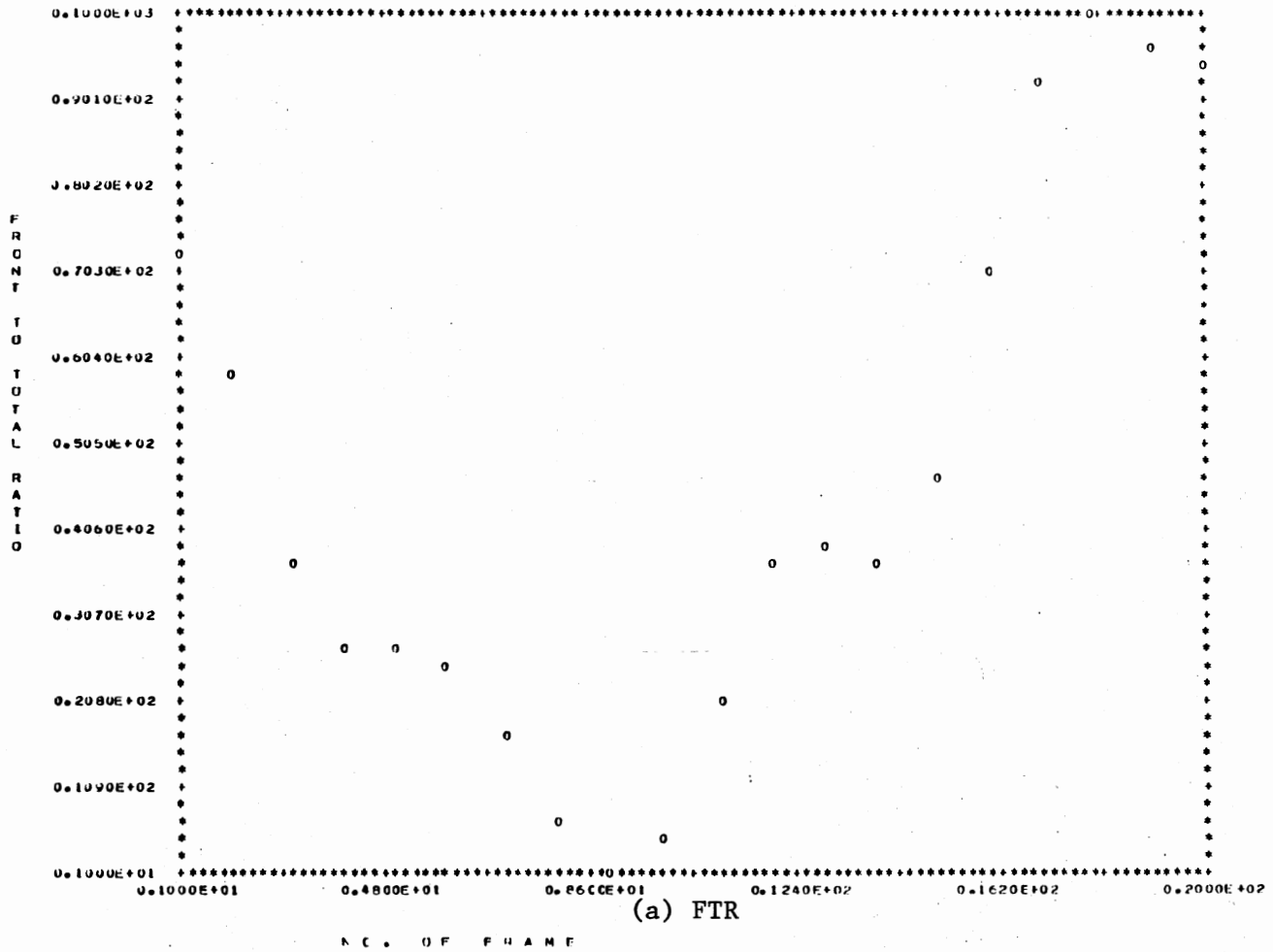
(b) CTR

Figure 56. (Continued)



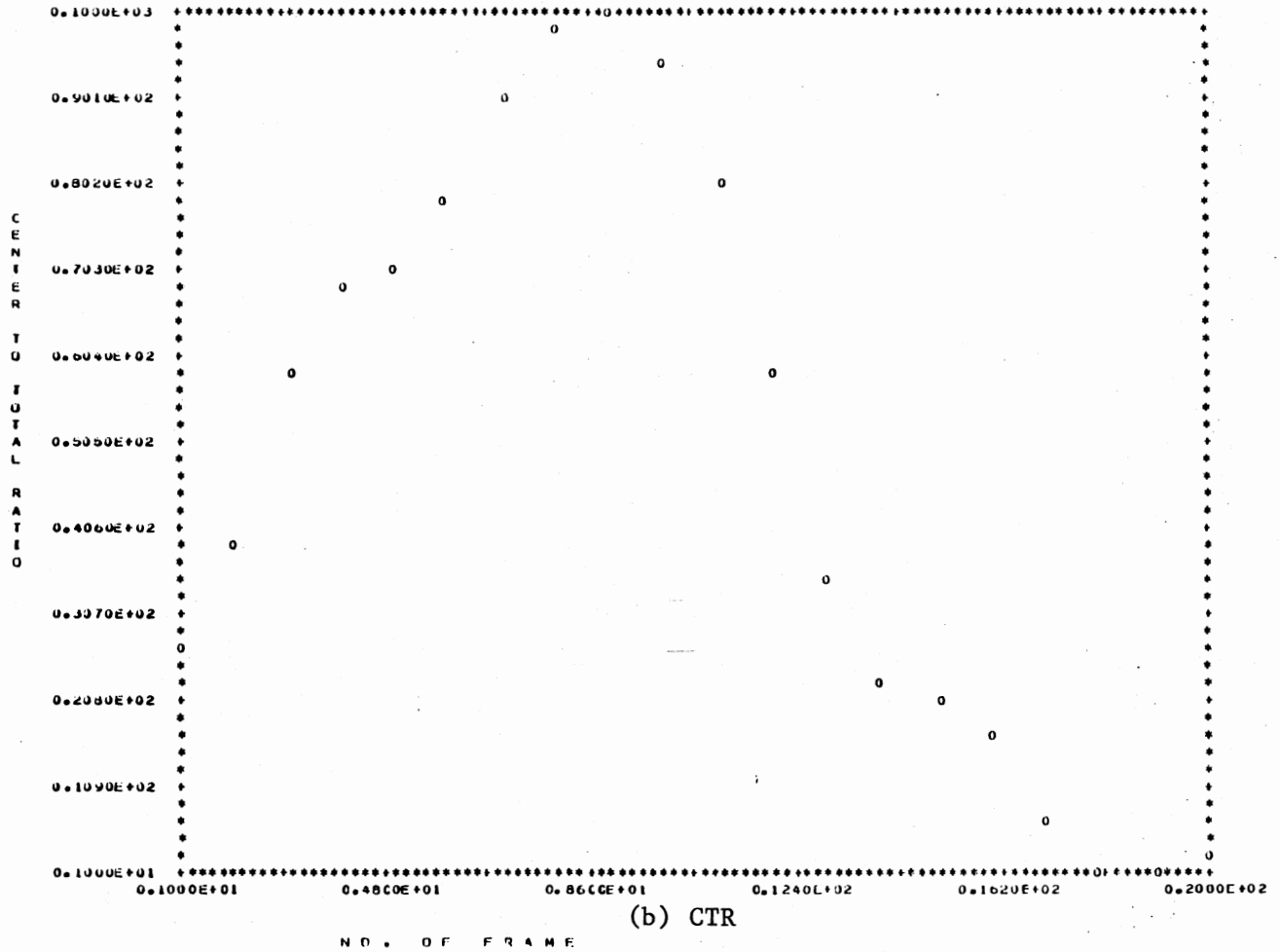
(c) BTR

Figure 56. (Continued)



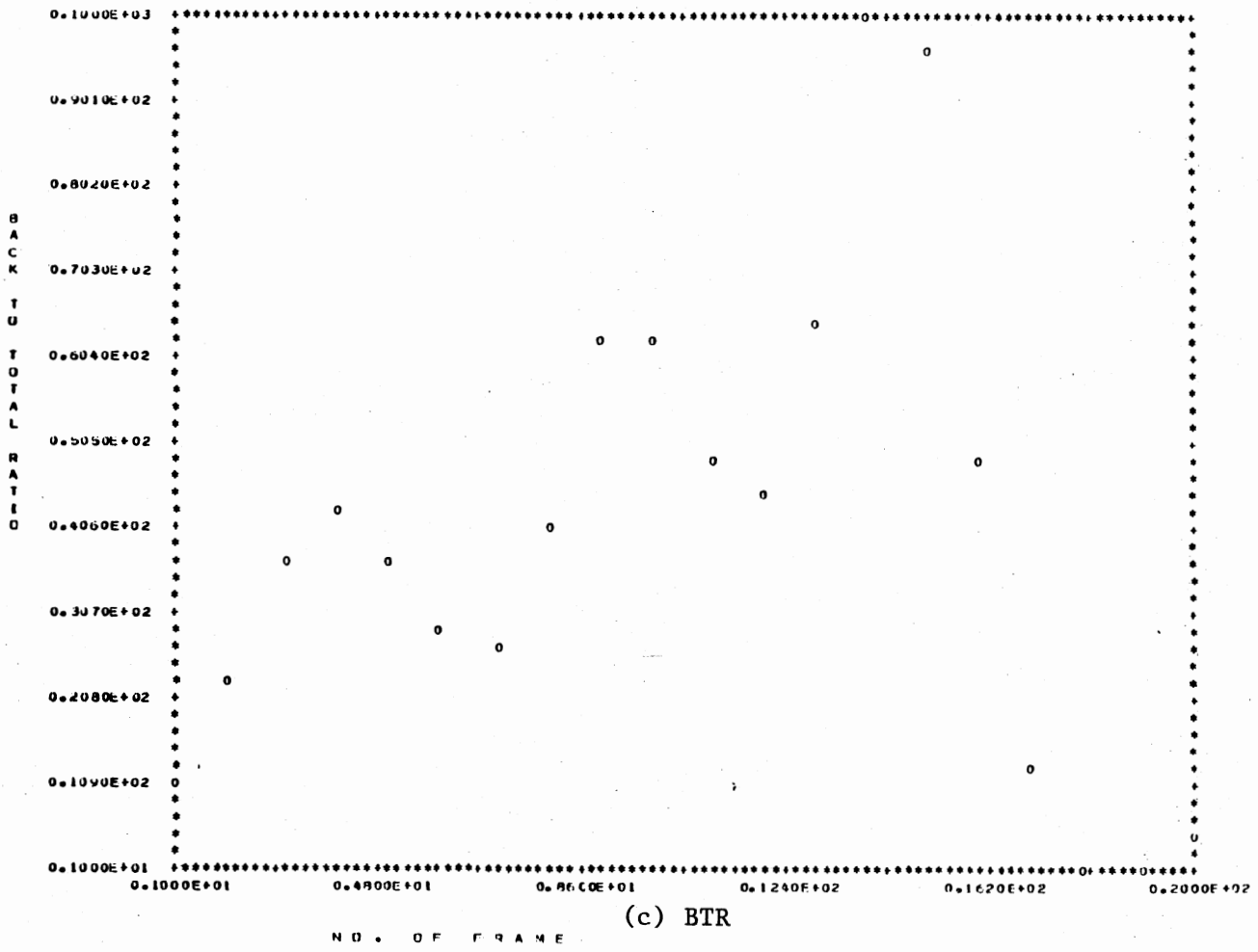
(a) FTR

Figure 57. Smoothed and Quantized Feature Parameters for Digit Eight Spoken in American English



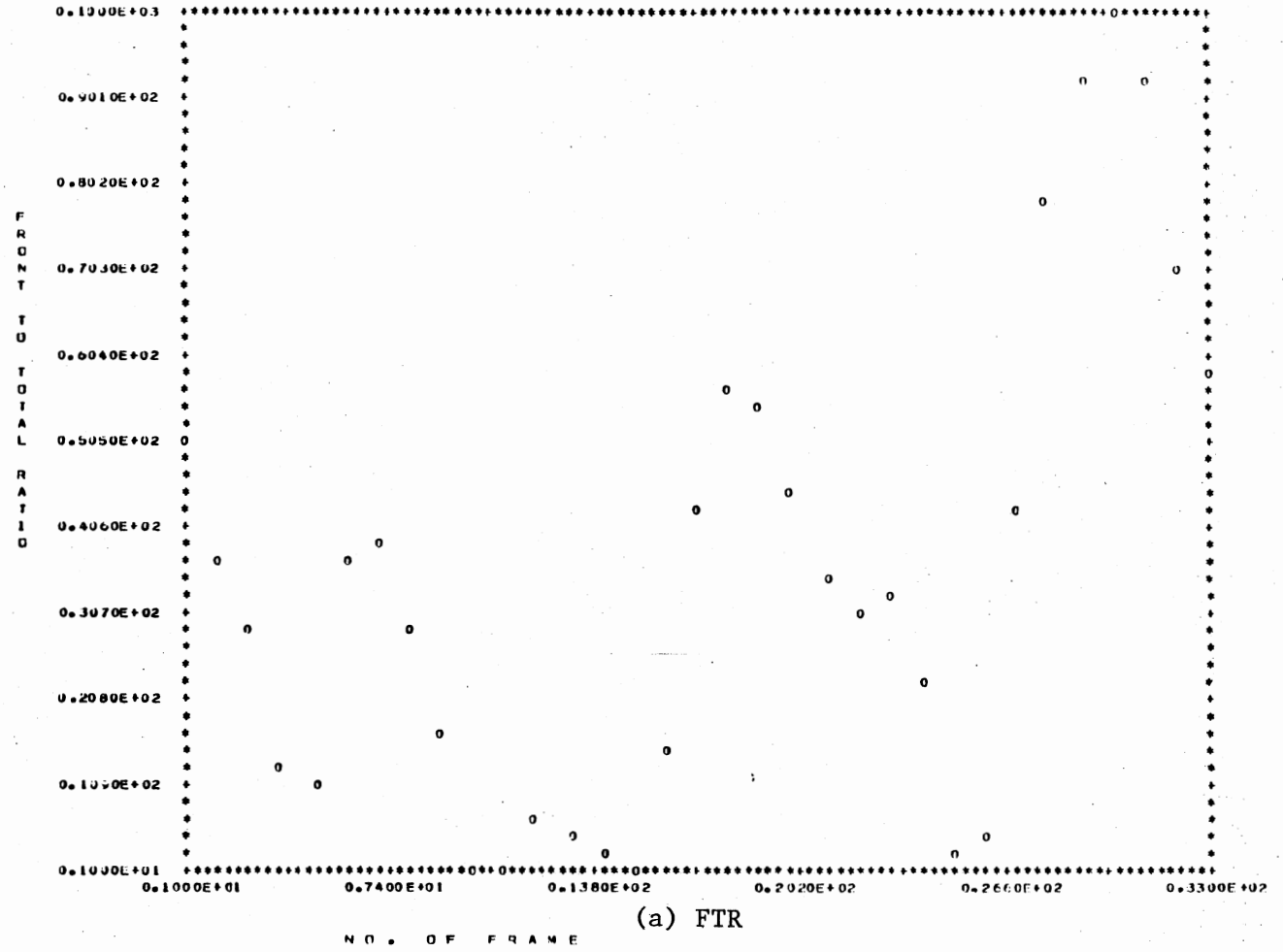
(b) CTR

Figure 57. (Continued)



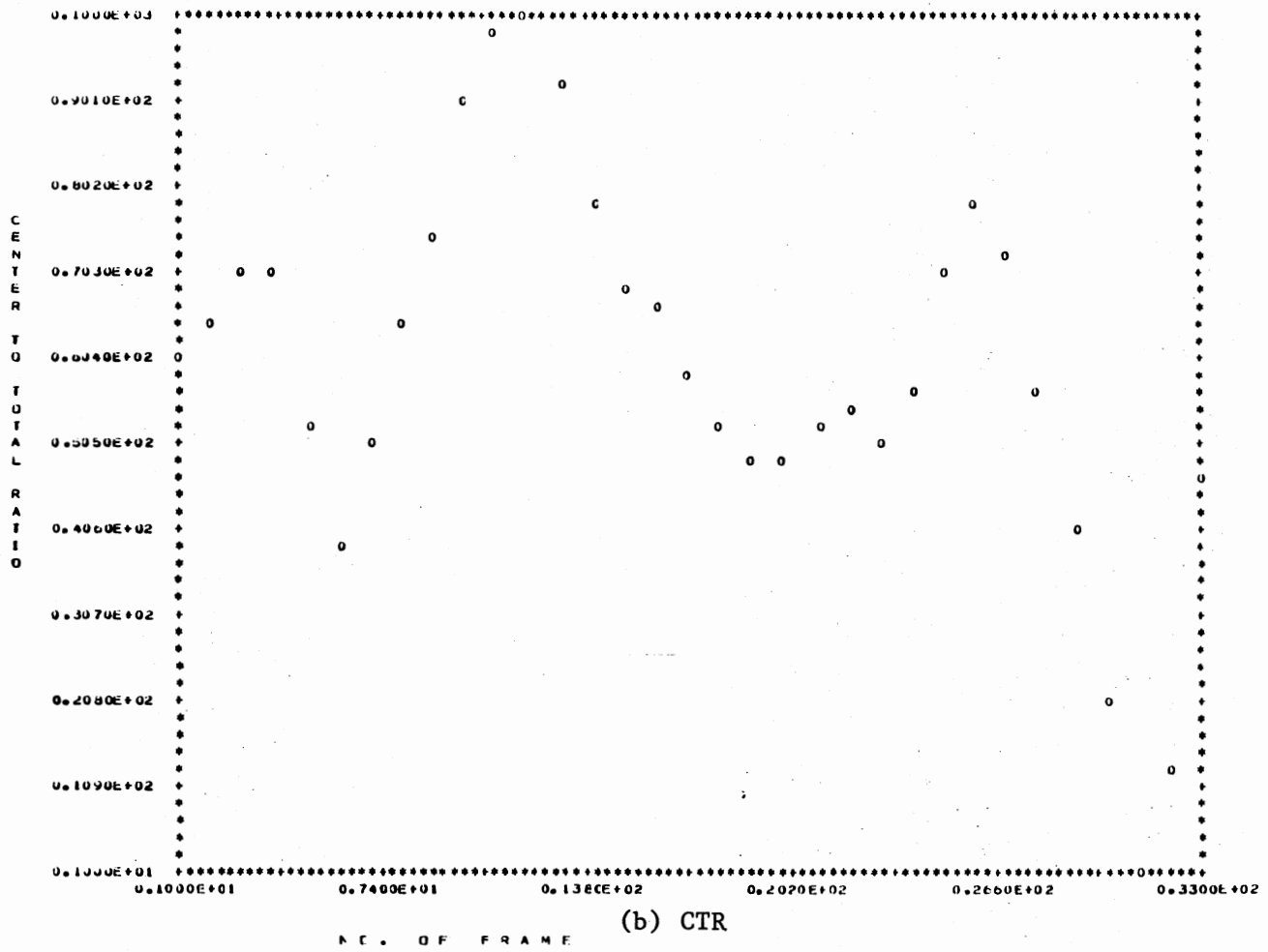
(c) BTR

Figure 57. (Continued)



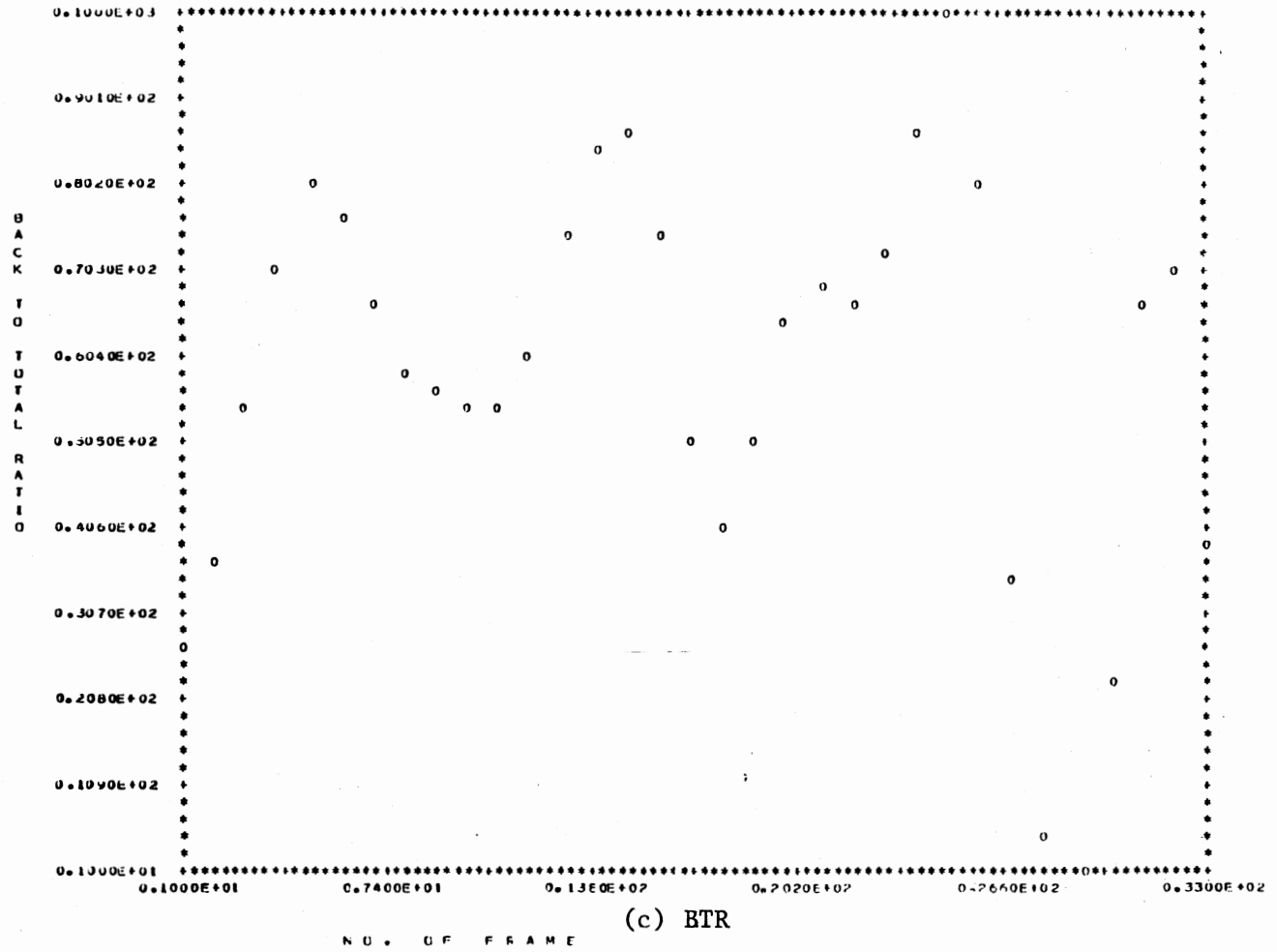
(a) FTR

Figure 58. Smoothed and Quantized Feature Parameters for Digit Nine Spoken in American English



(b) CTR

Figure 58. (Continued)



(c) BTR

Figure 58. (Continued)

schemes. In addition, the BTR is used to separate nasals from vowel segments, and the SFBR is used to separate turbulence noise segments from back vowel segments. The classification of each segment is stored for further usage.

The third stage in Figure 28 corresponds to the actual digit recognition. The discussion on peak ratios was presented earlier. A summary of the beginning, middle and ending sound classification of digits spoken in American English is tabulated (Table VII). This Table is derived from Table II. The uniqueness aspects of the classifications are used in the final decision. The decision algorithm is based upon using Table VII and sound classification at the beginning and the end region of a particular digit. If a decision cannot be made, then the middle region is referred for identification. For example, the digits zero, six, and eight have unique classification for beginning and ending sounds. These three digits are decided on the beginning and ending sound. It is interesting to point out that the digit seven classified as having two front vowels in the middle region. The digits one and nine have the same sound patterns for end regions. However, the middle region is different. In a similar manner, the sound classification of the remaining digits can be discussed.

Figure 59 shows the flow chart for the digit recognition as based upon the uniqueness aspects of Table VII. The beginning region is tested first for "VL" indication to separate zero and one and nine from the remaining digits. If the beginning sound is vowel-like, then the middle region is tested. A middle vowel to front vowel indication is made to distinguish between one and nine. Middle

TABLE VII

SEQUENCE OF SOUND CLASS REGIONS OF DIGITS SPOKEN IN AMERICAN ENGLISH

Digit	RMS and BTR Decision		
	Beginning	Middle	End
/zIro/	VL	FV/BV	V
/wʌn/	VL	MV	VL
/tu/	NV	FV	V
/θri/	NV	VL	V
/fɔr/	NV	BV/MV	VL
/faIv/	NV	MV/FV	VL
/sIks/	NV	FV	NV
/sevɛn/	NV	FV/FV	VL
/eIt/	V		NV
/naIn/	VL	MV/FV	VL

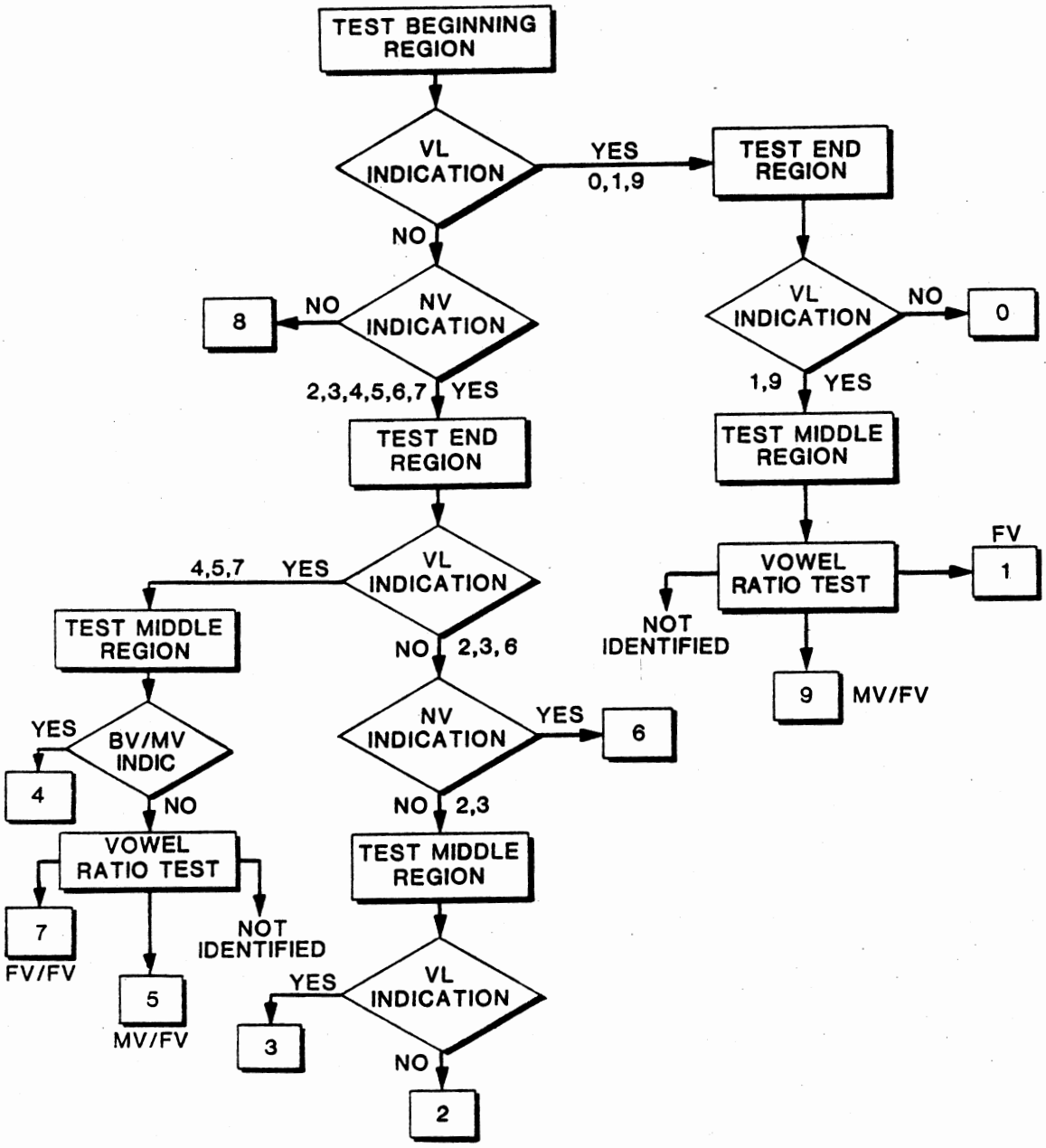


Figure 59. Decision Tree for Digit Identification Scheme for American English Language

vowel to front vowel indicated that digit nine is spoken. A front vowel indicates that digit one was spoken. For the other digits, the beginning region is tested for non-vowel "NV" segments. The remaining flow chart can be identified in a similar manner.

The above algorithm was tested for seven speakers both male and female. The accuracy rate was about 75 percent. Some of the speakers were international students. The results were much better for Americans and are accurate to about 95 percent, especially when the spoken digits were intact. By this method the digits two and four have the major problem in recognition. The reason for this is that both of these digits have only one essential vowel in the center region giving almost similar energy contour for most speakers. The digit four has a back vowel in the center region followed by mid-vowel in the end region since the semivowel /r/ is not pronounced distinctly by the Americans and is stressed by internationals. If both paths in Figure 28 give the same digit result, then the recognition is completed. If the two paths give different results, or that no decision has been made, then the following procedure is used.

Pattern Recognition Scheme

The final step in the recognition scheme is based on the cross-correlation coefficients in Equation (43) in Chapter III. The correlation of digits i and j are considered for the digits zero through nine. For the seven speakers, correlation tables are constructed. Some are given in Tables VIII - XI. Since the seven sets of digits are spoken by seven people having different dialects and accents, the

TABLE VIII

RMS ENERGY CORRELATION TABLE FOR DIGITS SPOKEN IN AMERICAN ENGLISH
(MAXIMUM NUMBER OF FRAMES USED)

SET NUMBER : 4

RMS CORRELATION

ZERO	ONE	TWO	THREE	FOUR	FIVE	SIX	SEVEN	EIGHT	NINE
1.000	0.206	0.389	0.003	0.224	0.033	0.724	-0.004	-0.085	-0.016
0.206	1.000	0.280	0.186	0.464	0.003	0.391	0.017	-0.116	0.410
0.389	0.280	1.000	0.151	0.693	0.292	0.614	0.395	0.622	0.605
0.003	0.186	0.151	1.000	0.253	0.039	-0.057	0.115	0.147	0.076
0.224	0.464	0.693	0.253	1.000	0.147	0.677	0.679	0.212	0.800
0.033	0.003	0.292	0.039	0.147	1.000	0.156	-0.057	0.249	0.447
0.724	0.391	0.614	-0.057	0.677	0.156	1.000	0.384	0.062	0.444
-0.004	0.017	0.395	0.115	0.679	0.057	0.384	1.000	0.239	0.617
-0.085	-0.116	0.622	0.147	0.212	0.249	0.062	0.239	1.000	0.287
-0.016	0.410	0.605	0.076	0.800	0.447	0.444	0.617	0.287	1.000

TABLE IX

RMS ENERGY CORRELATION TABLE FOR DIGITS IN AMERICAN ENGLISH
(MAXIMUM NUMBER OF FRAMES USED)

SET NUMBER : 5

RMS CORRELATION

ZERO	ONE	TWO	THREE	FOUR	FIVE	SIX	SEVEN	EIGHT	NINE
1.000	-0.095	0.131	0.786	0.400	0.569	0.559	0.745	0.573	0.599
-0.095	1.000	0.650	0.047	0.428	-0.155	0.593	0.173	0.493	0.651
0.131	0.650	1.000	0.044	0.584	-0.147	0.690	0.280	0.776	0.573
0.786	0.047	0.044	1.000	0.379	0.762	0.394	0.581	0.410	0.715
0.400	0.428	0.584	0.379	1.000	0.414	0.225	0.174	0.481	0.613
0.569	-0.155	-0.147	0.762	0.414	1.000	0.091	0.419	0.092	0.457
0.559	0.593	0.690	0.394	0.225	0.091	1.000	0.598	0.747	0.665
0.745	0.173	0.280	0.581	0.174	0.419	0.598	1.000	0.300	0.408
0.573	0.493	0.776	0.410	0.481	0.092	0.747	0.300	1.000	0.635
0.599	0.651	0.573	0.715	0.613	0.457	0.665	0.408	0.635	1.000

TABLE X

BTR CORRELATION TABLE FOR DIGITS SPOKEN IN AMERICAN ENGLISH
(MAXIMUM NUMBER OF FRAMES USED)

SET NUMBER : 4

BTR CORRELATION

ZERO	ONE	TWO	THREE	FOUR	FIVE	SIX	SEVEN	EIGHT	NINE
1.000	-0.103	0.027	0.157	-0.333	-0.125	-0.347	-0.002	-0.119	0.192
-0.103	1.000	-0.639	-0.298	-0.724	0.489	0.143	-0.239	-0.405	0.428
0.027	-0.639	1.000	0.045	0.423	0.196	-0.465	-0.524	0.320	0.139
0.157	-0.298	0.045	1.000	-0.123	-0.249	-0.224	-0.437	-0.532	-0.169
-0.333	-0.724	0.423	-0.123	1.000	0.648	0.244	-0.138	0.199	0.134
-0.125	0.489	0.196	-0.249	0.648	1.000	0.360	0.006	0.351	0.519
-0.347	0.143	-0.465	-0.224	0.244	0.360	1.000	0.429	0.165	0.145
-0.002	-0.239	-0.524	-0.437	-0.138	0.006	0.429	1.000	0.003	0.071
-0.119	-0.405	0.320	-0.532	0.199	0.351	0.165	0.003	1.000	0.353
0.192	0.428	0.139	-0.169	0.134	0.519	0.145	0.071	0.353	1.000

TABLE XI

BTR CORRELATION TABLE FOR DIGITS SPOKEN IN AMERICAN ENGLISH
(MAXIMUM NUMBER OF FRAMES USED)

SET NUMBER : 5

BTR CORRELATION

ZERO	ONE	TWO	THREE	FOUR	FIVE	SIX	SEVEN	EIGHT	NINE
1.000	-0.400	-0.547	-0.003	0.318	-0.067	0.032	-0.062	0.158	-0.241
-0.400	1.000	0.339	0.161	0.568	0.354	0.426	-0.100	0.338	-0.182
-0.547	0.339	1.000	-0.099	0.787	0.532	0.264	-0.167	0.835	0.225
-0.003	0.161	-0.099	1.000	0.007	-0.268	0.133	0.085	0.153	-0.496
0.318	0.568	0.787	0.007	1.000	0.922	0.508	-0.034	0.767	0.224
-0.067	0.354	0.532	-0.268	0.922	1.000	0.444	-0.362	0.698	-0.021
0.032	0.426	0.264	0.133	0.508	0.444	1.000	0.284	0.212	-0.108
-0.062	-0.100	-0.167	0.085	-0.034	-0.362	0.284	1.000	0.144	0.111
0.158	0.338	0.835	0.153	0.767	0.698	0.212	0.144	1.000	0.363
-0.241	-0.182	0.225	-0.496	0.224	-0.021	-0.108	0.111	0.363	1.000

correlation data gives in some way, the general pattern for the correlation of the spoken digits. Tables VIII - XI indicate that, some digits are heavily correlated.

In the following, pattern matching is used to classify the digits that have not been recognized by the procedure discussed earlier. Recognition of these spoken digits is achieved by cross-correlating the pattern of the unknown spoken digit with the stored test patterns of the digits zero through nine, as shown by Figures 60 and 61. The standard patterns used are that of the RMS and BTR parameters. The digit which has the highest correlation is selected. This system utilizes no linguistic information, but uses procedures of acoustic characteristics impeded in the RMS energy and BTR parameter derived from the equivalent area function. This completes the digit recognition scheme.

The above method was applied for sample digits and the results were very good as shown by Tables XII and XIII. However, more sample digits have to be tested before the recognition rate can be given. Figures 60 and 61 can be used for further research for testing a digit recognition scheme.

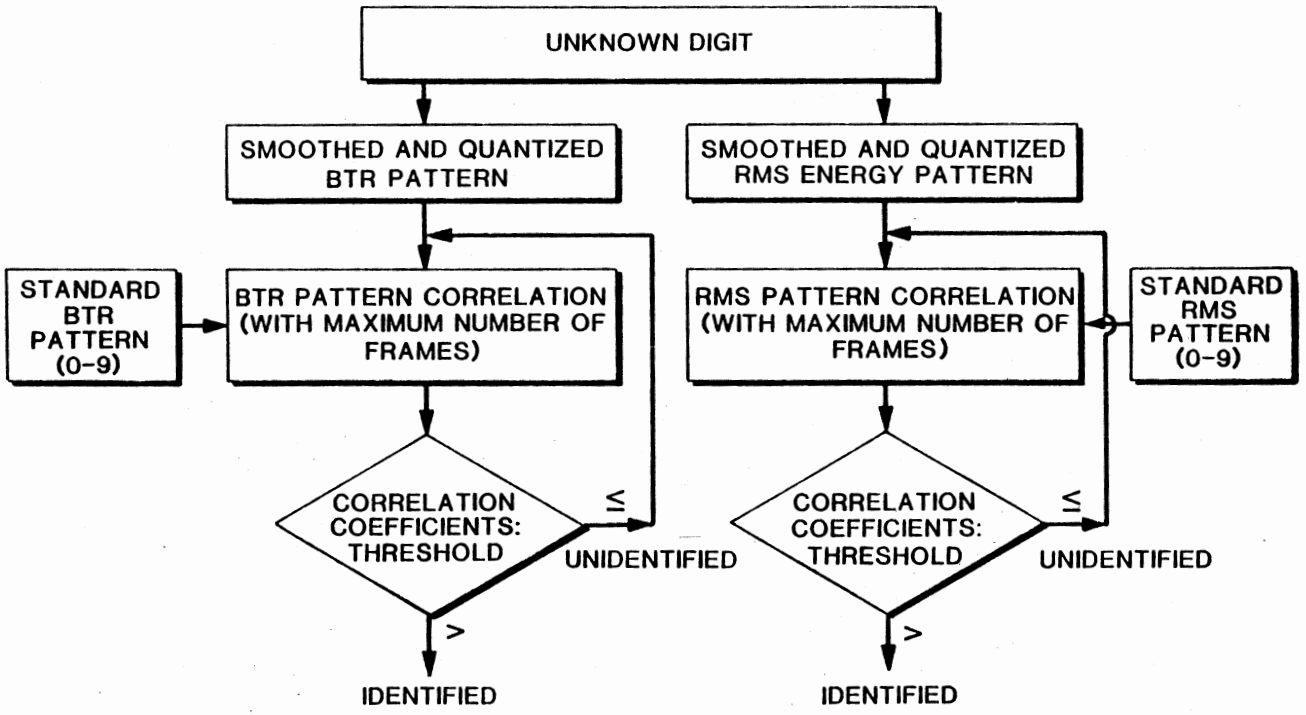


Figure 60. RMS and BTR Correlation Flow Diagram for Digit Recognition

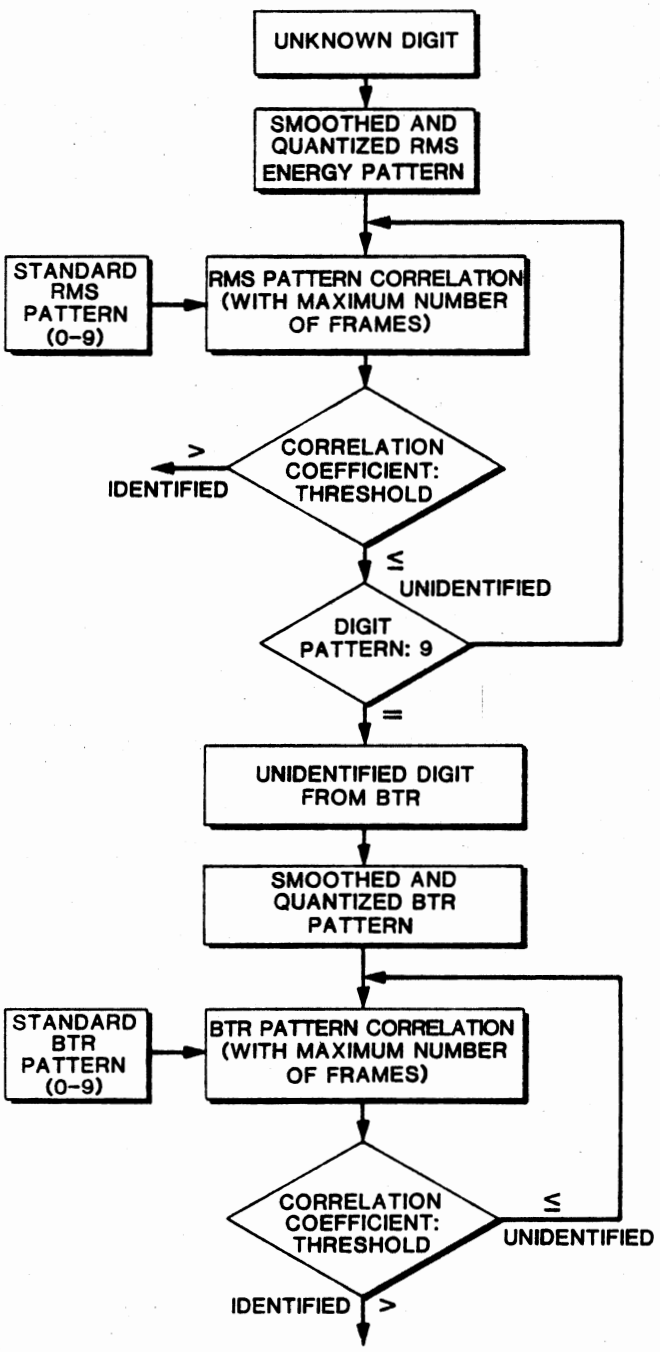


Figure 61. Future RMS and BTR Correlation Flow Diagram for an Efficient Digit Recognition System

TABLE XII

RMS ENERGY CORRELATION TABLE FOR TWO SETS OF DIGITS SPOKEN IN AMERICAN ENGLISH BY TWO DIFFERENT SPEAKERS

RMS CORRELATION

ZERO	ONE	TWO	THREE	FOUR	FIVE	SIX	SEVEN	EIGHT	NINE
0.791	-0.107	0.076	0.149	0.379	-0.040	0.212	0.471	-0.130	0.251
-0.107	0.681	0.172	0.584	0.195	0.616	0.355	0.229	0.961	-0.005
0.076	0.172	0.961	0.841	0.476	0.781	0.988	0.901	0.045	0.604
0.149	0.584	0.841	0.949	0.504	0.856	0.887	0.818	0.489	0.571
0.379	0.195	0.476	0.504	0.855	0.185	0.614	0.805	-0.166	0.828
-0.040	0.616	0.781	0.856	0.185	0.960	0.818	0.698	0.663	0.242
0.212	0.355	0.988	0.887	0.614	0.818	0.987	0.858	0.287	0.470
0.471	0.229	0.901	0.818	0.805	0.698	0.858	0.793	0.342	0.451
-0.130	0.961	0.045	0.489	-0.166	0.663	0.287	0.342	0.972	-0.077
0.251	-0.005	0.604	0.571	0.828	0.242	0.470	0.451	-0.077	0.954

TABLE XIII

BTR CORRELATION TABLE FOR THE SAME TWO SETS OF DIGITS USED IN TABLE XII

			BTR CORRELATION						
ONE	TWO	THREE	FOUR	FIVE	SIX	SEVEN	EIGHT	NINE	TEN
0.662	0.078	-0.491	-0.070	-0.678	-0.553	-0.375	-0.400	0.088	-0.403
0.078	0.745	-0.224	-0.267	0.165	0.144	-0.136	-0.021	0.677	0.283
-0.491	-0.224	0.758	0.385	0.630	0.143	0.530	0.338	0.310	-0.399
-0.070	-0.267	0.385	0.713	0.153	-0.082	0.620	-0.022	0.309	-0.569
-0.678	0.165	0.630	0.153	0.801	-0.047	0.305	0.580	0.336	-0.110
-0.553	0.144	0.143	-0.082	-0.047	0.832	0.526	0.160	0.650	0.550
-0.375	-0.136	0.530	0.620	0.305	0.526	0.709	-0.216	0.582	0.465
-0.400	-0.021	0.338	-0.022	0.580	0.160	-0.216	0.656	0.307	-0.143
0.088	0.677	0.310	0.309	0.336	0.650	0.582	0.307	0.762	0.377
-0.403	0.283	-0.339	-0.559	-0.110	0.550	0.465	-0.143	0.377	0.756

CHAPTER V

ACOUSTIC PHONEMIC DIGIT RECOGNITION SCHEME FOR DIGITS SPOKEN IN ARABIC

Introduction

In the last chapter, the phonemic digit recognition was discussed based upon acoustic patterns of phonemes. Obviously, a similar approach can be used in recognizing digits in other languages. The only difference is that the phonemes for a given digit in two languages will generally be different. Also, some of the phonemes used in some languages may not be in other languages. For example, the glottal phoneme in Arabic is not in English. In this chapter, the ideas in Chapter IV are extended for digits spoken in Arabic and is shown that the phonemic Arabic digit recognition can be accomplished in the general frame work of speech processing.

Arabic Phonemes

There are about 57 phonemes in Arabic Language, but only 24 phonemes are used in the digits 0 to 9 in Arabic. Table XIV shows the Arabic alphabet and its equivalent translation into American English [70]. Also Figure 9 and Table IV classify the phonemes used in digits spoken in Arabic. There are ten vowels, three of which are short vowels and three can be long vowels, where it is really pronounced long as in /θaɫâθðh/ (three), i.e. the phoneme /â/ is a long vowel. The short

TABLE XIV

TRANSLITERATION OF ARABIC WORDS AND NAMES

ا ء	Consonantal sound	a	ط	.....	t
ا	Long vowel *	ā	ظ	.....	ẓ
ب	.....	b	ع	.....	ʿ (Inverted apostrophe)
ت	.....	t	غ	.....	ġ
ث	.....	th	ف	.....	f
ج	.....	j	ق	.....	q
ح	.....	h	ك	.....	k
خ	.....	kh	ل	.....	l
د	.....	d	م	.....	m
ذ	.....	ẓ	ن	.....	n
ر	.....	r	ه ة	.....	h
ز	.....	z	و	consonant	w
س	.....	s	و	long vowel *	ū
ش	.....	sh	و	diphthong	au
ص	.....	s	ي	consonant	y
ض	.....	dh	ي	long vowel *	ī
			ي	diphthong	ai

Short vowels: ـَ (*fatḥa*) a
ـِ (*kasra*) i
ـُ (*dhamma*) u

vowels are /a/, /i/, and /u/. For example /a/ as in /ωahid/, (one). The long vowels are /â/, /û/, and /î/, as shown by Figure 9 and 11. Dipthongs do not appear in Arabic digits.

There are nine basic sounds which fundamentally differ from any sound in American English. These are /ḥ/, /kh/, /ṣ/, /dh/, /ṭ/, /Ẓ/, /ρ/ or /Ḥ/, /ġ/, and /q̣/. Since these phonemes are very difficult to be produced by a non-Arabic speaking person, a brief description of the articulatory problems will be given below for interest.

The Emphatics

This group of Arabic sounds share some common features: 1) they are all produced with the back of the tongue raised towards the back roof of the mouth; 2) all of them have non-emphatic counterparts, from which they should be clearly distinguished; and 3) in producing echo, these sounds produce more echo and 'thickness of voice' than their counterparts.

Both vowel and consonant productions may involve the simultaneous activity of many articulators. For vowels, the tongue, velum, lips, and larynx are all in operation. It is difficult to say that one articulator's movements is more important than another's. In languages other than American English, the lips and tongue may form simultaneous vocal-tract stoppages [19][21]. Multiple articulation is important in Arabic.

Velarized Phonemes [67]

The Arabic /ṣ/ is a velarized alveolar /s/. It is alveolar, which means that it is produced not at the teeth, like the /s/ as in 'seen',

but further back in the mouth. It is velarized, which means that the back part of the tongue is made tense, with some raising up toward the soft palate or velum, as illustrated by Figure 62, giving the /s/ sound a velar effect. Thus the phoneme /s/ is quite different from the phoneme /s/.

The Arabic phoneme /dh/ or /d/ pronounced as 'daad', has no American English equivalent. It is a velarized voiced alveolar stop. From another view point, it is like the phoneme /d/, with the addition of velarization [67]. It should be pointed out that one side of the tongue gets closer to the side of the mouth that touching the molars.

The phoneme /t/, pronounced as 'taa' is the emphatic counterpart of the phoneme /t/. The consonant /t/ is a voiceless alveolar stop. It differs from the plain American English /t/, which is often aspirated (i.e., it is produced with a slight puff of breath); while /t/ is not aspirated [69].

The phoneme /ð/ pronounced as 'thaa' is a velarized voiced interdental fricative; it is like the American phoneme /ð/ as in that, with the addition of velarization (the tensing up of the base of the tongue).

Throat and Back-of-Mouth Sounds

The phoneme /ρ/, i.e. /A/ pronounced as 'Aayn' is not generally used in the English Language. It is a voiced pharyngeal fricative. The 'Aayn' is similar to the sound one produces sometimes to express feeling of being strangled. This sound is produced deep in the throat contracting the muscles in there and forcing the air through, like

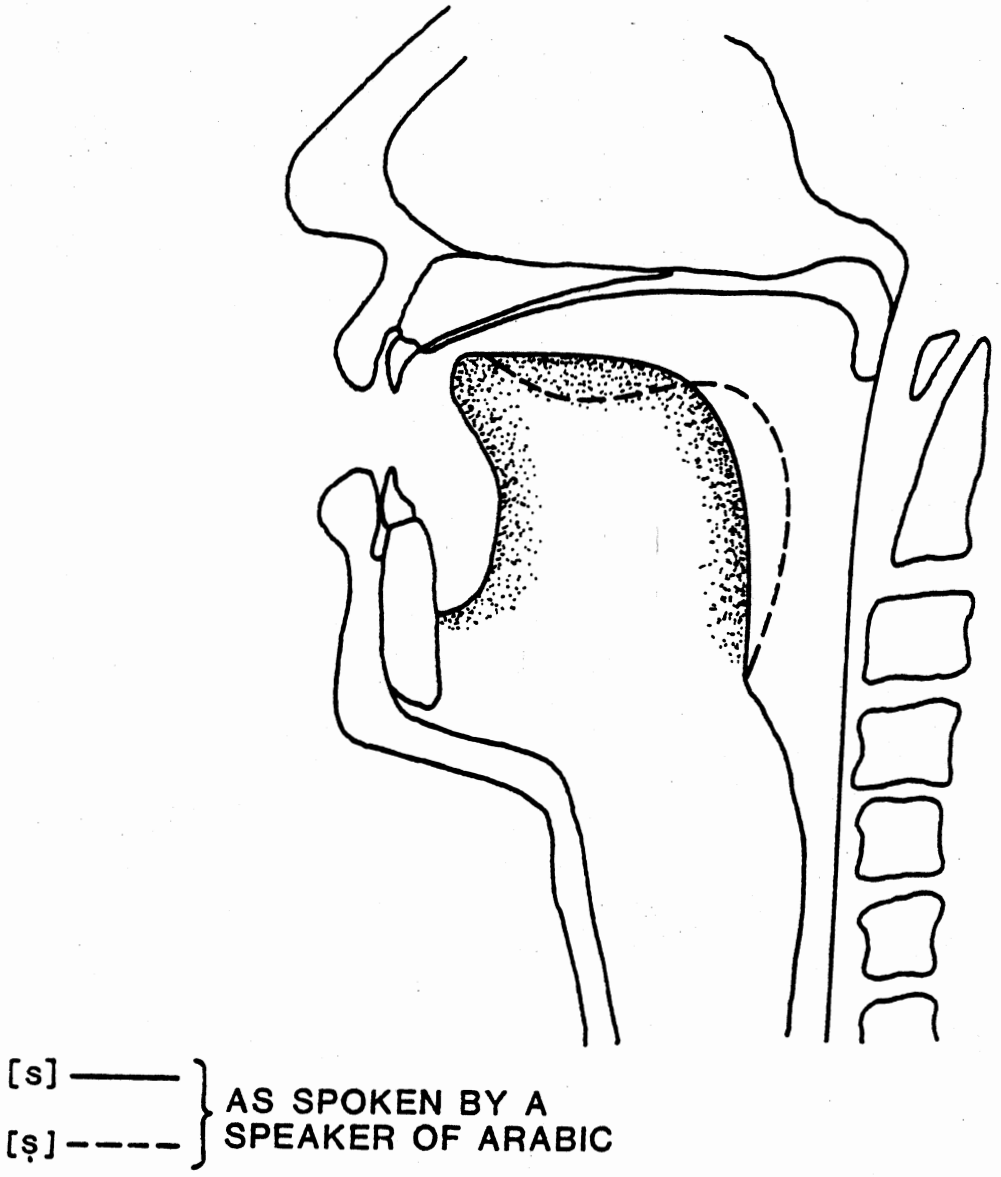


Figure 62. Tongue Position for the Emphatic-Nonemphatic [s] vs [s] Fricative Consonants in Arabic

'arbaɋðh' in Arabic for digit four. It is important to distinguish this sound from 'Haa' as it causes some echo in the mouth, which could be clearly noticed if one closes one's ears when producing the sound 'Aayn'.

The phoneme /ġ/, pronounced as 'gayn' is very close in pronunciation to standard French 'r'. It is like the noise one makes when gargling without water. It is a voiced velar fricative. This sound should be distinguished from American English 'g' for which the air stream is completely interrupted in the area of the back roof of the mouth.

The phoneme /q̇/, pronounced as 'qaaf' is produced in the area deeper and more to the back than k. It has a very clear echo in the mouth, as distinct from k or g in American English. The closest sound in English is heard in pronunciation of words like 'caught', but it is still deeper.

The phoneme /ḥ/ like /ωâḥid/ for Arabic digit one; it is a voiceless pharyngeal fricative. It is produced with the base of the tongue near the back of the pharynx (throat) and the pharynx walls strongly constricted. There is no contact whatsoever between the base of the tongue and the velum.

The phoneme /hk/ is a voiceless velar fricative. It is produced by narrowing the passageway between the back of the tongue and the velum, so creating friction as the air passes through; the vocal cords are at rest. The digit (five) in Arabic is /khʌmaðh/ has the phoneme /kh/ which is sometimes written in the form /x/ which has nothing to do with the American English sound x. The phoneme transcribed as /kh/

sound is produced with a continuous stream of air, as opposed to k which has an interrupted stream of air in pronunciation. It is closer to German ch in 'nacht', Scotch 'ch' in 'loch' and Latin American pronunciation of 'j' in 'mejor'.

Minor Differences

The phoneme /r/, pronounced as 'raa' should be distinguished from its counterpart in American English. In Arabic, it is produced by the tip of the tongue rapidly touching the alveolar ridge (or the area just over the gums of the upper front teeth). Occasionally, we hear a very similar sound in the pronunciation of the t and d in American English 'latter' and 'ladder'. It is also similar to British English r in 'very' and Spanish r too. The tip of the tongue is not curled when pronouncing the phoneme /r/ as in /ṣefr/.

The phoneme /ɫ/, pronounced 'laam' as in digit /θaɫâθðh/ in Arabic, meaning three, has a unique case in Arabic. Usually it is like in western languages in general and American English /ɫ/ in words like 'leaf', 'lip', but not 'lot' or 'low' .

There is an important difference between the Arabic sound /ɫ/ and the American English /ɫ/. Most speakers of American English pronounce the phoneme /ɫ/ with back of the tongue raised somewhat toward the velum, resulting in a velarized /ɫ/. This velarized quality of the English /ɫ/ is especially noticeable at the end of the word. For example as in feel or bell. The Arabic /ɫ/ has a non-velarized or clear sound. The clear /ɫ/ results when the back of the tongue is relaxed and not raised.

The Arabic phoneme /s/ corresponds to the American English /s/ as in 'see'. The Arabic /s/ is dental pronounced with the tongue tip at the upper teeth as in the Arabic spoken digit /khʌmsəh/, i.e. five, other examples are shown in Table III and IV. It is interesting to point out that the American English /s/ is alveolar, pronounced slightly behind the teeth, giving slightly lower-pitched /s/.

The Arabic phoneme /h/ as in /θaɬâθəh/, is like the American English /h/ as in 'hat' and 'heat'; it is a voiceless glottal fricative. It is generally considered as whisper. The Arabic /h/ differs from the American English /h/ in the following ways: 1) it is pronounced with more force than in the English /h/; 2) it can be pronounced at the end of a syllable word; 3) also it can be held twice as long.

The Arabic phoneme /θ/, as in /θaɬâθəh/, digit three in Arabic, is like the American phoneme /θ/, as in digit /θri/ in American English. The only difference is that the phoneme /θ/ is followed by a front vowel in digit /θri/ spoken in English, while it is followed by a back vowel in the same digit spoken in Arabic. Similarly the phonemes /f/, /ω/, /d/, /n/, /m/, /b/ and /y/ have no noticeable differences when used only in digits spoken in Arabic.

Finally an important feature in Arabic phonology is that of germination, which means that the consonant is doubled in pronunciation as in the digit /sittəh/, i.e. six. Furthermore, the feature impeded in the spoken digit depends considerably on the degree of constriction and tongue hump position, as shown by Table IV and discussed in detail in Chapter II. In addition a simple comparison spoken by American English and Arabic is shown in Table XV.

TABLE XV

COMPARISON BETWEEN BEGINNING AND END OF ENGLISH AND ARABIC DIGITS

Digit	Start	End	Digit	Start	End
/zIro/	Voiced f	B.V	/serf/	UV.f	Semi Vowel
/wʌn/	Semi Vowel	Nasal	/wâhid/	Semi Vowel	Voiced Stop
/tu/	UV.f	FV-BV	/iθnân/	F.V	Nasal
/θri/	UV.f	F.V	/θalâθəh/	UV.f	M.V*
/fɔr/	UV.f	Semi Vowel	/arbaɸə/	M.V	M.V*
/faIv/	UV.f	Voiced f	/xʌmsəh/	Fricative	M.V*
/sIks/	UV.f	UV.f	/sIttəh/	UV.f	M.V*
/sevɛn/	UV.f	Nasal	/sʌbɸəh/	UV.f	M.V*
/eIt/	Dipthong	UV. Stop	/θamânyəh/	UV.f	M.V*
/naIn/	Nasal	Nasal	/tIsɸəh/	UV. Stop	M.V*

* These mid-vowels (M.V) are followed by a whisper phoneme /h/ as shown in Table IV.

Segmentation

The segmentation scheme, used previously in Chapter IV, is utilized for locating the boundaries of digits spoken in Arabic, because it is based on silence, voiced and unvoiced decisions. In addition, the end-point detection scheme discussed earlier is based on the assumption that all digits spoken in American English have no silence intervals among them, with the exception of digits zero and seven. However, the segmentation algorithm will account for the number of boundaries in digits spoken in Arabic provided that the number of spoken digits are known. In addition the number of phonemes for each digit must be known. This is because seven of the digits spoken in Arabic have silence interval in the middle region and the rest have voiced interval in the middle region. Table IV illustrates the sequence of sound classes for the digits spoken in Arabic. The middle region dips, for the digits that have silence region, can be seen from the smoothered RMS energy plot. Successful segmentation is achieved provided that the ZCR and energy threshold for silence, unvoiced and voiced segments was properly set. This method of segmentation is discussed in detail in Chapters III and IV.

Digit Recognition Flow Chart

Once the digit boundaries are located, the RMS energy per frame is computed for the spoken digit as discussed earlier and shown by the flow chart of Figure 28. The difference between English and Arabic digit recognition is that the digit phoneme tree is different. The

RMS energy is smoothed and then quantized to a maximum level of 100, which essentially normalizes the data. Following the flow chart, the smoothed quantized RMS peaks are computed and stored for the uttered digit. RMS plots are shown in Figures 63-83. For each uttered digit the ratio of the largest two peaks is calculated, so that it is always less than unity. As a first step towards phonemic digit recognition, the computed values for the peak ratios for each uttered digit, zero through nine is then stored. A threshold level is adjusted according to an empirical value. Tables XVI and XVII give the range for the largest two peaks and the value of their ratio. The computed ratio is first compared to the emperically established threshold. If the peak ratio of the unknown spoken digit does not match the established range, then the first idea based on RMS energy does not work. It has been found from measurements, based on five sets of data, that the actual digit is recognized more than half the time. The other part of the time it indicates that no digit is identified. It can be concluded that the peak ratio value is speaker dependent. This completes the first digit recognition idea.

Proceding in the second path of the flow chart in Figure 28, the signal is Hamming windowed, where the window length is assumed to be 150 points. Fourteenth order LPA model is computed using 128 points per frame. As before, the parameters signed FBR, the smoothed BTR, CTR, and FTR are computed. Plots are then obtained for these parameters, as shown by Figures 84-93. The RMS dip-classification is applied in the phonetic feature detection section followed by the two vowel, vowel-like and non-vowel decisions, which are based on the RMS and BTR contours. An 'OR' decision is finally made using the above two classification

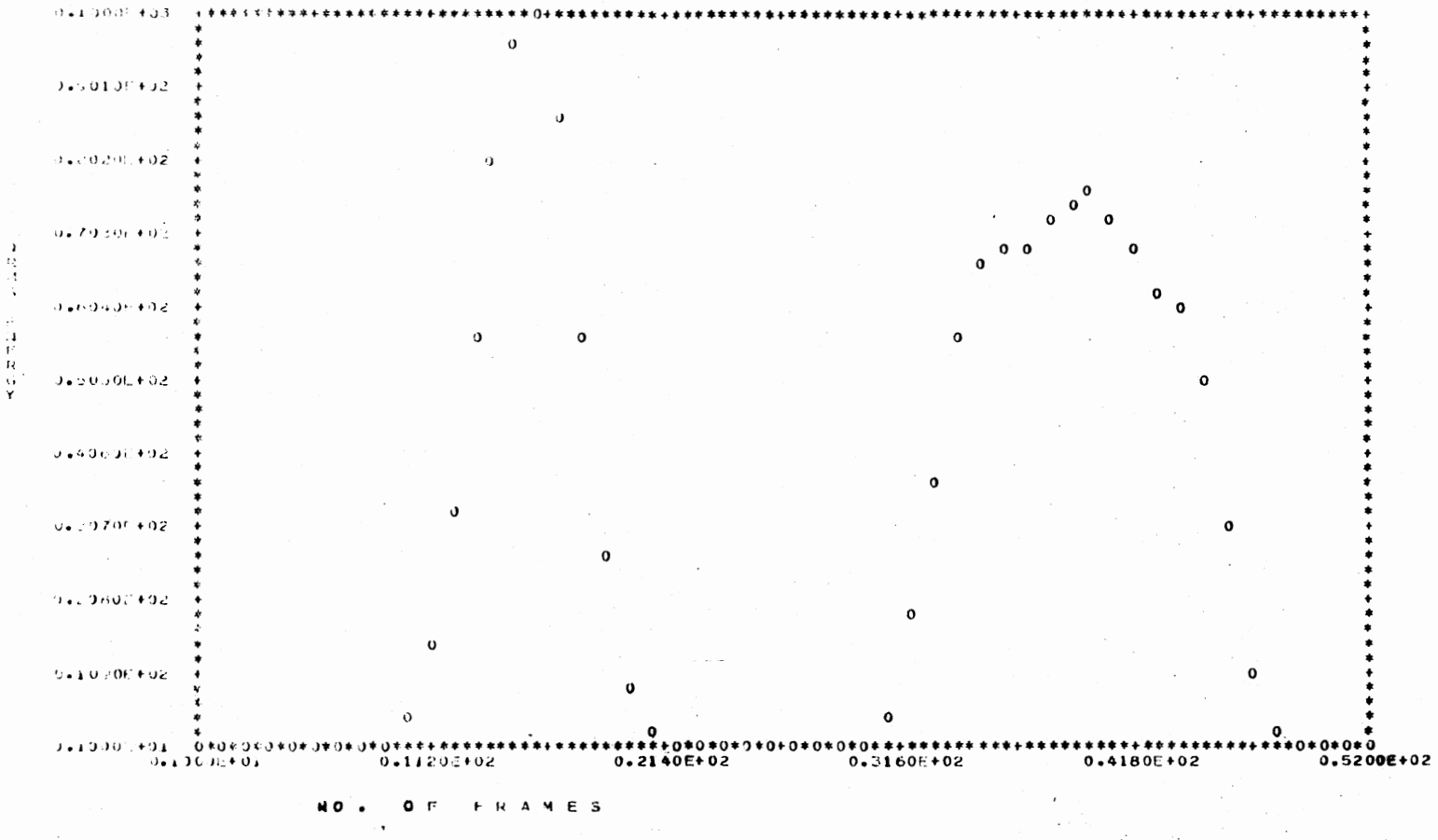


Figure 63. Smoothed and Quantized RMS Energy Contour for Digit Zero, i.e. /sefr/ Spoken in Arabic, Sample 1

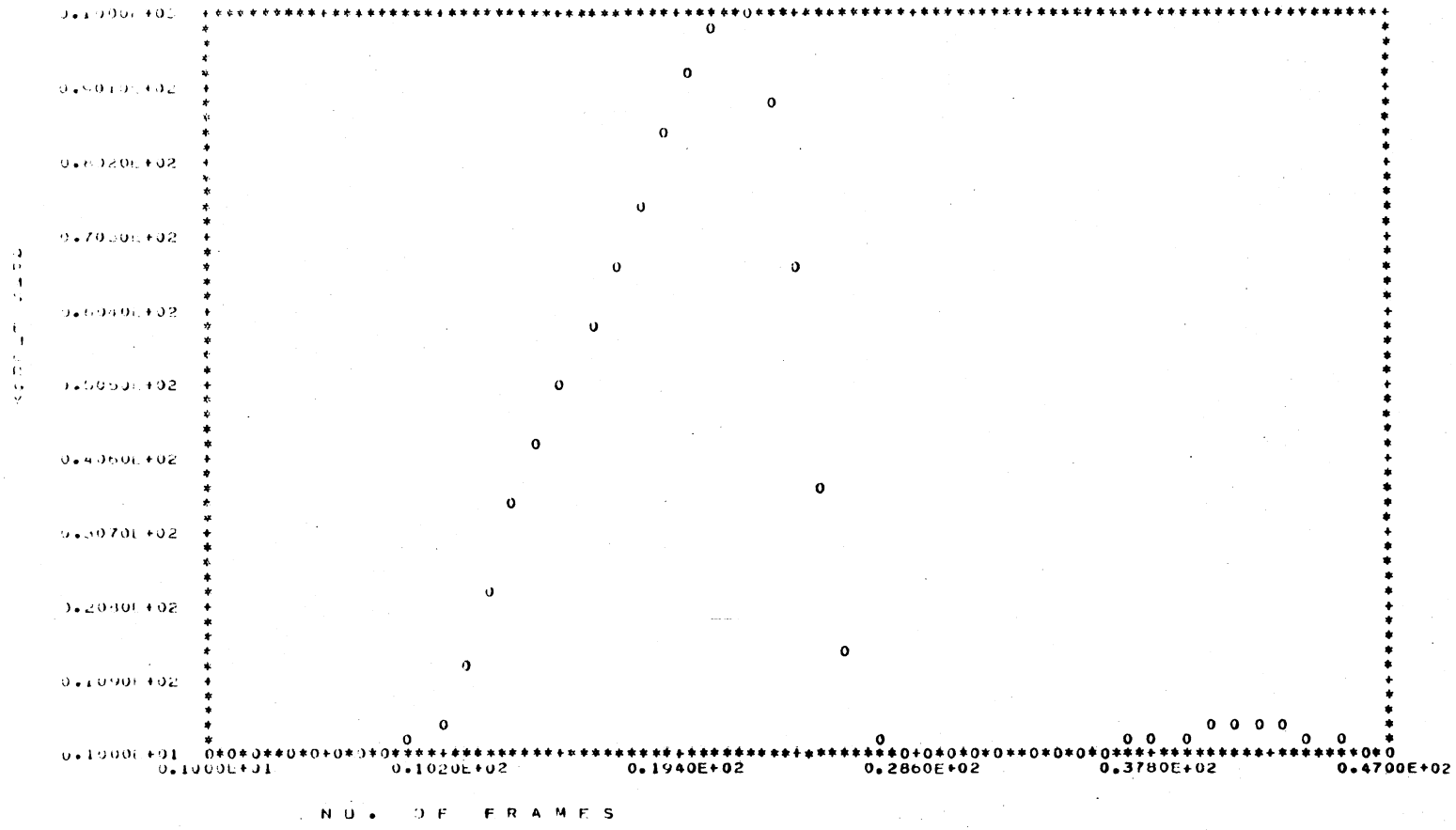


Figure 64. Smoothed and Quantized RMS Energy Contour for Digit One, i.e. /ωâhid/ Spoken in Arabic, Sample 1

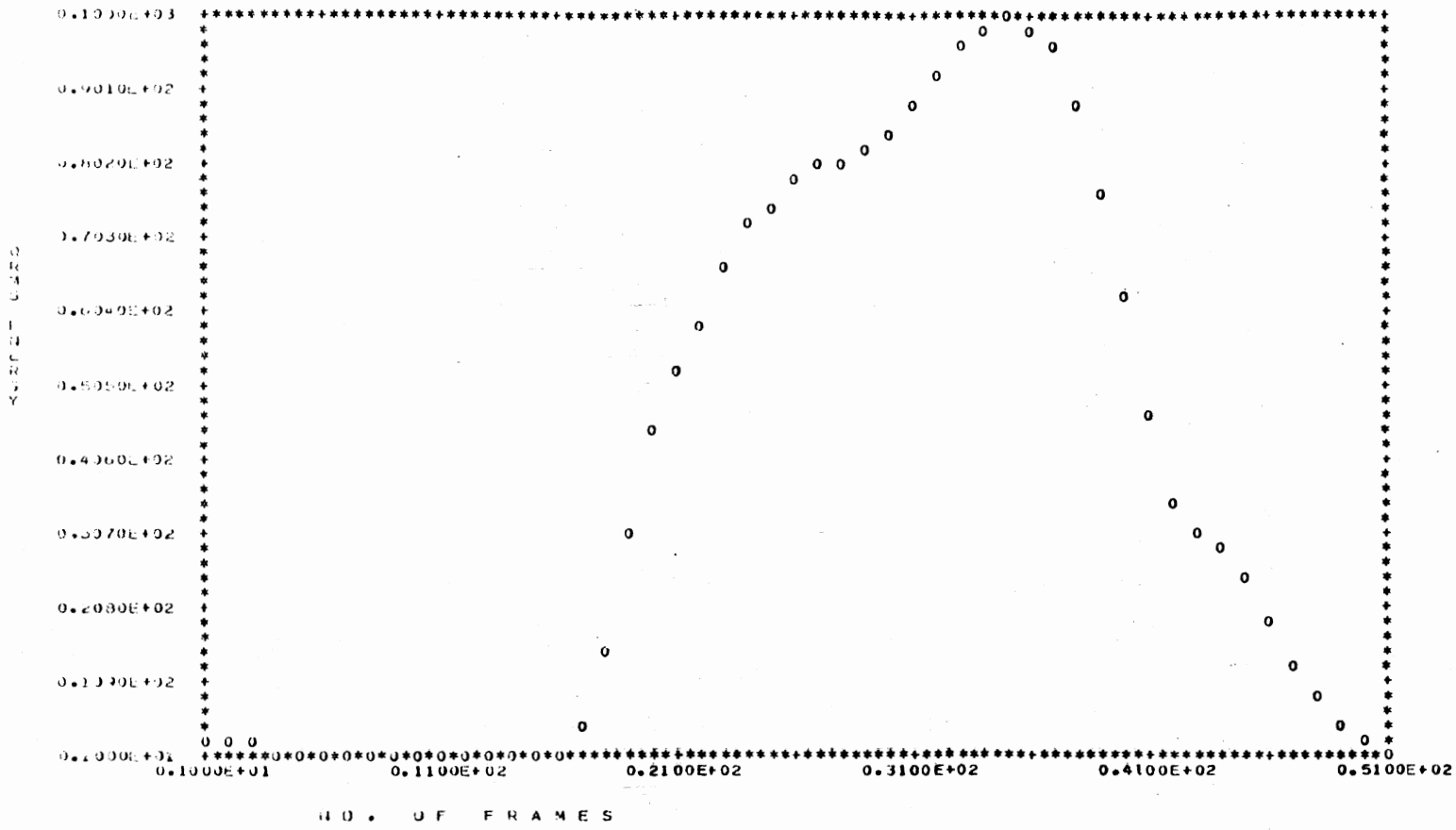


Figure 65. Smoothed and Quantized RMS Energy Contour for Digit Two, i.e. /iθnân/ Spoken in Arabic, Sample 1

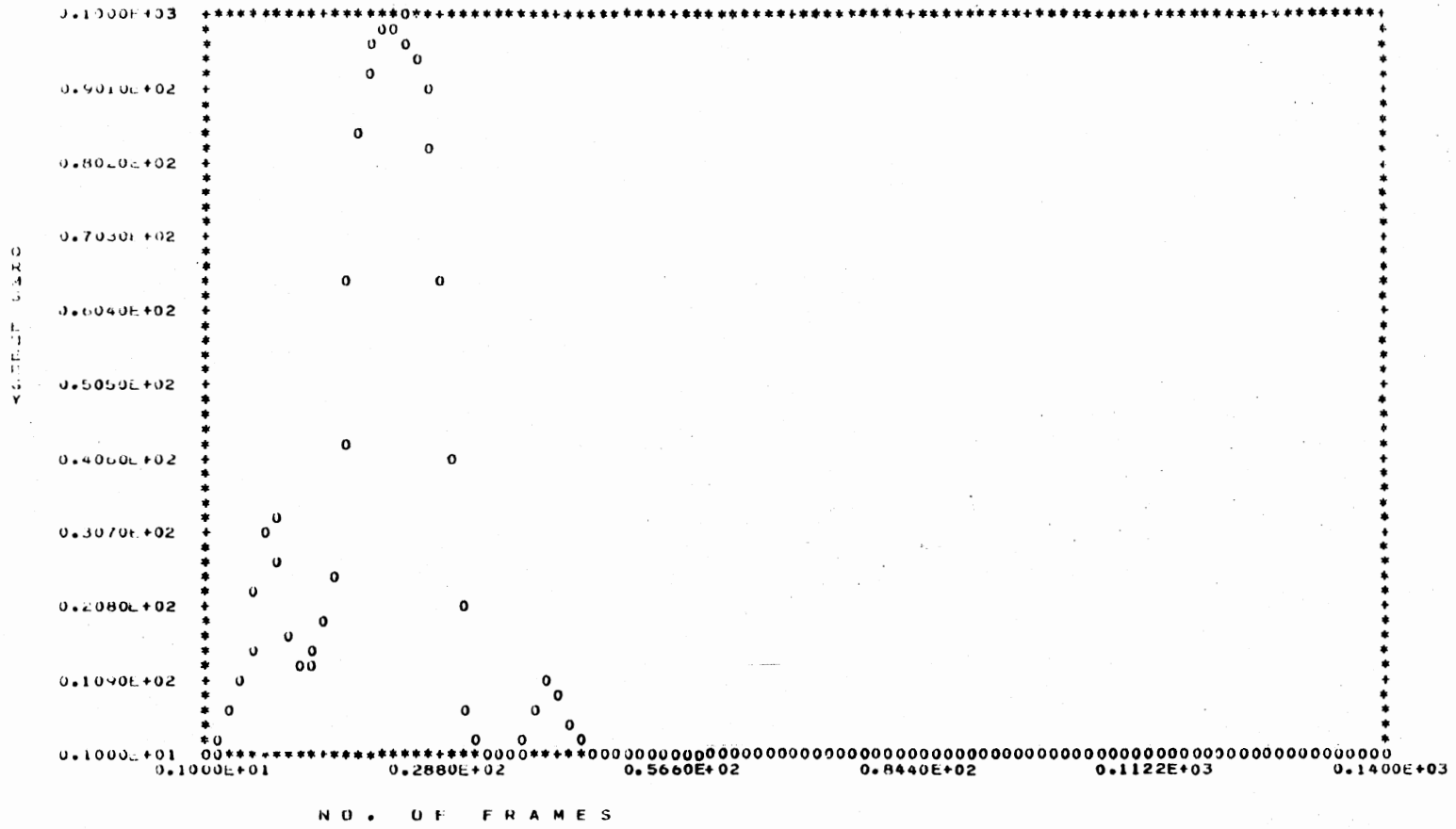


Figure 66. Smoothed and Quantized RMS Energy Contour for Ditit Three, i.e. /θalâθðh/ Spoken in Arabic Showing Wrong End Point Detection, Sample 1

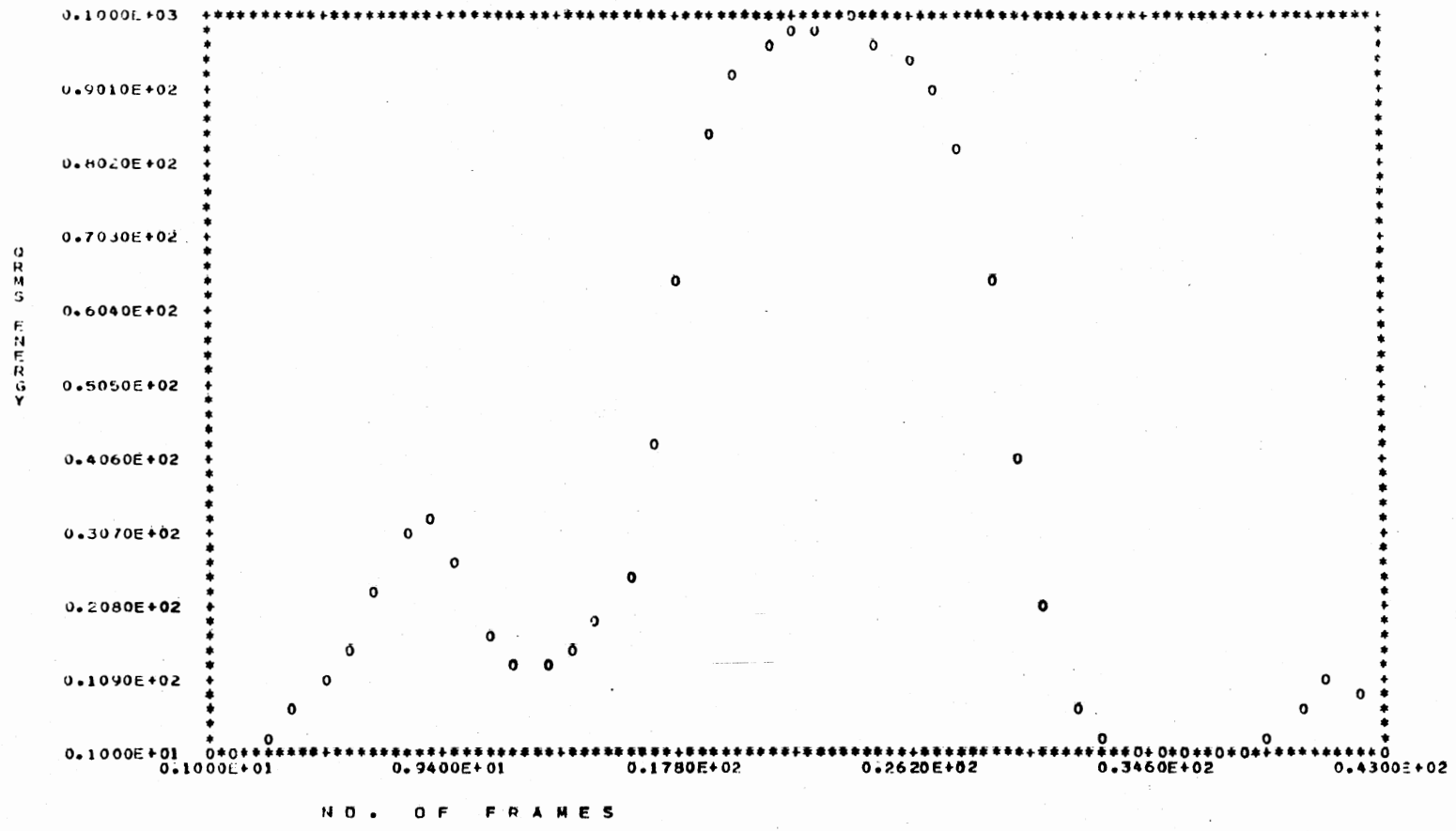


Figure 67. Smoothed and Quantized RMS Energy Contour for Digit Three, i.e. /θaɫâθðh/ Spoken in Arabic, Sample 1

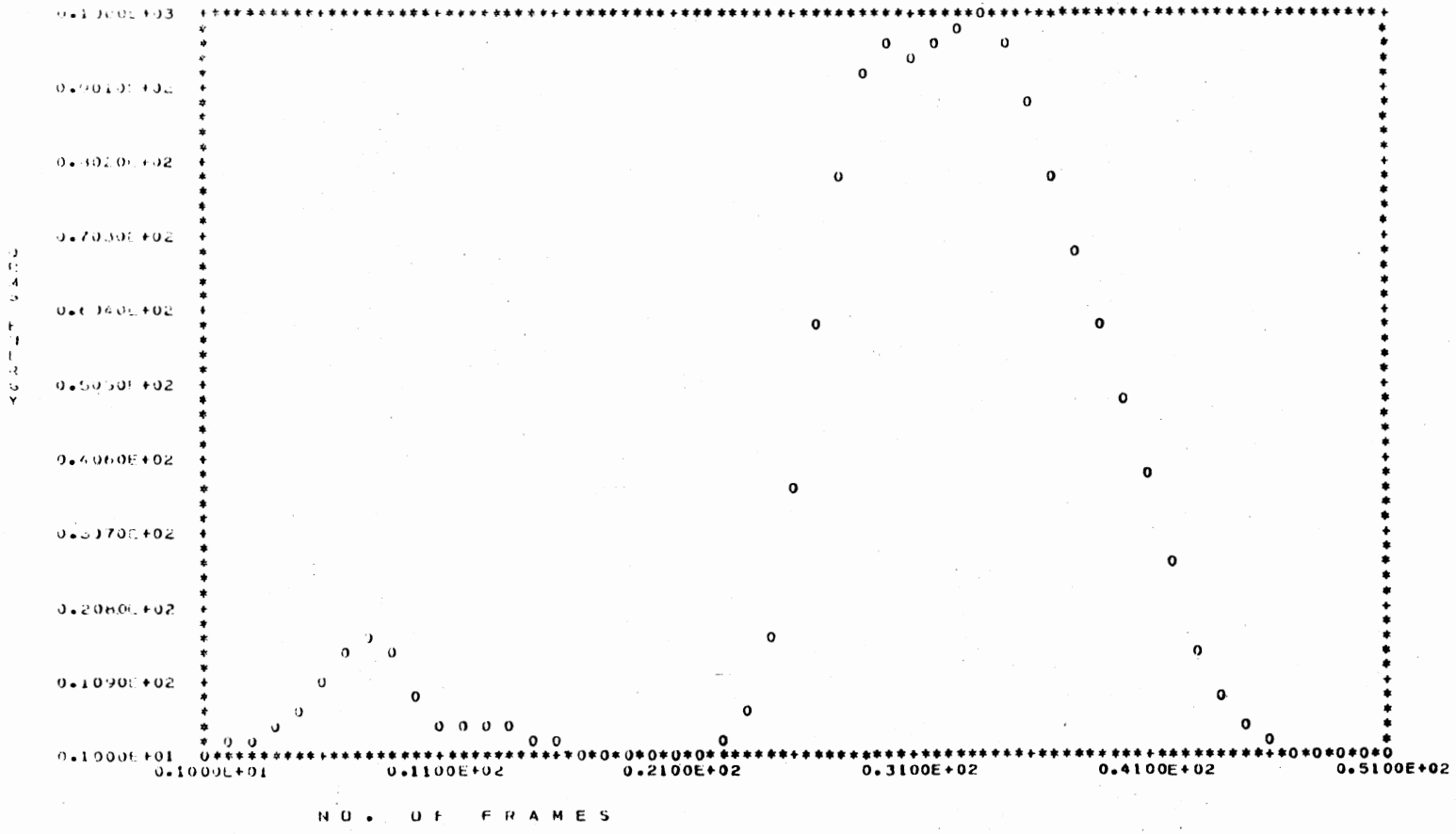


Figure 68. Smoothed and Quantized RMS Energy Contour for Digit Four, i.e. /arbaɒðh/ Spoken in Arabic, Sample 1

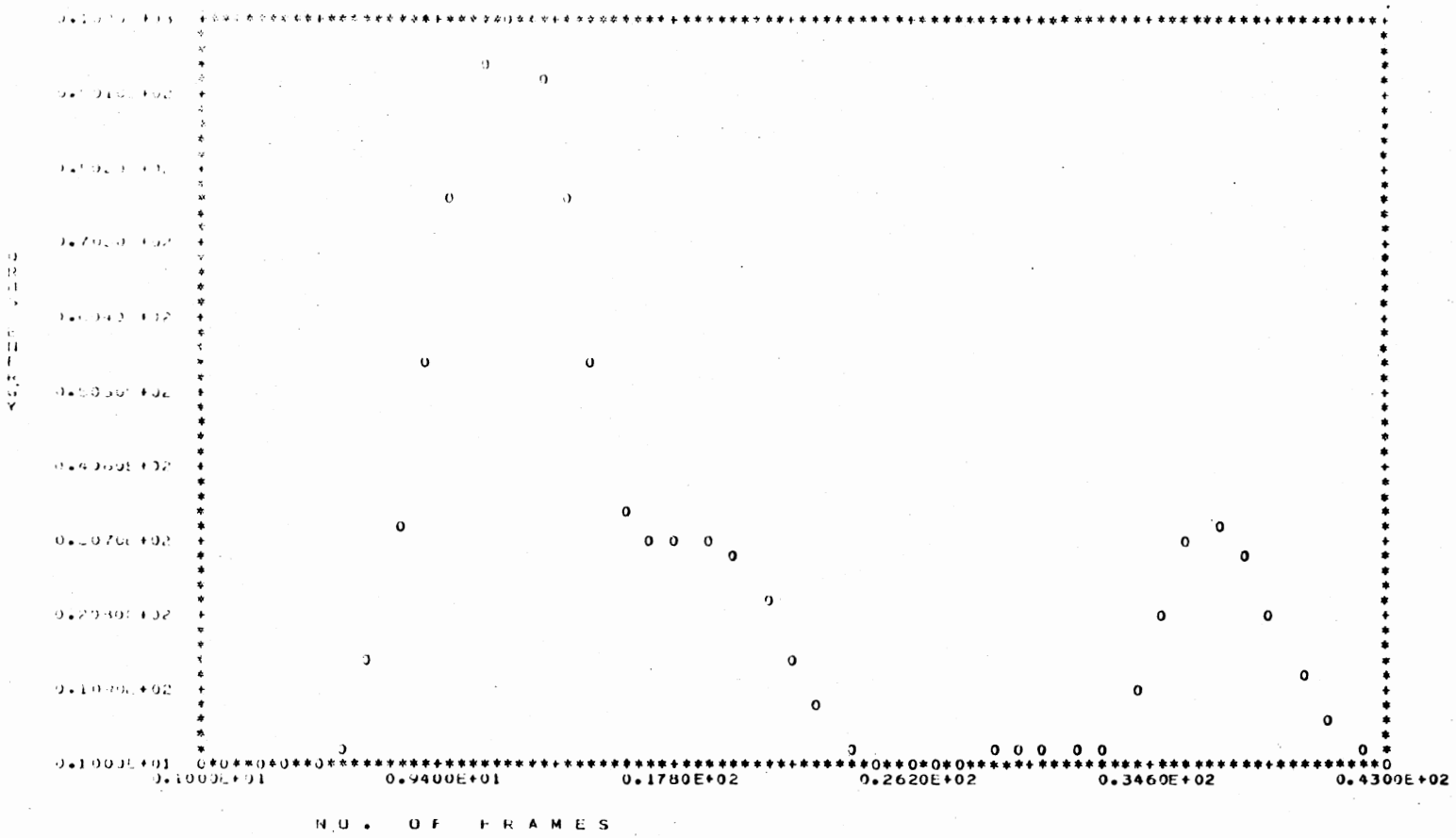


Figure 69. Smoothed and Quantized RMS Energy Contour for Digit Five, i.e. /khΛms∂h/ Spoken in Arabic, Sample 1

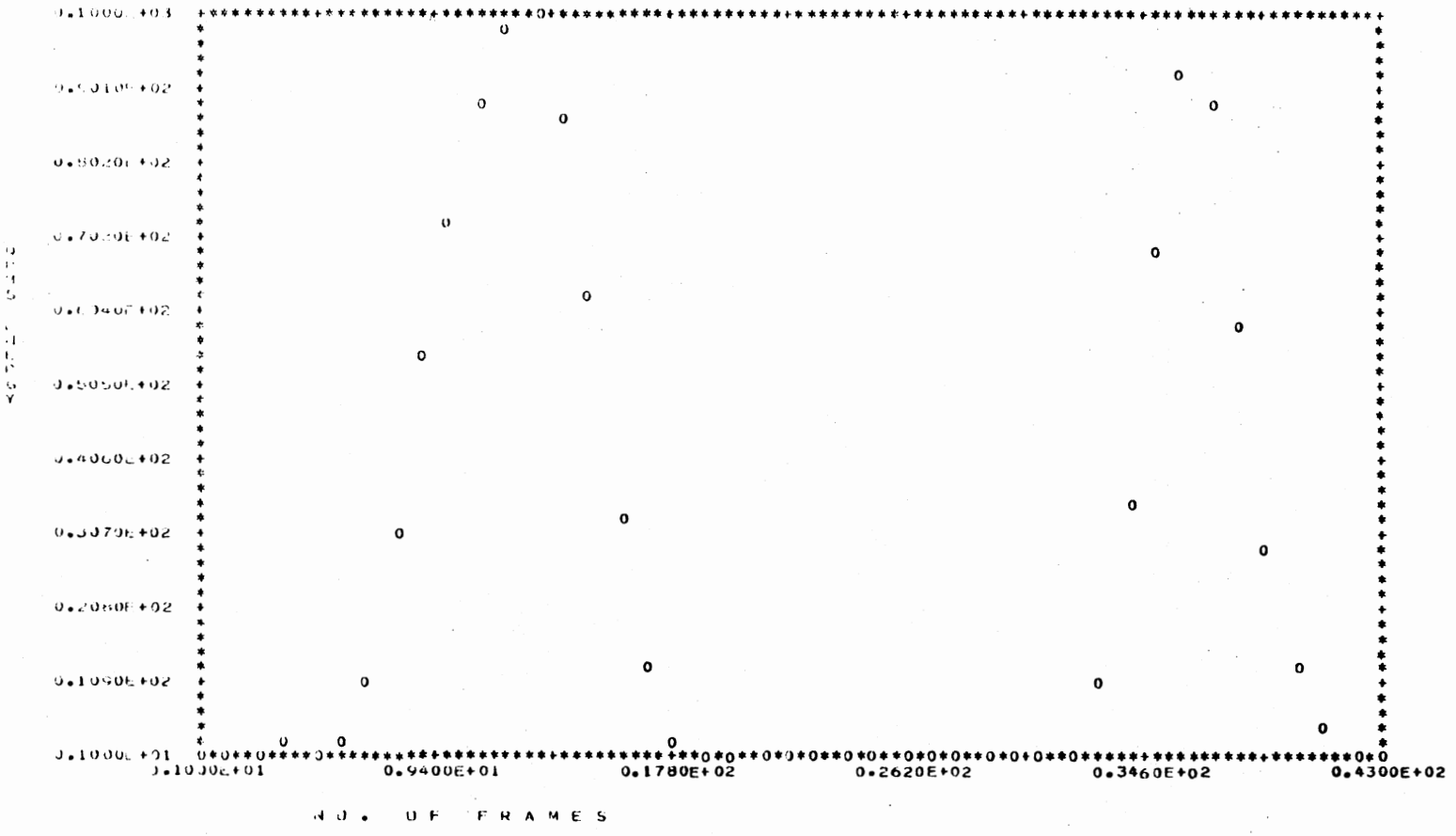


Figure 70. Smoothed and Quantized RMS Energy Contour for Digit Six, i.e. /sIttəh/ Spoken in Arabic, Sample 1

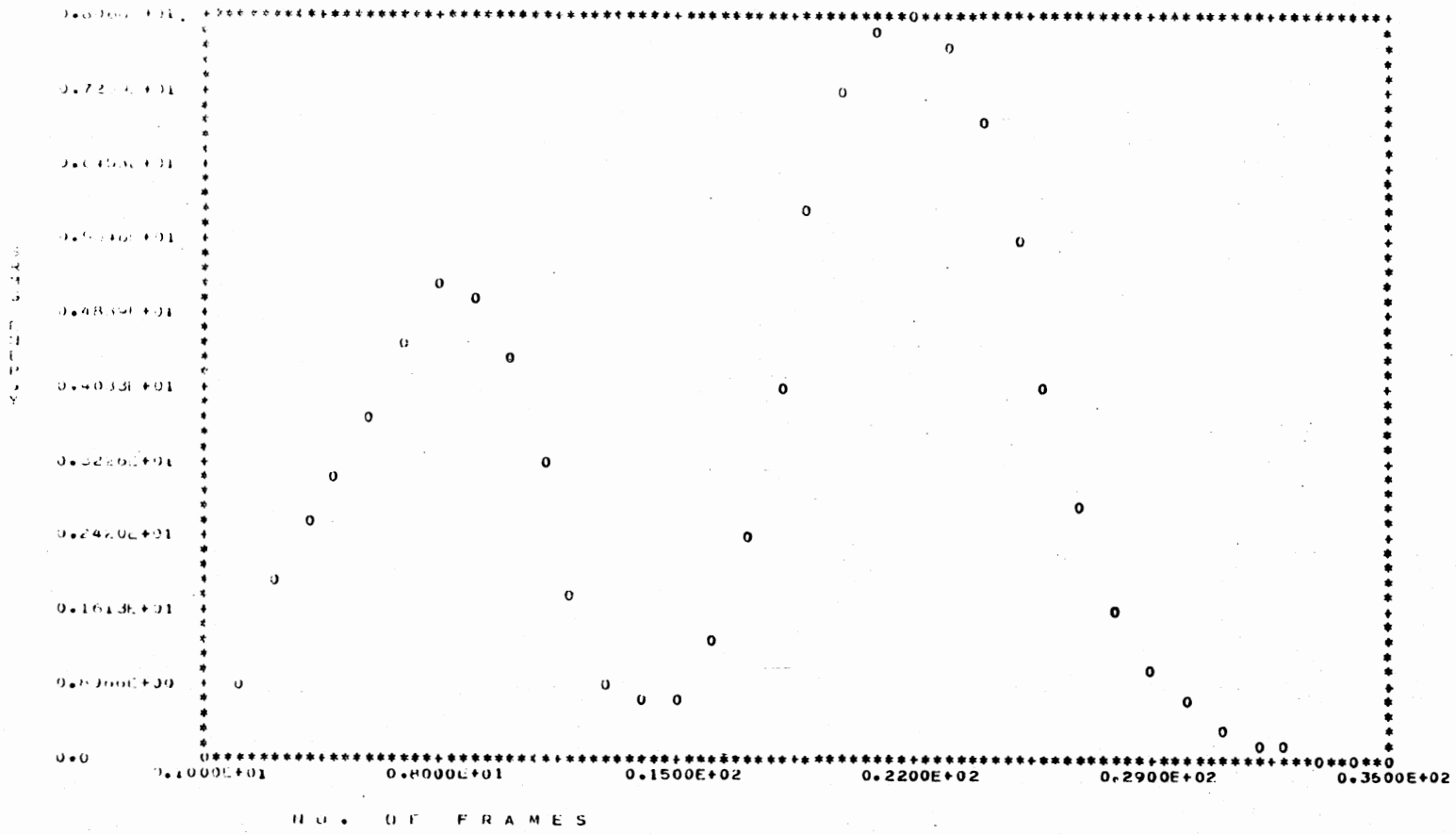


Figure 71. Smoothed and Quantized RMS Energy Contour for Digit Seven, i.e. /sʌbɒðh/ Spoken in Arabic, Sample 1

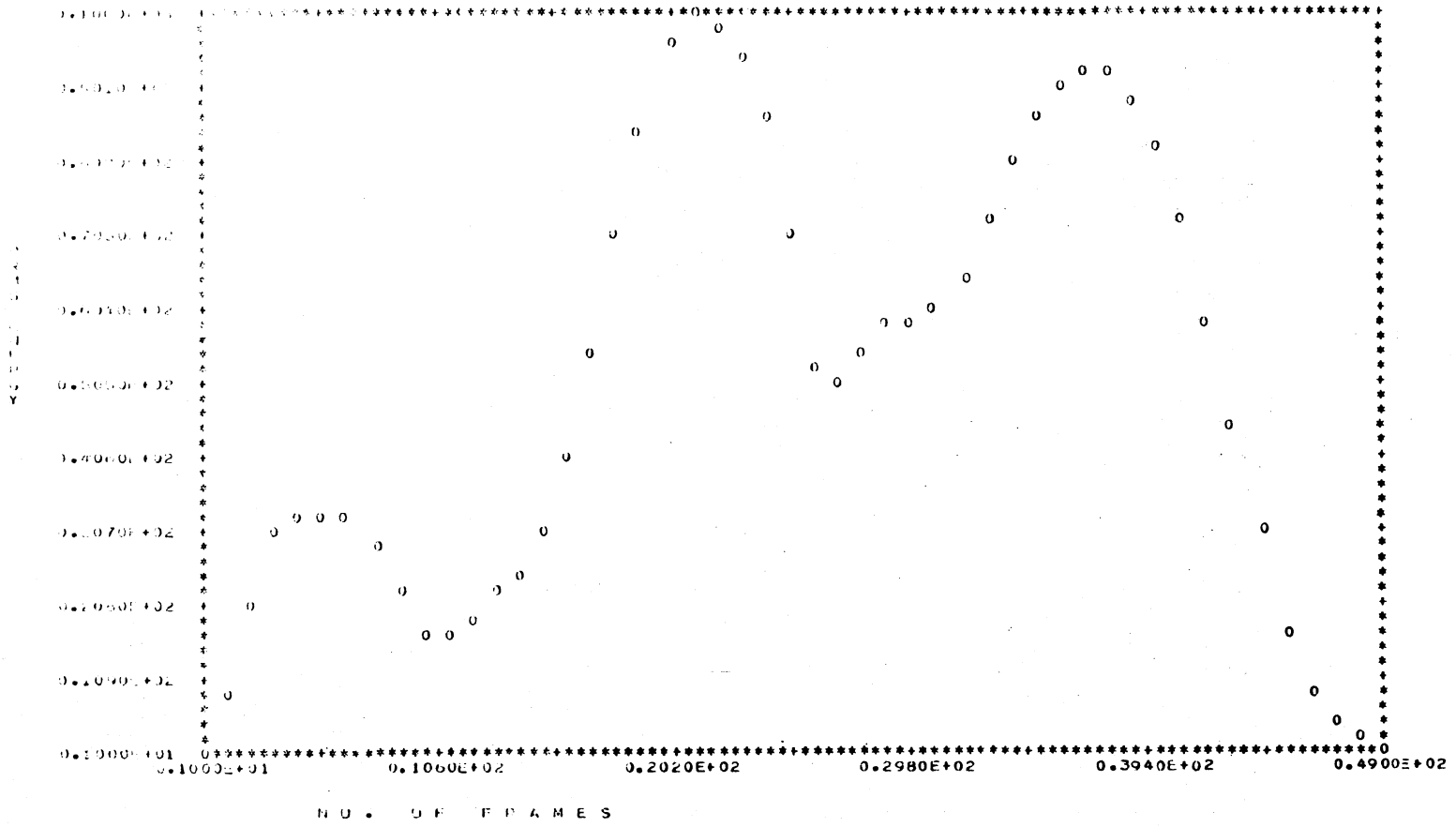


Figure 72. Smoothed and Quantized RMS Energy Contour for Digit Eight, i.e. /θamâny∂h/ Spoken in Arabic , Sample 1

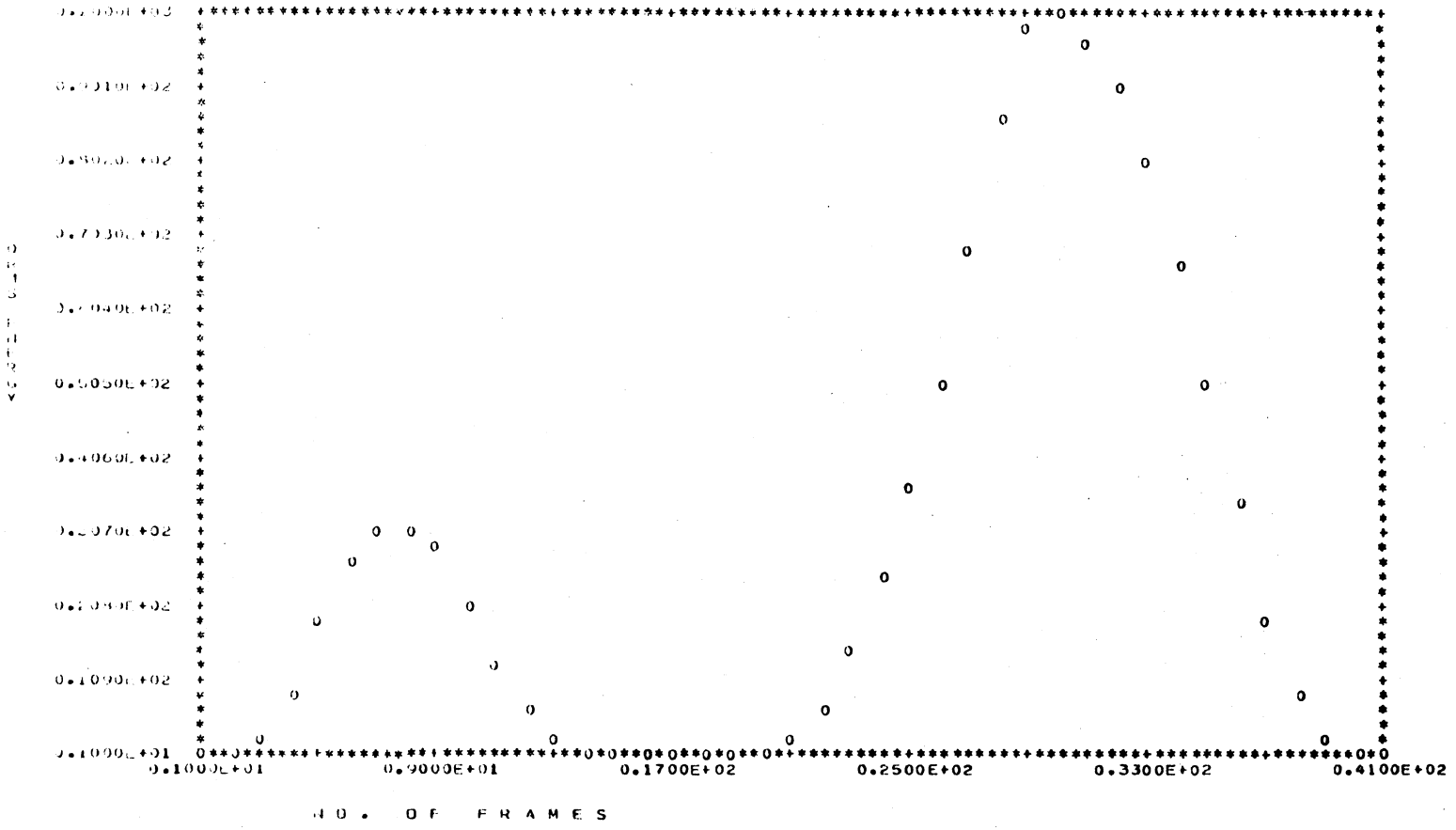


Figure 73. Smoothed and Quantized RMS Energy Contour for Digit Nine, i.e. /tIsɒðh/ Spoken in Arabic, Sample 1

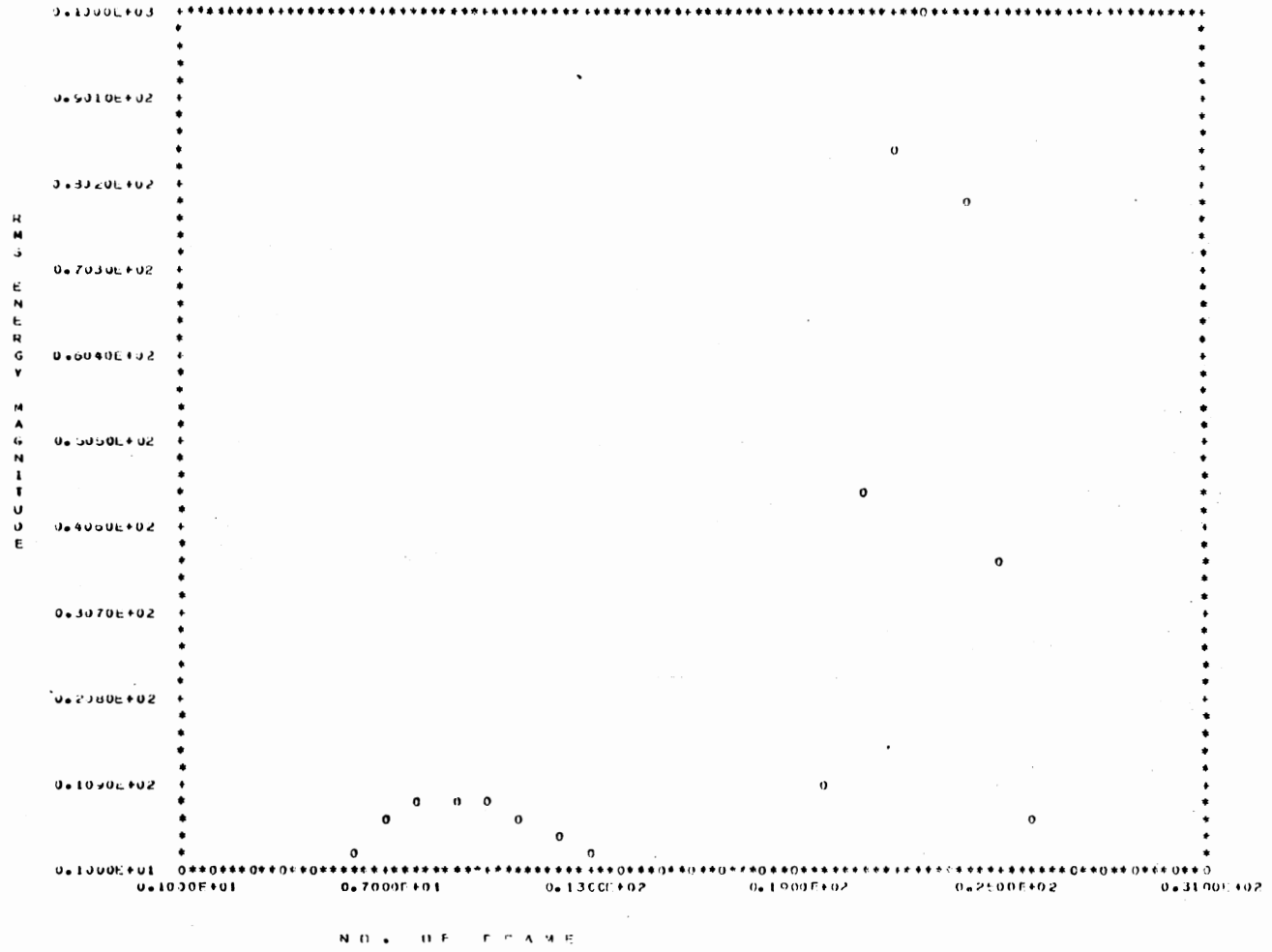


Figure 74. Smoothed and Quantized RMS Energy Contour for Digit Zero, i.e. /sefr/, Spoken in Arabic, Sample 2

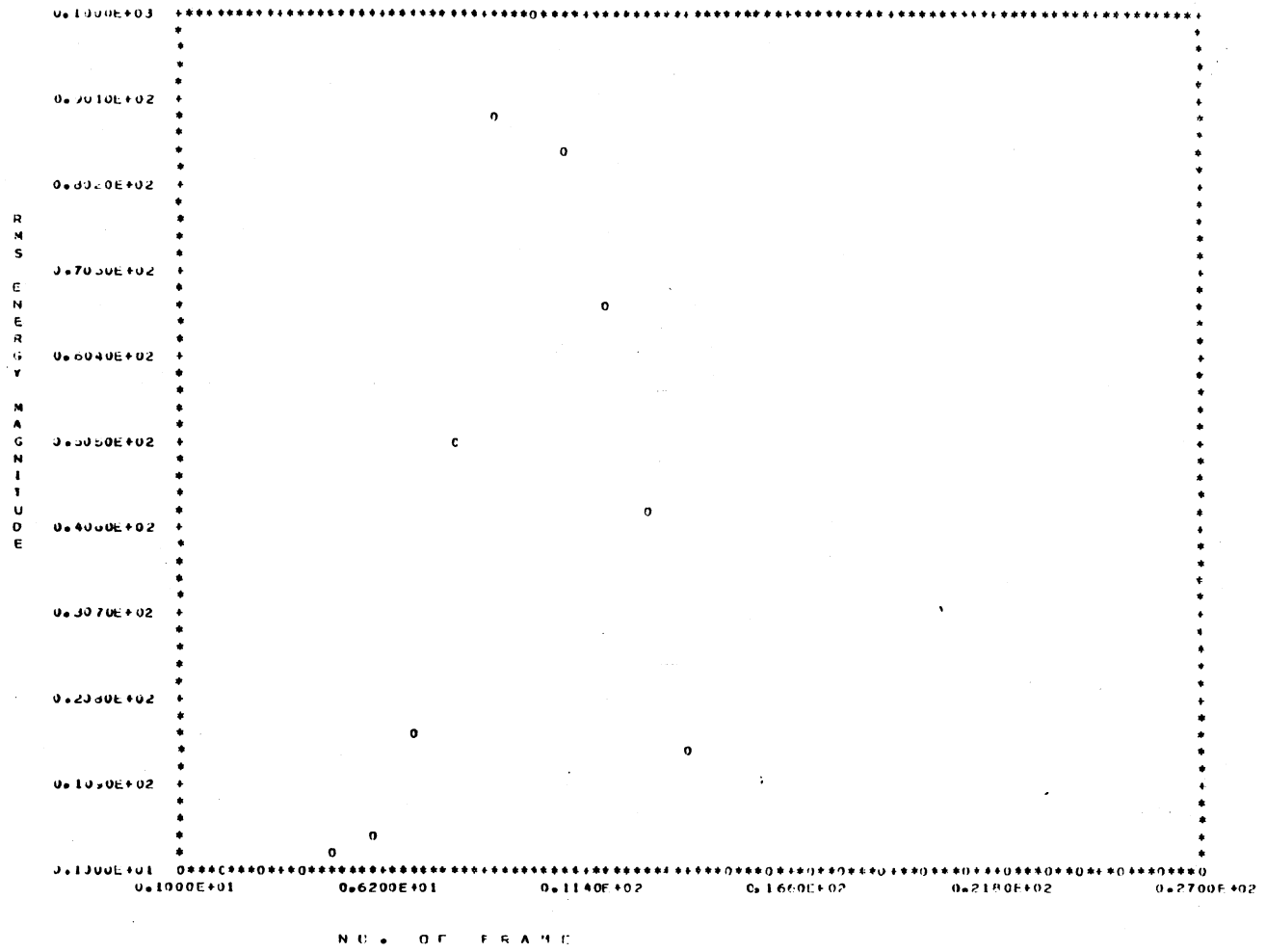


Figure 75. Smoothed and Quantized RMS Energy Contour for Digit One, i.e. /ωâhid/ Spoken in Arabic, Sample 2

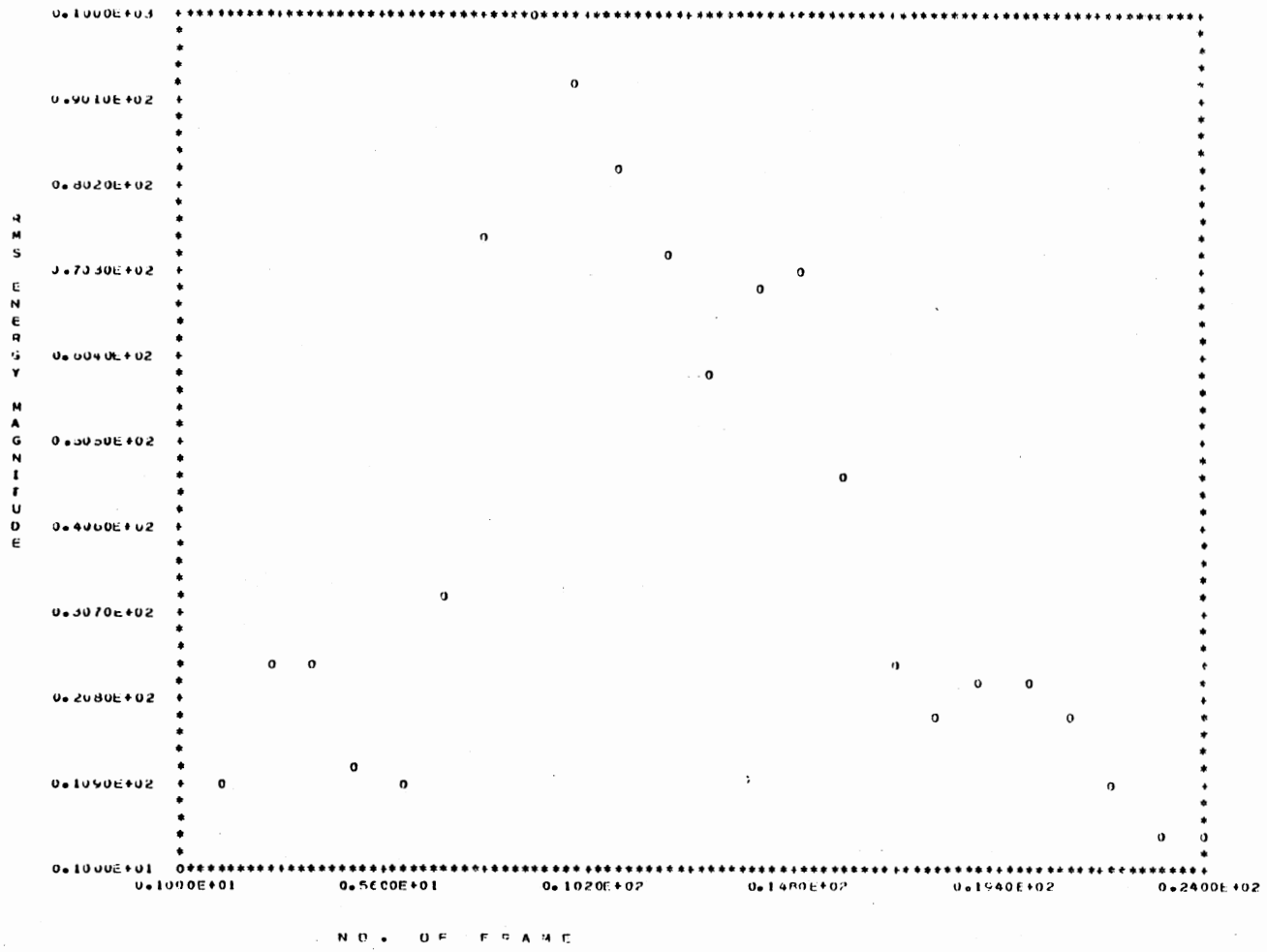


Figure 76. Smoothed and Quantized RMS Energy Contour for Digit Two, i.e. /iθnân/, Spoken in Arabic, Sample 2

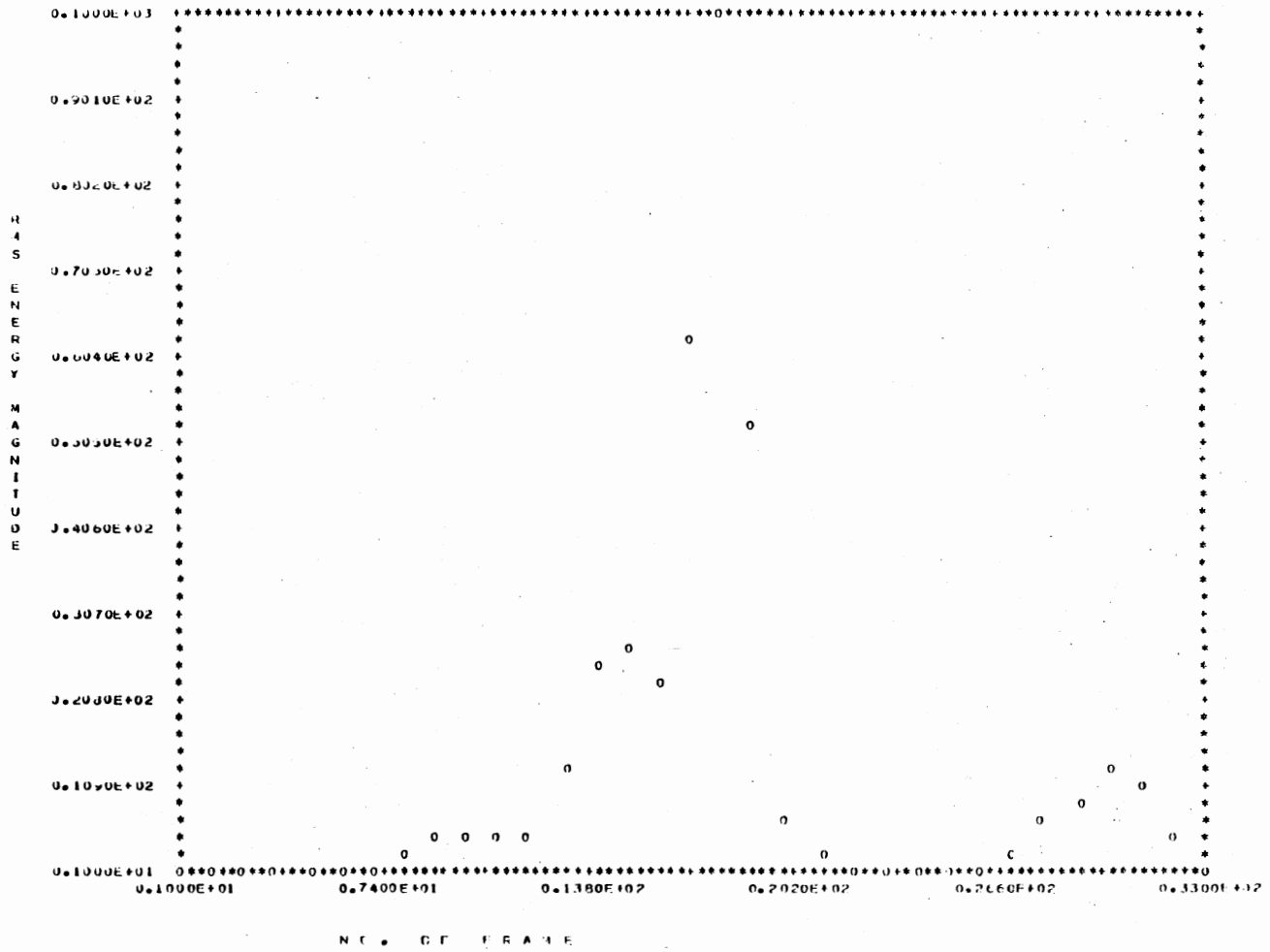


Figure 77. Smoothed and Quantized RMS Energy Contour for Digit Three, i.e. /θaℓâθðh/ Spoken in Arabic, Sample 2

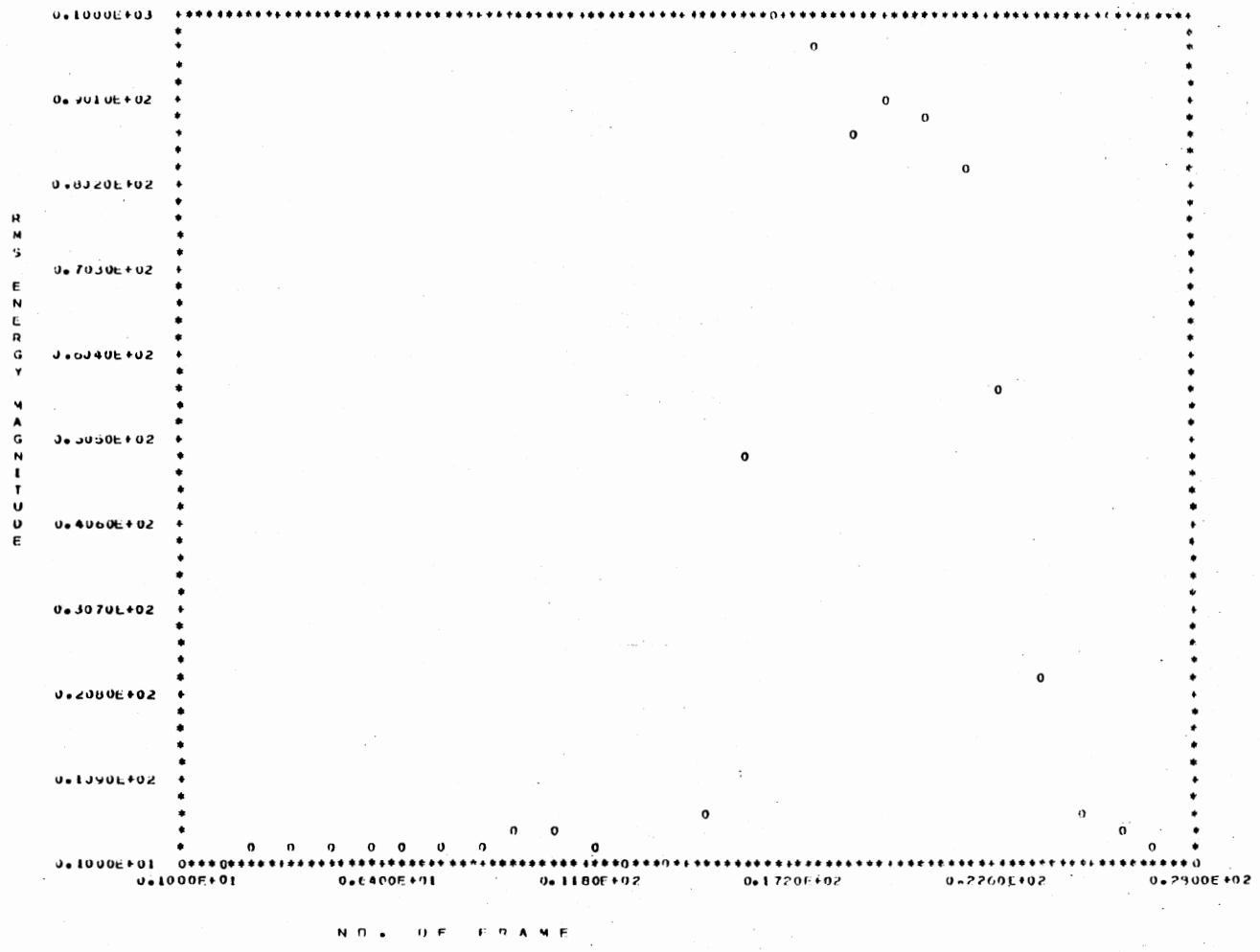


Figure 78. Smoothed and Quantized RMS Energy Contour for Digit Four, i.e. /arbaɒðḥ/ Spoken in Arabic, Sample 2

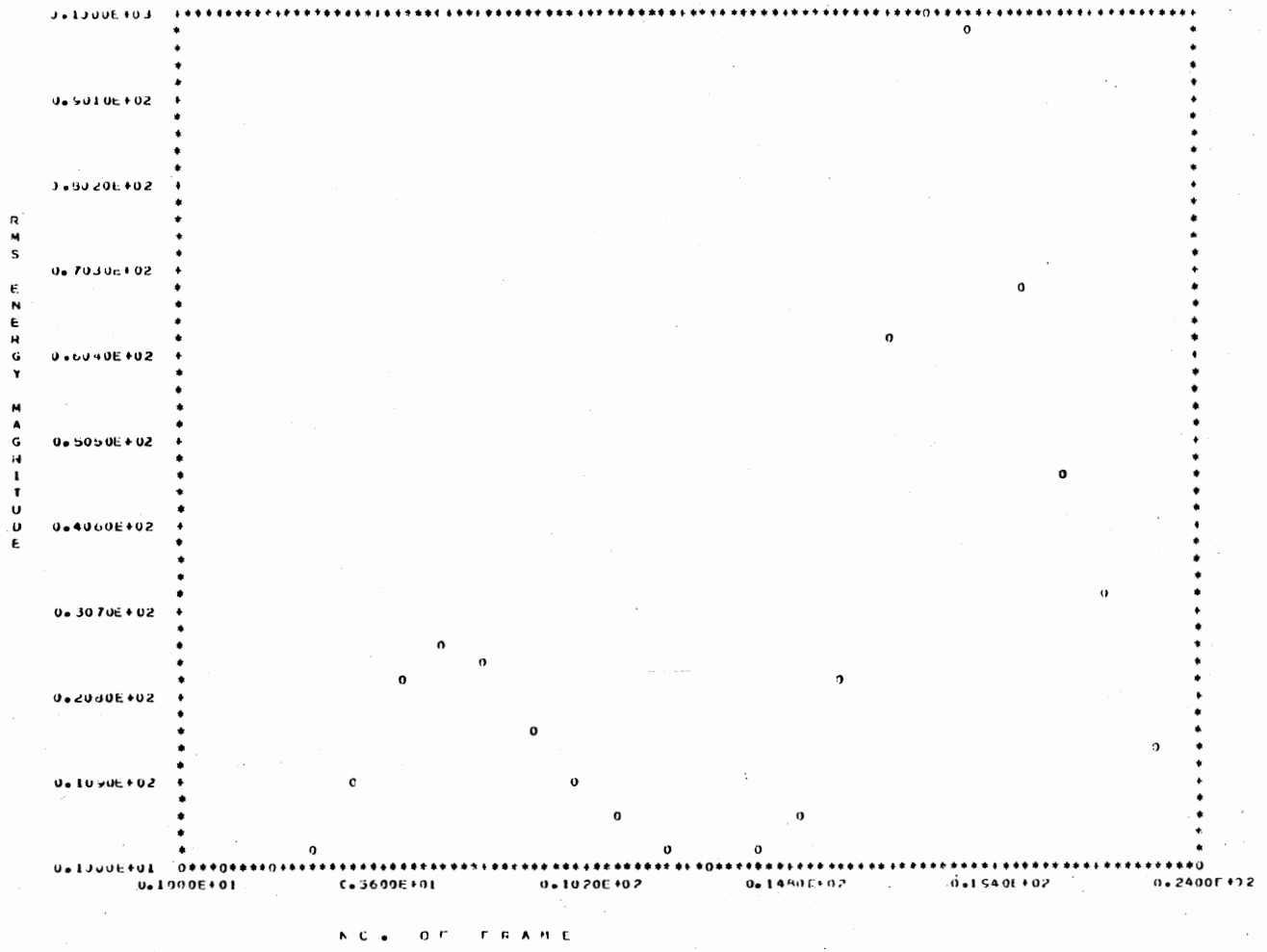


Figure 79. Smoothed and Quantized RMS Energy Contour for Digit Five, i.e. /khΛms∂h/ Spoken in Arabic, Sample 2

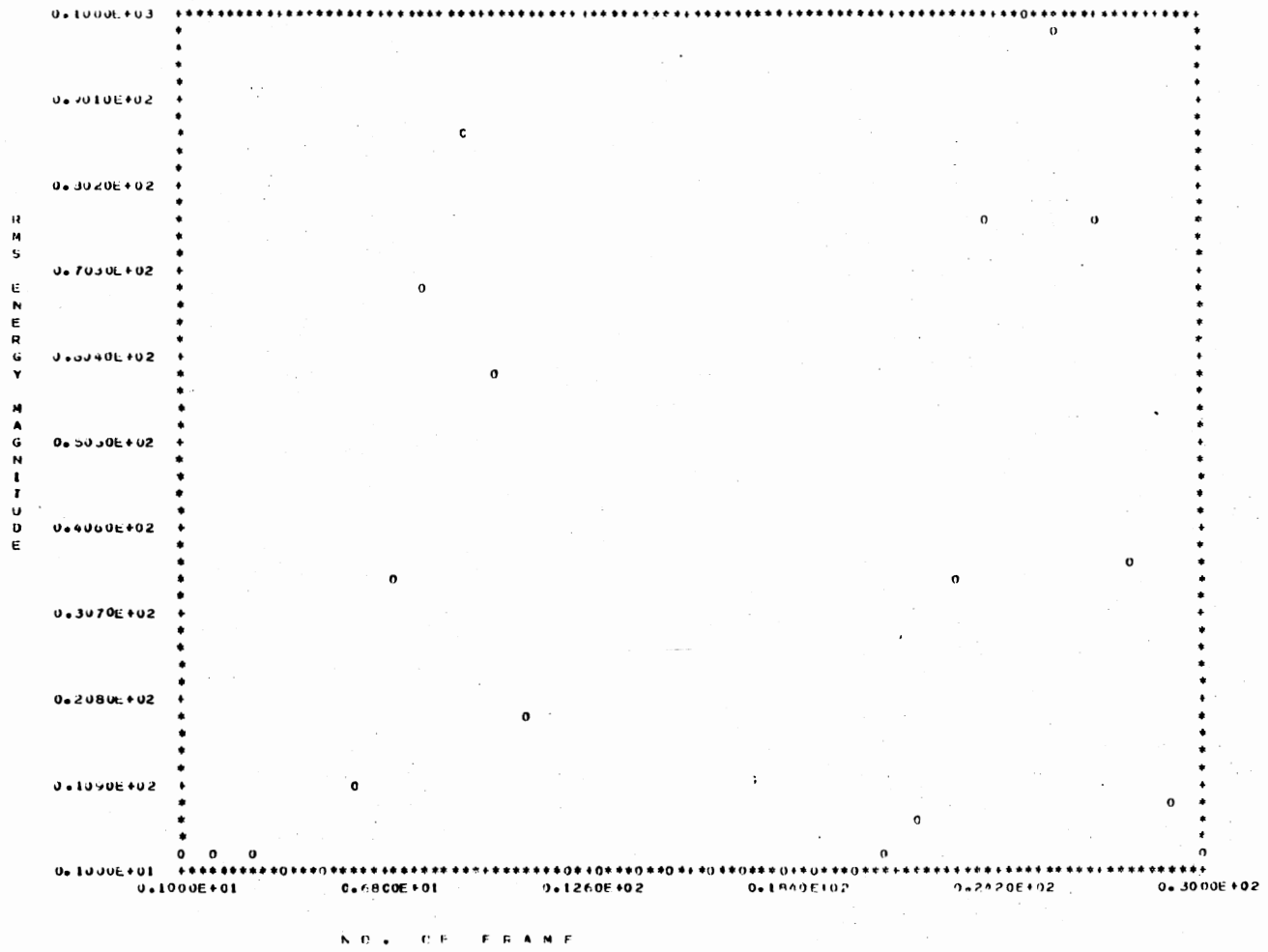


Figure 80. Smoothed and Quantized RMS Energy Contour for Digit Six, i.e. /sIttəh/, Spoken in Arabic, Sample 2

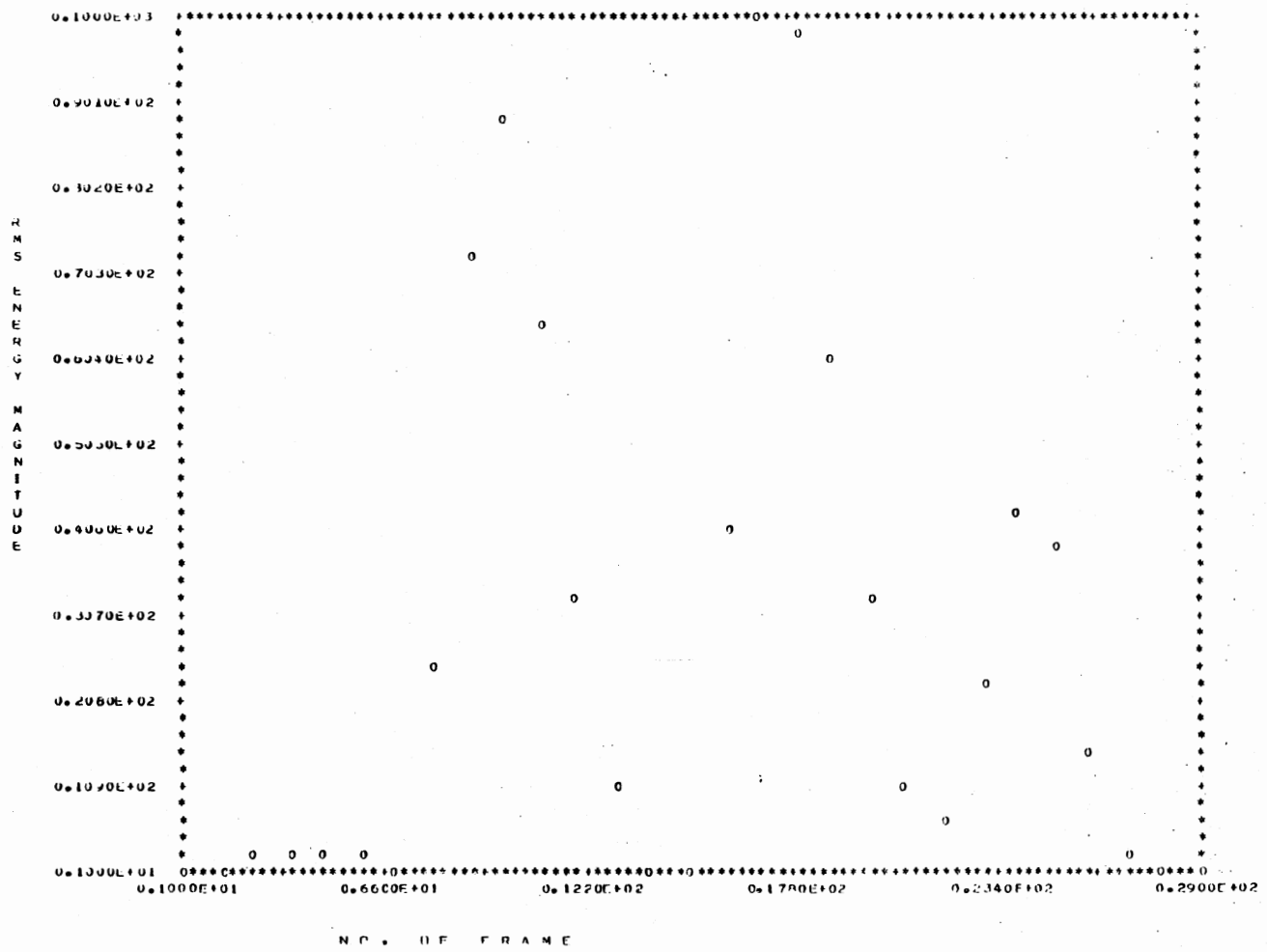


Figure 81. Smoothed and Quantized RMS Energy Contour for Digit Seven, i.e. /sʌbɒə̀h/ Spoken in Arabic, Sample 2

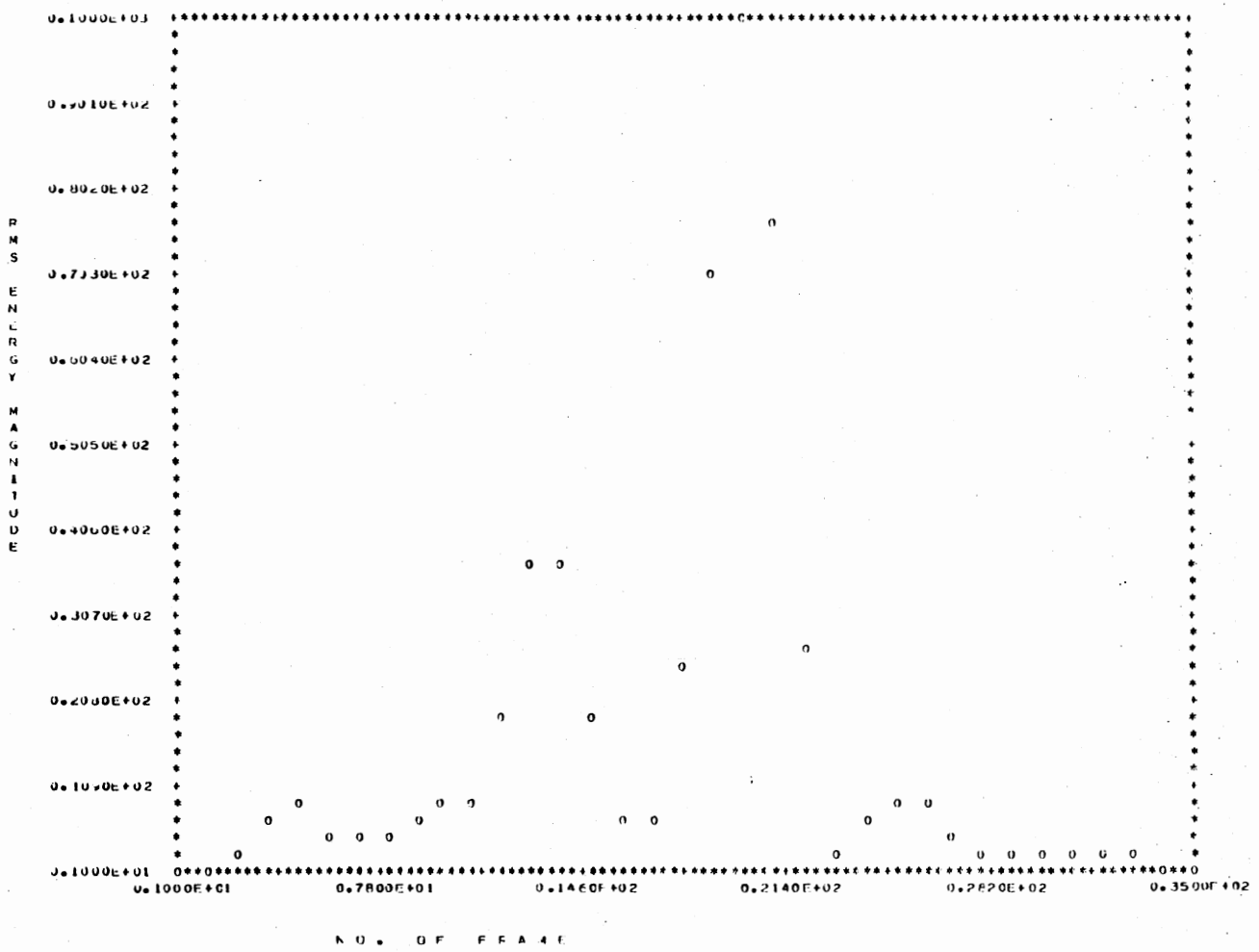


Figure 82. Smoothed and Quantized RMS Energy Contour for Digit Eight, i.e. /θamânyəh/ Spoken in Arabic, Sample 2

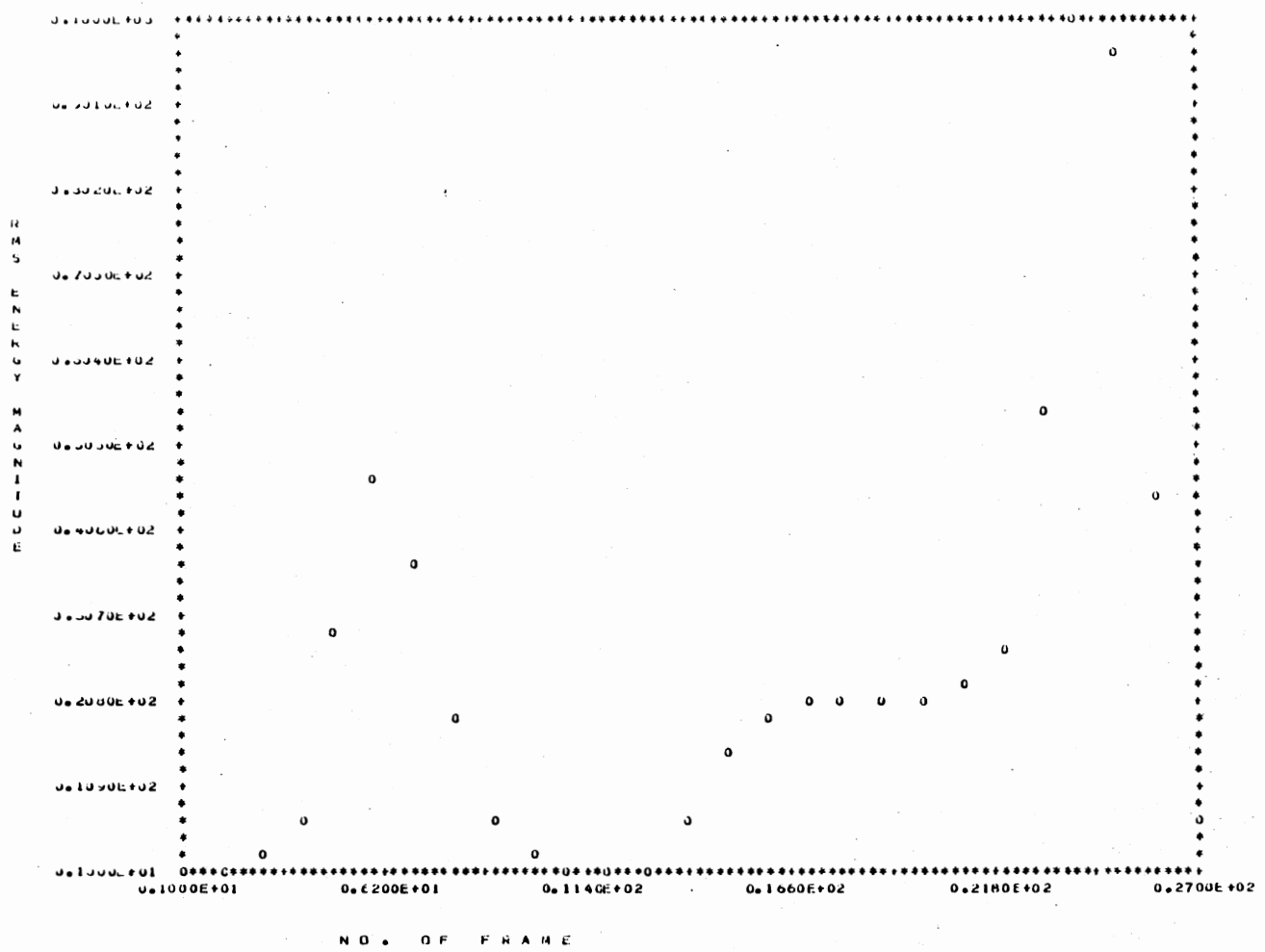


Figure 83. Smoothed and Quantized RMS Energy Contour for Digit Nine, i.e. /tIsɒðh/ Spoken in Arabic, Sample 2

TABLE XVI

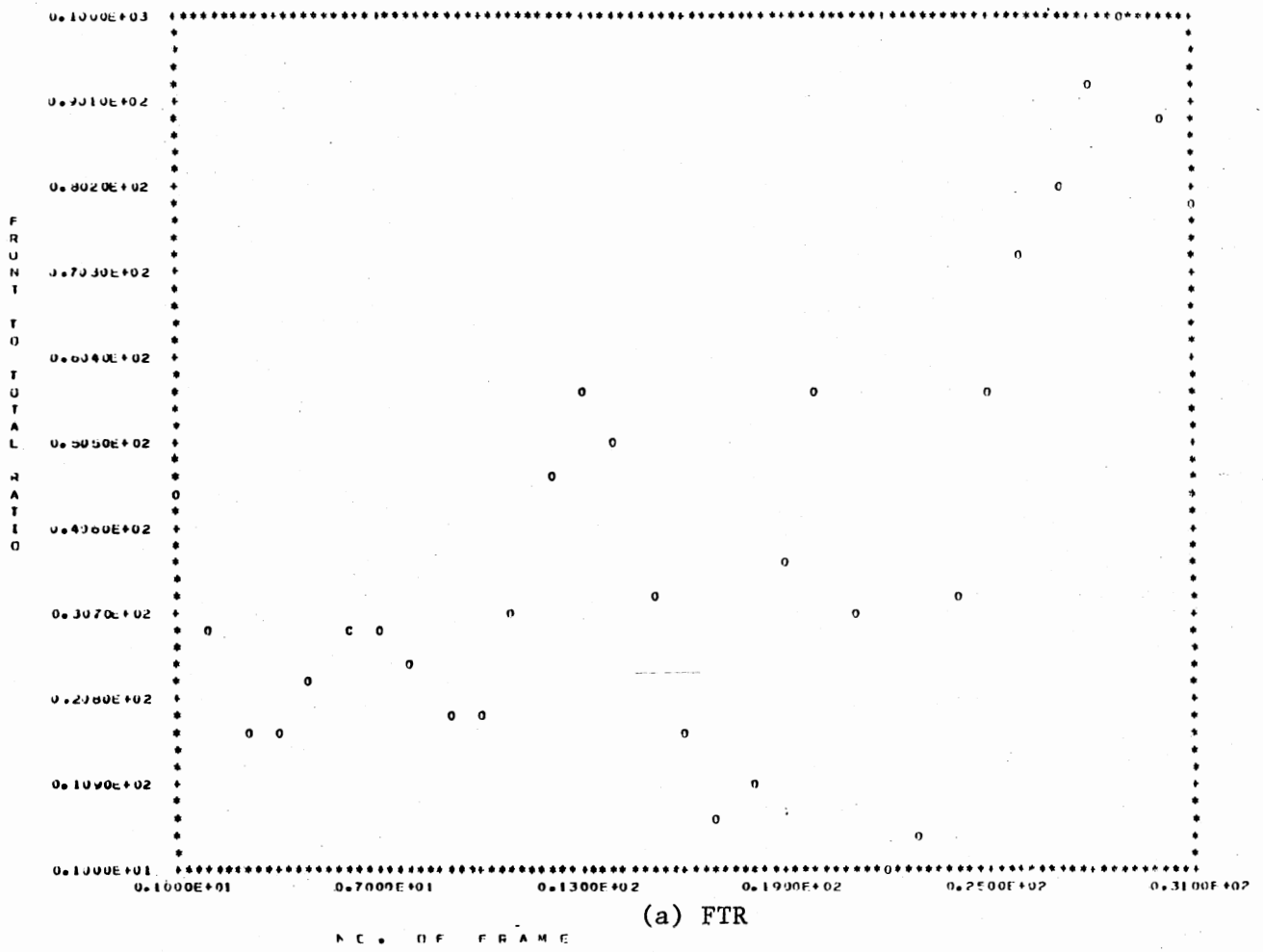
SMOOTHED RMS ENERGY PEAKS AND THEIR RATIO'S FOR DIGITS SPOKEN IN ARABIC WITH VARYING AMPLITUDE AND FIXED MEAN

Digit	P_1	P_2		Ratio
/sefr/	11.35	8.64		0.76
/ωâhid/	12.39			
/iθnân/	23.11			
/θaɫâθəh/	27.38	100.67		0.27
/arbaρəh/	1.52	9.48		0.16
/khʌmsəh/	46.98	15.97		0.34
/sIttəh/	34.85	32.41		0.93
/sʌbρəh/	5.1	8.07		0.63
/θamânyəh/	5.54	16.97	14.27	0.84
/tIsρəh/	4.05	13.34		0.30

TABLE XVII

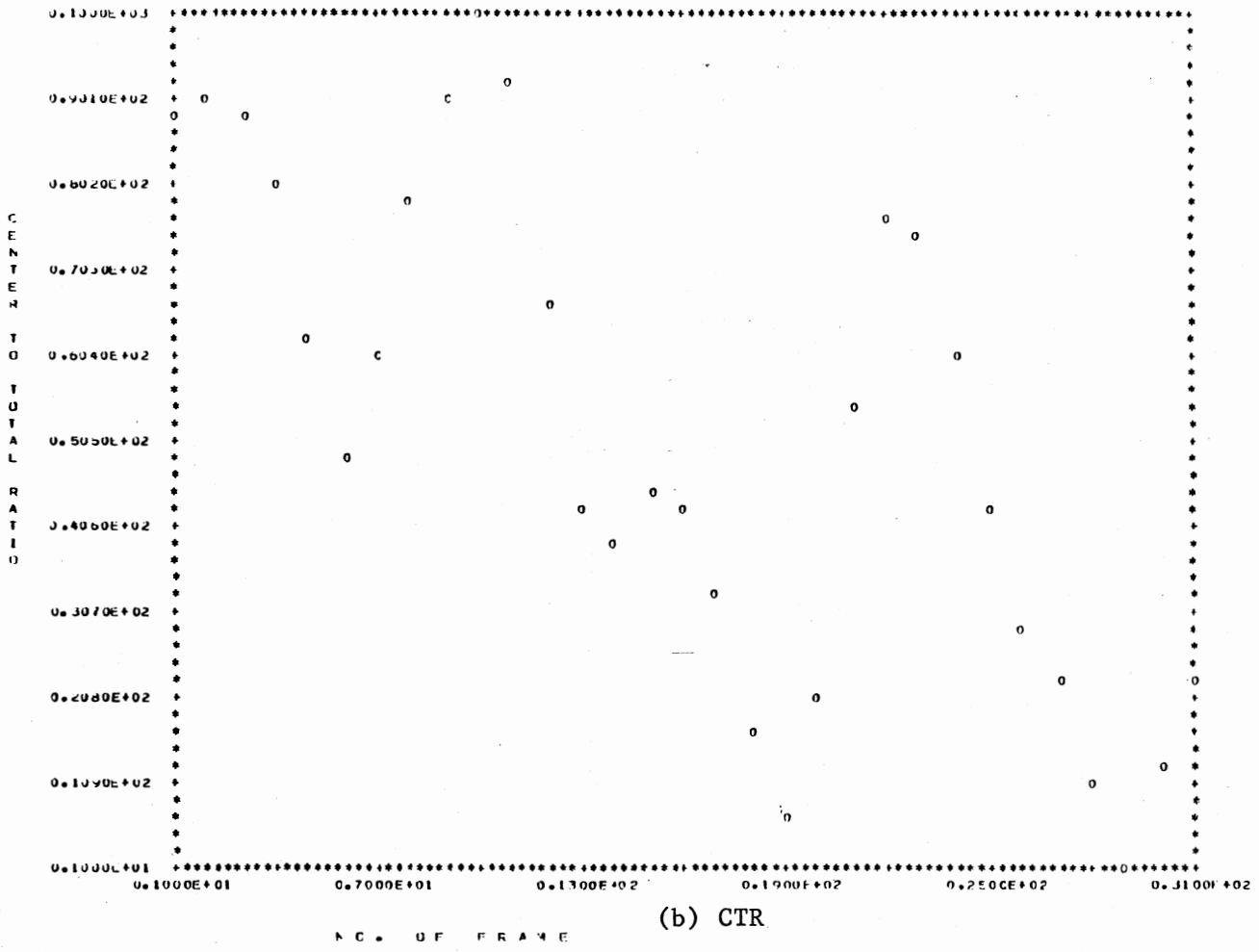
SMOOTHED AND QUANTIZED RMS ENERGY PEAK RATIO'S
FOR DIGITS SPOKEN IN ARABIC

Digit	P_1	P_2	Ratio
/sefr/	10.9	100	0.11
/ω̂ahid/	100		
/iθn̂an/	70	100	0.7
/θaɫâaθəh/	100	17	0.17
/arbaɾəh/	100	90	0.9
/kh̲ʌmsəh/	27	100	0.27
/sɪt̲təh/	87	100	0.87
/sʌbɾəh/	89	100	0.89
/θam̂anyəh/	37	100	0.37
/tIsɾəh/	46	100	0.46



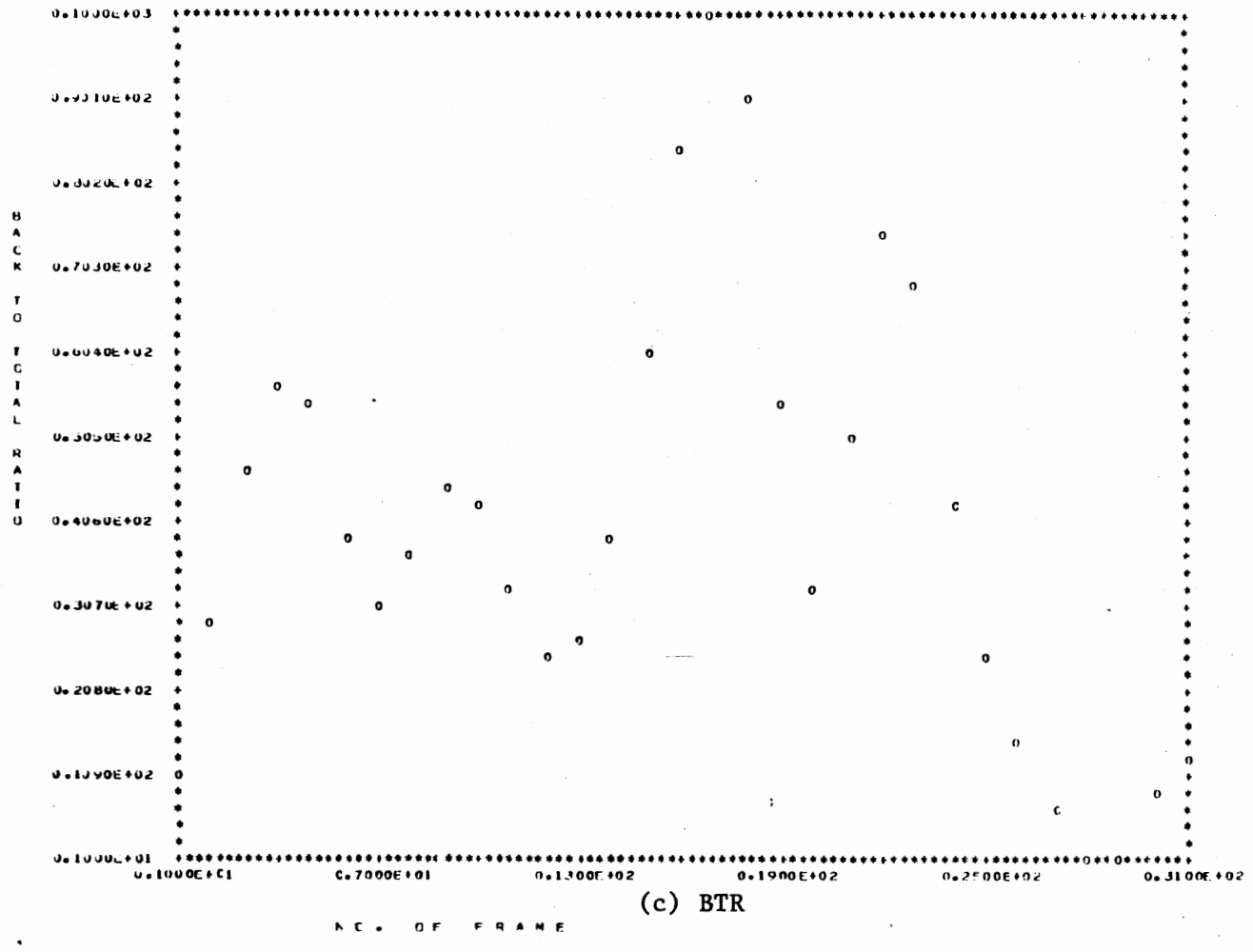
(a) FTR

Figure 84. Smoothed and Quantized Feature Parameter for Digit Zero, i.e. /sefr/ Spoken in Arabic



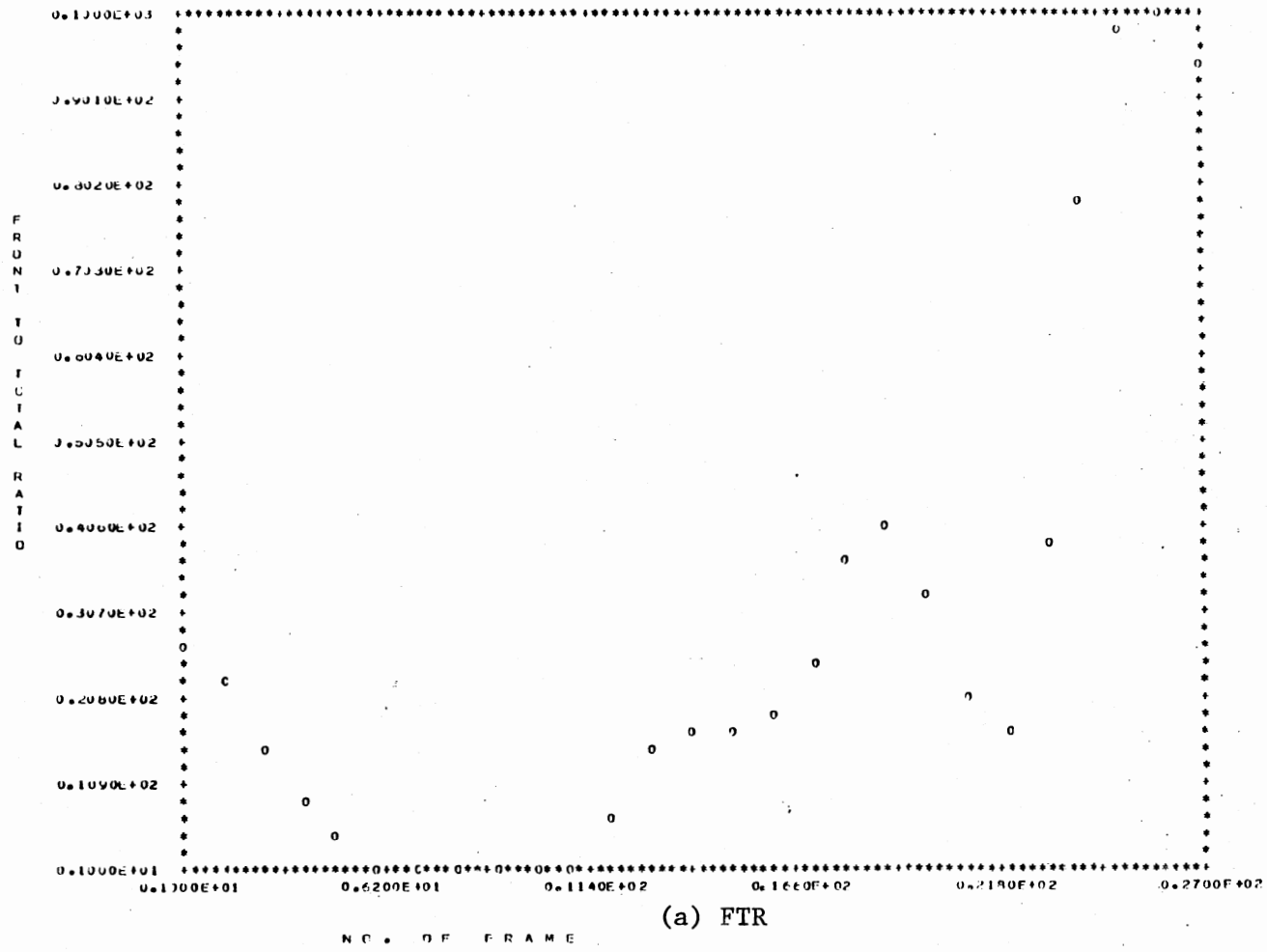
(b) CTR

Figure 84. (Continued)



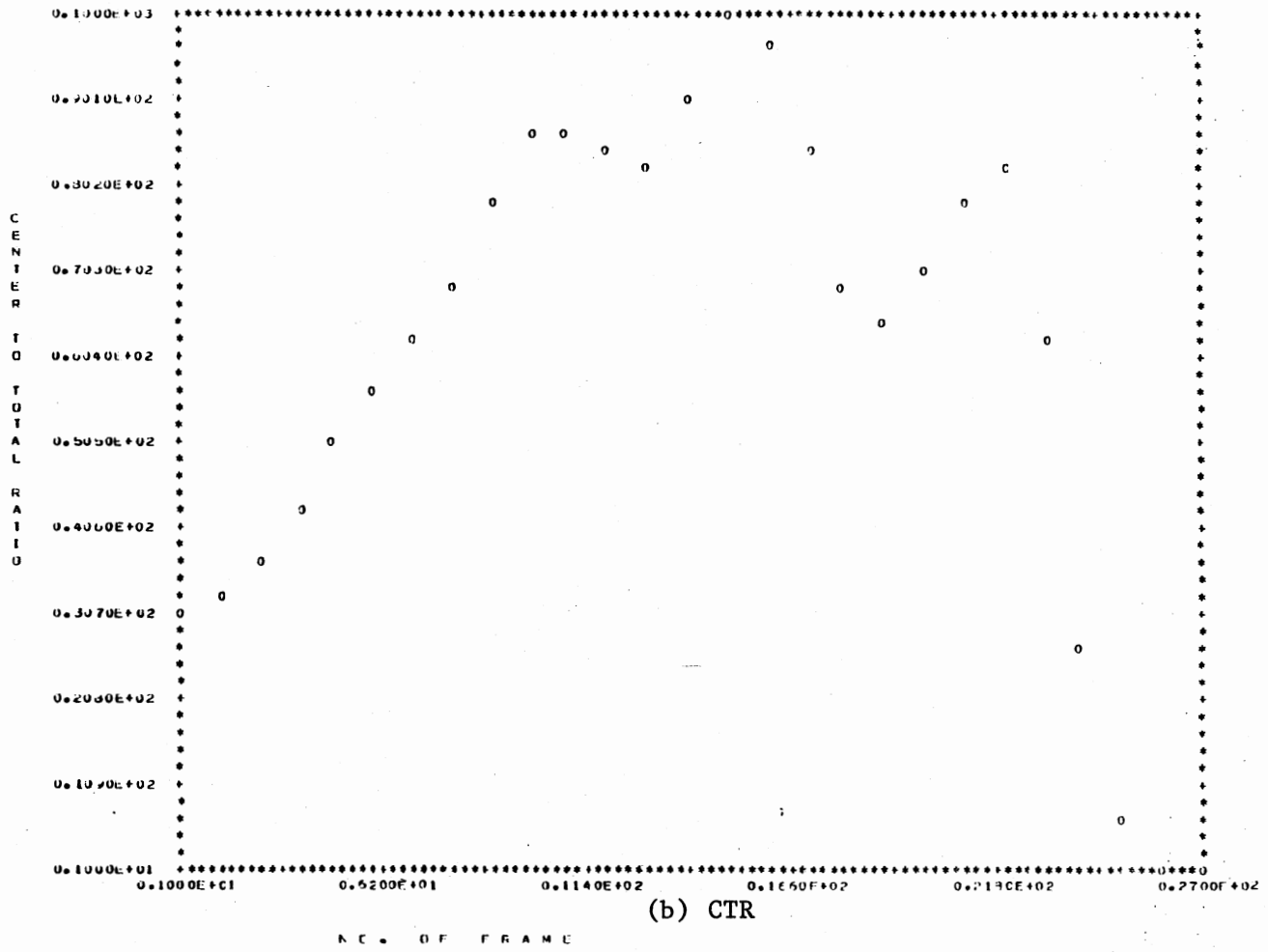
(c) BTR

Figure 84. (Continued)



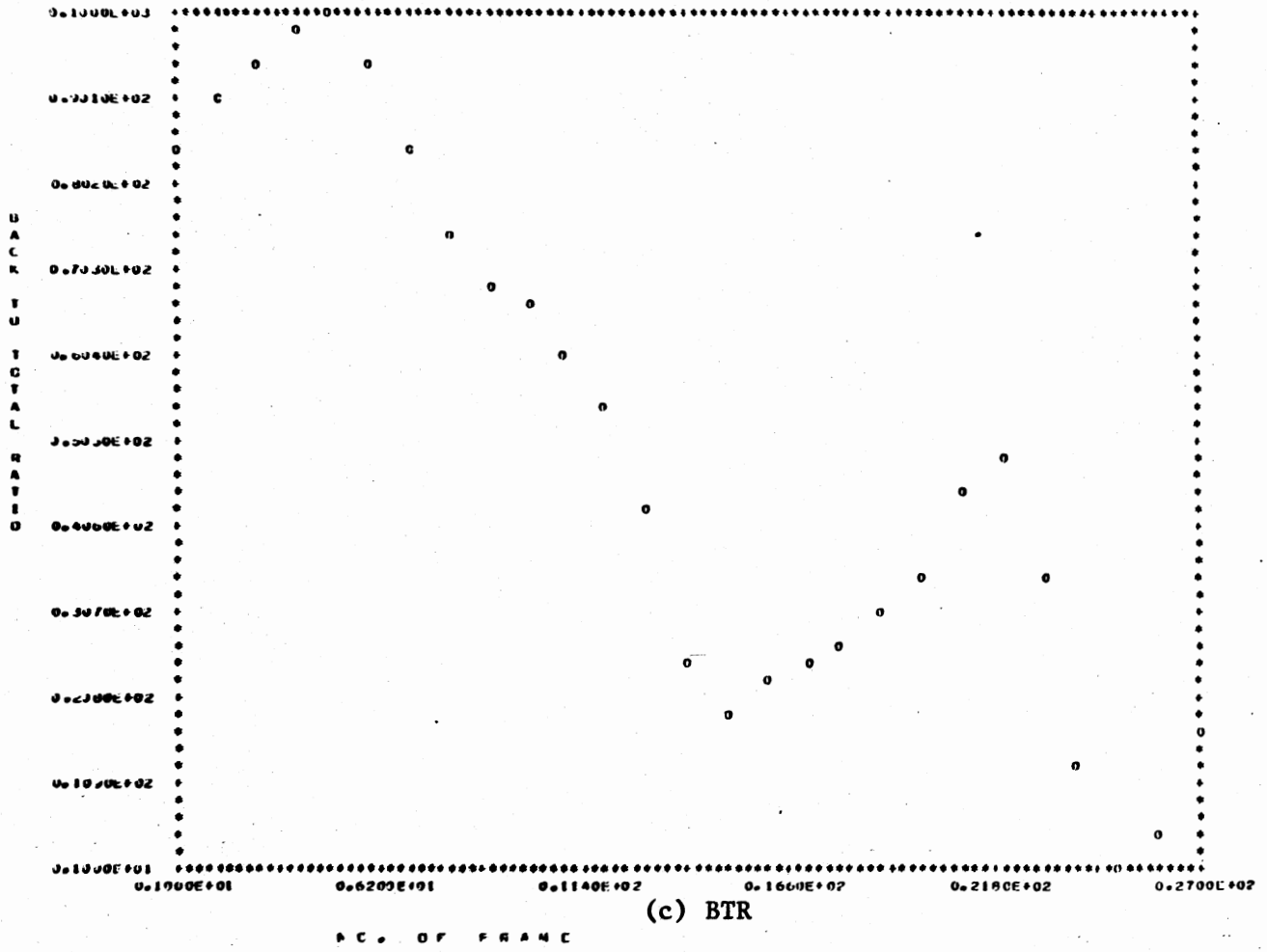
(a) FTR

Figure 85. Smoothed and Quantized Feature Parameter for Digit One, i.e. /ωâhid/ Spoken in Arabic



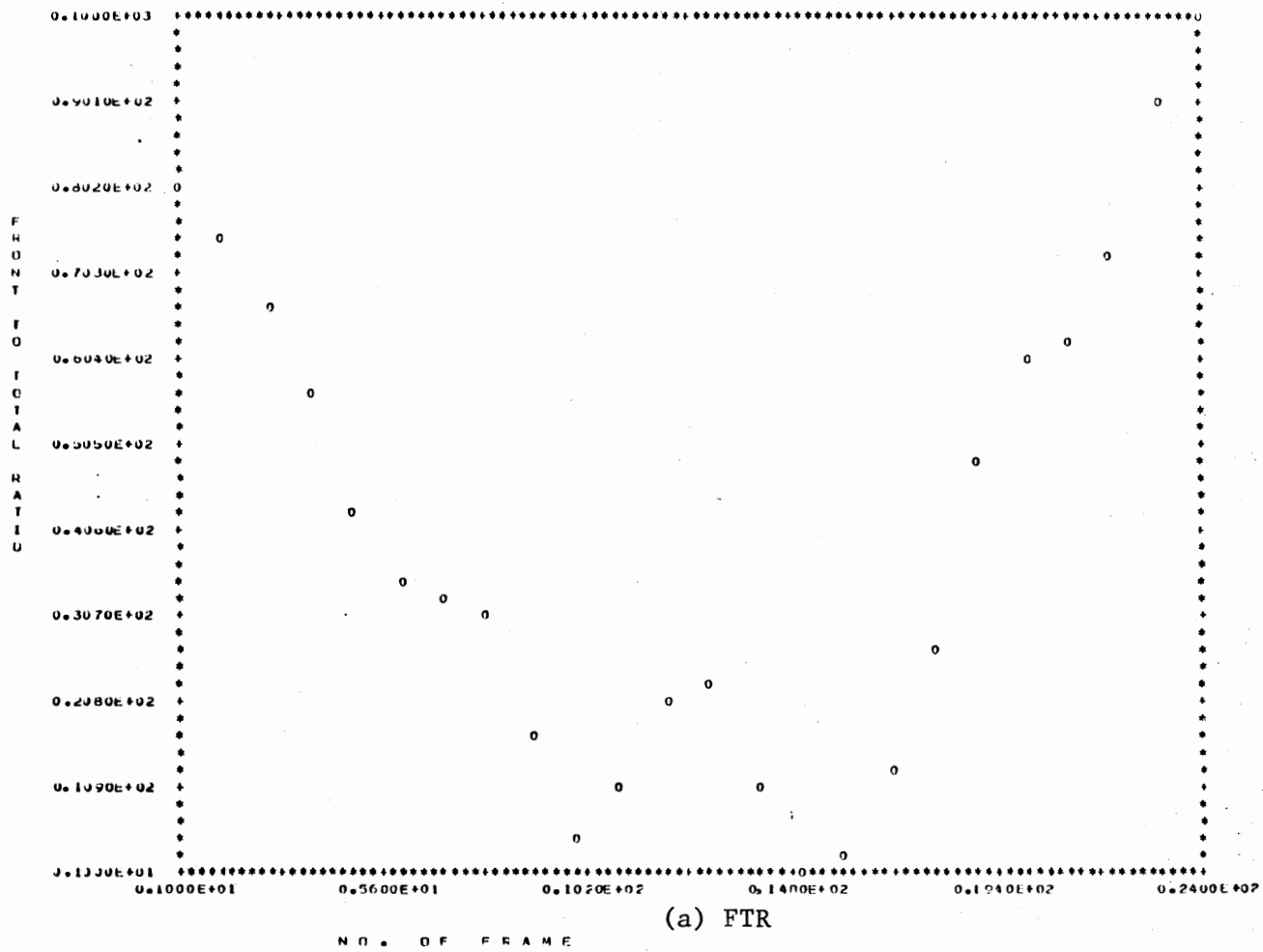
(b) CTR

Figure 85. (Continued)



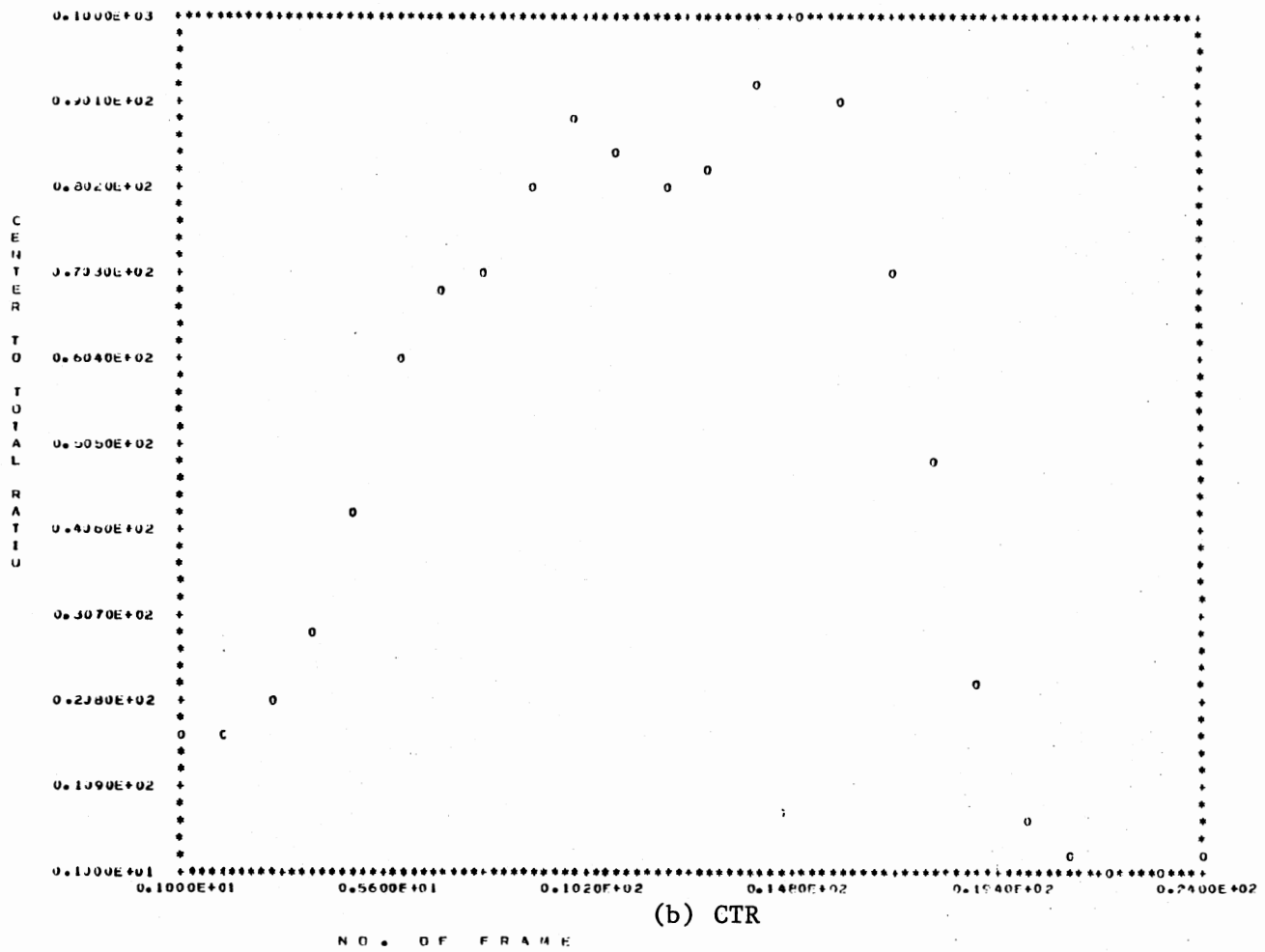
(c) BTR

Figure 85. (Continued)



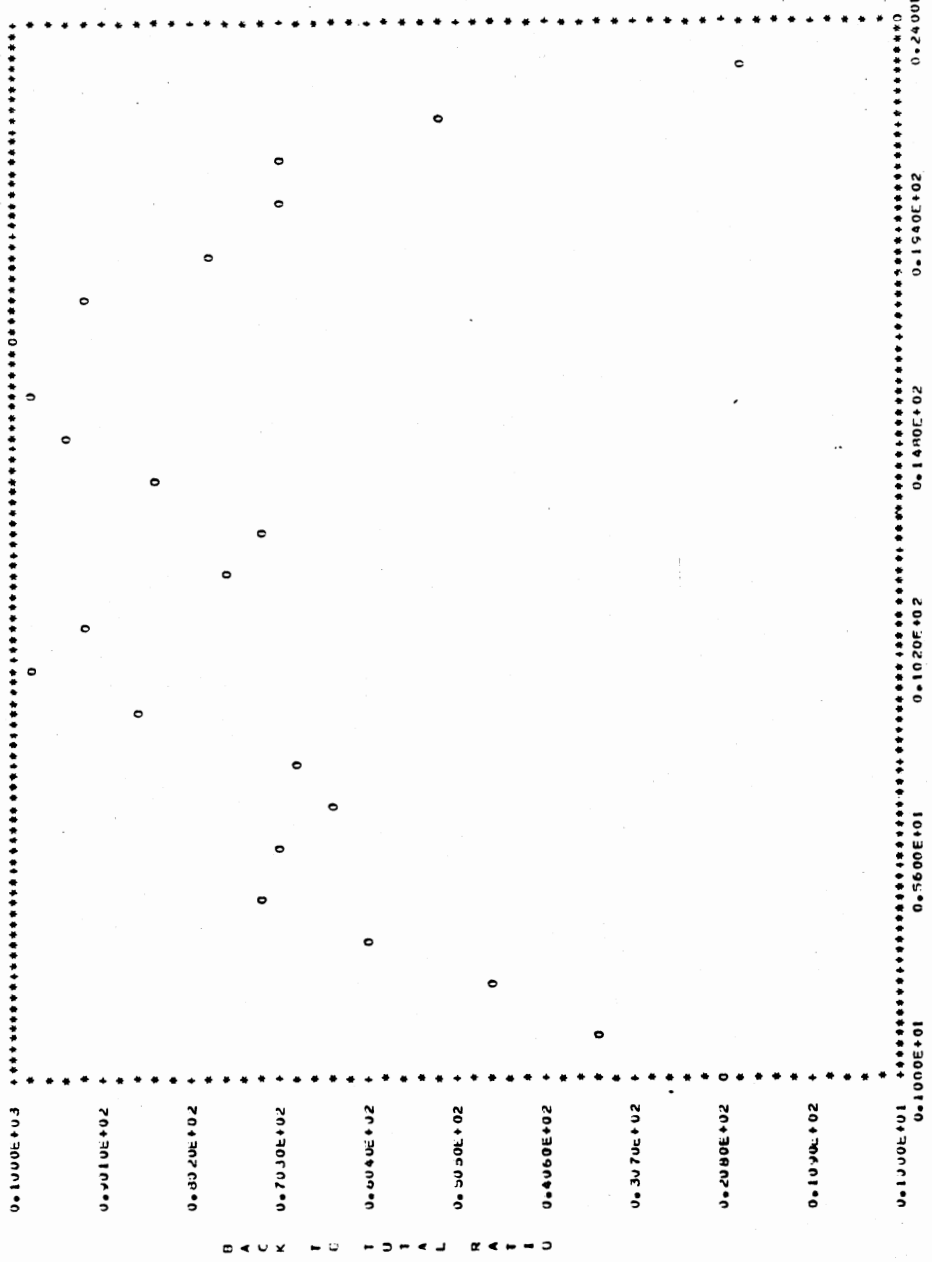
(a) FTR

Figure 86. Smoothed and Quantized Feature Parameter for Digit Two, i.e. /iθnân/ Spoken in Arabic



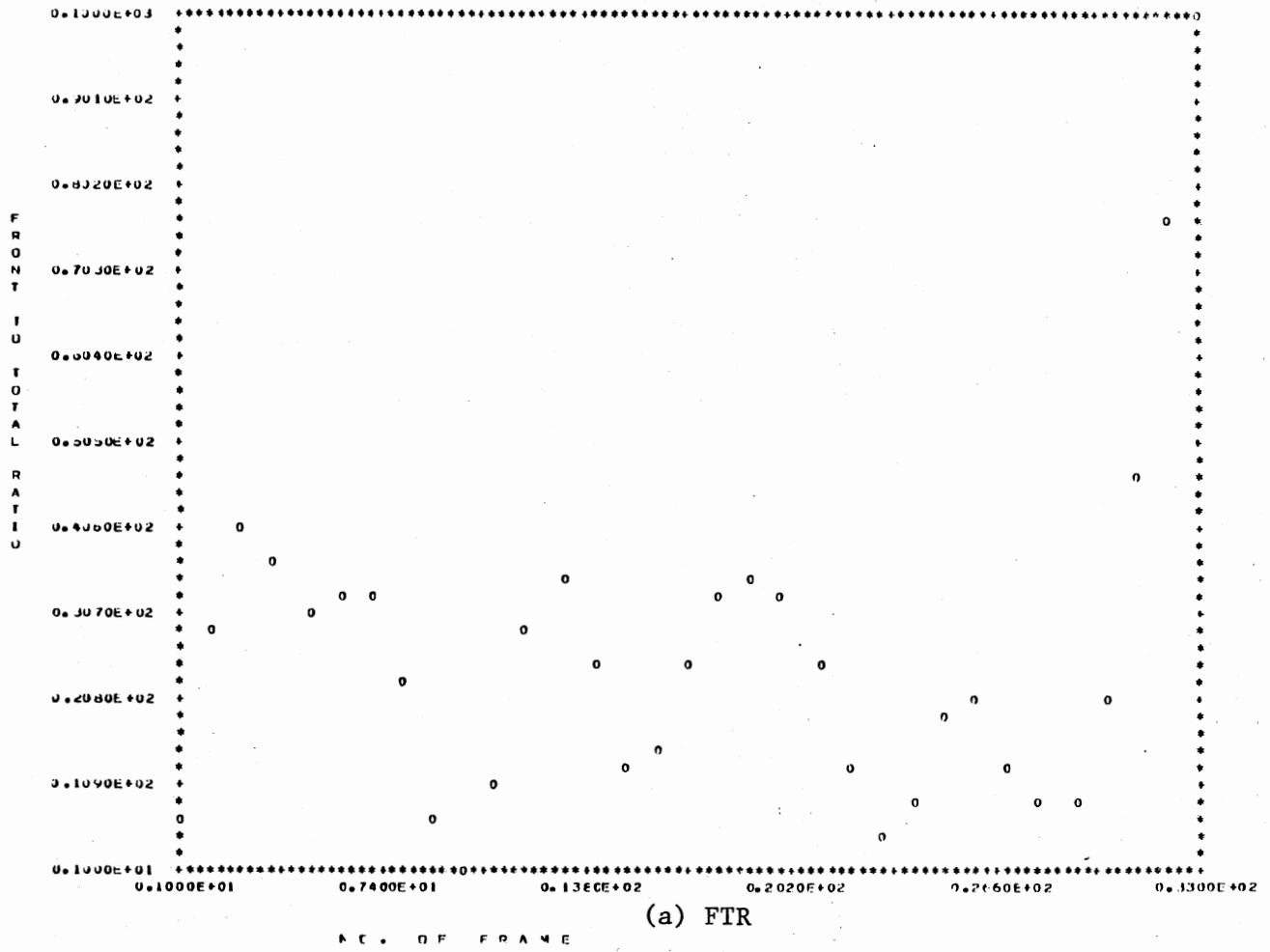
(b) CTR

Figure 86. (Continued)



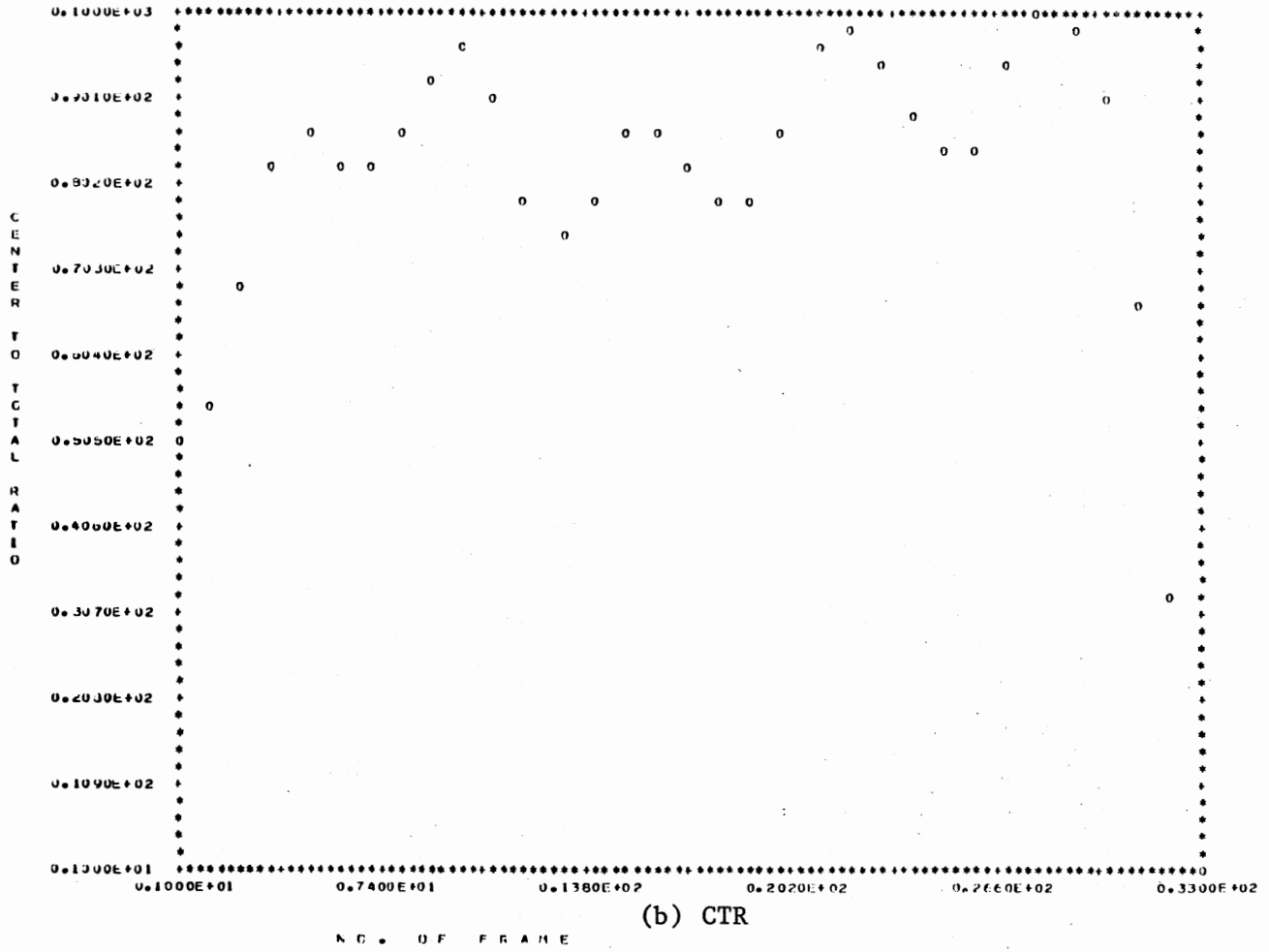
(c) BTR

Figure 86. (Continued)



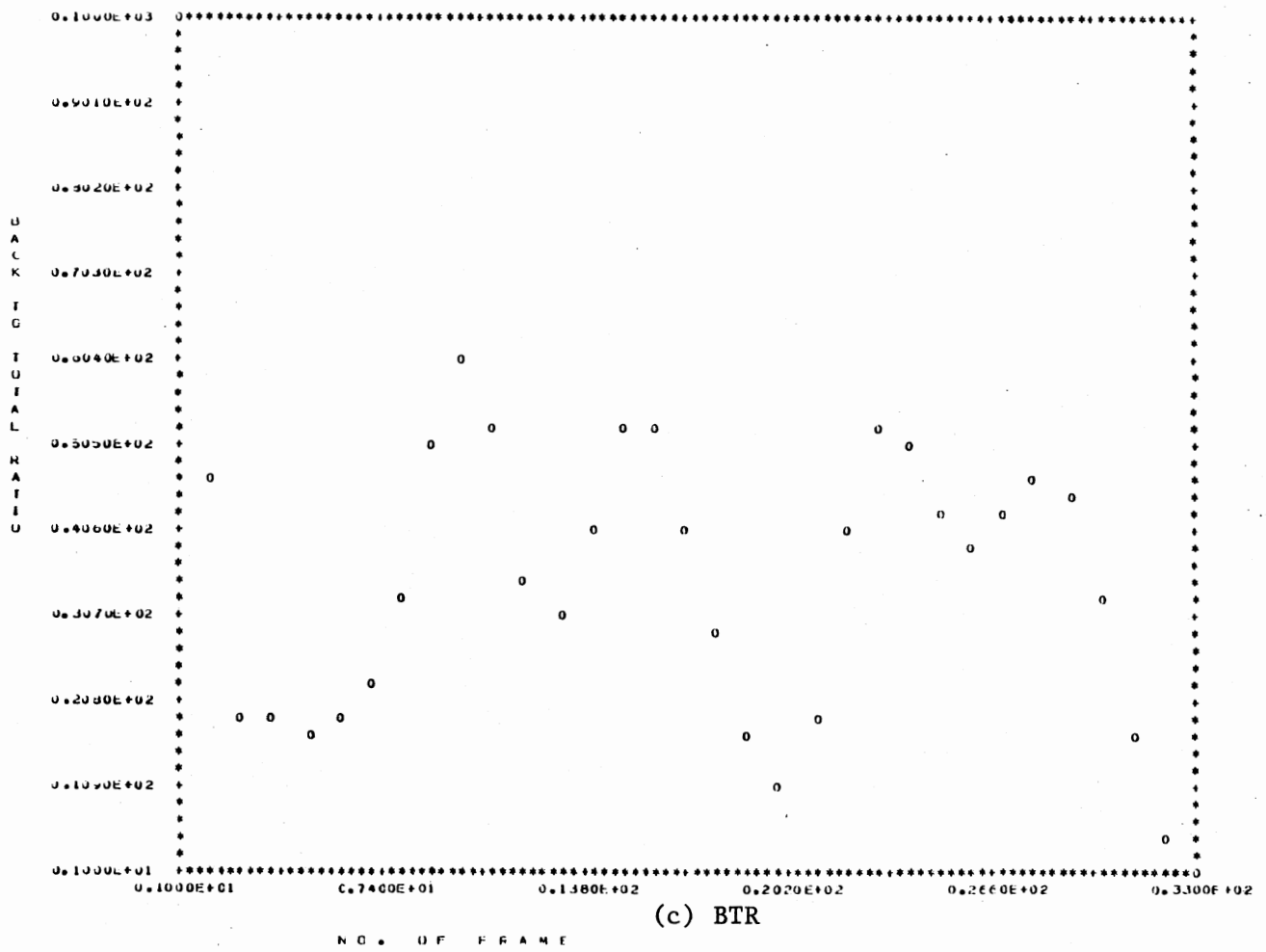
(a) FTR

Figure 87. Smoothed and Quantized Feature Parameter for Digit Three, i.e./θaℓaθðh/ Spoken in Arabic



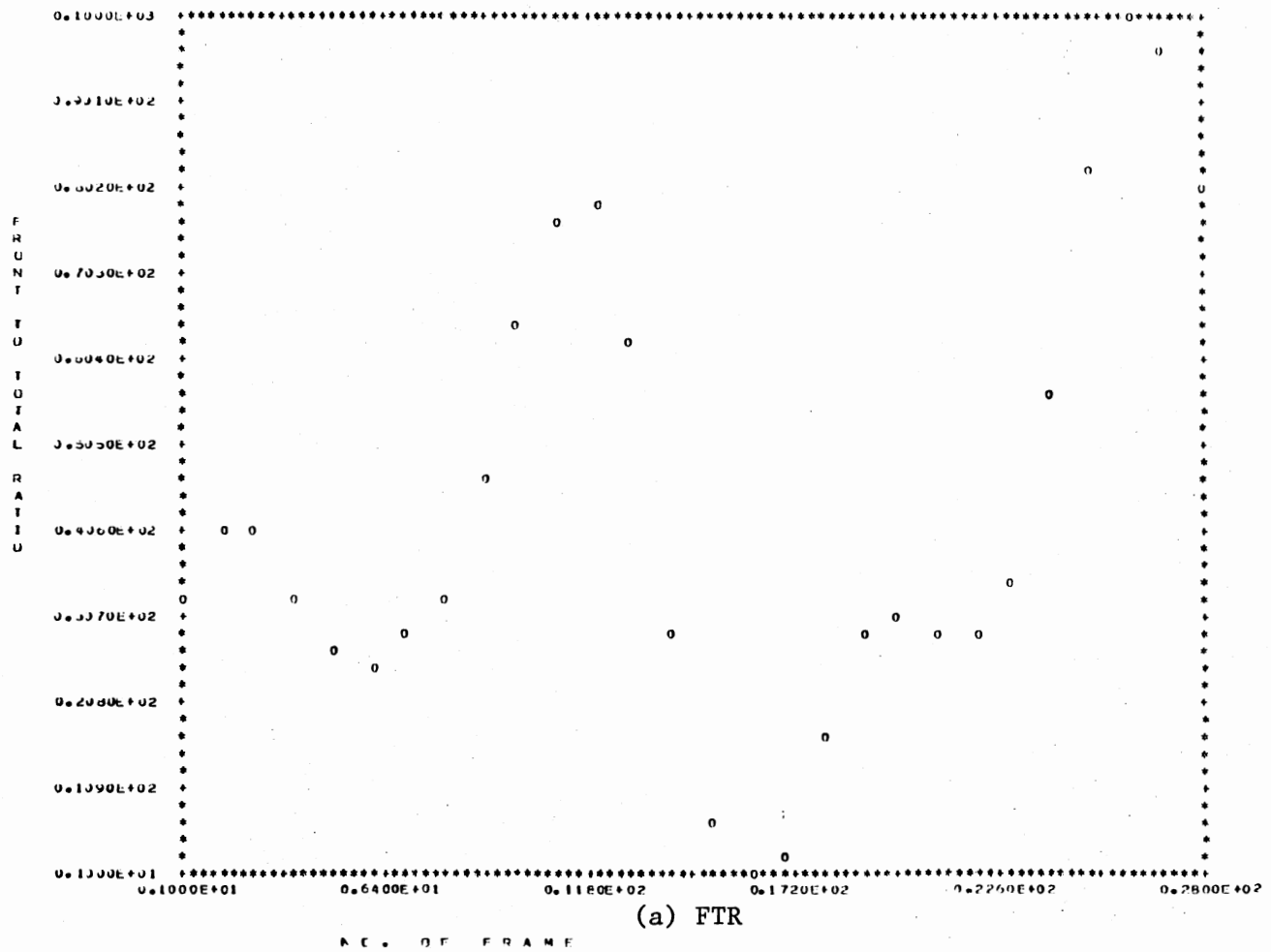
(b) CTR

Figure 87. (Continued)



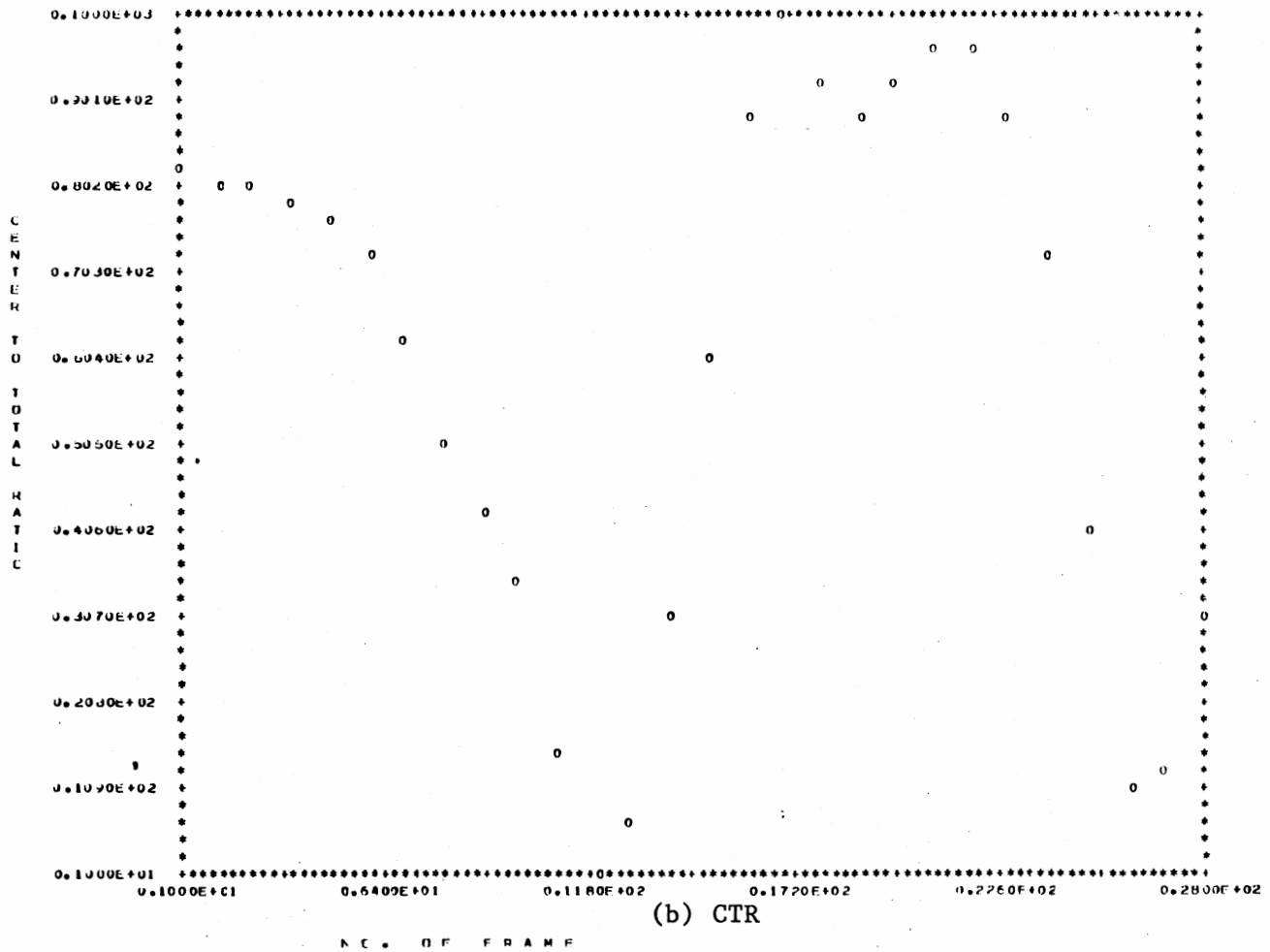
(c) BTR

Figure 87. (Continued)



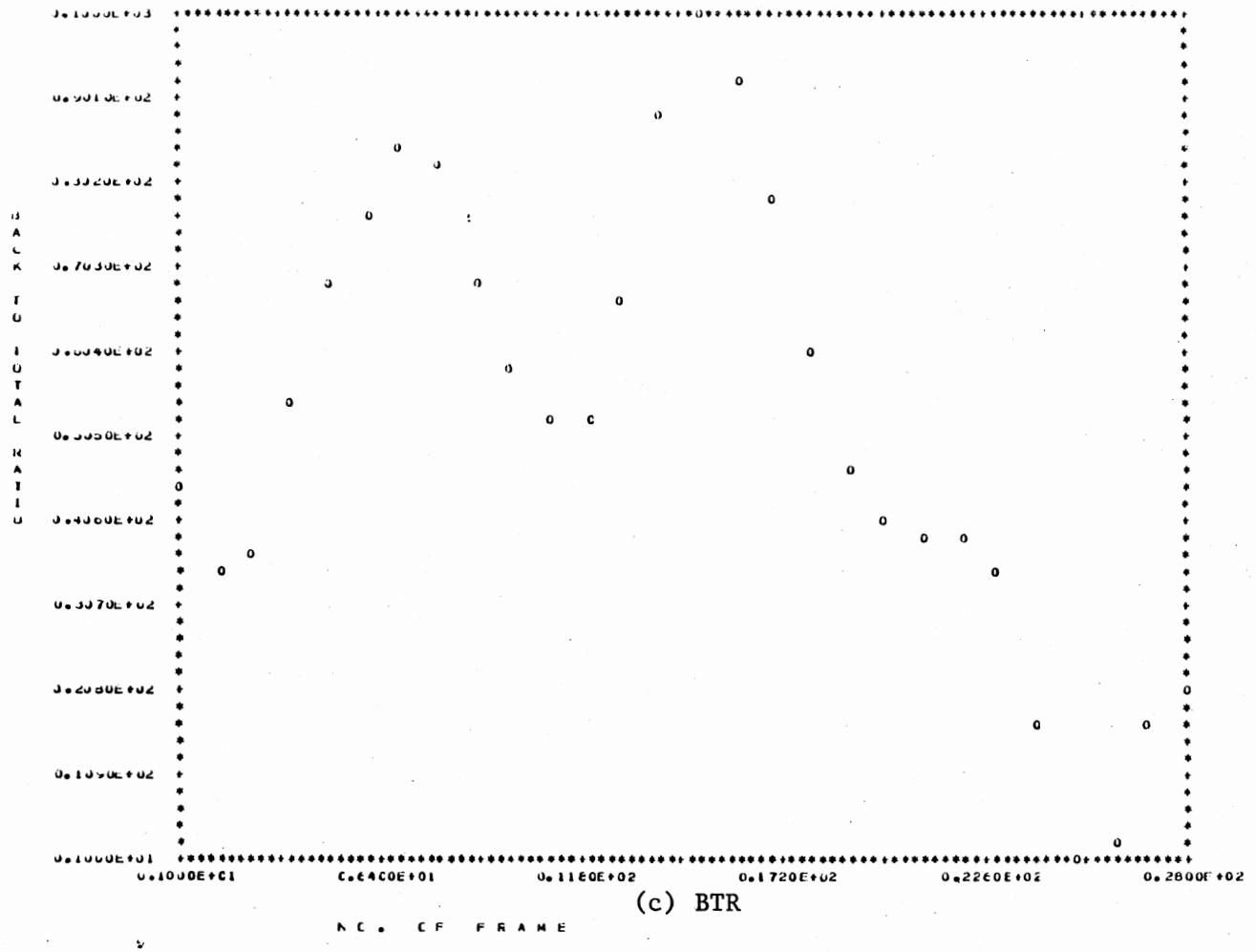
(a) FTR

Figure 88. Smoothed and Quantized Feature Parameter for Digit Four, i.e. /arbaɒðh/ Spoken in Arabic



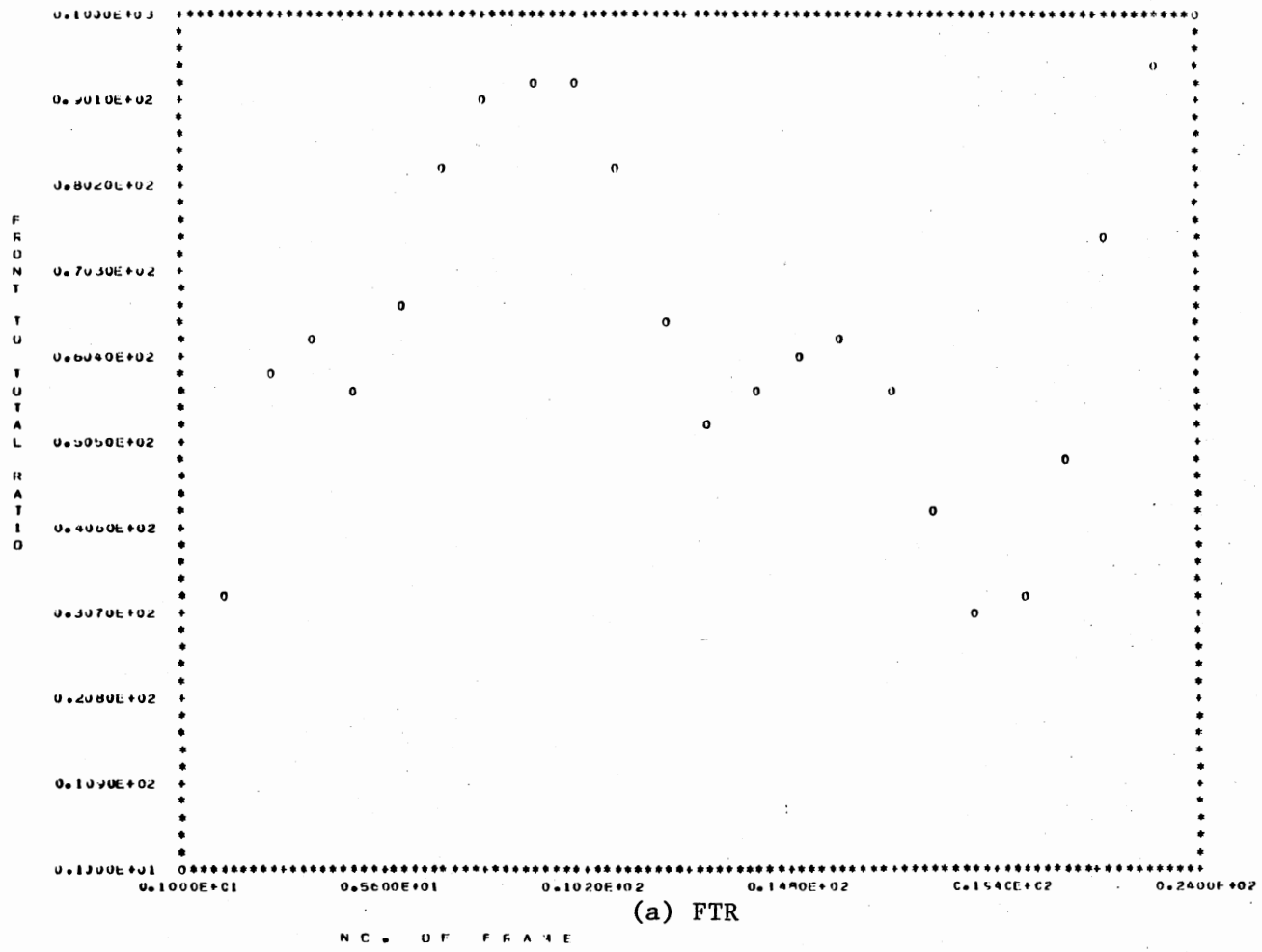
(b) CTR

Figure 88. (Continued)



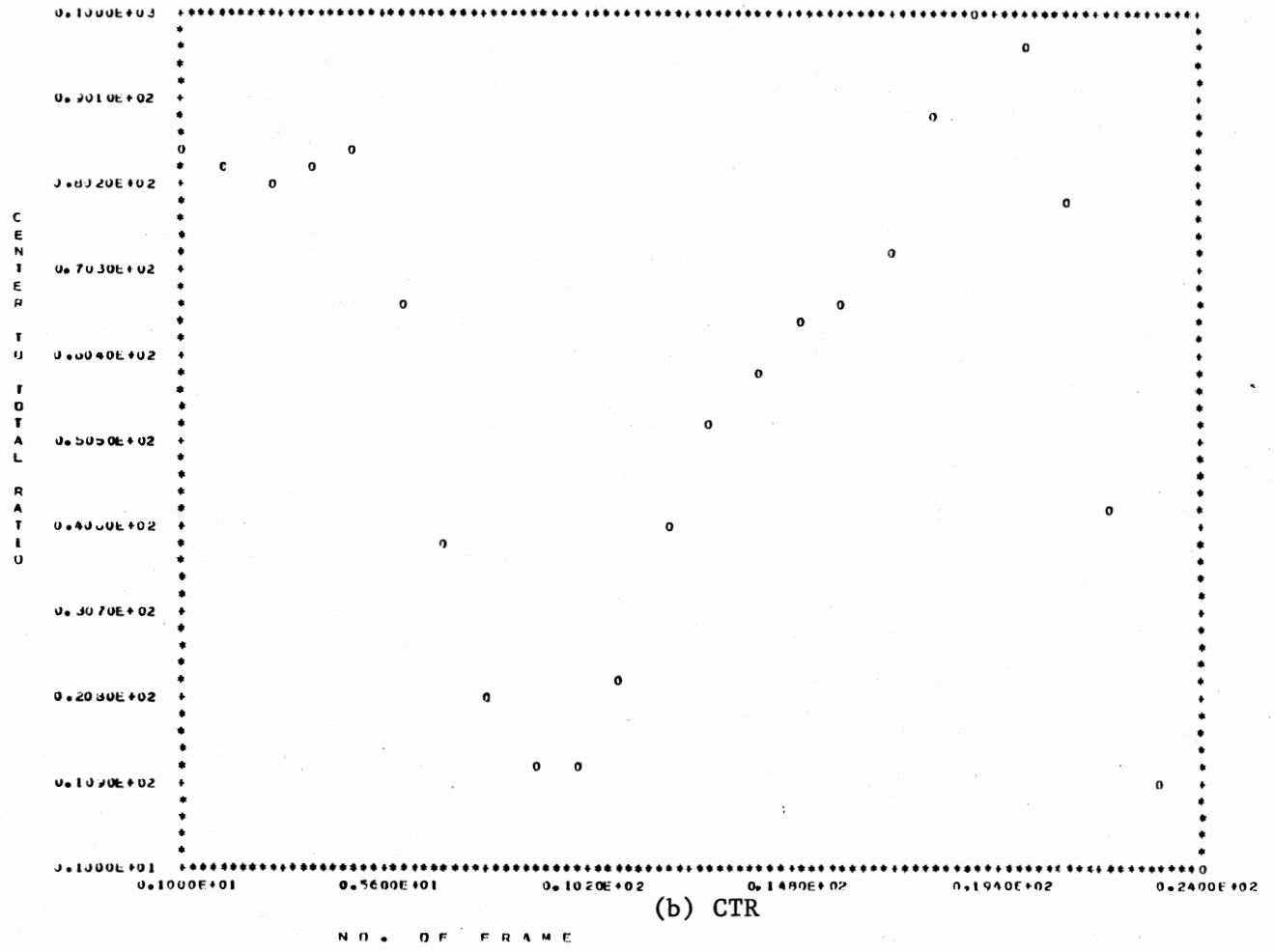
(c) BTR

Figure 88. (Continued)



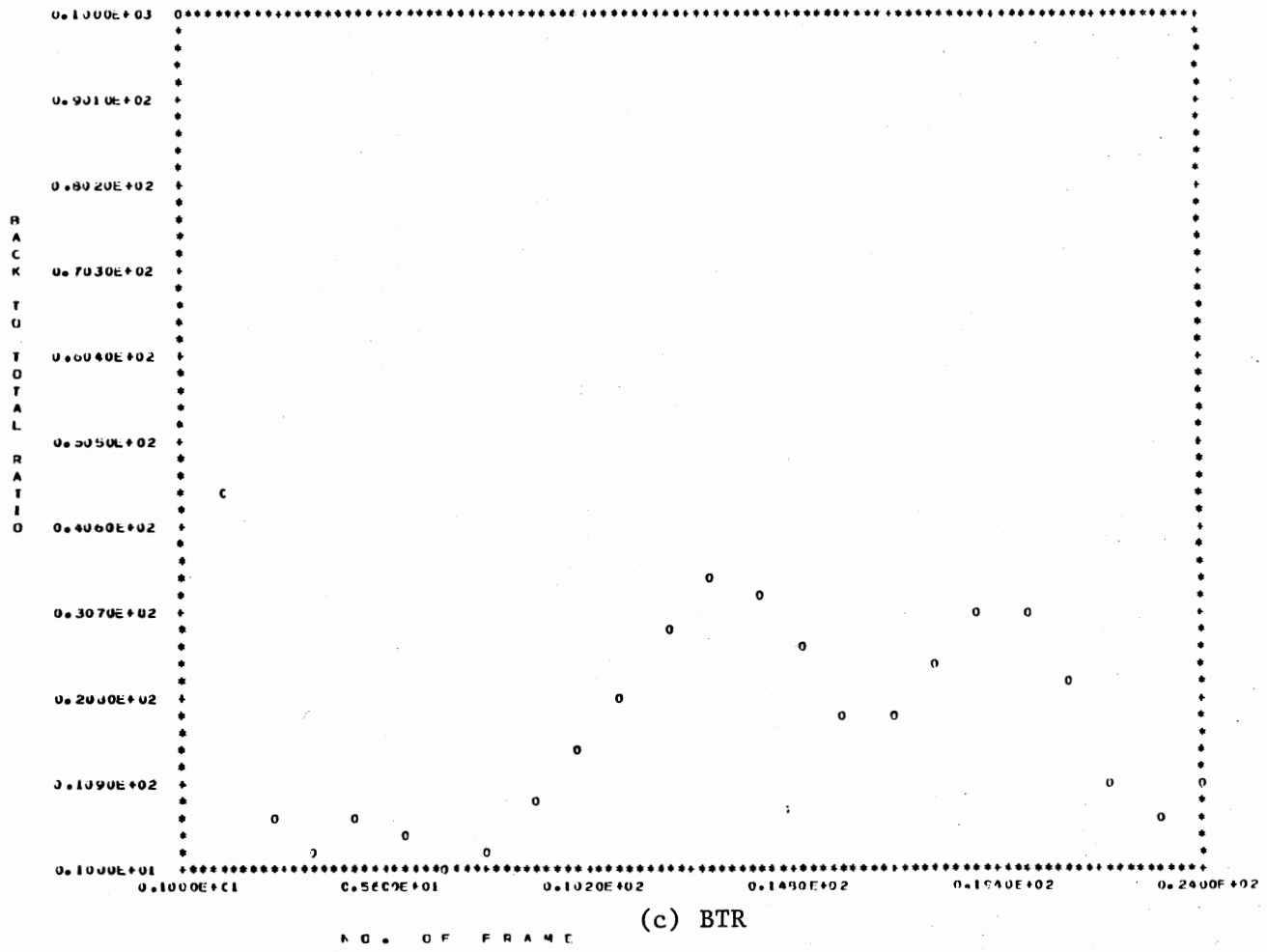
(a) FTR

Figure 89. Smoothed and Quantized Feature Parameter for Digit Five, i.e. /khʌmsəh/ Spoken in Arabic



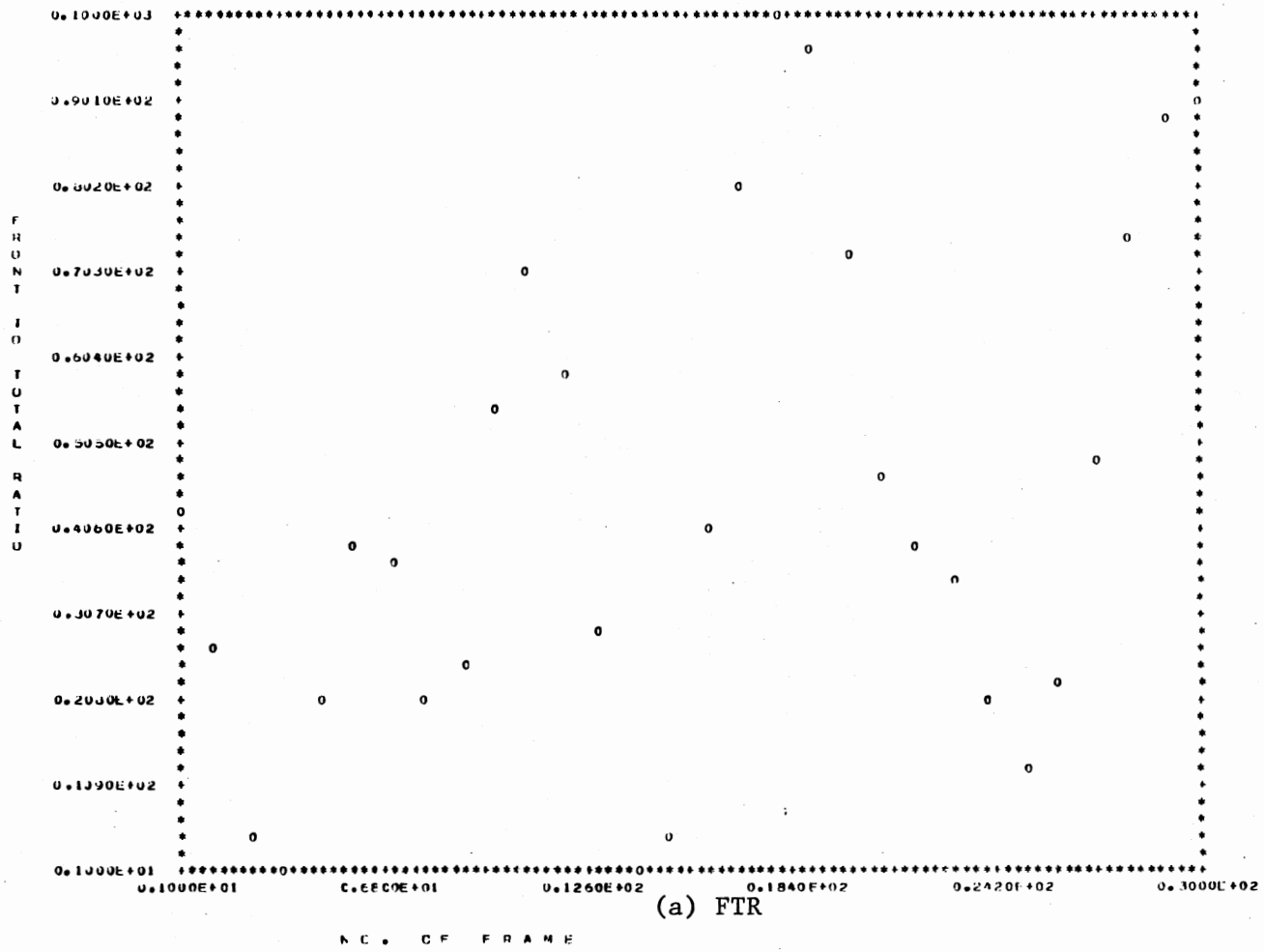
(b) CTR

Figure 89. (Continued)



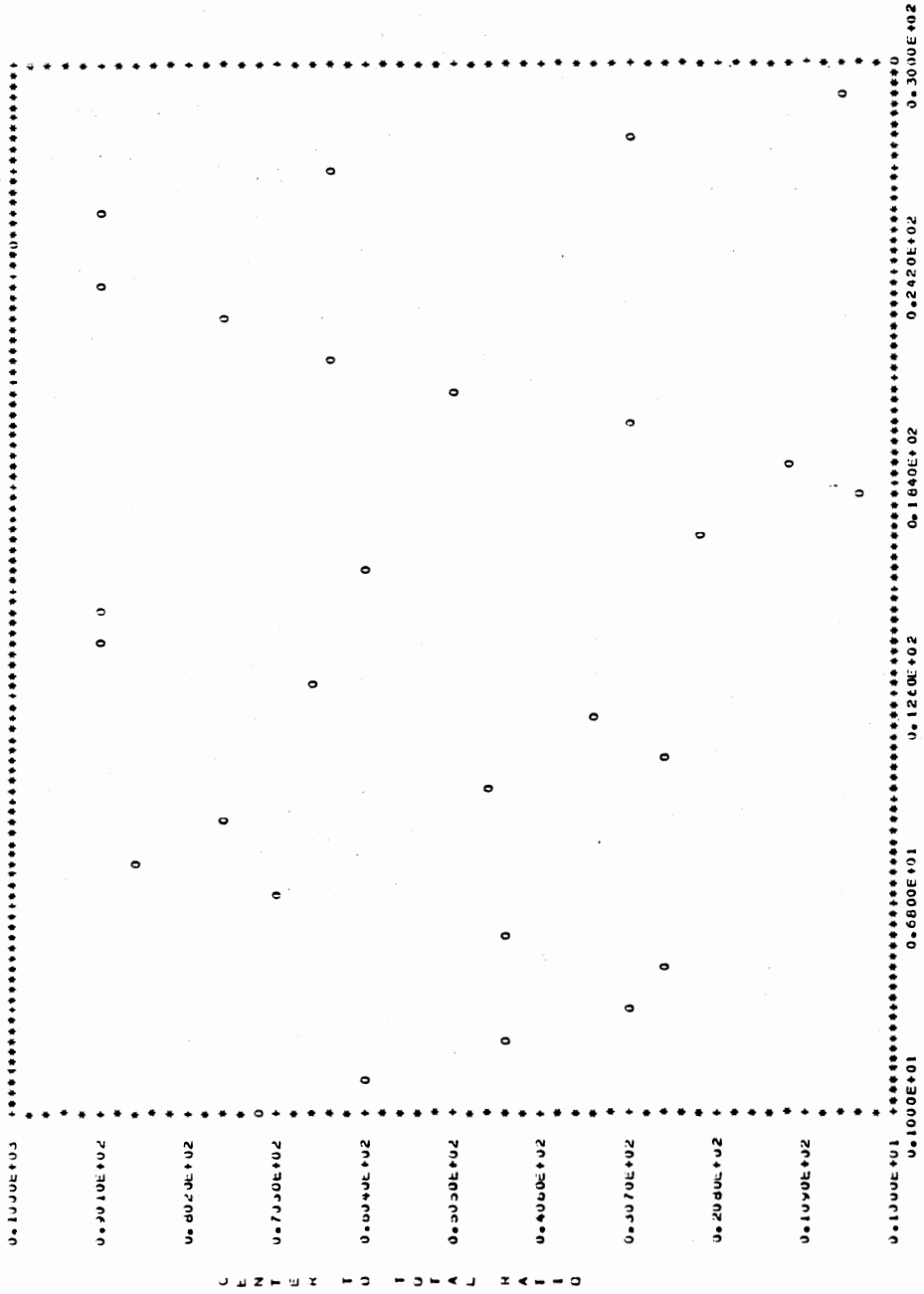
(c) BTR

Figure 89. (Continued)



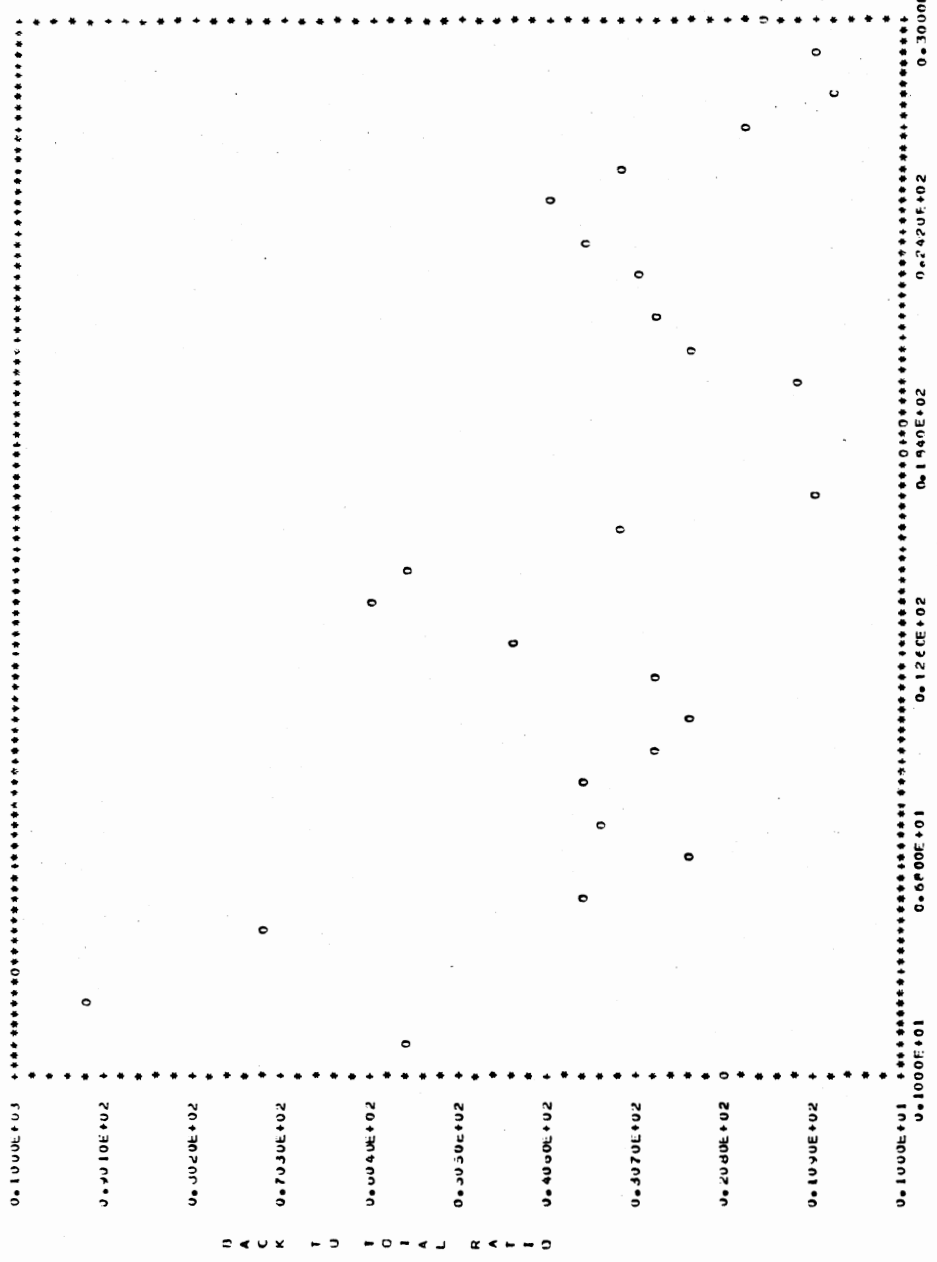
(a) FTR

Figure 90. Smoothed and Quantized Feature Parameters for Digit Six, i.e. /sIttəh/ Spoken in Arabic



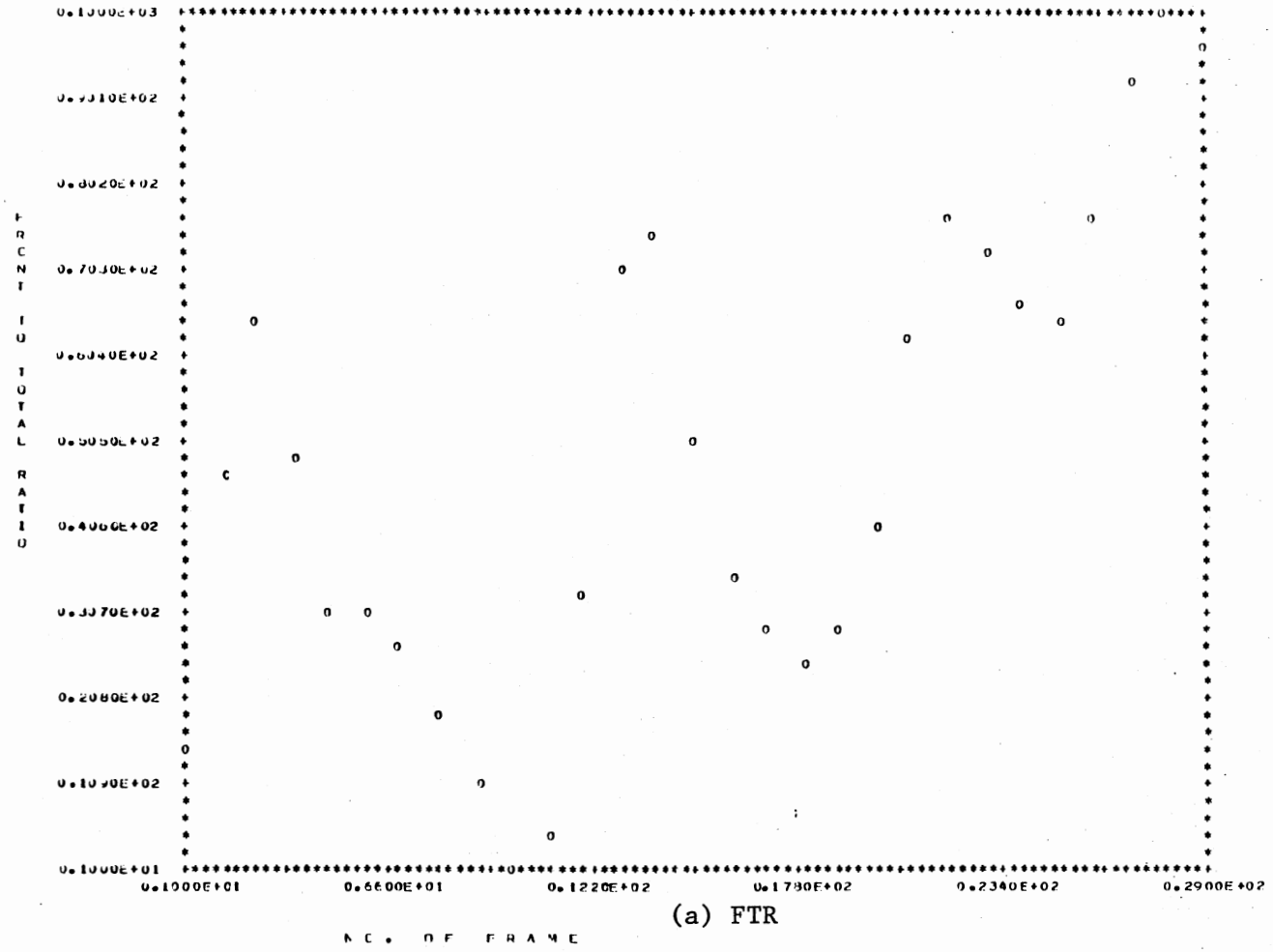
(b) CTR

Figure 90. (Continued)



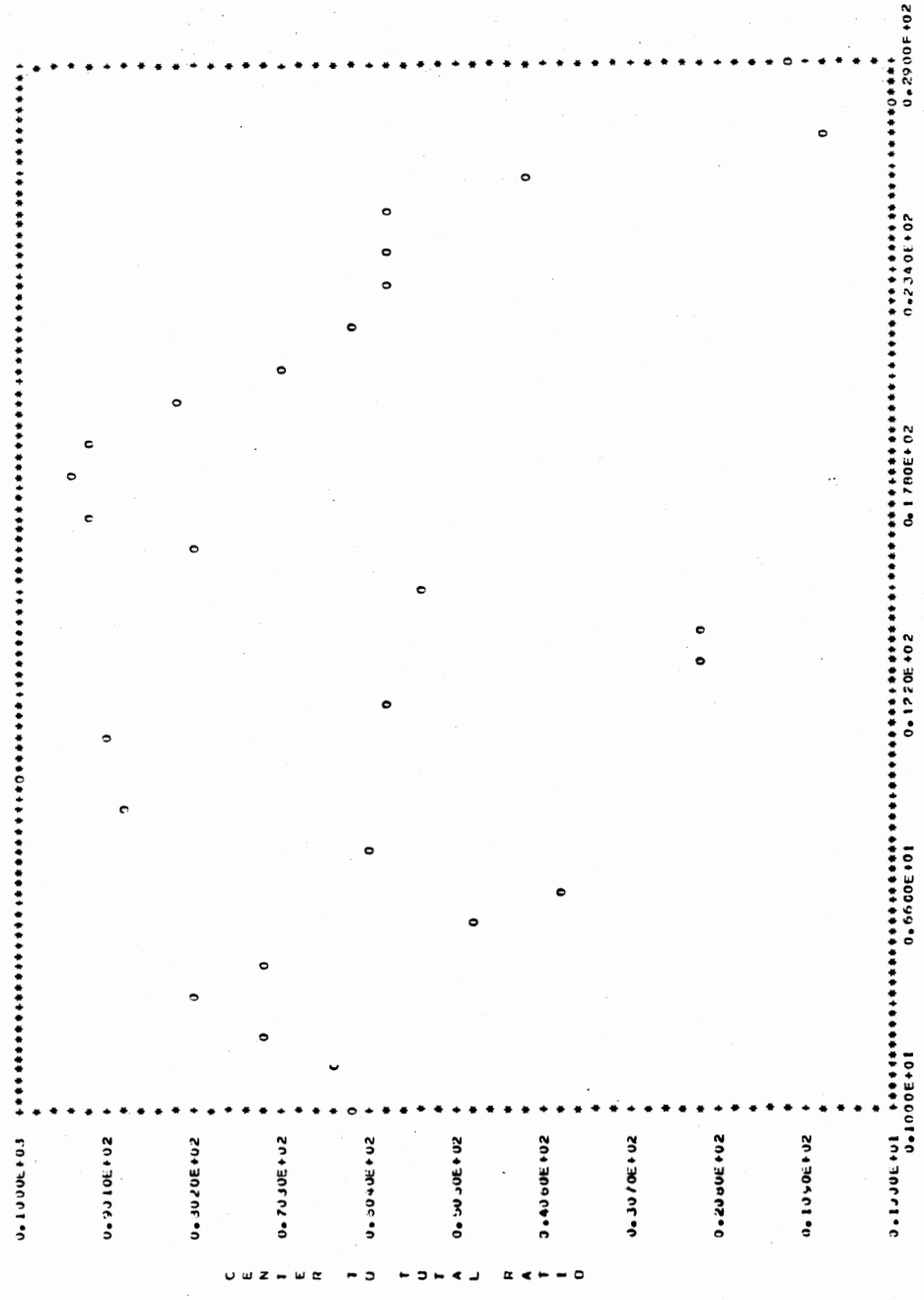
(c) BTR

Figure 90. (Continued)



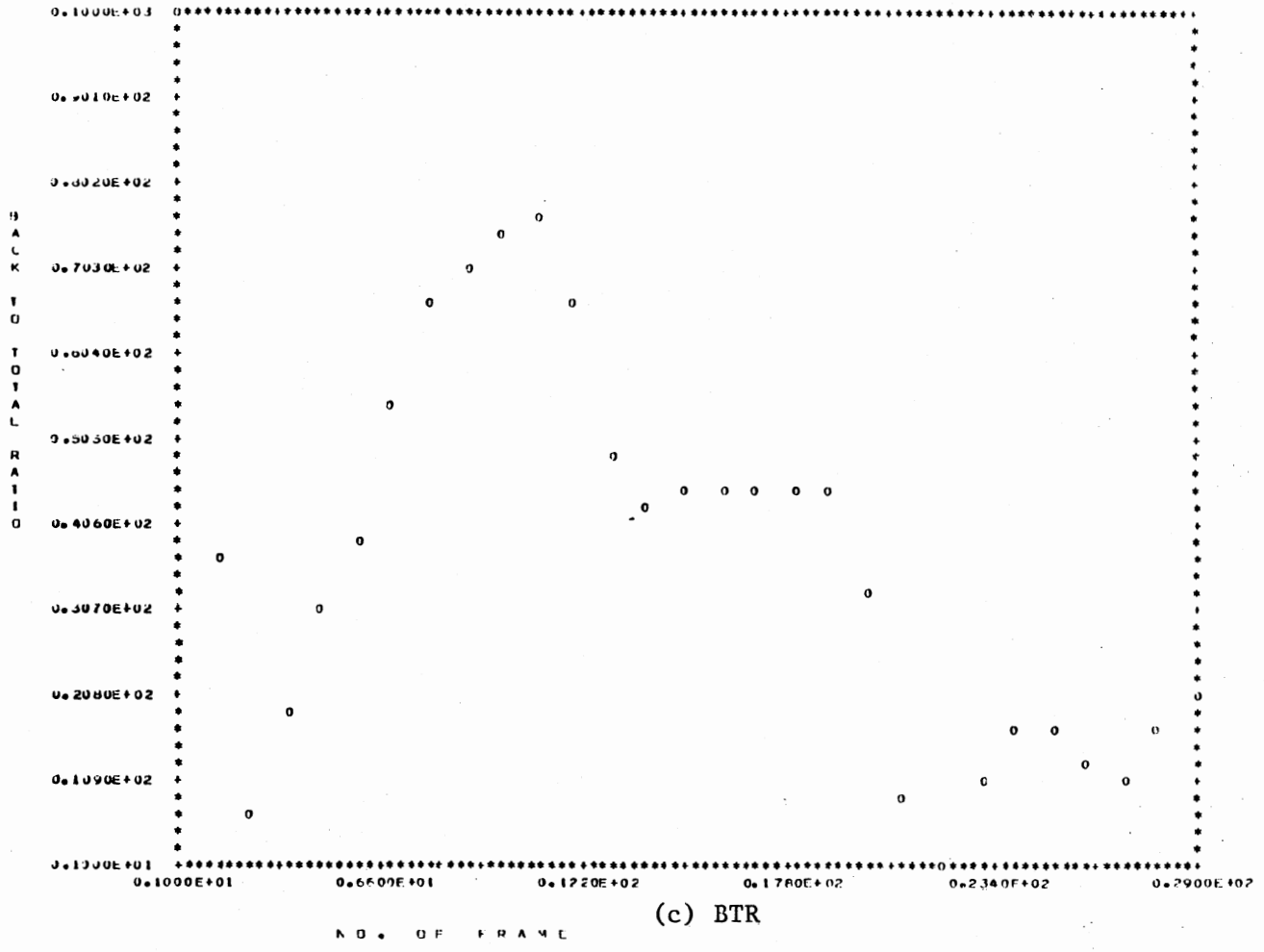
(a) FTR

Figure 91. Smoothed and Quantized Feature Parameter for Digit Seven, /sʌbɒðh/ Spoken in Arabic



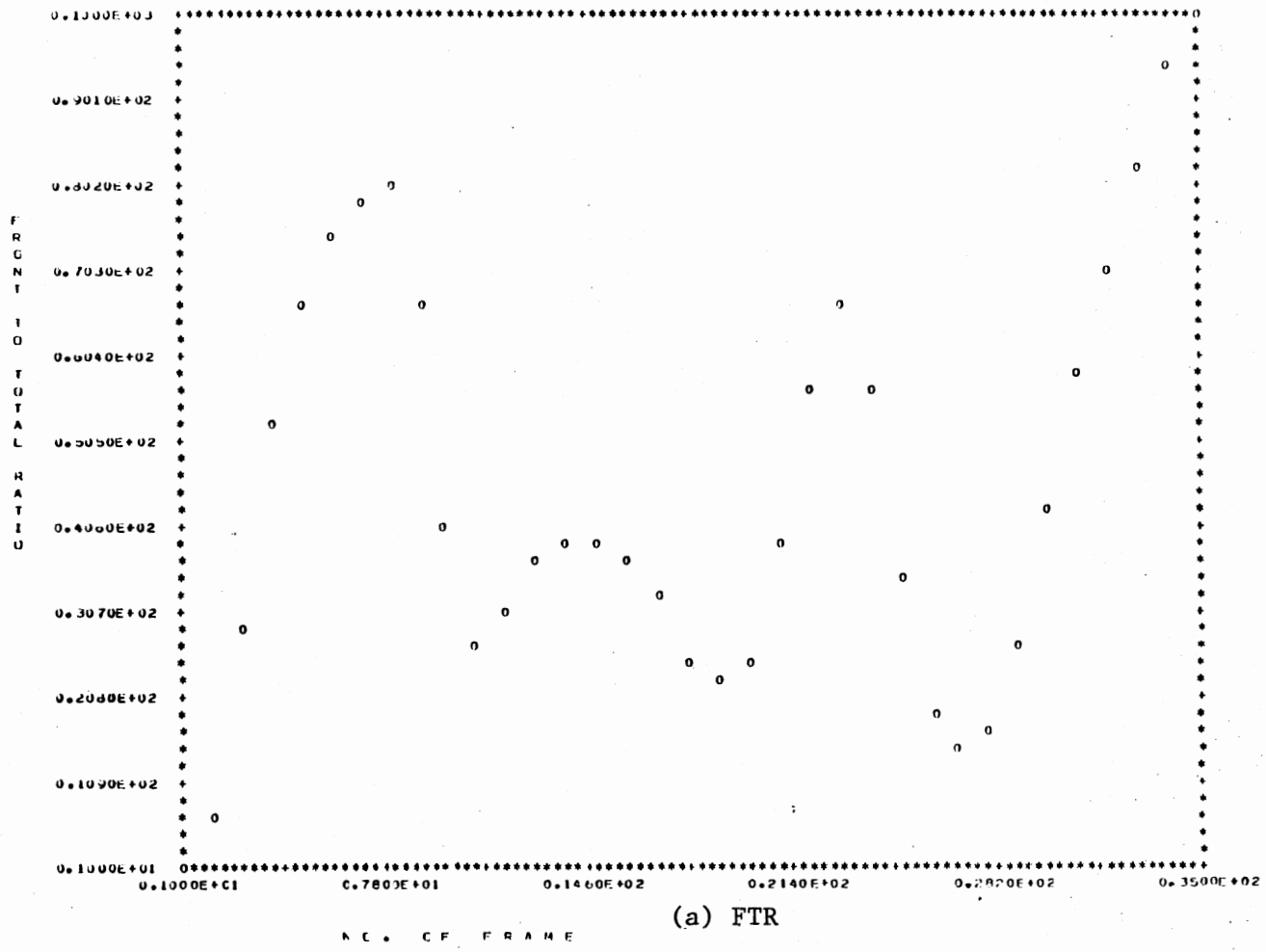
(b) CTR

Figure 91. (Continued)



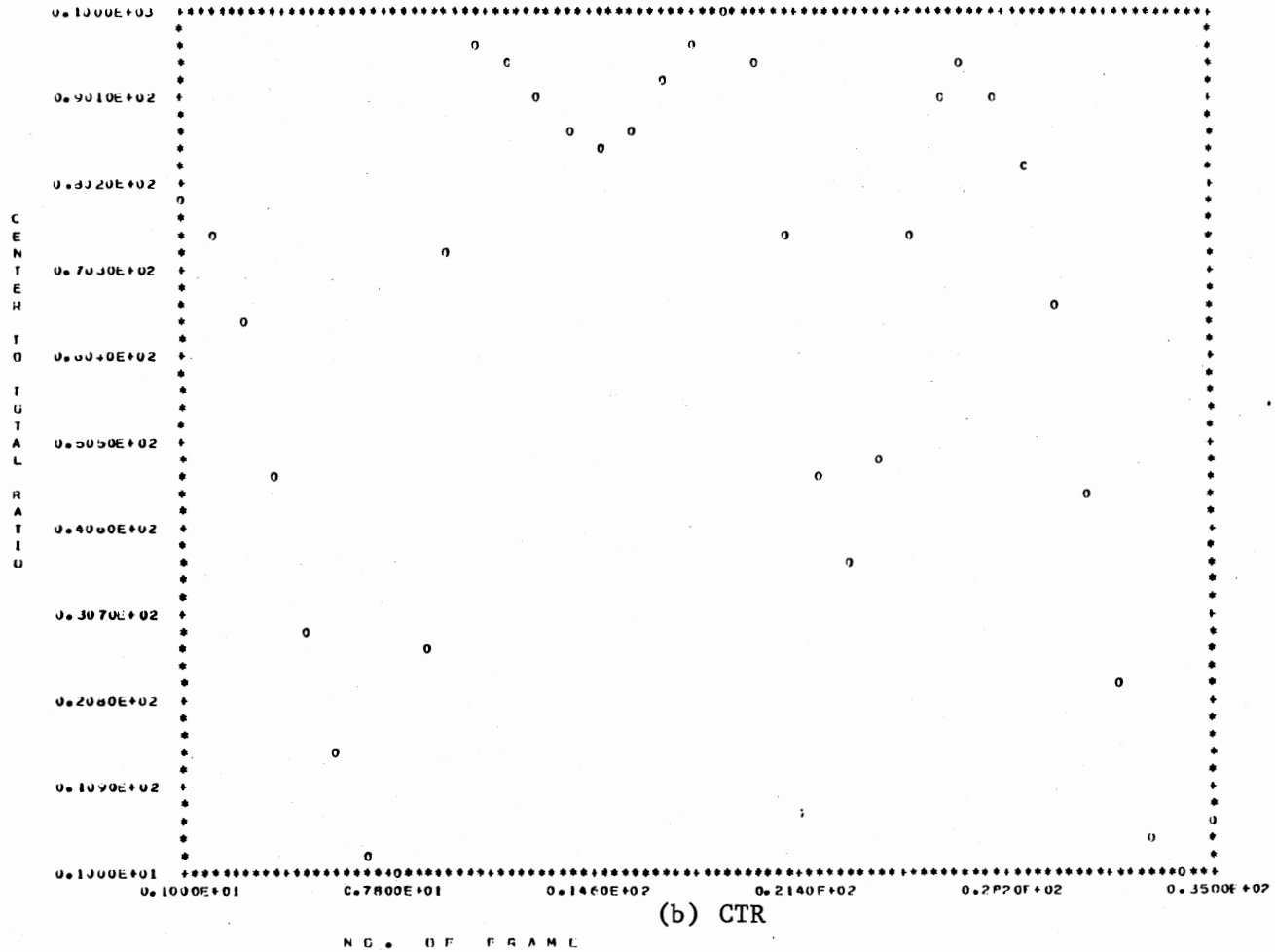
(c) BTR

Figure 91. (Continued)



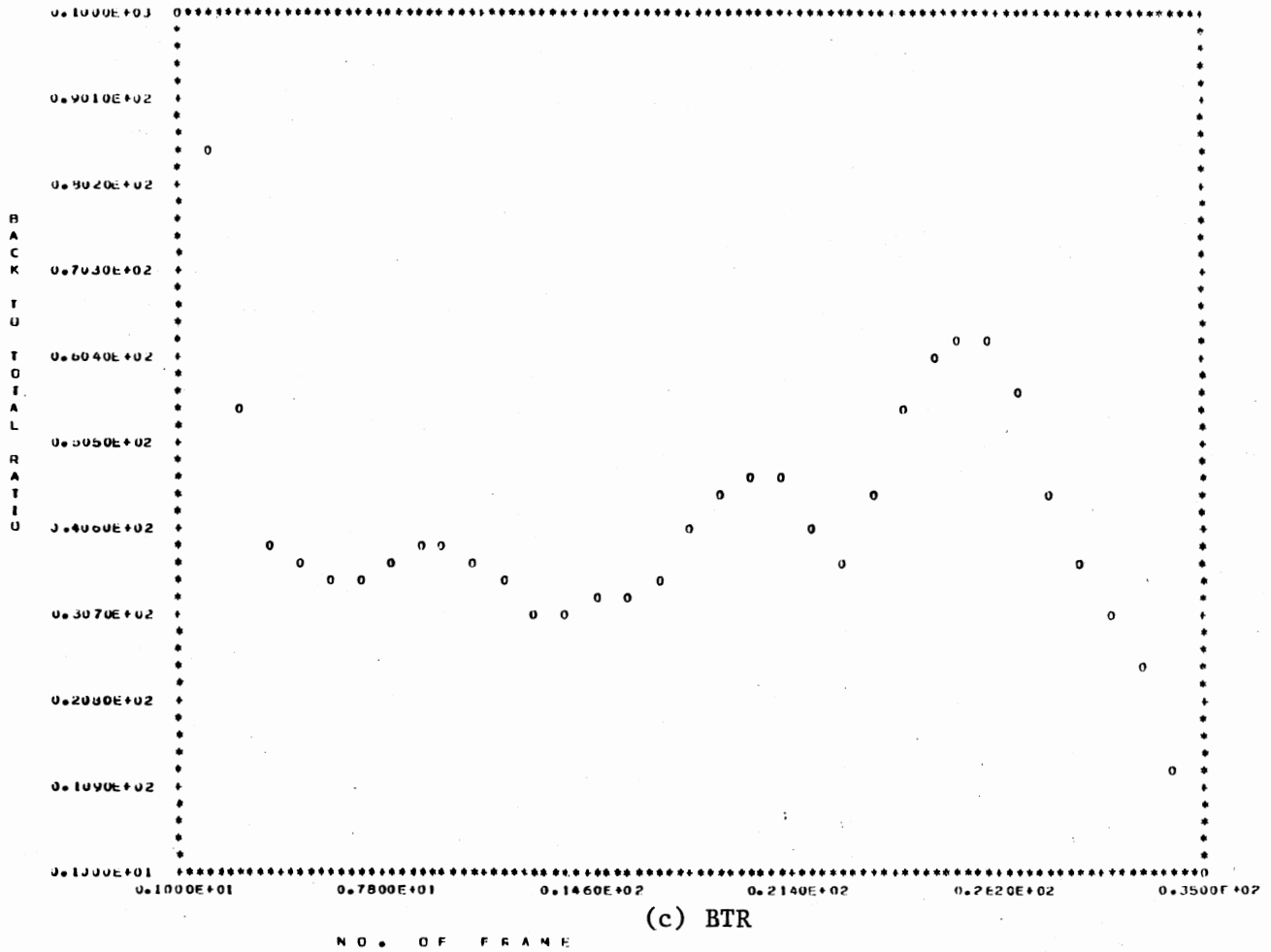
(a) FTR

Figure 92. Smoothed and Quantized Feature Parameter for Digit Eight, i.e. /θamâny∂h/ Spoken in Arabic



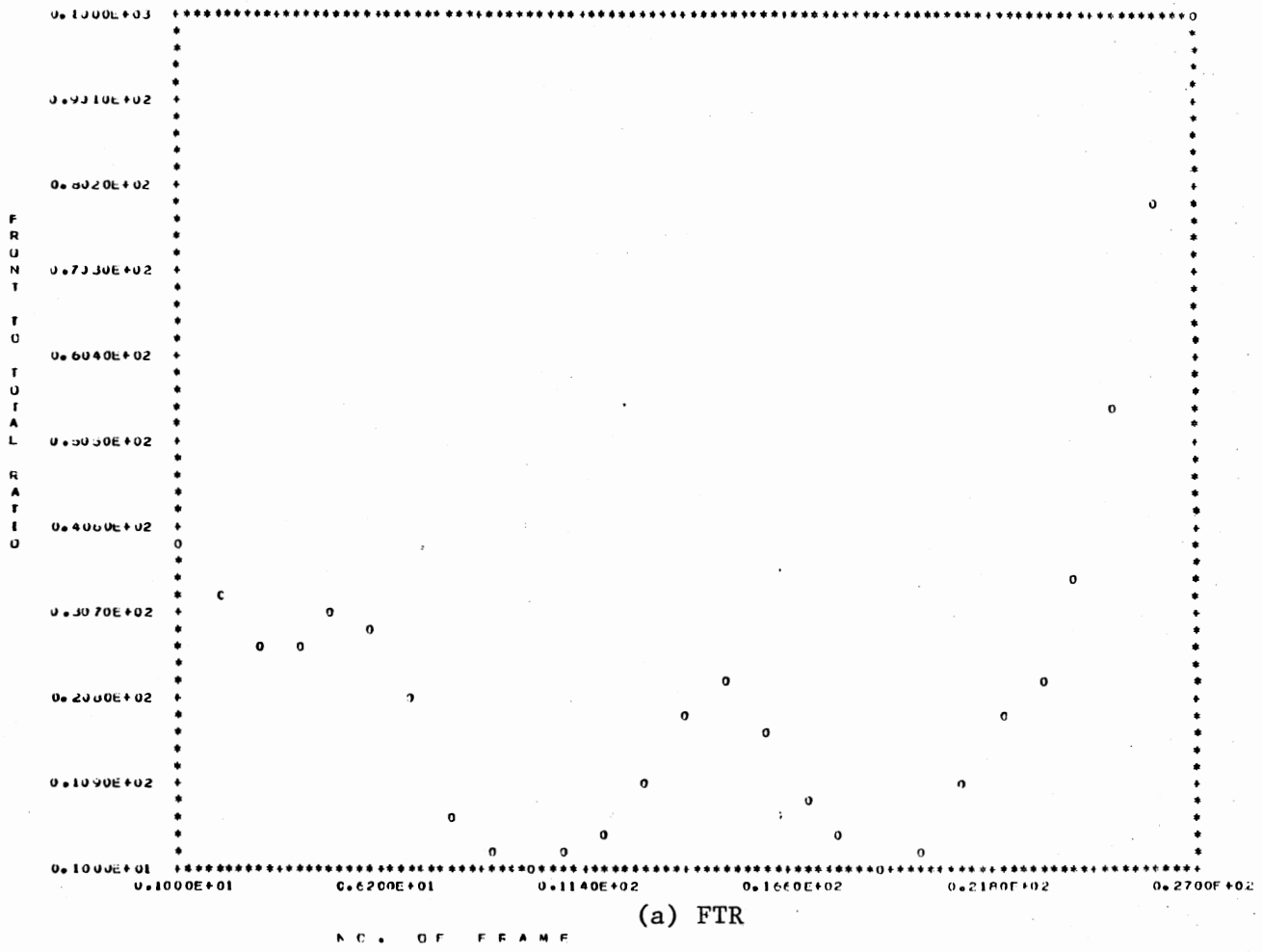
(b) CTR

Figure 92. (Continued)



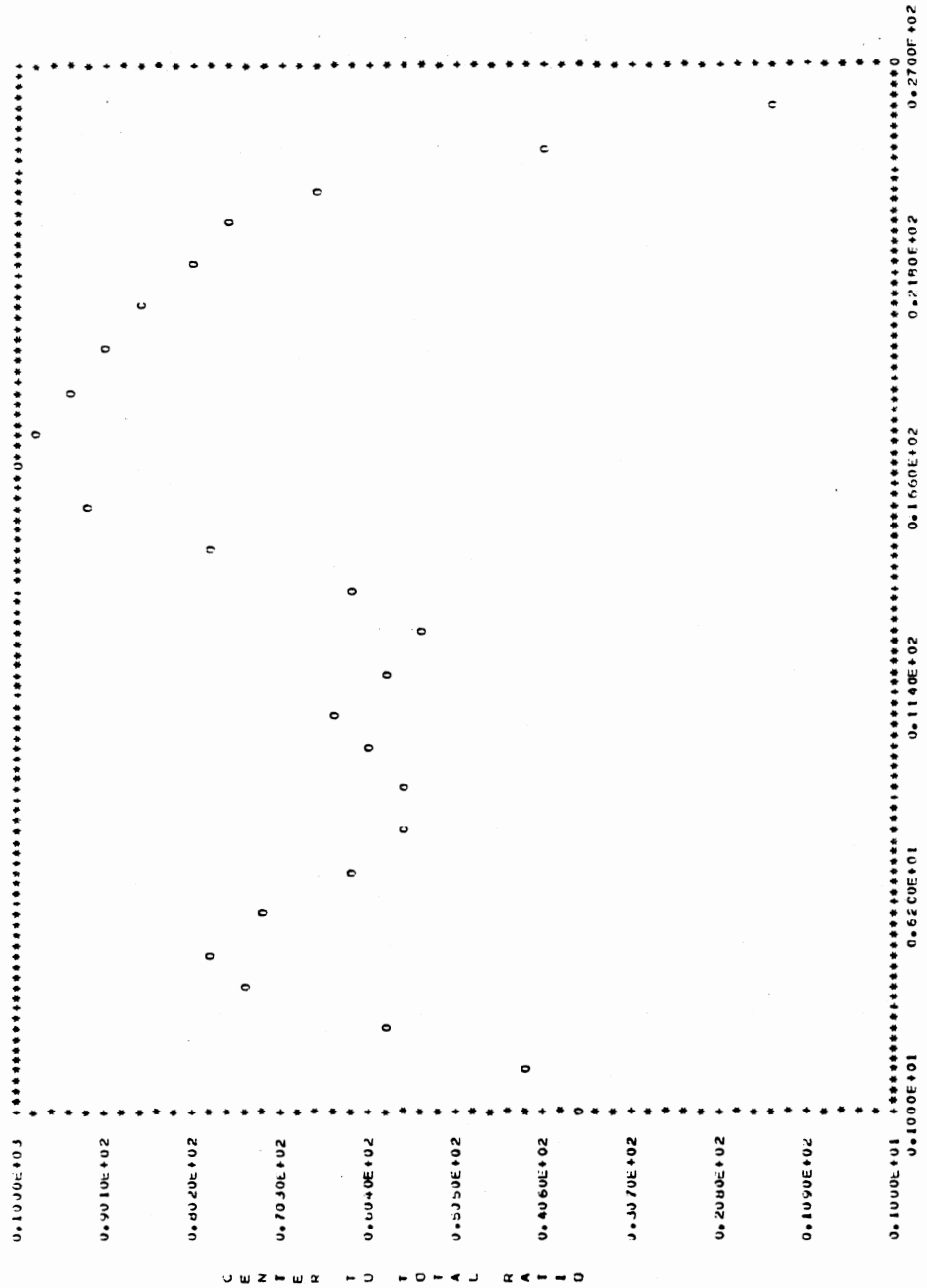
(c) BTR

Figure 92. (Continued)



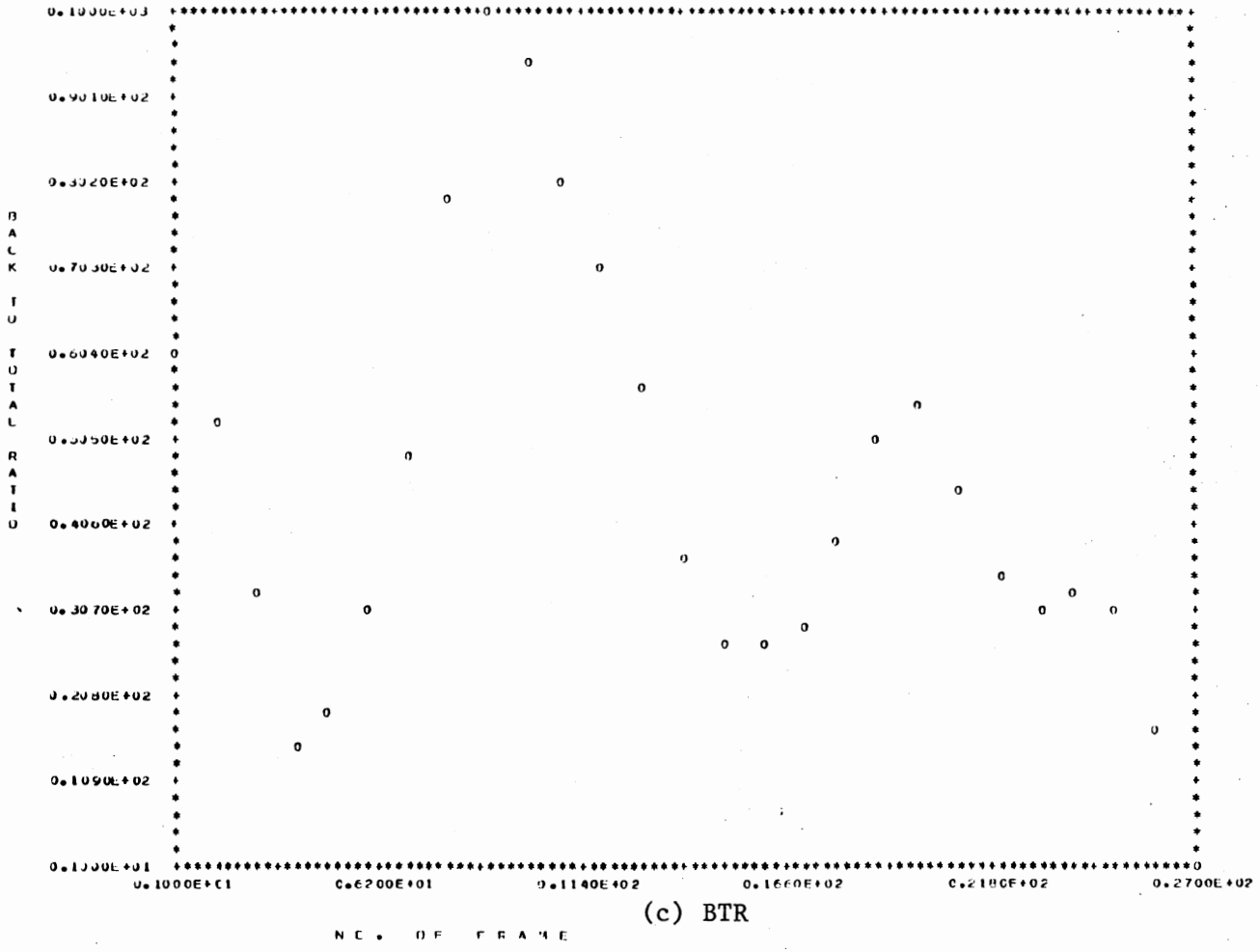
(a) FTR

Figure 93. Smoothed and Quantized Feature Parameter for Digit Nine, i.e. /tIsρðh/ Spoken in Arabic



(b) CTR

Figure 93. (Continued)



(c) BTR

Figure 93. (Continued)

schemes. Furthermore, the SFBR parameter and the BTR parameter are used to separate turbulence noise segments from back vowel segments and nasals segments from vowel segments respectively. The classification of each segment is stored, so that it can be utilized in the third stage, for actual digit recognition.

The second path in Figure 28 is implemented as before. Table XVIII gives a summary of the beginning, middle, and ending sound classification of digits spoken in Arabic. This table is derived from Table IV. The decision algorithm is based upon the uniqueness aspects of the classifications of a particular digit. This is illustrated by Table XVIII. For example, the digits zero, one, two and four have unique classification for beginning and ending sounds. These four digits are therefore decided on the beginning and ending sounds. The digits seven and eight can be separated from the remaining digits using the non-vowel uniqueness aspects in the middle region. The remaining digits thru, five, six and seven have the same uniqueness aspects at front and end region, but the middle region is different. The flow chart in Figure 94 gives this procedure for recognizing the digits spoken in Arabic.

The above algorithm was tested for five male speakers whose native tongue is Arabic. The accuracy rate was about 70 percent. The speakers have different accents, which accounts for the inconsistency of the peak ratios among different speakers. The results were much better when the spoken digits were intact, and the accuracy ratio was about 95 percent. By this method, the digits six and nine have the major problem. The reason is due to the fact that different speakers tend to stress different phonemes in the spoken digit. This fact affects

TABLE XVIII

SEQUENCE OF SOUND CLASS REGIONS OF DIGITS SPOKEN IN ARABIC

Digit	RMS and BTR Decision		
	Beginning	Middle	End
/sefr/	NV	FV/NV	VL
/ωâhid/	VL	BV/FV	NV
/iθnân/	FV	VL/BV	VL
/θaℓâθəh/	NV	VL/BV/NV	MV
/arbaρəh/	MV	VL/MV	MV
/khʌmsəh/	NV	MV/VL/NV	MV
/sIttəh/	NV	FV/NV	MV
/sʌbρəh/	NV	MV/VL	MV
/θamânyəh/	NV	MV/BV	MV
/tIsρəh/	NV	VL/BV/VL	MV

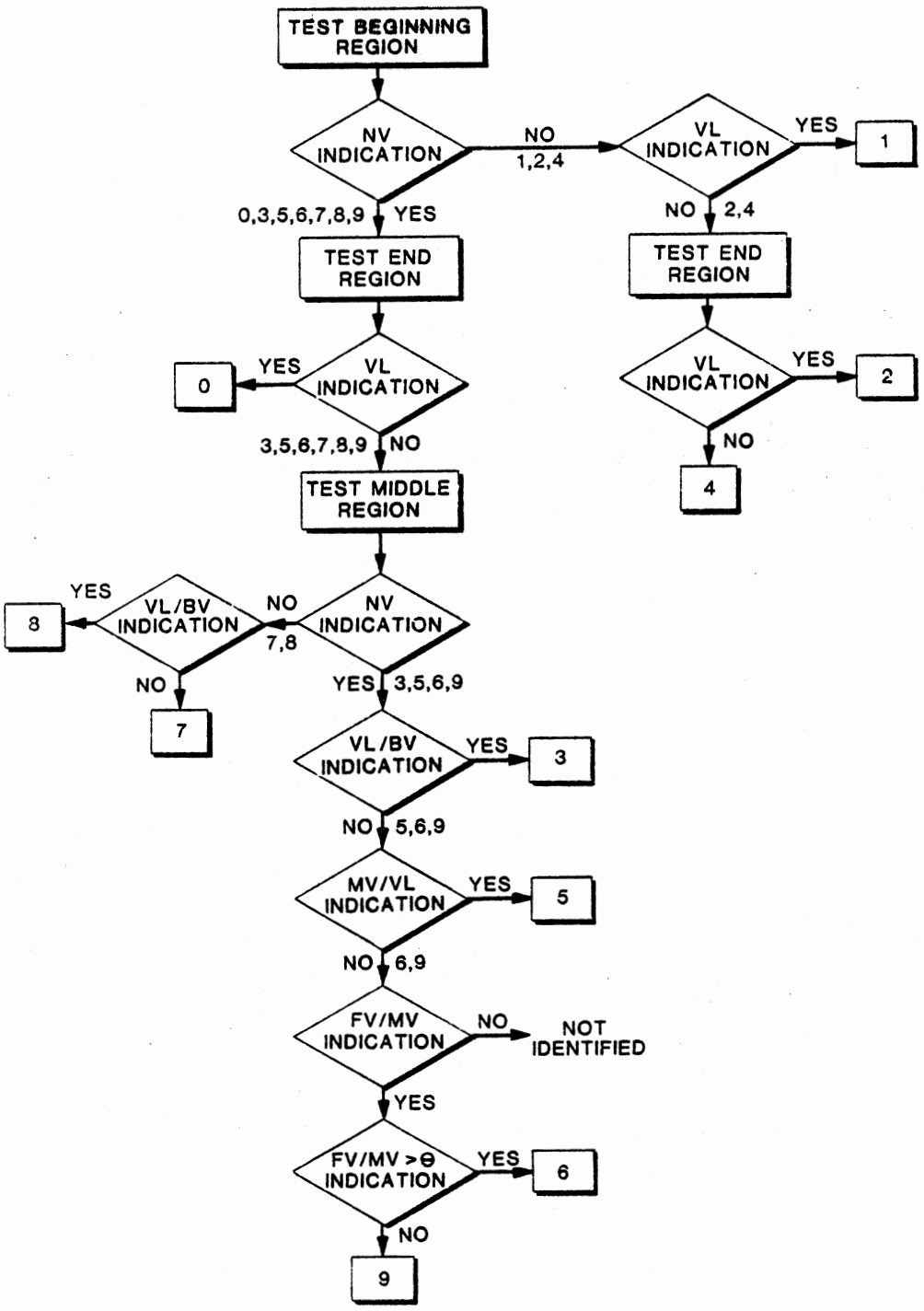


Figure 94. Decision Tree for Digit Identification for Arabic Language

the value range of FV-MV indication for different speakers. It is interesting to point out that if the spoken Arabic digit has a long vowel followed by a consonant (VVC) then that vowel is always stressed. It is pronounced louder than the other vowels; otherwise the first vowel of the digit is stressed. This can be seen from Table IV, which shows digit nine, /tIsɒðh/, have VCCV, where as digit six have either VccV or VCV, depending on how the /t/ is stressed. In addition, these two digits have almost similar characteristics in all the three regions. In fact the digit six, /sIttðh/, indicates that it has VCCV in the center because there is double /t/, i.e. the duration of the /t/ is doubled. Even though the second recognition idea is more reliable, the recognition result is based on the two paths for robustness. If the two paths give different results, or that no decision has been made, the pattern recognition procedure will be used. This is discussed next.

Pattern Recognition Scheme

In order to ascertain that the recognition scheme is robust, the cross-correlation coefficients are used in a pattern recognition scheme. The correlation coefficients is computed for the spoken digit using Equation (43). The flow chart of Figure 60 shows that a standard RMS and BTR pattern of the digits zero to nine are first selected using emperical rule, then stored. The RMS pattern correlation coefficients of the unknown digit is computed with maximum number of frames, then compared with the standard pattern. The correlation coefficient threshold is chosen emperically. If no recognition results, then the unknown digit is correlated with the

patterns of the remaining digit. The digit which has the highest correlation is selected. The same procedure is followed using the BTR pattern. The BTR pattern of the spoken digit is matched by the cross-correlation process. The RMS energy and BTR correlations for digits spoken in Arabic are shown in Tables XIX-XXIV. The cross-correlation process involves selecting the best match by locating the reference pattern which gives the greatest correlation coefficients with the pattern to be recognized. This system utilizes no linguistic information, but procedures of acoustic characteristics impeded in the RMS energy and the BTR parameter as discussed in the previous chapter.

The above scheme was applied for sample digits. The results, as shown in Tables XXV and XXVI, were good. However, before the recognition rate can be given, more digit patterns have to be listed. The computer programs are given in detailed form in Appendix B. This completes the discussion on the digit recognition scheme for digits spoken in Arabic.

TABLE XIX

RMS ENERGY CORRELATION TABLE FOR DIGITS SPOKEN IN ARABIC
(MAXIMUM NUMBER OF FRAMES USED)

SET NUMBER : 1

RMS CORRELATION

ZERO	ONE	TWO	THREE	FOUR	FIVE	SIX	SEVEN	EIGHT	NINE
1.000	-0.304	-0.399	-0.385	-0.276	-0.213	-0.286	-0.015	-0.329	-0.291
-0.304	1.000	-0.089	-0.258	0.421	-0.226	0.132	0.721	0.183	0.927
-0.399	-0.089	1.000	0.305	-0.253	0.306	-0.231	-0.116	-0.305	-0.115
-0.385	-0.258	0.305	1.000	0.251	0.425	-0.145	-0.394	0.194	-0.294
-0.276	0.421	-0.253	0.251	1.000	0.153	0.109	0.267	0.329	0.399
-0.213	-0.226	-0.306	0.425	0.153	1.000	0.588	-0.135	0.710	-0.284
-0.286	0.132	-0.231	-0.145	0.109	0.588	1.000	0.297	0.795	0.043
-0.015	0.721	-0.116	-0.394	0.267	-0.135	0.297	1.000	0.418	0.725
-0.329	0.183	-0.305	0.194	0.329	0.710	0.795	0.418	1.000	0.121
-0.291	0.927	-0.115	-0.294	0.399	-0.284	0.043	0.725	0.121	1.000

TABLE XX

RMS ENERGY CORRELATION TABLE FOR DIGITS SPOKEN IN ARABIC
(MAXIMUM NUMBER OF FRAMES USED)

SET NUMBER : 2

RMS CORRELATION

ZERO	ONE	TWO	THREE	FOUR	FIVE	SIX	SEVEN	EIGHT	NINE
1.000	-0.137	-0.236	-0.223	0.355	0.007	0.244	0.008	0.110	0.615
-0.137	1.000	0.845	-0.144	-0.339	-0.214	0.260	0.442	-0.050	-0.367
-0.236	0.845	1.000	0.053	-0.255	-0.261	0.023	0.364	0.127	-0.498
-0.223	-0.144	0.053	1.000	0.598	0.740	-0.291	0.610	0.245	-0.066
0.355	-0.339	-0.255	0.598	1.000	0.855	-0.305	0.464	0.634	0.123
0.007	-0.214	-0.261	0.740	0.855	1.000	-0.291	0.546	0.632	-0.017
0.244	0.260	0.023	-0.291	-0.305	0.291	1.000	0.166	-0.229	0.503
0.008	0.442	0.364	0.610	0.464	0.546	0.166	1.000	0.163	0.076
0.110	-0.050	0.127	0.245	0.634	0.632	-0.229	0.163	1.000	0.037
0.615	-0.367	-0.498	-0.066	0.123	-0.017	0.503	0.076	0.037	1.000

TABLE XXI

RMS ENERGY CORRELATION TABLE FOR DIGITS SPOKEN IN ARABIC
(MAXIMUM NUMBER OF FRAMES USED)

SET NUMBER : 3

RMS CORRELATION

ZERO	ONE	TWO	THREE	FOUR	FIVE	SIX	SEVEN	EIGHT	NINE
1.000	-0.253	-0.166	-0.234	0.137	-0.150	-0.132	-0.419	0.033	-0.440
-0.253	1.000	0.104	0.795	-0.187	0.843	-0.156	0.103	-0.068	0.501
-0.166	0.104	1.000	0.149	0.817	-0.018	-0.189	0.139	0.449	0.197
-0.234	0.795	0.149	1.000	-0.104	0.928	-0.290	-0.040	0.103	0.202
0.137	-0.187	0.817	-0.104	1.000	-0.253	-0.124	0.109	0.213	-0.009
-0.150	0.843	-0.018	0.928	-0.253	1.000	-0.124	-0.064	0.050	0.202
-0.132	-0.156	-0.189	-0.290	-0.124	-0.124	1.000	0.000	-0.038	0.182
-0.419	0.103	0.139	-0.040	0.109	-0.064	0.000	1.000	-0.367	0.773
0.033	-0.068	0.449	0.103	0.213	0.050	-0.038	-0.367	1.000	-0.348
-0.440	0.501	0.197	0.202	-0.009	0.202	0.182	0.773	-0.348	1.000

TABLE XXII

BTR CORRELATION TABLE FOR DIGITS SPOKEN IN ARABIC
(MAXIMUM NUMBER OF FRAMES USED)

SET NUMBER : 1

BTR CORRELATION

ZERO	ONE	TWO	THREE	FOUR	FIVE	SIX	SEVEN	EIGHT	NINE
1.000	-0.282	-0.123	-0.227	-0.294	-0.158	-0.235	-0.212	-0.138	0.084
-0.282	1.000	0.470	-0.409	0.317	-0.092	0.263	0.136	0.559	0.485
-0.123	0.470	1.000	-0.306	0.268	-0.213	0.287	0.155	0.130	0.391
-0.227	-0.409	-0.306	1.000	0.196	-0.185	-0.166	0.022	-0.009	-0.021
-0.294	0.317	0.268	0.196	1.000	0.582	0.567	0.472	0.475	0.163
-0.158	-0.092	-0.213	-0.185	0.582	1.000	0.429	0.324	0.085	-0.354
-0.235	0.263	0.287	-0.166	0.567	0.429	1.000	-0.050	0.617	-0.205
-0.212	0.136	0.155	0.022	0.472	0.324	-0.050	1.000	-0.002	0.335
-0.138	0.559	0.130	-0.009	0.475	0.085	0.617	-0.002	1.000	0.172
0.084	0.485	0.391	-0.021	0.163	-0.354	-0.205	0.335	0.172	1.000

TABLE XXIII

BTR CORRELATION TABLE FOR DIGITS SPOKEN IN ARABIC
(MAXIMUM NUMBER OF FRAMES USED)

SET NUMBER : 2

BTR CORRELATION

ZERO	ONE	TWO	THREE	FOUR	FIVE	SIX	SEVEN	EIGHT	NINE
1.000	0.172	0.653	0.064	0.564	0.025	0.149	0.055	-0.106	0.148
0.172	1.000	0.190	0.073	0.323	0.109	0.526	0.347	0.237	0.294
0.653	0.190	1.000	0.034	0.627	-0.185	0.092	0.503	-0.201	0.407
0.064	0.073	0.034	1.000	0.171	0.530	0.044	0.549	0.621	0.352
0.564	0.323	0.627	0.171	1.000	0.133	0.223	0.528	-0.120	0.276
0.025	0.109	-0.185	0.530	0.133	1.000	-0.105	0.570	0.512	0.289
0.149	0.526	0.092	0.044	0.223	-0.105	1.000	-0.147	0.152	-0.052
0.055	0.347	0.503	0.549	0.528	0.570	-0.147	1.000	0.234	0.741
-0.106	0.237	-0.201	0.621	-0.120	0.512	0.152	0.234	1.000	0.143
0.148	0.294	0.407	0.352	0.276	0.289	-0.052	0.741	0.143	1.000

TABLE XXIV

BTR CORRELATION TABLE FOR DIGITS SPOKEN IN ARABIC
(MAXIMUM NUMBER OF FRAMES USED)

SET NUMBER : 3

BTR CORRELATION

ZERO	ONE	TWO	THREE	FOUR	FIVE	SIX	SEVEN	EIGHT	NINE
1.000	0.527	0.205	0.041	0.127	0.117	0.027	0.292	-0.128	0.283
0.527	1.000	-0.380	0.480	0.105	0.413	0.269	0.566	0.459	0.371
0.205	-0.380	1.000	-0.340	0.023	-0.132	-0.072	0.086	-0.520	0.057
0.041	0.480	-0.340	1.000	0.694	-0.104	0.097	-0.397	0.551	0.206
0.127	0.105	0.023	0.694	1.000	-0.002	0.512	0.128	0.256	0.227
0.117	0.413	-0.132	-0.104	-0.002	1.000	0.520	0.551	0.756	0.585
0.027	0.269	-0.072	0.097	0.512	0.520	1.000	0.750	0.597	0.157
0.292	0.566	0.086	-0.397	0.128	0.551	0.750	1.000	0.372	0.206
-0.128	0.459	-0.520	0.551	0.256	0.756	0.597	0.372	1.000	0.702
0.283	0.371	0.057	0.206	0.227	0.585	0.157	0.206	0.702	1.000

TABLE XXV

RMS ENERGY CROSS-CORRELATION TABLE FOR TWO SETS OF DIGITS SPOKEN IN ARABIC BY TWO DIFFERENT SPEAKERS

			RMS CORRELATION						
ZERO	ONE	TWO	THREE	FOUR	FIVE	SIX	SEVEN	EIGHT	NINE
0.929	0.394	-0.039	-0.360	0.243	-0.065	0.140	0.681	0.328	-0.444
0.394	0.965	0.720	-0.131	-0.107	0.340	0.005	0.399	0.470	-0.297
-0.039	0.720	0.861	0.007	-0.164	0.502	-0.026	0.196	0.250	-0.316
-0.360	-0.131	0.007	0.916	0.358	-0.144	-0.205	-0.012	0.626	0.810
0.243	-0.107	-0.164	0.358	0.886	-0.020	0.143	0.533	0.225	0.369
-0.065	0.340	0.502	-0.144	-0.020	0.820	-0.094	0.159	-0.119	-0.130
0.140	0.005	-0.026	-0.205	0.143	-0.094	0.967	0.413	-0.525	-0.385
0.681	0.399	0.196	-0.012	0.533	0.159	0.413	0.575	0.580	-0.108
0.328	0.470	0.250	0.626	0.225	-0.119	-0.525	0.580	0.795	0.735
-0.444	-0.297	-0.316	0.810	0.369	-0.130	-0.385	-0.108	0.735	0.911

TABLE XXVI

BTR CROSS-CORRELATION TABLE FOR THE SAME DIGITS
USED IN TABLE XXV

BTR CORRELATION									
ZERO	ONE	TWO	THREE	FOUR	FIVE	SIX	SEVEN	EIGHT	NINE
0.567	0.009	0.708	0.141	0.694	-0.163	0.359	0.112	-0.350	0.297
0.009	0.721	0.389	-0.344	0.114	-0.143	0.269	0.508	0.080	0.751
0.708	0.389	0.798	0.026	0.340	-0.150	0.097	0.264	-0.165	0.292
0.141	-0.344	0.026	0.688	-0.134	-0.077	-0.235	-0.399	0.330	0.001
0.694	0.114	0.340	-0.134	0.821	0.104	-0.051	-0.038	-0.060	0.212
-0.163	-0.143	-0.150	-0.077	0.104	0.556	0.105	0.141	-0.193	-0.109
0.359	0.269	0.097	-0.235	-0.051	0.105	0.454	-0.132	-0.079	0.031
0.112	0.508	0.264	-0.399	-0.038	0.141	-0.132	0.493	0.064	0.407
-0.350	0.080	-0.165	0.330	-0.060	-0.193	-0.079	0.064	0.443	-0.047
0.297	0.751	0.292	0.001	0.212	-0.109	0.031	0.407	-0.047	0.554

CHAPTER VI

SUMMARY AND SUGGESTION FOR FURTHER STUDY

Summary

In this thesis a Robust Phonetic Digit Recognition System for digits spoken in American English and Arabic has been developed. A three-stage recognition scheme has been developed based on the RMS energy, the equivalent area functions and pattern analysis. The first stage uses rectangular window for segmenting the spoken digit into 128 data points per frame. The RMS energy is computed per frame, followed by a smoothing routine, to remove any undesirable ripples without loosing any useful features. The smoothed RMS energy is then quantized to a maximum level of 100. This form of normalization made the RMS peaks to be more consistent with different speakers uttering the same digit than using a fixed mean and varying amplitudes. The later type of data scaling is avoided, because the amplitude value depends considerably on the number of data points which depends on the end point detection algorithm. The largest two peaks are computed from the smoothed RMS energy and their ratio is computed, which is assumed to be less than unity. An emperical threshold value is established from seven different speakers for American English digits and five different speakers for Arabic digits.

In the second stage, the data is first Hamming windowed, then a first order low-frequency pre-emphasis is used with a gain of (-1.0). A 14th order linear prediction analysis model is applied to the windowed signal. Useful measures are extracted from the smoothed and quantized RMS energy functions for effectively distinguishing vowel intervals from non-vowel intervals. The vowel, vowel-like and non-vowel intervals detection is based on the RMS dip-classification. In addition, the parameters FTR, CTR, and BTR are computed from the equivalent vocal-tract area via LPA. These parameters have some useful measures, acoustical and phonemic features. However, the BTR is not only found to be reliable, but sufficient for distinguishing vowel, and non-vowel intervals than the CTR. An 'OR' decision control is used based on the RMS dip-classification and BTR parameters for correctly identifying vowel, vowel-like, and non-vowel segments. Two different digit recognition trees are then used, based on vowel, vowel-like and non-vowel decision scheme for each language.

In the first stage, the digit recognition rate is about 60 percent (55 percent) for digits spoken in American English (Arabic), whereas in the second stage the digit recognition rate is about 75 percent (70 percent) for digits spoken in American English (Arabic). However, a digit recognition rate of 95 percent is obtained with American English, speaking in a sound proof chamber. The low recognition accuracy for digits spoken in both languages is due to that all the speakers have different accents. In addition most international speakers tend to stress a vowel or a consonant with a spoken digit. Another important reason is that some of the spoken digits are not intact. A lower recognition rate is obtained for digits spoken in

Arabic, because some Arabic digits have silence interval in the middle region of spoken digits. Wrong boundary detection resulted whenever the digit is not intact.

In the final stage, the digit recognition algorithm, based on the correlation coefficients of the RMS and BTR parameters, is used if no digit is identified by both procedures. A 97 percent recognition rate is obtained for digits spoken in both languages. The number of speakers used in this research is rather limited, indicating that the recognition scheme cannot be completely classified as speaker independent. The cross-correlation scheme improved the recognition rate accuracy considerably. In addition, it furnished a better understanding of the feature uniqueness among the digits. For example, it has been found that the digit seven is not heavily correlated with the same digit in another set. This is because international speakers tend to either emphasize the semi-vowel /v/ or mis-pronounce this vowel.

Similar recognition rate is achieved for digits spoken in Arabic. The RMS and BTR patterns are used so that, the spoken digit will always be identified, either by the BTR or by both parameters.

Suggestions for Further Research

Some extensions to the present research are proposed below.

The parameters derived from the equivalent area functions discussed in Chapter III need to be further studied. That is parameter normalization, where no prior knowledge about the input speakers is required, needs to be further investigated. Especially the vocal-tract

length and area functions may be first estimated from the acoustic speech waveform, and then the area function is normalized to an acoustic tube of the same shape having a certain reference length.

The two dimensional plots of BTR versus CTR may give a unique pattern for a given spoken digit after parameter normalization. Furthermore, three dimensional plots of BTR, CTR and FTR may give additional insight into the phonemic features of spoken words. Definite pattern uniqueness for the digits zero to nine may result after parameter normalization and boundary limit modification. An overlap limit for the front, central and back cavity may modify these parameters to be very effective for discriminating between front, middle, and back parameters.

In this study, spectrum analysis is avoided, in order to have an efficient recognition system. However, studying consonant features using selective Linear Prediction Analysis may improve detection of nasals, voiced fricatives and plosives. But normalization of the frequency scale is of major importance. This may lead to speaker verification system. Additional research in terms of the radius r_o at the constriction and its distance from the glottis may allow new results in word recognition.

This research has opened up a new area in speech and speaker recognition in Arabic. Obviously, this area is wide open.

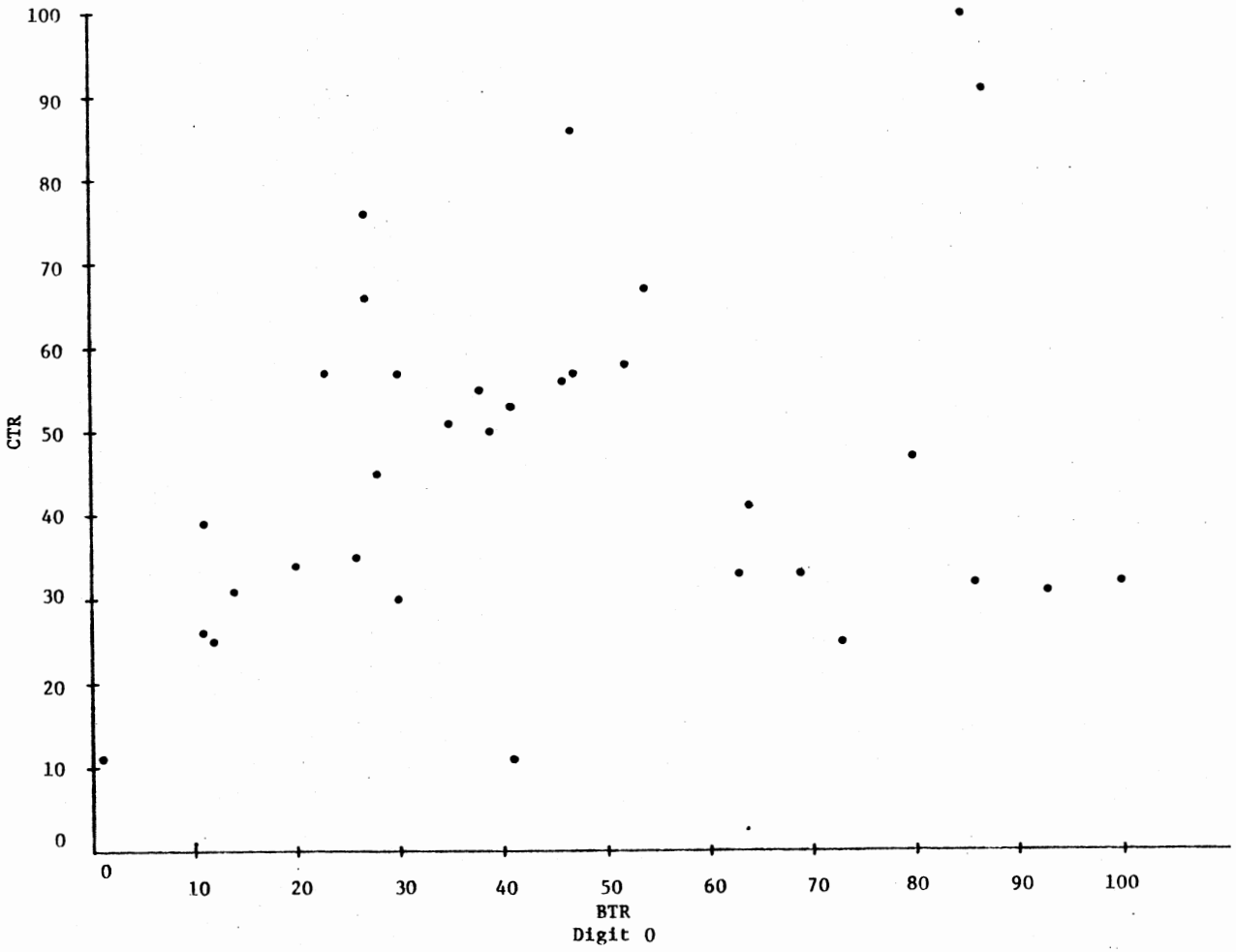


Figure 95. Plots of CTR vs BTR for Digit Zero

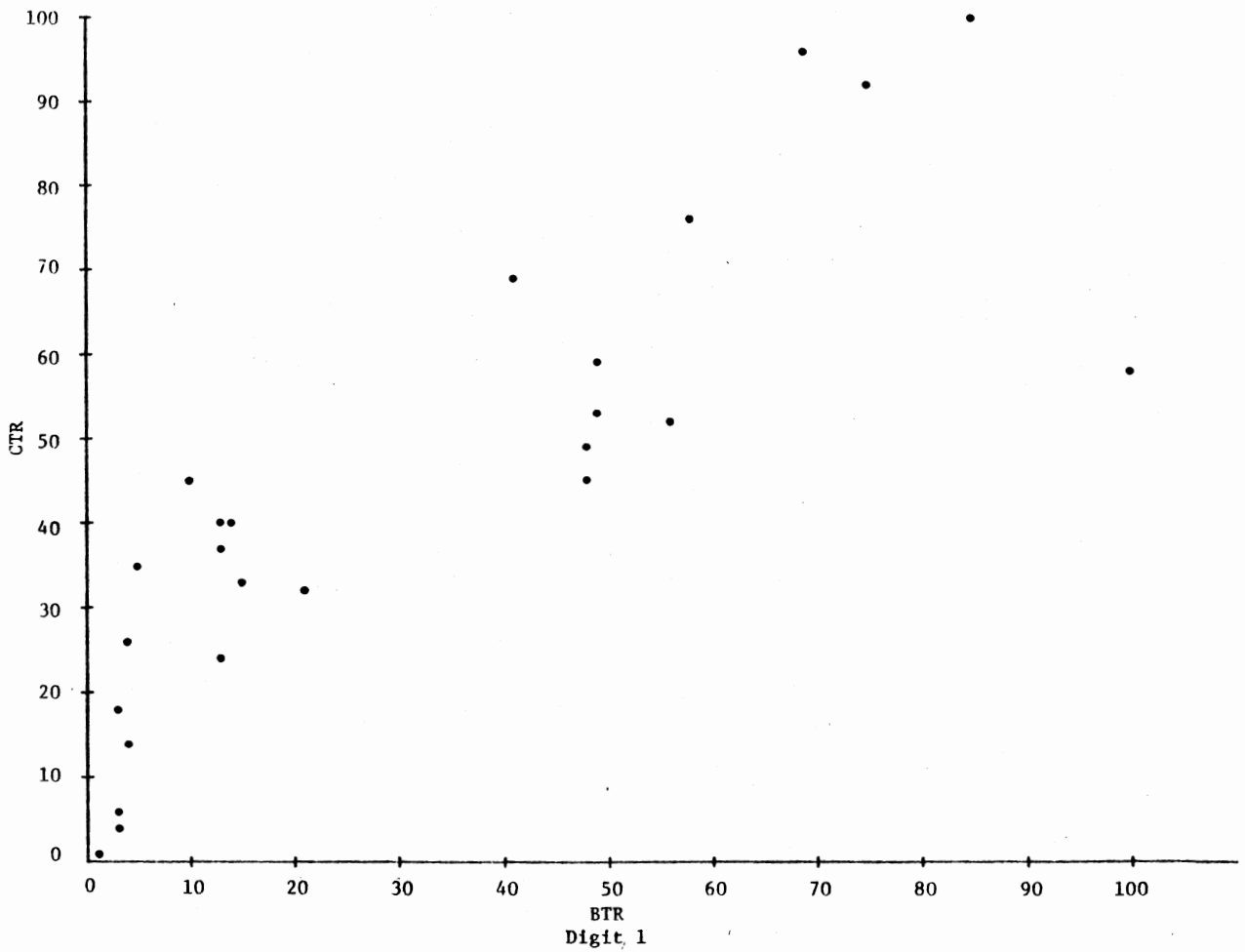


Figure 96. Plots of CTR vs BTR for Digit One

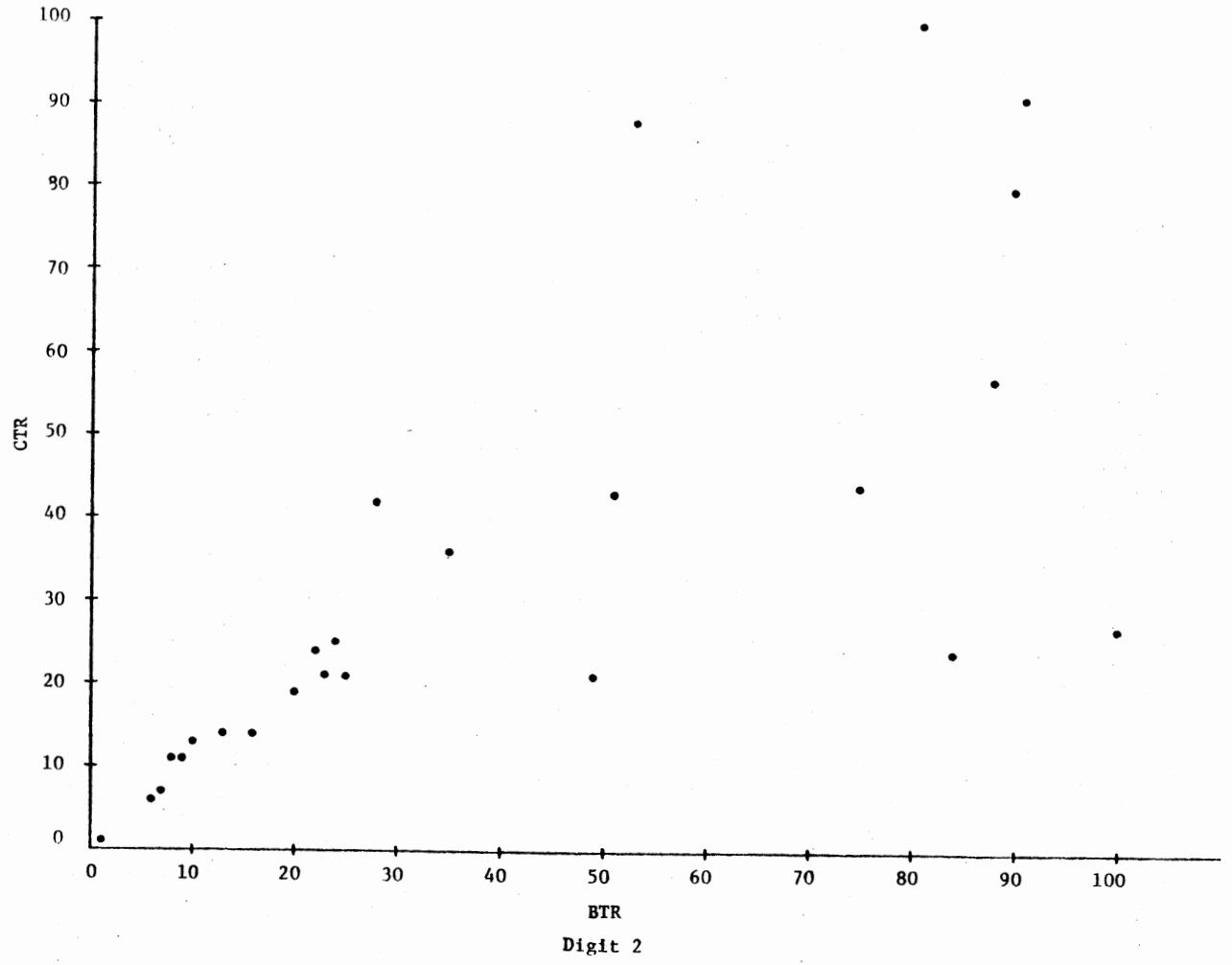


Figure 97. Plots of CTR vs BTR for Digit Two

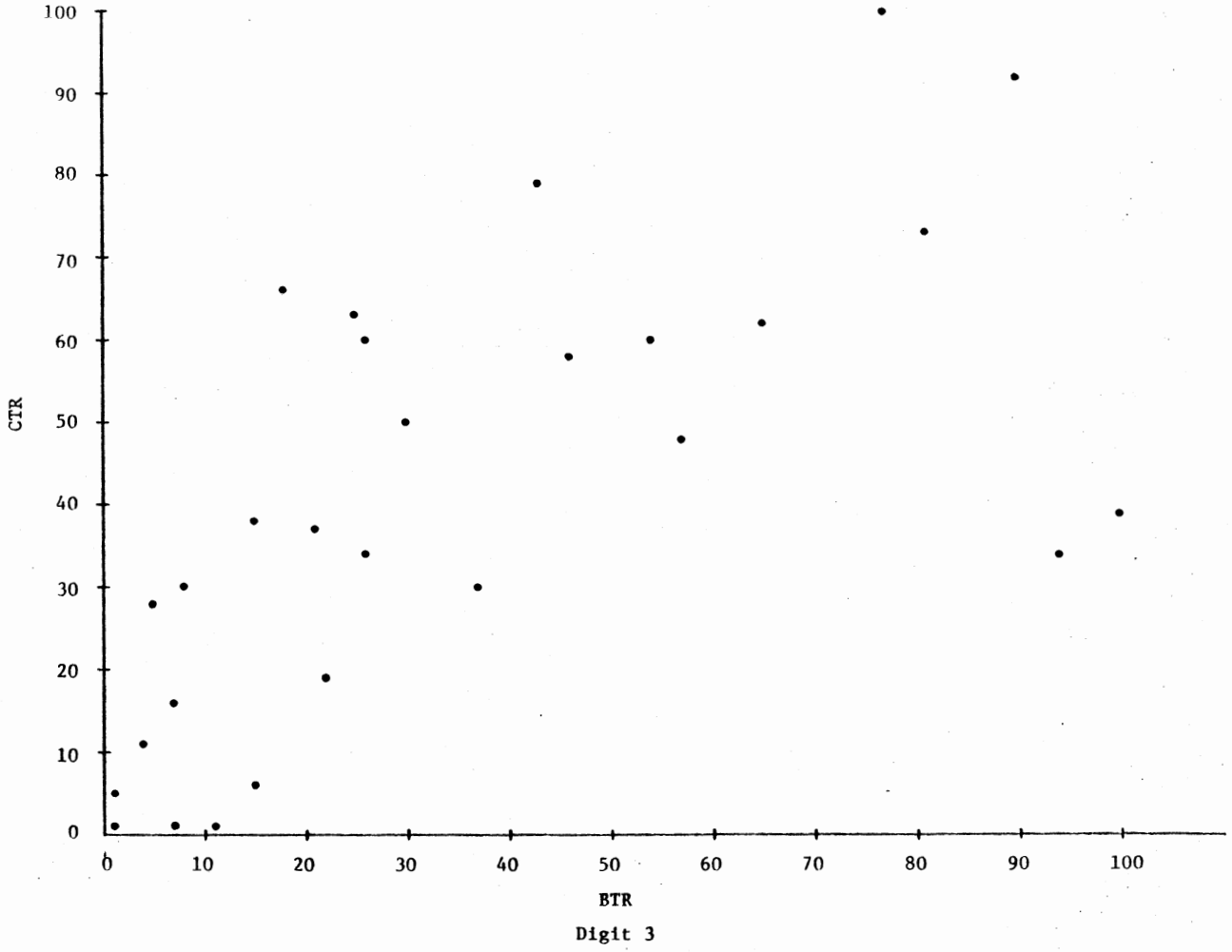


Figure 98. Plots of CTR vs BTR for Digit Three

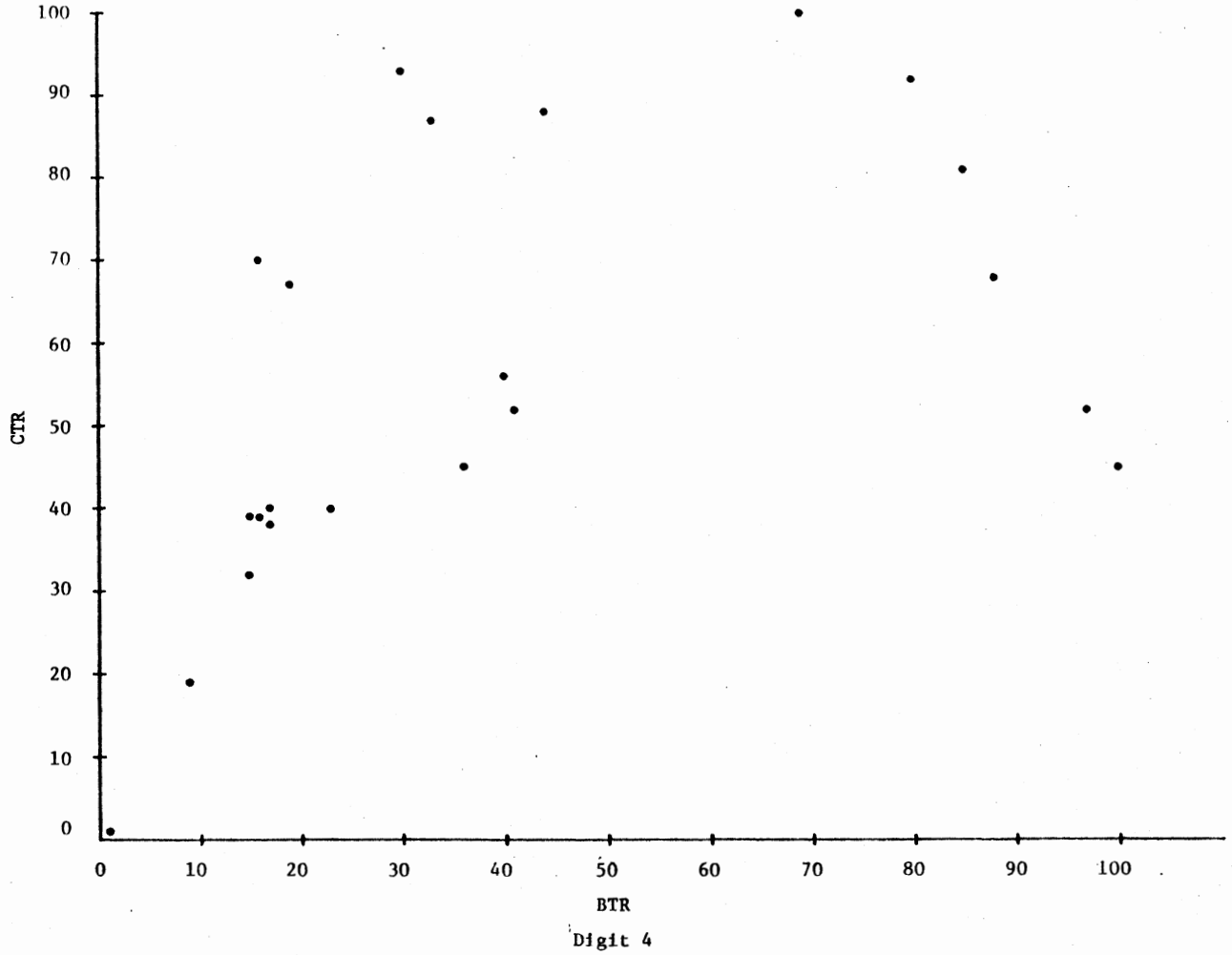


Figure 99. Plots of CTR vs BTR for Digit Four

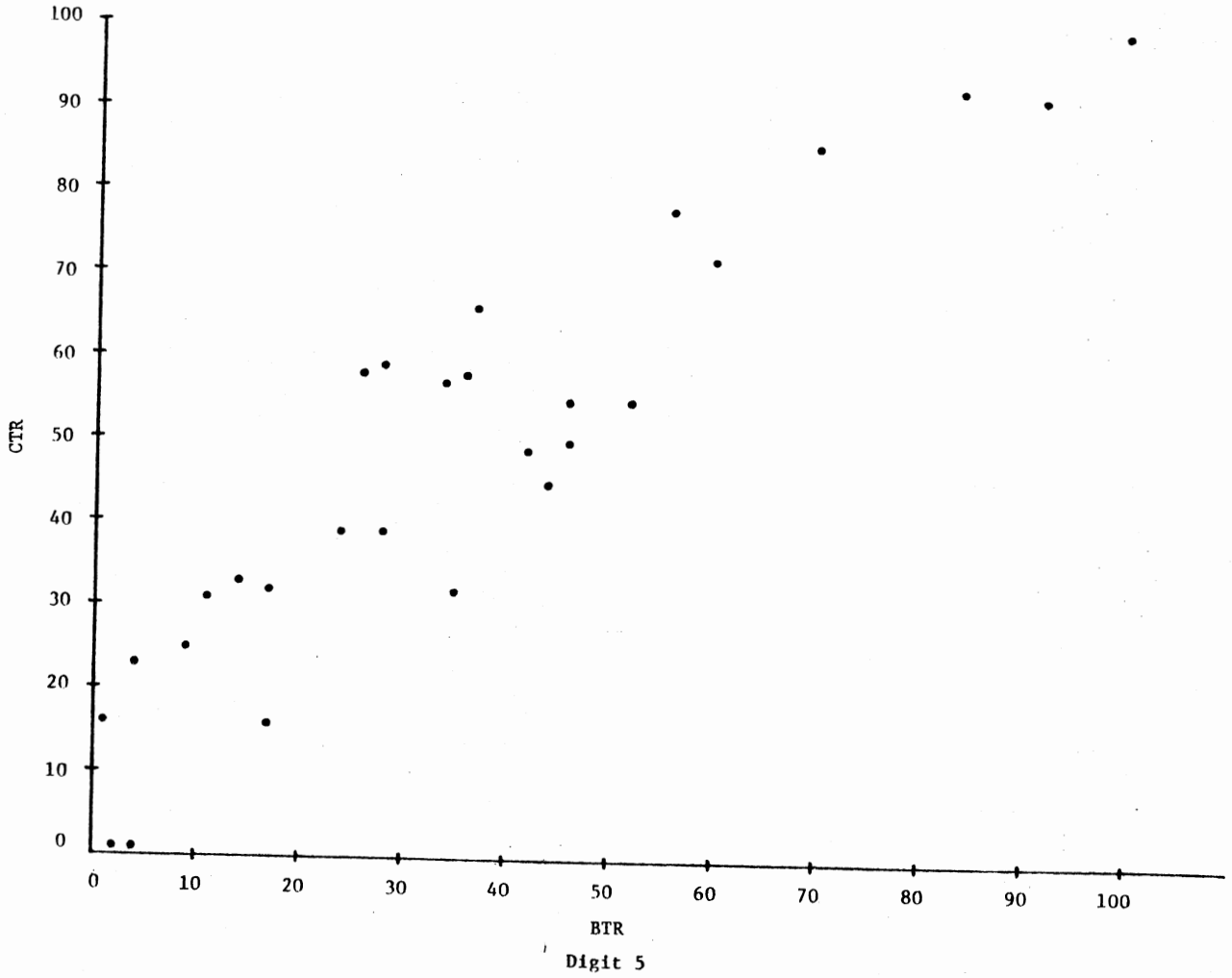


Figure 100. Plots of CTR vs BTR for Digit Five

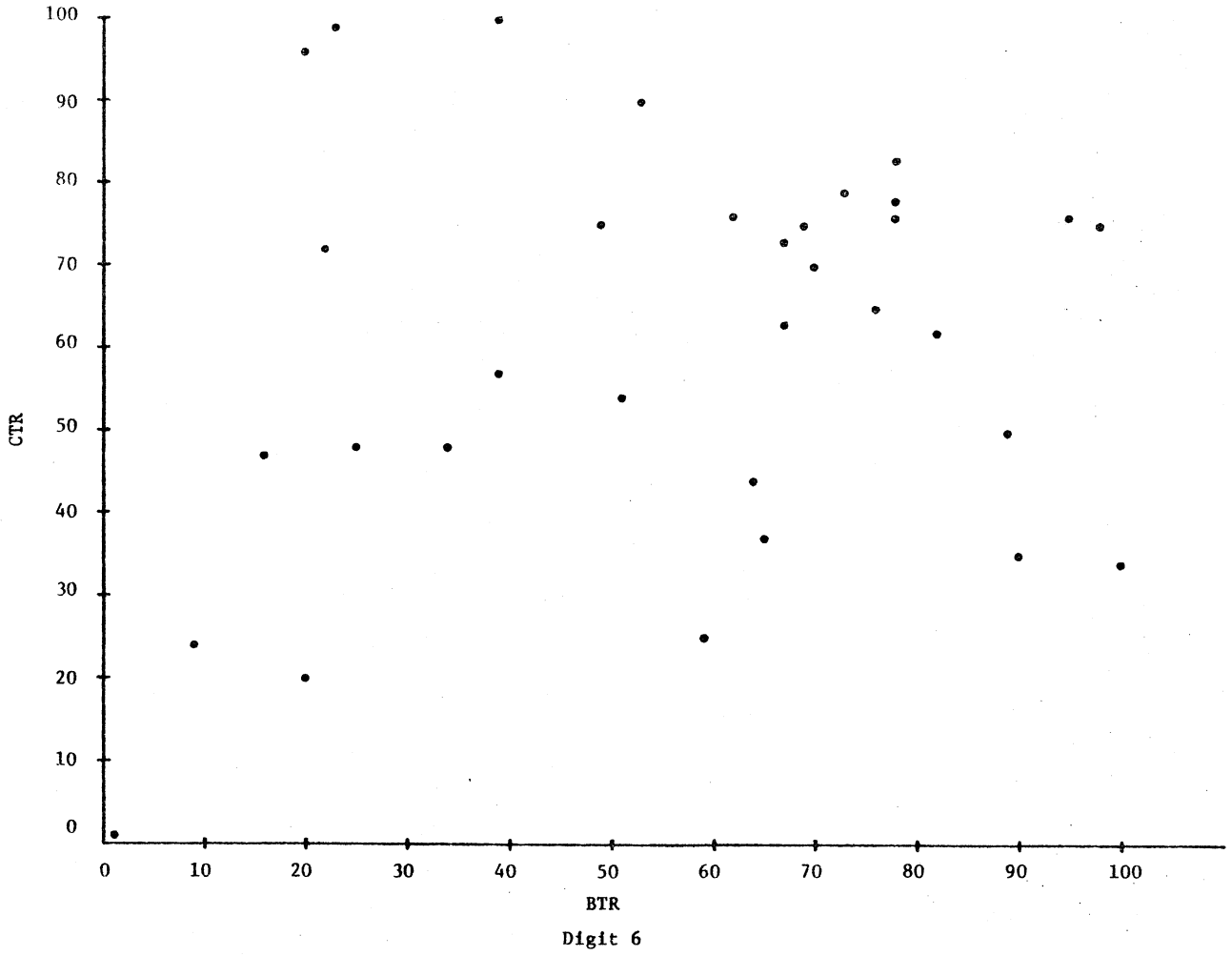


Figure 101. Plots of CTR vs BTR for Digit Six

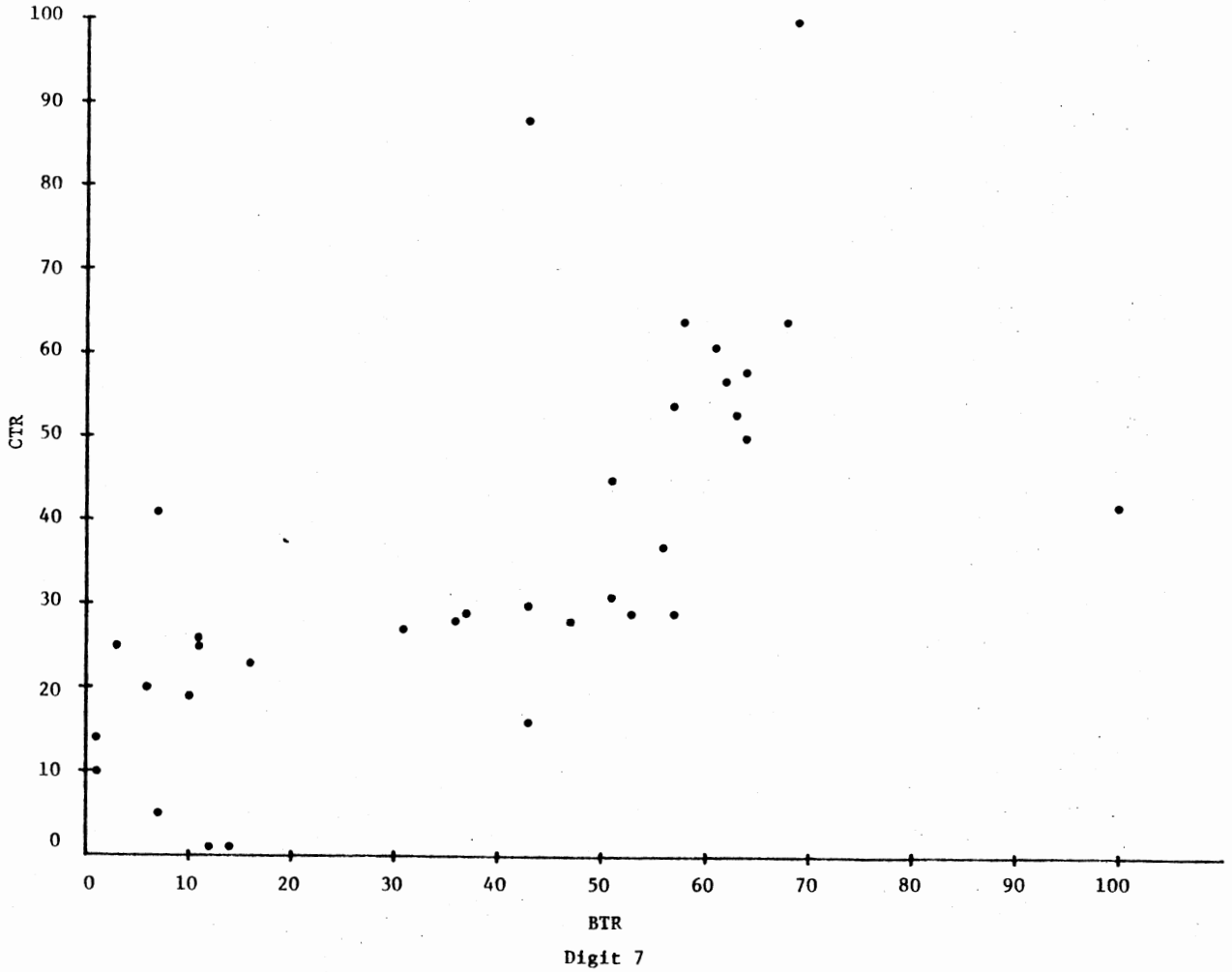


Figure 102. Plots of CTR vs BTR for Digit Seven

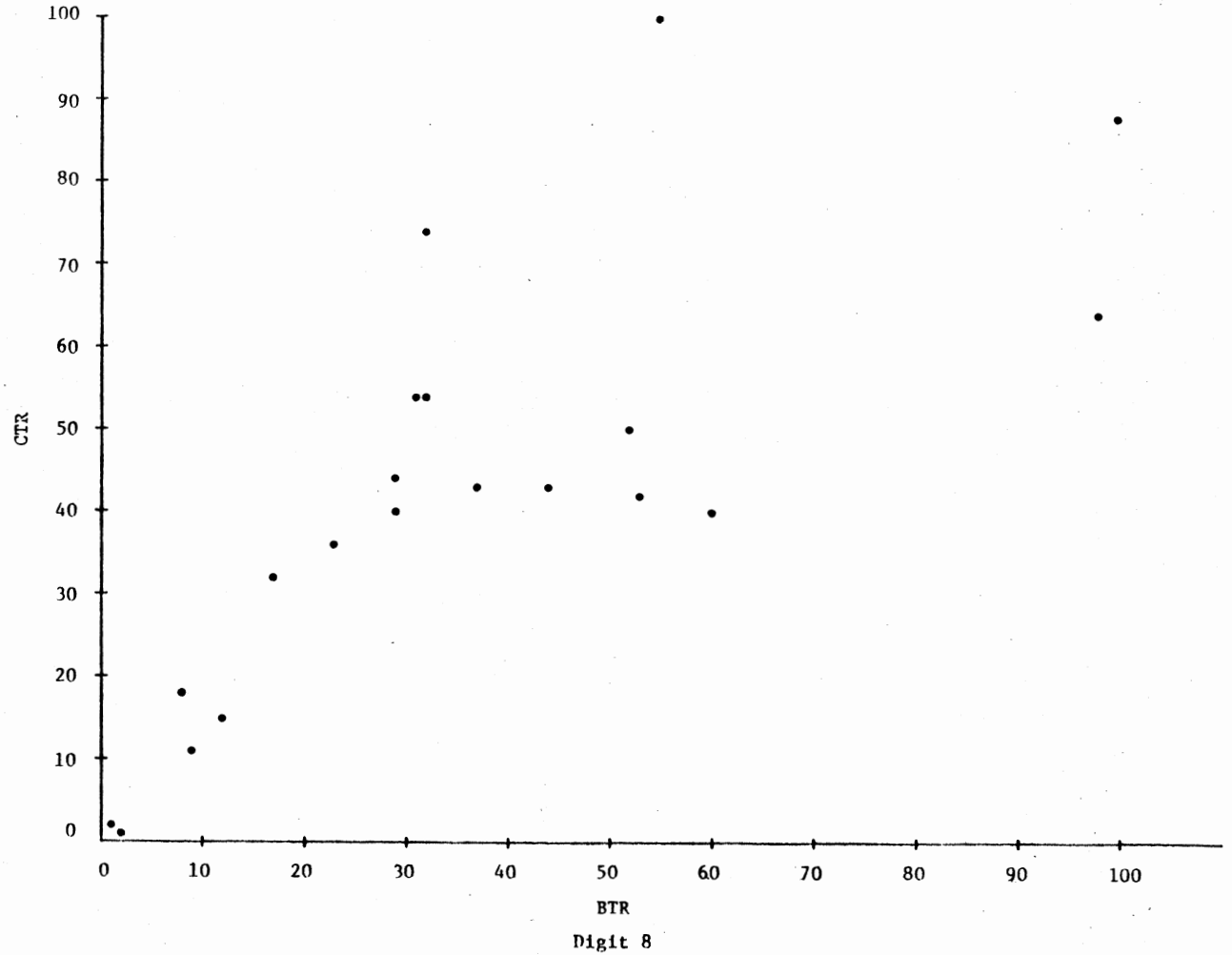


Figure 103. Plots of CTR vs BTR for Digit Eight

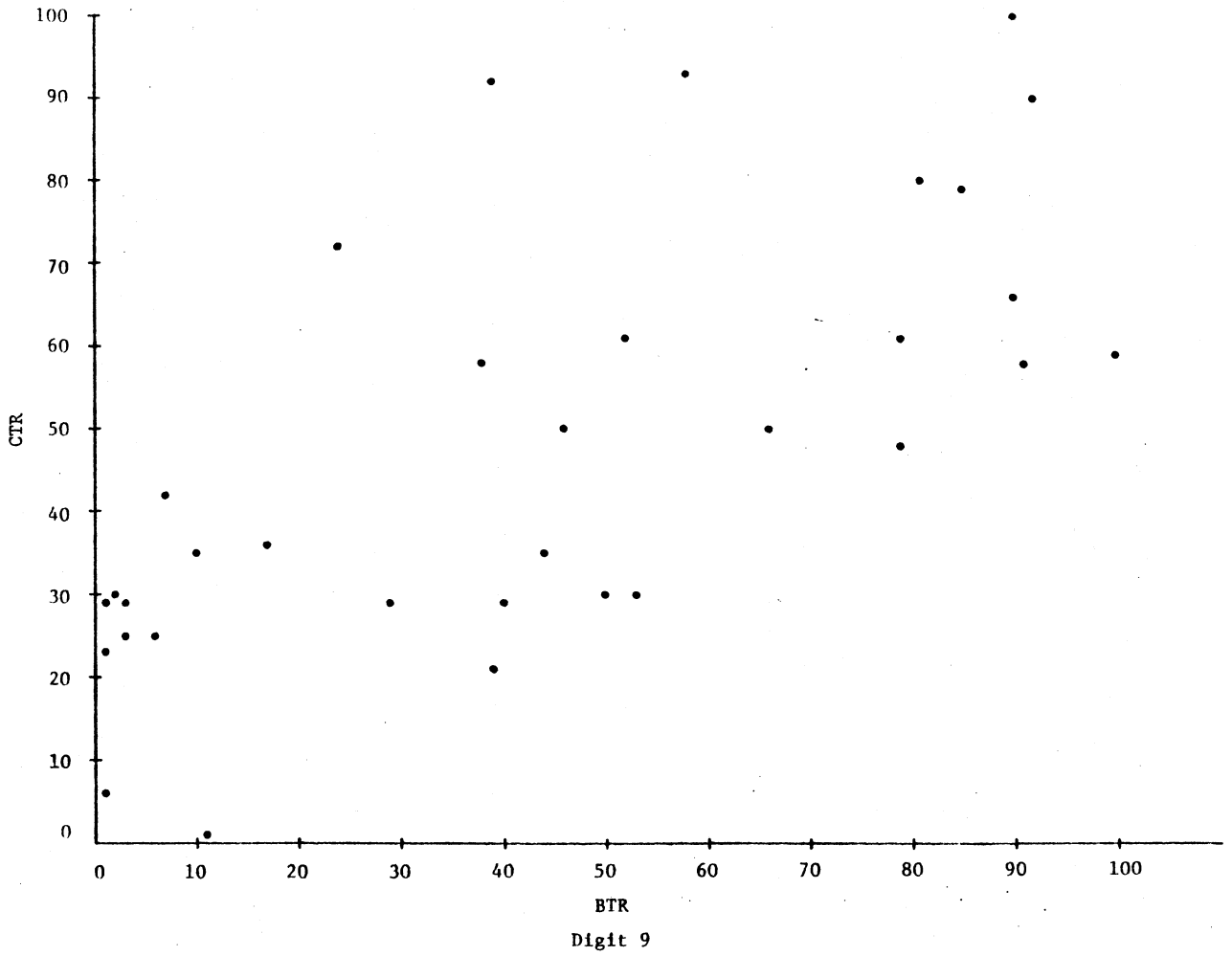


Figure 104. Plots of CTR vs BTR for Digit Nine

REFERENCES

[1] L. R. Rabiner and R. W. Schafer, "Digital Processing of Speech Signals," Signal Processing Series, R. V. Oppenheim (Series Ed.). Englewood Cliffs: Prentice Hall, Inc. 1978.

[2] D. R. Reddy, "Speech Recognition by Machines," Proceedings of the IEEE, Vol. 64, No. 4, 501-531, Apr. 1976.

[3] P. Denes and M. W. Mathews, "Spoken Digit Recognition Using Time-Frequency Pattern Matching," J. Acoust. Soc. Am., Vol. 32, 1450-1455, 1960.

[4] M. R. Sambur and L. R. Rabiner, "A Speaker-Independent Digit-Recognition System," Bell System Technical Journal, Vol. 54, No. 1, 81-102, Jan. 1975.

[5] P. W. Ross, "A Limited-Vocabulary Adaptive Speech-Recognition System," Journal of the Audio Engineering Society, Vol. 15, No. 4, 414-418, Oct. 1967.

[6] H. R. Dixon and H. F. Silverman, "A General Language-Operated Decision Implementation System (GLODIS), Its Application to Continuous Speech Segmentation," IEEE Trans. on Acous. Sp. Sig. Proc., Vol. ASSP-24, No. 2, 137-162, Apr. 1976.

[7] D. R. Reddy, "Segmentation of Speech Sounds," J. Acoust. Soc. Am., Vol. 40, No. 2, 307-312, March 1966.

[8] H. Kasuya and H. Wakita, "An Approach to Segmenting Speech into Vowel- and Nonvowel-like Intervals," IEEE Trans. on Acous. Sp. and Sig. Proc., Vol. ASSP 27 , No. 4, 319-327, Aug. 1979.

[9] L. R. Rabiner and M. R. Sambur, "An Algorithm for Determining the Endpoints of Isolated Utterances," Bell System Technical Journal, Vol. 54, No. 2, 297-315, Feb. 1975.

[10] W. A. Lea, Trends in Speech Recognition. Englewood Cliffs, New Jersey: Prentice Hall, Inc., 1980.

[11] J. L. Flanagan, Speech Analysis, Synthesis and Perception, Vol. 3. New York: Springer-Verlag, 1972.

[12] L. R. Rabiner and M. R. Sambur, "Some Preliminary Experiments in the Recognition of Aonnected Digits," IEEE Trans. on Acous. Sp. and Sig. Proc., Vol. ASSP 24, No. 2, 244-256, Apr. 1976.

[13] K. H. Davis, R. Biddulph and S. Balshek, "Automatic Recognition of Spoken Digits," J. Acoust. Soc. Am., Vol. 24, No. 6, 637-642, Nov. 1952.

[14] T. G. von Keller, "An On-Line Recognition System for Spoken Digits," J. Acoust. Soc. Am., Vol. 49, No. 4 (Part 2), 1288-2196, 1971.

[15] P. N. Sholtz and R. Bakis, "Spoken Digit Recognition Using Vowel-Consonants Segmentation," J. Acoust. Soc. Am., Vol. 34, No. 1, 1-5, Jan. 1962.

[16] A. Cohen and S. G. Nooteboom, Structure and Process in Speech Perception. New York: Spring-Verlag, 1975.

[17] J. D. Markel and A. H. Gray, Linear Prediction of Speech. New York: Springer-Verlag, 1966.

[18] F. D. Minifie, T. J. Hixon and F. Williams, Normal Aspects of Speech, Hearing, and Language. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1973.

[19] W. R. Zemlin, Speech and Hearing Science Anatomy and Physiology. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1968.

[20] C. W. Kanliner and R. West, Phonetics. New York: Harper and Brothers Publishers, 1960.

[21] G. Fant, Speech Communication, Speech Wave Processing and Transmission, Vol. 1. New York: John Wiley and Sons, 1975.

[22] C. H. Coker, "A Model of Articulatory Dynamics and Control," Proceedings of the IEEE, Vol. 64, No. 4, 452-459, Apr. 1976.

[23] C. J. Weinstein, S. S. McCandless, L. F. Mondshein and V. W. Zue, "A System for Acoustic-Phonetic-Analysis of Continuous Speech," IEEE Trans. on Acous. Sp. and Sig. Proc., Vol. ASSP-23, No. 1, 54-67, Feb. 1975.

[24] K. N. Stevens and A. S. House, "Development of a Quantitative Description of Vowel Articulation," J. Acoust. Soc. Am., Vol. 27, No. 3, 484-492, May 1955.

[25] R. Schwartz and J. Makhoul, "Where the Phones are: Dealing with Ambiguity in Acoustic-Phonetic Recognition," IEEE Trans. on Acous. Sp. and Sig. Proc., Vol. ASSP-23, No. 1, 50-53, Feb. 1975.

[26] I. Lehiste and G. E. Peterson, "Transitions, Glides, and Diphthongs," J. Acoust. Soc. Am., Vol. 33, No. 3, 268-277, Mar. 1961.

[27] J. Makhoul, "Linear Prediction: A Tutorial Review," Proceedings of the IEEE, Vol. 63, 561-580, Apr. 1975.

[28] O. Fujimura, "Analysis of Nasal Consonants," J. Acoust. Soc. Am., Vol. 34, No. 12, 1865-1875, Dec. 1962.

[29] B. S. Atal and S. L. Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave," J. Accost. Soc. Am., Vol. 50, No. 2 (Part 2), 637-655, Aug. 1971.

[30] H. Wakita, "Direct Estimation of the Vocal Tract Shape by Inverse Filtering of Acoustic Speech Waveforms," IEEE Trans. Audio and Electroacoustics, Vol. AU-21, No. 25, 417-427, Oct. 1973.

[31] R. F. Purton, "Speech Recognition Using Autocorrelation Analysis," IEEE Trans. Audio and Electroacoustics, Vol. AU-16, No. 2, 235-239, June 1968.

[32] L. R. Rabiner, M. R. Sambur and C. E. Schmidt, "Applications of a Nonlinear Smoothing Algorithm to Speech Processing," IEEE Trans. on Acous. Sp. and Sig. Proc., Vol. ASSP-23, No. 6, 552-557, Dec. 1975.

[33] W. J. Hess, "Time-Domain, Digital Segmentation of Connected Natural Speech," International Conference on Artificial Intelligence, 4th TBILIST, USSR, 491-498, 1975.

[34] G. Fant, "The Acoustics of Speech," Proc. Third Intern. Congr. Acoust., 188-201, 1959.

[35] H. K. Dunn, "The Calculation of Vowel Resonances and an Electrical Vocal Tract," J. Acoust. Soc. Am., Vol. 22, 740-753, 1950.

[36] B. L. Bowerman and R. T. O'Connell, Time Series and Forecasting. Dunbury Press, A Division of Wadsworth, Inc., 1979.

[37] H. Lane, "The Motor Theory of Speech Perception: A Critical Review," Psychological Review, Vol. 72, No. 4, 275-309, 1965.

[38] R. A. Fisher, "The Use of Multiple Measurements in Taxonomic Problems," Ann. Eugen., Vol. 7, 1979-188, 1936.

[39] H. Suzuki, H. Kasuya and K. Kido, "The Acoustic Parameters for Vowel Recognition Without Distinction of Speakers," Proceedings of the Conf. on Speech Communication and Processing, Cambridge, England, 92-96, Mar. 1967.

[40] B. S. Atal, "Linear Prediction of Speech--Recent Advances with Applications to Speech Analysis," Reprinted from: Speech Recognition. New York: Academic Press, Inc., 221-230, 1975.

[41] J. D. Markel and A. Gray,"On Autocorrelation Equations as Applied to Speech Analysis," IEEE Trans. on Audio and Electroacoustics, Vol. AU-21, No. 2, 69-79, Apr. 1973.

[42] F. L. Wightman and D. M. Green, "The Perception of Pitch," American Scientist, Vol. 62, No. 2, 208-215, Mar.-Apr. 1974.

[43] J. B. Thomas and R. J. Niederjohn, "Enhancement of Speech Intelligibility at High Noise Levels by Filtering and Clipping," Journal of the Audio Engineering Society, Vol. 16, No. 4, 412-415, Oct. 1968.

[44] M. J. Hunt, J. S. Bridle and J. N. Holmes, "Interactive Digital Inverse Filtering and Its Relation to Linear Prediction Methods," International Conference on Acoustics, Speech and Signal Processing, 78CH1285-6 ASSP, 15-18, 1978.

[45] C. J. Weinstein, S. S. McCandless, L. F. Mondshein and V. W. Zue, "A System for Acoustic-Phonetic Analysis of Continuous Speech," IEEE Trans. on Acoust., Sp. and Sig. Proc., Vol. ASSP-23, No. 1, 54-67, Feb. 1975.

[46] B. T. Oshika, V. W. Zue, R. V. Weeks, H. N. Neu, and J. Aurbach, "The Role of Phonological Rules in Speech Understanding Research," IEEE Trans. on Acous. Sp. and Sig. Proc., Vol. ASSP-23, No. 1, 104-112, Feb. 1975.

[47] H. F. Silverman and N. R. Dixon, "A Parametrically Controlled Spectral Analysis System for Speech," IEEE Trans. on Acoust., Sp. and Sig. Proc., Vol. ASSP-22, No. 5, 362-381, Oct. 1974.

[48] N. R. Dixon and H. F. Silverman, "A General Language-Operated Decision Implementation System (GLODIS): Its Application to Continuous-Speech Segmentation," IEEE Trans. on Acoust., Sp. and Sig. Proc., Vol. ASSP-24, No. 2, 137-162, Apr. 1976.

[49] H. Dudley and S. Balashek, "Automatic Recognition of Phonetic Patterns in Speech," J. Acous. Soc. Am., Vol. 30, No. 8, 721-732, Aug. 1958.

[50] P. Ladefoged and D. E. Broadbent, "Information Conveyed by Vowels," J. Acous. Soc. Am., Vol. 29, No. 1, 98-104, Jan. 1957.

[51] G. W. Hughes and M. Halle, "Spectral Properties of Fricative Consonants," J. Acous. Soc. Am., Vol. 28, No. 2, 303-310, Mar. 1956.

[52] D. R. Reddy and P. J. Vicens, "A Procedure for the Segmentation of Connected Speech," Journal of the Audio Engineering Society, Vol. 16, No. 4, 404-411, Oct. 1968.

[53] A. S. House, "On Vowel Duration in English," J. Acous. Soc. Am., Vol. 33, No. 9, 1174-1178, Sept. 1961.

[54] B. S. Atal and L. R. Rabiner, "A Pattern Recognition Approach to Voiced-Unvoiced-Silence Classification with Applications to Speech Recognition," IEEE Trans. on Acoust., Sp. and Sig. Proc., Vol. ASSP-24, No. 3, 201-212, June 1976.

[55] G. E. Peterson and H. L. Barney, "Control Methods Used in a Study of the Vowels," J. Acous. Soc. Am., Vol. 24, No. 2, 175-184, Mar. 1952.

[56] R. Viswanathan and J. Markhoul, "Quantization Properties of Transmission Parameters in Linear Predictive Systems," IEEE Trans. on Acoust., Sp. and Sig. Proc., Vol. ASSP-23, No. 3, 309-321, June 1975.

[57] J. D. Markel, "Digital Inverse Filtering--A New Tool for Forman Trajectory Estimation," IEEE Transactions on Audio and Electroacoustics, Vol. AU-20, No. 2, 129-137, June 1972.

[58] A. E. Rosenberg and M. R. Sambur, "New Techniques for Automatic Speaker Verification," IEEE Trans. on Acoust., Sp. and Sig. Proc., Vol. ASSP-23, No. 2, 1969-175, Apr. 1975.

[59] S. S. McCandless, "An Algorithm for Automatic Formant Extraction Using Linear Prediction Spectra," IEEE Trans. on Acoust., Sp. and Sig. Proc., Vol. ASSP-22, No. 2, Apr. 1974.

[60] R. W. Schafer, "A Survey of Digital Speech Processing Techniques," IEEE Transactions on Audio and Electroacoustics, Vol. AU-20, No. 1, 28-35, Mar. 1972.

[61] J. Markhoul, "Spectral Linear Prediction: Properties and Applications," IEEE Trans. on Acoust., Sp. and Sig. Proc., Vol. ASSP-23, No. 3, 283-296, June 1975.

[62] L. Molho, "Automatic Acoustic-Phonetic Analysis of Fricatives and Plosives," IEEE International Conference on Acoustics, Speech and Signal Processing, 182-189, 1976.

[63] S. Brooks and F. Fallside, "A Technique for Converting the Linear Prediction Areas Model of Speech to a Simple Articulatory Model," IEEE International Conference on Acoustics, Speech, and Signal Processing, 71-78, 1976.

[64] W. Bezdel and H. J. Chandler, "Results of an Analysis and Recognition of Vowels by Computer Using Zero-Crossing Data," Proceedings of the IEEE, Vo. 112, No. 11, 2060-2066, 1965.

[65] W. Bezdel and J. S. Bridle, "Speech Recognition Using Zero-Crossing Measurements and Sequence Information," Proceedings of the IEEE, Vol. 116, No. 4, 617-622, 1969.

[66] I. B. Thomas and R. J. Niederjohn, "A Preliminary Analysis Technique for Speech Sound Classification," Proceedings of the National Electronics Conference, Vol. 25, 685-690, 1969.

[67] P. F. Abboud, N. A. Bezirgan, W. M. Erwin, M. A. Khouri, E. N. McCarus, and R. M. Rammuny, Introduction to Modern Standard Arabic Pronunciation and Writing. Ann Arbor, Mich.: University of Michigan, Department of Eastern Studies, 1968.

[68] A. E. Rosenberg, "Effect of Glottal Pulse Shape on the Quality of Natural Vowels," J. Acoust. Soc. Am., Vol. 49, No. 2 (Part 2), 583-590, 1971.

[69] M. Abdul-Rauf, "Arabic for English Speaking Students," Published by The Iclamic Center, Manufactured by McGregor and Warner, Inc., Library of Congress Catalog Number 76-59592, 4th. Ed., 1979.

[70] A. Y. Ali, "The Holy Quran," Published by American Trust Publishers for The Muslim Students' Association of the United State and Canada, Library of Congress Catalog Number 77-78098, ISBN No. 0-89259-006-8, pp. xvii, June 1977.

APPENDIX A

DATA ACQUISITION

Data Aquisition

It would, of course, be more desirable to develop a digit recognition system for digits spoken by randomly selected untrained speakers rather than for a trained speaker or group of trained speakers. In persuit of this objective, seven speakers were selected, two American males, two American females and three international males. Each person spoke the digits zero through nine in American English. Also several combinations of three connected digits were spoken by the seven speakers. In addition, three sets of digits zero through nine were spoken in a sound proof chamber. Comparably, five Arabic speaking people were randomly selected and they spoke the digits zero through nine in Arabic.

The analog-to-digital conversion system used to prepare the data for computer processing is shown in Figure 105. A low-pass second order filter with a gain of 2 and 4 KHz cut-off frequency is used as shown in Figure 106. An 8 KHz sampling frequency is used. Furthermore, a 10 bit quantizer is used with an analog signal range of $\pm$ 10.24 volt. Therefore 20 mV/Bit resolution was obtained.

The speakers were instructed to depress a button as they began their speech. This caused a pulse to be entered on tape. The data entry procedure is not begun until this, triggering pulse, is detected in order to activate the system. Following detection of the triggering pulse, the voice data entry computer routine is activated, but the A/D conversion does not begin until the speech energy exceeds a preset threshold level. When this level is exceeded, the voice

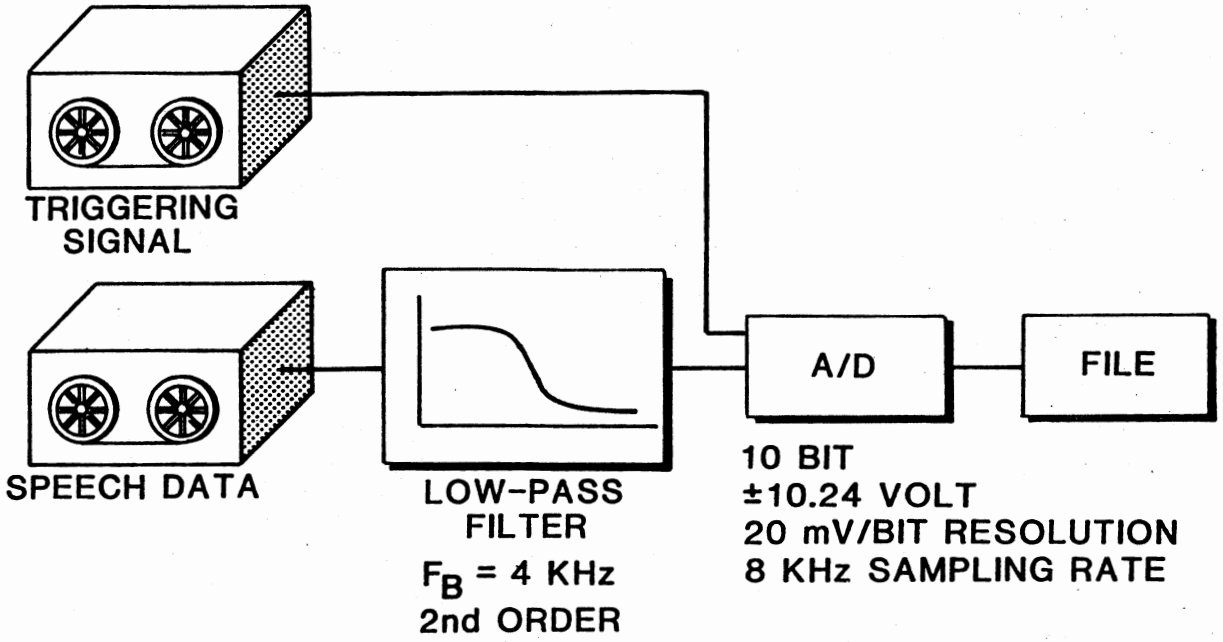
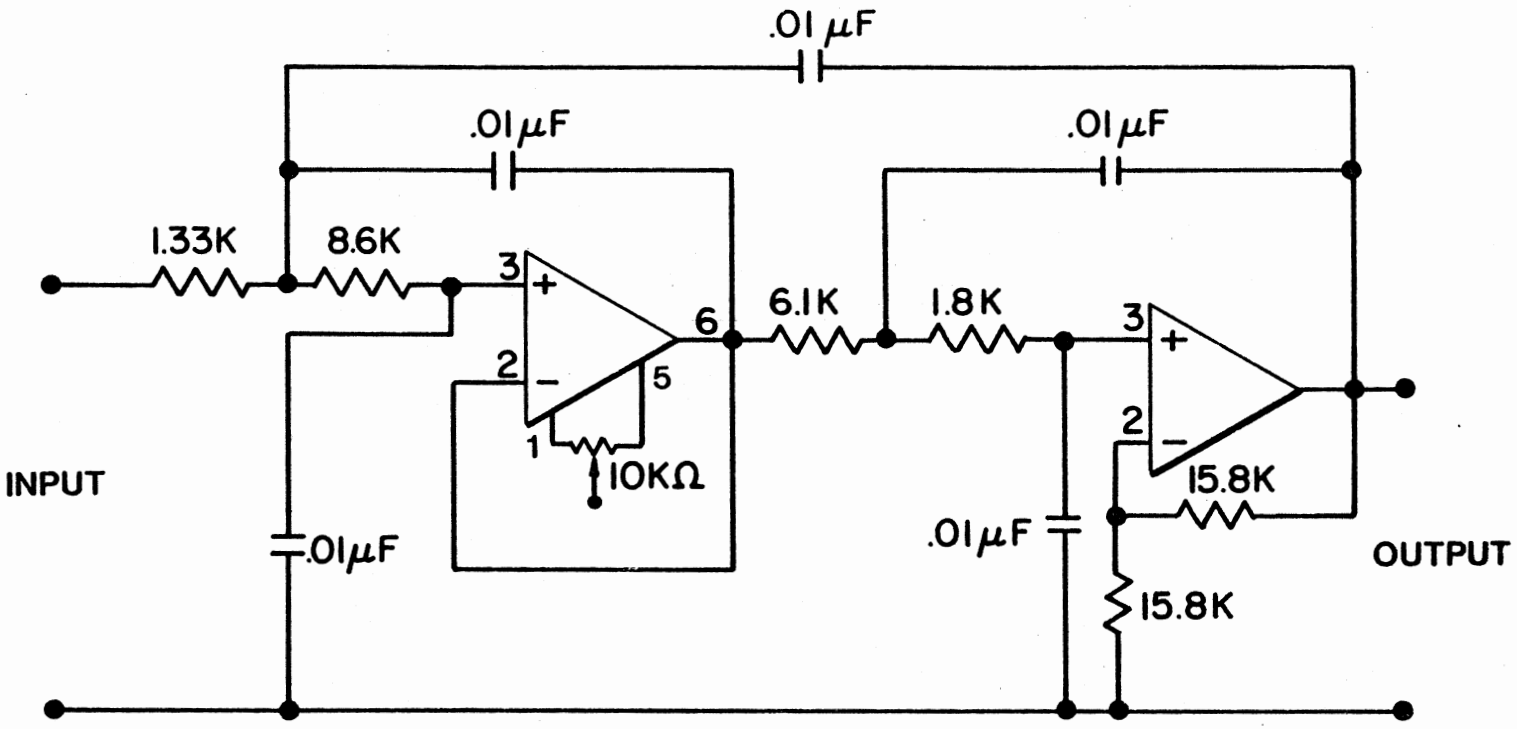


Figure 105. Data Acquisition



Pin [4] V_{cc-}

Pin [7] V_{cc+}

Pin [1] and [5] input for offset voltage (Null Circuit)

Figure 106. Second Order Low-Pass Filter

APPENDIX B

COMPUTER SUBROUTINES

Subroutine FRMS

This subroutine computes the RMS energy using 128 data points per frame. The details are explained in Chapter III. A flow chart is given in Figure 107.

Subroutine FBTR

In this routine the area functions from the reflection coefficients are computed. The BTR, CTR and FTR and SFBR are then estimated as discussed in Chapter III. A flow chart is given in Figure 108.

Subroutine Smooth

This routine performs 3 and 5 point median smoothing of a discretized input signal. A 3-point Hanning window smoother is used in conjunction with the median smoothers. A detailed discussion is given in Chapter III.

Subroutine Quantz

This routine performs the linear quantization of the data in an array. The resulting values are in the range 1-100 as explained in Chapter IV.

Subroutine FDIP

The maxima and the minima in the RMS energy contours are located and then passed to the dip-classification routine. A flow chart is given in Figure 109.

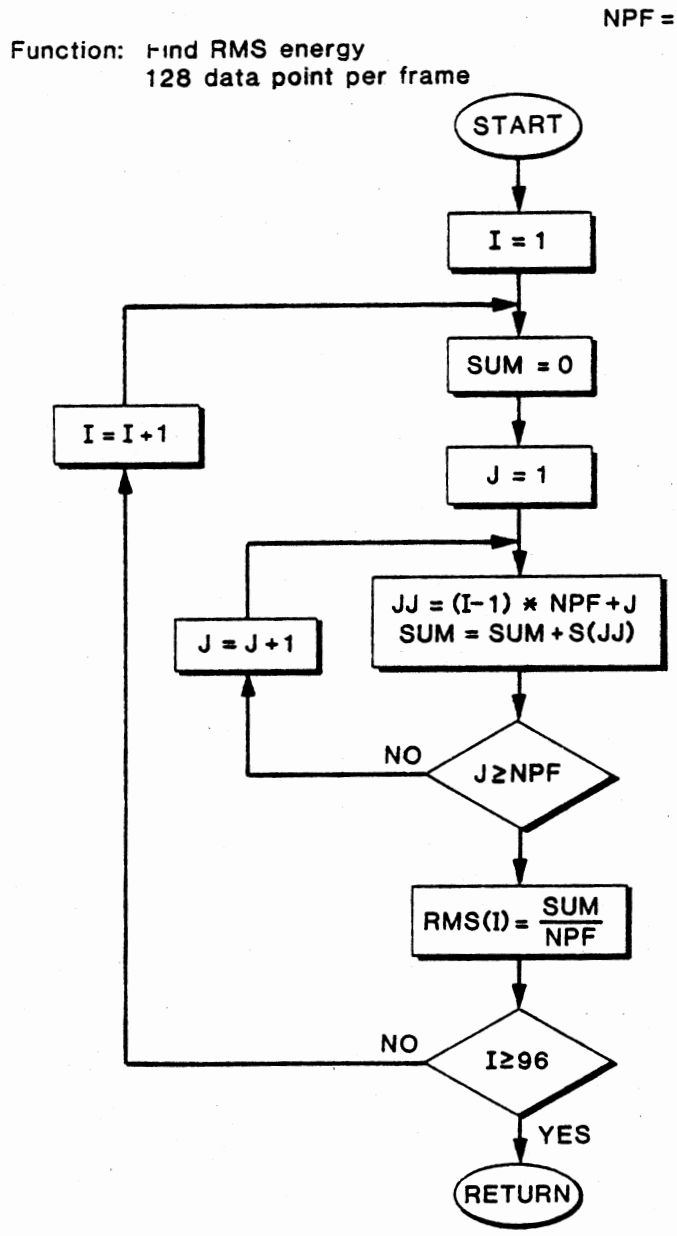


Figure 107. Subroutine FRMS

Function: Find the BTR and SFBR from the reflection coefficients

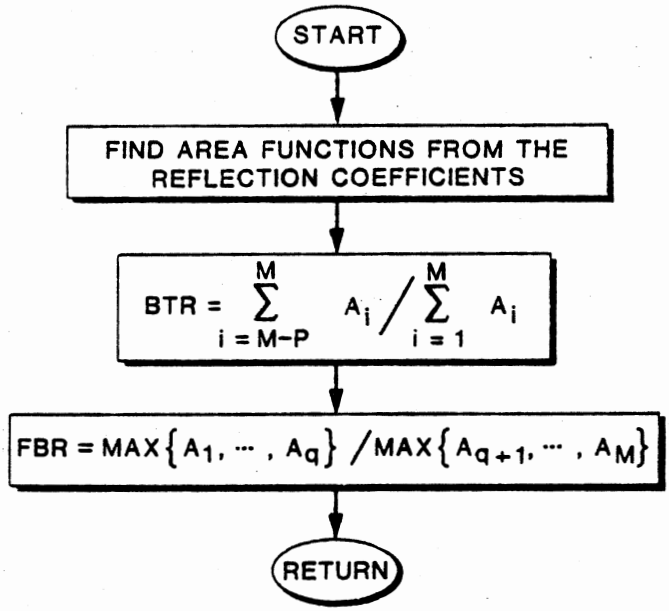


Figure 108. Subroutine FBTR

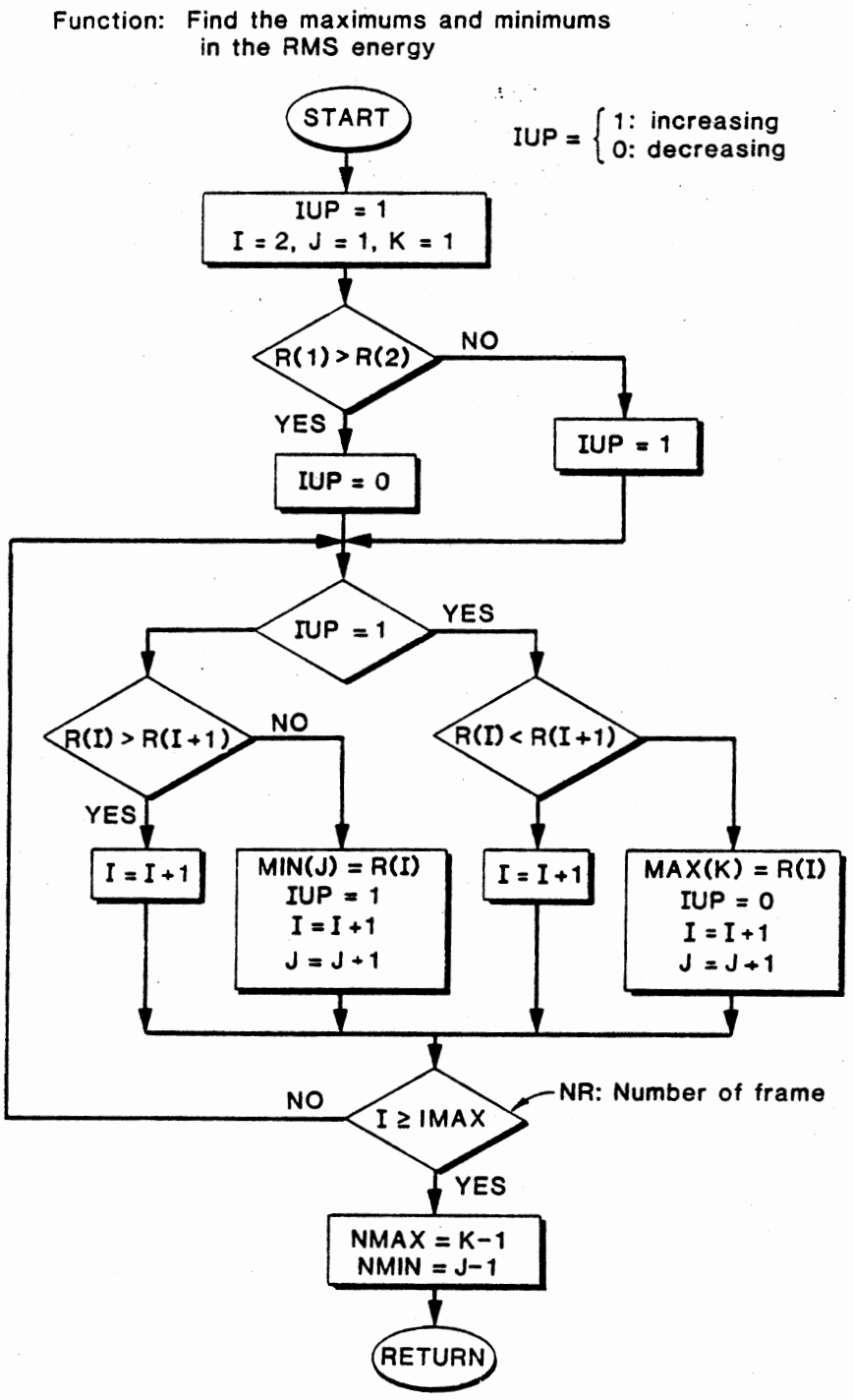


Figure 109. Subroutine FDIP

Subroutine FCSD

It finds the statistical distribution of $V_i^+ - V_j^-$ and $V_{i+1}^+ - V_j^-$, which will be used later to determine the slicing functions SL2 and SL2, as explained in Chapter III.

Subroutine Class

This routine classifies and identifies the three types of dips according to the slicing functions and the sign of Z_1, Z_2, and Z_3 as discussed in Chapter III. A flow chart is given in Figure 110.

Subroutine DWIND

This subroutine windows the data prior to low-frequency pre-emphasis. It uses a 150 point Hamming window.

Subroutine Auto [17]

This routine is for implementing the autocorrelation method of linear prediction, and uses 128 points.

Subroutine DIGIT

This routine recognizes the spoken digit using decisions based on phoneme sequence as discussed in Chapters II, IV, and V.

Subroutine Spect

This routine finds the ratio of the energy in the high frequency to that in the low frequency.

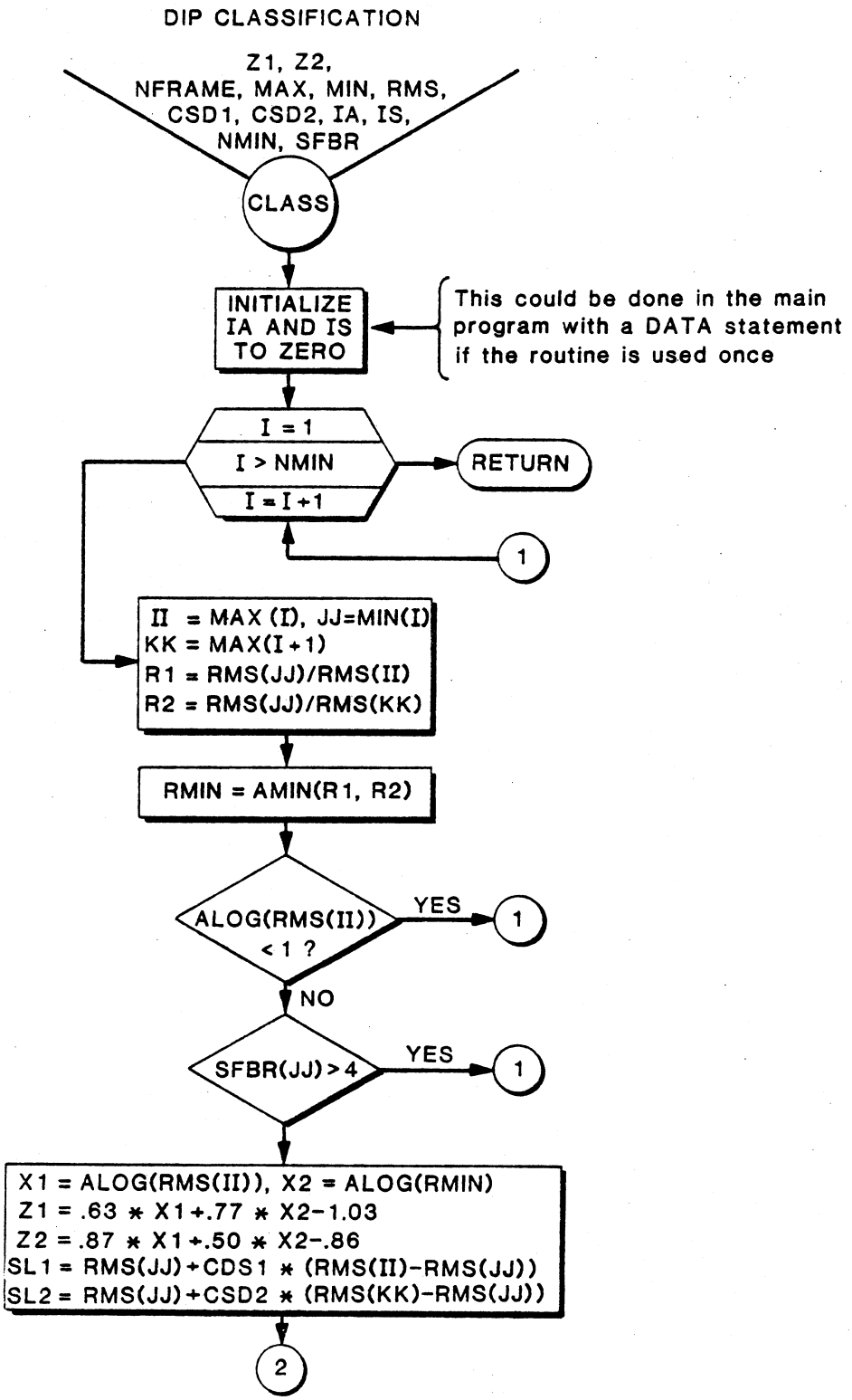


Figure 110. Subroutine Class

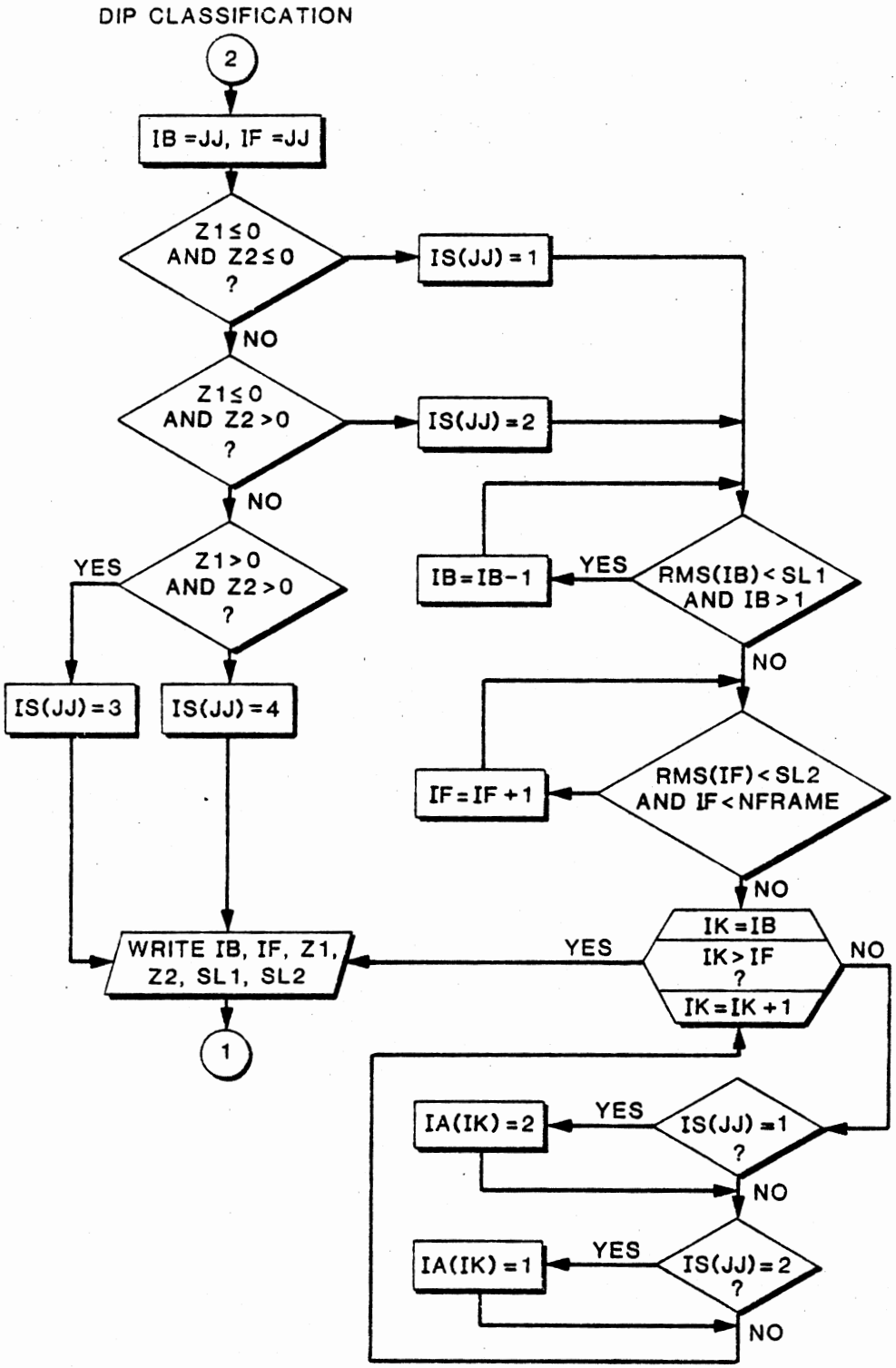


Figure 110. (Continued)

Subroutine UGETIO

This routine plots the RMS energy, BTR, CTR, FTR and the decision for vowel, vowel-like and non-vowel intervals.

```
C                                                                       00000010
C                                                                       00000020
C.....MAIN PROGRAM.........                                             00000030
C                                                                       00000040
C                                                                       00000050
       INTEGER EV                                                       00000060
      DIMENSION IX(10000)                                               00000070
      DIMENSION INOISE(400)                                             00000080
      DIMENSION RMS(400),SRMS(400),QRMS(400)                            00000090
      DIMENSION B(256),RANGE(4),EVLVL(7)                                00000100
      DIMENSION BTR(400) , SFBR(400) , S(15000) , SI(256) , SISM(512)   00000110
      DIMENSION IS(400)                                                 00000120
      DIMENSION IA(400) , MAX(100) , MIN(100)                           00000130
      DIMENSION X(400)                                                  00000140
      DIMENSION KC(21)                                                  00000150
      DIMENSION ALRA(400) , EV(7)                                       00000160
      INTEGER V,VL,ATYPE,BTYPE,STYPE,MTYPE,STYPE1,STYPE2,FINTYP         00000170
      DATA LP/6/,NV,V,VL,NVL,ELNK/'NV','V','VL','NVL',' '/              00000180
      DATA RANGE/16.,13.,7.,1./                                         00000190
C                                                                       00000200
C                                                                       00000210
C                                                                       00000220
C.....READ THE INPUT DATA.....                                          00000230
      READ(5,100) ISTART,ISTOP                                          00000240
C                                                                       00000250
      READ(3,1) IFILE,INUMR                                             00000260
100   FORMAT(2I6)                                                       00000270
      WRITE(LP,111) IFILE,INUMR                                         00000280
111   FORMAT('1',10X,'FILE -- ',I4,10X,'NUMBER OF POINTS',I6)           00000290
      READ(8,2)(IX(K),K=1,ISTOP)                                        00000300
  2      FORMAT (20I4)                                                  00000310
      IDIF = ( ISTOP - ISTART + 1 )                                     00000320
      NF=IDIF/NPS                                                       00000330
      IF (NF*NPS.NE.IDIF) NF=NF+1                                       00000340
      NDIM=NF*NPS+ISTART                                                00000350
      IK=ISTOP+1                                                        00000360
      DO 3 I=IK,NDIM                                                    00000370
3           IX(I)=0                                                     00000380
C                                                                       00000390
C                                                                       00000400
14     YI = 0                                                           00000410
      DO 19  I=1,512                                                    00000420
  19  YI = FLOAT(IX(I)) + YI                                            00000430
      DCLEV = YI / 512                                                  00000440
C                                                                       00000450
         WRITE (LP,*) DCLEV                                             00000460
C                                                                       00000470
C                                                                       00000480
C                                                                       00000490
C                                                                       00000500
C.....SCALE THE GIVEN DATA.....                                         00000510
          J=0                                                           00000520
          DO 6 I=ISTART,NDIM                                            00000530
               J=J+1                                                    00000540
               S(J)=(FLOAT(IX(I))+.3)*SCALE                             00000550
6         CONTINUE                                                      00000560
C                                                                       00000570
C                                                                       00000580
       CALL FRMS (NF,NPS,S,RMS)                                         00000590
C                                                                       00000600
          WRITE(LP,193)(RMS(J),J=1,NF)                                  00000610
193       FORMAT('1'///11X,'UNSMOOTHED RMS VALUES -- '//               00000620
     $              (11X,5E20.7))                                       00000630
          CALL PLOTER(NF,RMS)                                           00000640
          CALL SMOOTH (RMS,SRMS,NF)                                     00000650
          CALL PLOTER(NF,SRMS)                                          00000660
          WRITE(LP,197)(SRMS(J),J=1,NF)                                 00000670
```

```
      WRITE(LP,197)(SRMS(J),J=1,NF)                                     C0000670
197   FORMAT('1'///11X,'SMOOTHED RMS VALUES --'//                       00000680
     $                 (11X,5E20.7))                                    00000690
      CALL QUANTZ(SRMS,QRMS,NF,RMAX,RMIN)                               00000700
      CALL PLOTER(NF,QRMS)                                              00000710
      WRITE(LP,199)(QRMS(J),J=1,NF)                                     00000720
199   FORMAT('1'///11X,'QUANTIZED RMS VALUES --'//                      00000730
     $                 (11X,5E20.7))                                    00000740
C                                                                       00000750
C                                                                       00000760
C                                                                       00000770
      WRITE (LP,7)                                                      00000780
    7 FORMAT (1H1)                                                      00000790
C                                                                       00000800
C                                                                       00000810
C                                                                       00000820
      CALL POLP(SRMS,MAX,MIN,NMAX,NMIN,NF)                              00000830
C     CALL PCSD(NMAX,NMIN,RMS,MAX,MIN,CSD1,CSD2)                        00000840
C                                                                       00000850
      WRITE (6,7)                                                       00000860
C                                                                       00000870
      WRITE(6,1010) NMAX                                                00000880
      WRITE(6,1020) NMIN                                                00000890
      WRITE(6,1030)(MAX(J),J=1,NMAX)                                    00000900
      WRITE(6,1040)(MIN(J),J=1,NMIN)                                    00000910
      WRITE(6,1050) CSD1,CSD2                                           00000920
1010  FORMAT(' NMAX= ',(1X,10I10))                                      00000930
1020  FORMAT(' NMIN= ',(1X,10I10))                                      00000940
1030  FORMAT(' MAX= ',/,(1X,10I10))                                     00000950
1040  FORMAT(' MIN= ',/,(1X,10I10))                                     00000960
1050  FORMAT(' CSD1= ',1PE14.6,/,' CSD2= ',1PE14.6)                     00000970
C                                                                       00000980
C                                                                       00000990
       NPW=256                                                          00001000
      DO 5  I=1,NF                                                      00001010
          DO 4  J=1,NPW                                                 00001020
             K = J + (I-1)*NPS                                          00001030
             S1(J) = S(K)                                               00001040
    4     CONTINUE                                                      00001050
       SAVE=S(K)                                                        00001060
      M1 = 14                                                           00001070
      IP1 = IPT                                                         00001080
C                                                                       00001090
      CALL DWINDF (S1,NPW,10.,1.0,XPRE)                                 00001100
       XPRE=SAVE                                                        00001110
      CALL SLP (NPS,S1,M1,IP1,K1,K2,ALPHA,RC)                           00001120
         CALL FBTR(RC,14,3,4,ANS1,ANS2)                                 00001130
C                                                                       00001140
         BTR(I)=ANS1                                                    00001150
         SFBR(I)=ANS2                                                   00001160
C                                                                       00001170
          CALL SMOOTH(S1,S1SM,NPS)                                      00001180
C         CALL SPECT(S1SM,HLR)                                          00001190
C     HLRA(I) = HLR                                                     00001200
C   5 CONTINUE                                                          00001210
C     WRITE (6,7)                                                       00001220
C         WRITE(6,17)  HLR                                            00001230
17        FORMAT(1H0,10X,'HLR --',F10.3)                                00001240
C                                                                       00001250
      WRITE(6,1060) (BTR(KL),KL=1,NF)                                   00001260
1060  FORMAT(' BTR= ',/,(1X,1P8E14.6))                                  00001270
      WRITE (6,7)                                                       00001280
      WRITE(6,1070) (SFBR(KL),KL=1,NF)                                  00001290
1070  FORMAT(' SFBR=',/,(1X,1P8E14.6))                                  00001300
      WRITE (6,7)                                                       00001310
       CALL PLOTER(NF,BTR)                                              00001320
       CALL PLOTER(NF,SFBR)                                             00001330
```

```
      CALL PLOTER(NF,SFBR)                                              00001330
      GO TO 99                                                          00001340
C                                                                       00001350
         CSD1 = 0.30                                                    00001360
         CSD2 = 0.17                                                    00001370
      CALL CLASS (NF,MAX,MIN,QRMS,CSD1,CSD2,IA,IS,NMIN,SFER)            00001380
C                                                                       00001390
C          PRELIMENARY DECISIONS                                        00001400
C                                                                       00001410
C         PRINT THE BTR, SFBR, AND HLR VALUES AND THEIR TYPE            00001420
C                                                                       00001430
490        WRITE (LP,41)                                                00001440
   41    FORMAT('1'///' ',T21,'** PRELIMENARY DECISIONS **'///' ',      00001450
     $    T13,'FRAME',T27,'RMS',T37,'IA',T43,'ATYPE',T57,'BTR',T63,     00001460
     $    'BTYPE',T77,'SFBR',T83,'STYPE',T97,'HLR',T103,'HTYPE')        00001470
C                                                                       00001480
         DO 1600 J=1,NF                                                 00001490
C                                                                       00001500
              ATYPE=BLNK                                                00001510
           IF (IA(J) .EQ. 2)  ATYPE=NV                                  00001520
           IF (IA(J) .EQ. 1)  ATYPE=VL                                  00001530
           IF (IA(J) .EQ. 0)  ATYPE=V                                   00001540
C                                                                       00001550
500           BTYPE=BLNK                                                00001560
           IF (BTR(J) .GT. 0.15) BTYPE=NV                               00001570
           IF (BTR(J) .LE. 0.15) BTYPE=V                                00001580
C                                                                       00001590
              IF(SFBR(J).GT. (-2.5)) STYPE=NV                           00001600
              IF(SFBR(J).LE. (-2.5)) STYPE=V                            00001610
             IF (SFBR(J) .GT. 0.)  STYPE=VL                             00001620
C                                                                       00001630
              HTYPE=BLNK                                                00001640
              IF(HLRA(J).LE. .2) HTYPE=NVL                              00001650
              IF (HLRA(J) .GT. .2) HTYPE=VL                             00001660
C                                                                       00001670
C                                                                       00001680
           WRITE (LP,43) J,SRMS(J),IA(J),ATYPE,BTR(J),BTYPE,            00001690
     $                          SFBR(J),STYPE,HLRA(J),HTYPE             00001700
   43      FORMAT(' ',T11,I5,T21,F10.3,T36,I3,T44,A3,T51,F10.3,         00001710
     $         T64,A3,T71,F10.3,T84,A3,T91,F10.3,T104,A3)               00001720
C                                                                       00001730
1600       CONTINUE                                                     00001740
C                                                                       00001750
C          FIRST  SECONDARY  DECISIONS                                  00001760
C         PRELIMINARY DECISION BASED ON SFBR AND HLR                    00001770
C                                                                       00001780
         WRITE(LP,47)                                                   00001790
   47   FORMAT('1'///' ',T13,'FRAME',T27,'SFBR',T41,'HLR',T49,'STYPE1') 00001800
C                                                                       00001810
         DO 1800 J=1,NF                                                 00001820
              STYPE1=NVL                                                00001830
              IF(SFBR(J).LE. (-2.5) .OR. HLRA(J).GT. .2) STYPE1=VL      00001840
              WRITE(LP,51) J,SFBR(J),HLRA(J),STYPE1                     00001850
51            FORMAT(' ',T11,I5,T21,F10.2,T36,F10.2,T49,A3)             00001860
1800       CONTINUE                                                     00001870
C                                                                       00001880
           WRITE (6,7)                                                  00001890
           DO 1801  J=1,NF                                              00001900
           STYPE1=VL                                                    00001910
              IF(SFBR(J).GT. (-2.5) .OR. HLRA(J).LE. .2) STYPE1=NVL     00001920
        WRITE(LP,51) J,SFBR(J),HLRA(J),STYPE1                           00001930
1801       CONTINUE                                                     00001940
C                                                                       00001950
C         SECONDARY DECISION                                            00001960
C                                                                       00001970
         WRITE(LP,59)                                                   00001980
59       FORMAT('1'///' ',T21,'*** SECONDARY DECISION ***'//            00001990
```

```
59          FORMAT('1'///' ',T21,'*** SECONDARY DECISION ***'//                00001990
     $             ' ',T13,'FRAME',T28,'IA',T43,'BTR',T57,'STYPE2'//)          00002000
C                                                                               00002010
            DO 1900  K=1,NF                                                     00002020
            STYPE2=V                                                            00002030
          IF ((IA(K) .EQ. 2 .OR. BTR(K).GT. .15)  STYPE2=NV                     00002040
                WRITE(LP,61)K,IA(K),BTR(K),STYPE2                               00002050
61              FORMAT(' ',T11,I5,T25,I7,T39,F10.3,T53,A3)                      00002060
1900        CONTINUE                                                            00002070
C                                                                               00002080
            WRITE (6,7)                                                         00002090
            DO 1901  K=1,NF                                                     00002100
            STYPE2=NVL                                                          00002110
          IF ((IA(K) .EQ. 1 .OR. BTR(K).LE. .15)  STYPE2=VL                     00002120
                WRITE(LP,61)K,IA(K),BTR(K),STYPE2                               00002130
1901        CONTINUE                                                            00002140
C                                                                               00002150
            WRITE (6,7)                                                         00002160
            DO 1902  K=1,NF                                                     00002170
            STYPE2=NV                                                           00002180
          IF ((IA(K) .EQ. 0 .OR. BTR(K).LE. .15)  STYPE2=V                      00002190
                WRITE(LP,61)K, IA(K),BTR(K),STYPE2                              00002200
1902        CONTINUE                                                            00002210
C                                                                               00002220
C           FINAL DECISION                                                      00002230
C                                                                               00002240
            WRITE(LP,63)                                                        00002250
63          FORMAT('1'///' ',T21,'*** FINAL DECISION ***'///' ',T13,            00002260
     $             'FRAME',T27,'RMS',T35,'FINTYP'//)                           00002270
C                                                                               00002280
            DO 2100 K=1,NF                                                      00002290
                FINTYP=VL                                                       00002300
        IF ((IA(K).EQ.2 .OR. BTR(K).GT. .15 .OR. SFBR(K).GT.(-2.5)              00002310
     $  .OR. HERA(K).LE. .2) FINTYP=NVL                                         00002320
                WRITE(LP,67)K,SRMS(K),FINTYP                                    00002330
67              FORMAT(' ',T11,I5,T21,F10.2,T36,A3)                             00002340
2100        CONTINUE                                                            00002350
C                                                                               00002360
                    WRITE (LP,7)                                                00002370
            DO 2101  K=1,NF                                                     00002380
            FINTYP=NVL                                                          00002390
        IF ((IA(K).EQ.1 .OR. BTR(K).LE. .15 .OR. SFBR(K).LE.(-2.5)              00002400
     $ .OR. HERA(K).GT. .2)  FINTYP=VL                                          00002410
                WRITE(LP,67)K, SRMS(K),FINTYP                                   00002420
2101        CONTINUE                                                            00002430
C                                                                               00002440
                    WRITE (LP,7)                                                00002450
            DO 2102  K=1,NF                                                     00002460
            FINTYP=NV                                                           00002470
        IF ((IA(K).EQ.0 .OR. BTR(K).LE. .15 .OR. SFBR(K).LE.(-2.5)              00002480
     $ .OR. HERA(K).GT. .2)  FINTYP=V                                           00002490
                WRITE(LP,67)K, SRMS(K),FINTYP                                   00002500
2102        CONTINUE                                                            00002510
C                                                                               00002520
C                                                                               00002530
C                                                                               00002540
C           FIND THE SUM AND AVERAGE OF THE SMOOTHED RMS VALUES                 00002550
C                                                                               00002560
C                                                                               00002570
            SUM=0.                                                              00002580
            DO 6000 J=1,NF                                                      00002590
C                                                                               00002600
                SUM=SUM+SRMS(J)                                                 00002610
6000        CONTINUE                                                            00002620
C                                                                               00002630
            AVE=SUM/FLOAT(NF)                                                   00002640
            WRITE(LP,205)SUM,NF,AVE,SMEAN,SCALE                                 00002650
```

```
      WRITE(LP,205)SUM,NF,AVE,SMEAN,SCALE                               00002650
205   FORMAT('1'///11X,'SUM OF SMOOTHED RMS VALUES',E20.7//             00002660
     $        11X,'NUMBER OF FRAMES',I10//                             00002670
     $        11X,'AVERAGE OF SMOOTHED RMS',E20.7//                    00002680
     $        11X,'MEAN SQUARE OF INPUT SIGNAL',E20.7//                00002690
     $        11X,'SCALING VALUE FOR INPUT SIGNAL',E20.7)              00002700
C                                                                       00002710
C                                                                       00002720
C                                                                       00002730
C                                                                       00002740
C     DETERMINE THE PEAK LEVELS                                         00002750
C                                                                       00002760
C                                                                       00002770
      DO 5500 J=1,NMAX                                                  00002780
C                                                                       00002790
           RMSPK=SRMS(MAX(J))/1.                                        00002800
C                                                                       00002810
           DO 5400 K=1,4                                                00002820
C                                                                       00002830
                IF(RMSPK .GE. RANGE(K)) GO TO 5450                      00002840
5400       CONTINUE                                                     00002850
C                                                                       00002860
           EV(J)=5                                                      00002870
           GO TO 5460                                                   00002880
C                                                                       00002890
5450       EV(J)=K                                                      00002900
5460       EVLVL(J)=RMSPK                                               00002910
C                                                                       00002920
5500  CONTINUE                                                          00002930
C                                                                       00002940
      EV(NMAX+1)=6                                                      00002950
C                                                                       00002960
C                                                                       00002970
C                                                                       00002980
      WRITE(LP,195)(EV(J),J=1,NMAX)                                     00002990
195   FORMAT('1'///11X,'THRESHOLDS (EV)'///11X,7I10)                    00003000
      WRITE(LP,196)(EVLVL(J),J=1,NMAX)                                  00003010
196   FORMAT(///11X,7E20.7)                                             00003020
C                                                                       00003030
C                                                                       00003040
C                                                                       00003050
510   CALL DIGIT(EV,BTR,JDIGIT)                                         00003060
      WRITE(5,151) JDIGIT                                               00003070
151   FORMAT('1'///10X,'RESULT - ',I5)                                  00003080
C                                                                       00003090
 99   STOP                                                              00003100
      END                                                               00003110
C                                                                       00003120
C                                                                       00003130
C                                                                       00003140
C                                                                       00003150
      SUBROUTINE FRMS(NFRAME,NPS,S,RMS)                                 00003160
      DIMENSION RMS(1) , S(1)                                           00003170
      DO 16 I= 1,NFRAME                                                 00003180
         SUM=0.                                                         00003190
         DO 15 J=1,NPS                                                  00003200
            JJ=(I-1)*NPS+J                                              00003210
            SUM=SUM+S(JJ)*S(JJ)                                         00003220
 15      CONTINUE                                                       00003230
         RMS(I)=SUM/FLOAT(NPS)                                          00003240
 16   CONTINUE                                                          00003250
      RETURN                                                            00003260
      END                                                               00003270
C                                                                       00003280
C                                                                       00003290
```

```
      SUBROUTINE PLOTER(NFRAME,ARRAY)                                   00003310
      DIMENSION ARRAY(1)                                                00003320
      DIMENSION X(400)                                                  00003330
      DIMENSION B(400,1)                                                00003340
      INTEGER IMAG4(5151),ITITLE(144),ICHAR(10)                         00003350
      REAL RANGE(4)                                                     00003360
      DATA ICHAR(1)/1H0/,RANGE/4*0.0/                                   00003370
C                                                                       00003380
      CALL JGETID(1,NIN,NOUT)                                           00003390
C                                                                       00003400
      READ (NIN,1001) (ITITLE(I),I=1,144)                               00003410
 1001 FORMAT (72A1)                                                     00003420
C                                                                       00003430
      DO 21  I=1 , NFRAME                                               00003440
      B(I,1) = ARRAY(I)                                                 00003450
      X(I) = FLOAT(I)                                                   00003460
   21 CONTINUE                                                          00003470
C                                                                       00003480
      INC=1                                                             00003490
      MM=1                                                              00003500
      IY = NFRAME                                                       00003510
      IOPT=1                                                            00003520
C                                                                       00003530
      CALL JSPLT(X,B,IY,NFRAME,MM, INC, ITITLE,RANGE,ICHAR, IOPT, IMAG4,IER)00003540
C                                                                       00003550
      RETURN                                                            00003560
      END                                                               00003570
C                                                                       00003580
C                                                                       00003590
C                                                                       00003600
      SUBROUTINE FDIP(RMS,MAX,MIN,KMAX,KMIN,NR)                         00003610
C                                                                       00003620
      DIMENSION RMS(1),MAX(100),MIN(100)                                00003630
C                                                                       00003640
      IUP=1                                                             00003650
      I=2                                                               00003660
      J=1                                                               00003670
      K=1                                                               00003680
      DO 165  MAA=1,100                                                 00003690
C                                                                       00003700
      MAX(MAA) = (NR + 1)                                               00003710
      MIN(MAA) = (NR + 1)                                               00003720
C                                                                       00003730
  165 CONTINUE                                                          00003740
      IF(RMS(1) .GT. RMS(2)) GO TO 99                                   00003750
         IUP=1                                                          00003760
         GO TO 100                                                      00003770
   99    IUP=0                                                         00003780
         MAX(K)=1                                                       00003790
         K=K+1                                                          00003800
   17 CONTINUE                                                          00003810
  100 IF (IUP .EQ. 1) GO TO 19                                          00003820
C                                                                       00003830
      IF (RMS(I) .GE. RMS(I+1)) GO TO 18                                00003840
         MIN(J)=1                                                       00003850
         IUP=1                                                          00003860
         J=J+1                                                          00003870
   18 GO TO 20                                                          00003880
   19 IF(RMS(I) .LE. RMS(I+1)) GO TO 20                                 00003890
         MAX(K)=I                                                       00003900
         IUP=0                                                          00003910
         K=K+1                                                          00003920
   20 I=I+1                                                             00003930
C                                                                       00003940
      IF (I .LT. NR) GO TO 17                                           00003950
```

```
                                                                        00003950
      NMAX=K-1                                                          00003960
      NMIN=J-1                                                          00003970
      RETURN                                                            00003980
      END                                                               00003990
C                                                                       00004000
C                                                                       00004010
C                                                                       00004020
C.....FIND THE STATISTICAL DISTRIBUTION OF VI-VJ.....                   00004030
C                                                                       00004040
      SUBROUTINE FCSD(NMAX,NMIN,RMS,MAX,MIN,CSD1,CSD2)                  00004050
      DIMENSION RMS(1),MAX(1),MIN(1)                                    00004060
      SUM=0.0                                                           00004070
C                                                                       00004080
      DO 244 I=1,NMAX                                                   00004090
         LL=MAX(I)                                                      00004100
         NN=MIN(I)                                                      00004110
         SUM=SUM+(RMS(LL)-RMS(NN))                                      00004120
  244 CONTINUE                                                          00004130
      CSD1=SUM/FLOAT(NMAX)                                              00004140
C                                                                       00004150
      SUM=0.0                                                           00004160
      NMA=NMAX-1                                                        00004170
      DO 25 I=1,NMA                                                     00004180
         LL=MAX(I+1)                                                    00004190
         NN=MIN(I)                                                      00004200
         SUM=SUM+(RMS(LL)-RMS(NN))                                      00004210
   25 CONTINUE                                                          00004220
         CSD2=SUM/FLOAT(NMA)                                            00004230
      RETURN                                                            00004240
      END                                                               00004250
C                                                                       00004260
C.....DIP CLASSIFICATION.....                                           00004270
C                                                                       00004280
      SUBROUTINE CLASS(NFRAME,MAX,MIN,RMS,CSD1,CSD2,IA,IS,NMIN,SFBR)    00004290
C                                                                       00004300
      DIMENSION MAX(1),MIN(1),RMS(1),IA(384),IS(384)                    00004310
      DIMENSION SFBR(1)                                                 00004320
C                                                                       00004330
      DO 22 I=1,NFRAME                                                  00004340
         IA(I)=0                                                        00004350
         IS(I)=0                                                        00004360
   22 CONTINUE                                                          00004370
C                                                                       00004380
      DO  24 I=1,NMIN                                                   00004390
         II=MAX(I)                                                      00004400
         JJ=MIN(I)                                                      00004410
         KK=MAX(I+1)                                                    00004420
         R1=RMS(JJ)/RMS(II)                                             00004430
          R2=RMS(JJ)/RMS(KK)                                            00004440
C                                                                       00004450
      RMIN=AMIN1(R1,R2)                                                 00004460
  190    IF(ALOG(RMS(II)) .LT. 1.0 .OR. SFBR(JJ).GT.4.0) GO TO  24      00004470
C                                                                       00004480
         X1=ALOG(RMS(II))                                               00004490
         X2=ALOG(RMIN)                                                  00004500
C                                                                       00004510
         Z1=0.63*X1+0.77*X2-1.03                                        00004520
         Z2=0.87*X1+0.50*X2-0.86                                        00004530
C                                                                       00004540
         SL1=RMS(JJ)+CSD1*(RMS(II)-RMS(JJ))                             00004550
         SL2=RMS(JJ)+CSD2*(RMS(KK)-RMS(JJ))                             00004560
C                                                                       00004570
C                                                                       00004580
         IB=JJ                                                          00004590
         IF=JJ                                                          00004600
C     ... DIP ...                                                       00004610
```

```
C     ... DIP ...                                                        00004610
      IF (Z1 .LE. 0.0 .AND. Z2 .LE. 0.0) GO TO  91                       00004620
C     ... DIP 2 ...                                                      00004630
      IF (Z1 .LE. 0.0 .AND. Z2 .GT. 0.0) GO TO 92                        00004640
C     ... DIP 3 ...                                                      00004650
      IF (Z1 .GT. 0.0 .AND. Z2 .GT. 0.0) GO TO  93                       00004660
C                                                                        00004670
      IS(JJ) = 4                                                         00004680
      GO TO  99                                                          00004690
C                                                                        00004700
   91 IS(JJ) = 1                                                         00004710
      GO TO  95                                                          00004720
C                                                                        00004730
   92 IS(JJ) = 2                                                         00004740
      GO TO  95                                                          00004750
C                                                                        00004760
   93 IS(JJ) = 3                                                         00004770
      GO TO  99                                                          00004780
C                                                                        00004790
   95 IF (RMS(IB) .LT. SL1 .AND. IB .GT. 1) GO TO  101                   00004800
      GO TO  102                                                         00004810
  101 IB = IB - 1                                                        00004820
      GO TO  95                                                          00004830
C                                                                        00004840
  102 IF (RMS(IF) .LT. SL2 .AND. IF .LT. NFRAME) GO TO  103              00004850
      GO TO  104                                                         00004860
  103 IF = IF + 1                                                        00004870
      GO TO  102                                                         00004880
C                                                                        00004890
  104 DO 105  IK = IB , IF                                               00004900
      IF (IS(JJ) .EQ. 1) IA(IK) = 2                                      00004910
      IF (IS(JJ) .EQ. 2) IA(IK) = 1                                      00004920
      IF (IS(JJ) .EQ. 3) IA(IK) = 0                                      00004930
105   CONTINUE                                                           00004940
C                                                                        00004950
99    WRITE(6,1000) IB,IF,Z1,Z2,SL1,SL2                                  00004960
1000  FORMAT(' IB= ',I10/                                                00004970
     1' IF= ',I10/                                                       00004980
     2' Z1= ',1PE14.6/                                                   00004990
     3' Z2= ',1PE14.6/                                                   00005000
     4' SL1= ',1PE14.6/                                                  00005010
     5' SL2= ',1PE14.6)                                                  00005020
C                                                                        00005030
24    CONTINUE                                                           00005040
      RETURN                                                             00005050
      END                                                                00005060
C                                                                        00005070
C                                                                        00005080
C                                                                        00005090
      SUBROUTINE DWINDF (Y,N,IHAM,PRE,XPRE)                              00005100
      DIMENSION Y(1)                                                     00005110
C                                                                        00005120
C         PREEMPHASIS AND WINDOWING OF A SIGNAL                          00005130
C                                                                        00005140
C         Y :   ARRAY CONTAINING THESIGNAL TO BE USED                    00005150
C         N :   NUMBER OF POINTS                                         00005160
C      IHAM :   ALPHABETIC 'Y' OR 'N' TO CONTROL WINDOW USE              00005170
C       PRE :   PREEMPHASIS FACTOR (-1.LE.PRE.LE.1)                      00005180
C      XPRE :   PREVIOUS SIGNAL POINT FOR PROPER PREEMPHASIS             00005190
C                                                                        00005200
C     IF PRE > 0 , HIGH FREQUENCY PREEMPHASIS IS USED                    00005210
C     IF PRE < 0 , LOW FREQUENCY PREEMPHASIS IS USED                     00005220
C     IF IHAM = 'Y ', A HAMMING WINDOW IS USED                           00005230
C     IF IHAM = 'N ', NO WINDOW IS USED                                  00005240
C/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 00005250
C                                                                        00005260
C                                                                        00005270
```

```
      NM1=J-1
C     RETURN                                                          00005260
C                                                                     00005270
      DATA ICHY/10/                                                   00005280
      HAM(I) = .54 - .46*COS((I-1)*CONST)                             00005290
      PI = 3.14159265                                                 00005300
      CONST = 2*PI/N                                                  00005310
C                                                                     00005320
      IF (IHAM.NE.ICHY) GO TO 50                                      00005330
C                                                                     00005340
C         HIGH FREQUENCY PREEMPHASIS AND HAMMING WINDOW               00005350
C                                                                     00005360
      IF (PRE.LT.0.) GO TO 30                                         00005370
      B = XPRE                                                        00005380
      DO 20  I=1,N                                                    00005390
      A = Y(I)                                                        00005400
      Y(I) = (A-PRE*B)*HAM(I)                                         00005410
   20 B = A                                                           00005420
      GO TO 100                                                       00005430
C                                                                     00005440
C         LOW FREQUENCY PREEMPHASIS AND HAMMING WINDOW                00005450
C                                                                     00005460
   30 B = 0.                                                          00005470
      DO 40  I=1,N                                                    00005480
      A = Y(I) - PRE*B                                                00005490
      Y(I) = A*HAM(I)                                                 00005500
   40 B = A                                                           00005510
      GO TO 100                                                       00005520
C                                                                     00005530
C         HIGH FREQUENCY PREEMPHASIS AND NO WINDOWING                 00005540
C                                                                     00005550
   50 IF (PRE.LT.0.) GO TO 80                                         00005560
      B = XPRE                                                        00005570
      DO 70  I=1,N                                                    00005580
      A = Y(I)                                                        00005590
      Y(I) = A - PRE*B                                                00005600
   70 B = A                                                           00005610
      GO TO 100                                                       00005620
C                                                                     00005630
C         LOW FREQUENCY PREEMPHASIS AND NO WINDOWING                  00005640
C                                                                     00005650
   80 B = 0.                                                          00005660
      DO 90  I=1,N                                                    00005670
      A = Y(I) - PRE*B                                                00005680
      Y(I) = A                                                        00005690
   90 B = A                                                           00005700
  100 RETURN                                                          00005710
      END                                                             00005720
C                                                                     00005730
C                                                                     00005740
      SUBROUTINE SLP (N,X,M,IP,K1,K2,ALPHA,RC)                        00005750
C                                                                     00005760
      DIMENSION X(1) , Y(512) , RC(1)                                 00005770
      DIMENSION A(21) , B(21) , R(21)                                 00005780
C                                                                     00005790
      PI = 3.141592653                                                00005800
      I = 2**IP                                                       00005810
      DO 10  J=1,I                                                    00005820
   10 Y(J) = 0.                                                       00005830
      NP = N + 1                                                      00005840
      DO 20  J=NP,I                                                   00005850
   20 X(J) = 0.                                                       00005860
C                                                                     00005870
      CALL FFT (X,Y,IP)                                               00005880
C                                                                     00005890
      L = K2 - K1                                                     00005900
      LP = L + 1                                                      00005910
      DO 30  J=1,LP                                                   00005920
      JK = J + K1                                                     00005930
```

```
      JK = J + KI                                                       00005930
C                                                                       00005940
   30 X(J) = X(JK)*X(JK) + Y(JK)*Y(JK)                                  00005950
C                                                                       00005960
      MP = M + 1                                                        00005970
      RL = X(LP)                                                        00005980
      DO 50  J=1,MP                                                     00005990
      R(J) = X(1) + RL                                                  00006000
      RL = -RL                                                          00006010
      LM = L - 1                                                        00006020
      DO 40  K=1,LM                                                     00006030
      ARG = (PI*K*(J-1)) / L                                            00006040
   40 R(J) = 2.*X(K+1)*COS(ARG) + R(J)                                  00006050
   50 R(J) = 0.5*R(J) / L                                               00006060
      ALPHA = R(1)                                                      00006070
      IF (R(1).NE.0.0) RC(1)=-R(2)/R(1)                                 00006080
      A(1) = 1.                                                         00006090
      A(2) = RC(1)                                                      00006100
      ALPHA = ALPHA - RC(1)*RC(1)*ALPHA                                 00006110
      DO 90  MINC=2,M                                                   00006130
      MF = MINC - 1                                                     00006140
      DO 60  J=1,MINC                                                   00006150
      JB = MINC - J + 1                                                 00006160
   60 B(J) = A(JB)                                                      00006170
      MF = MF + 1                                                       00006180
      S = 0.                                                            00006190
      DO 70  JP=1,MF                                                    00006200
      MIP = MF-JP+2                                                     00006210
   70 S = S + R(MIP)*A(JP)                                              00006220
      IF (ALPHA.NE.0.0) RC(M)=-S/ALPHA                                  00006230
      DO 80  JP=2,MF                                                    00006240
   80 A(JP) = A(JP) + RC(MF)*B(JP-1)                                    00006250
      A(MF+1) = RC(MF)                                                  00006260
      ALPHA = ALPHA - RC(MF)*RC(MF)*ALPHA                               00006270
   90 CONTINUE                                                          00006280
C                                                                       00006290
C                                                                       00006300
C     IF (X(J).NE.0.0) PRINT *,'POWER SPECTRUM= ',X                     00006310
C                                                                       00006320
      RETURN                                                            00006330
      END                                                               00006340
C                                                                       00006350
C                                                                       00006360
      SUBROUTINE FFT (X,Y,L)                                            00006370
      DIMENSION X(1) , Y(1)                                             00006380
C                                                                       00006390
C         RADIX - 2FFT                                                  00006400
C                                                                       00006410
      NP = 2**L                                                         00006420
      LMX = NP                                                          00006430
      SCL = 6.283185383/NP                                              00006440
      DO 20  LO=1,L                                                     00006450
      LIX = LMX                                                         00006460
      LMX = LMX/2                                                       00006470
      ARG = 0.                                                          00006480
      DO 10  LM=1,LMX                                                   00006490
      C = COS(ARG)                                                      00006500
      S = SIN(ARG)                                                      00006510
C                                                                       00006520
      ARG = ARG + SCL                                                   00006530
      DO 10  LI=LIX,NP,LIX                                              00006540
      J1 = LI - LIX + LM                                                00006550
      J2 = J1 + LMX                                                     00006560
      T1 = X(J1) - X(J2)                                                00006570
      T2 = Y(J1) - Y(J2)                                                00006580
      X(J1) = X(J1) + X(J2)                                             00006590
      Y(J1) = Y(J1) + Y(J2)                                             00006600
```

```
      X(J2) = C*T1*T2                                                   00006610
   10 Y(J2) = C*T2 - S*T1                                               00006620
   20 SCL = 2.*SCL                                                      00006630
C                                                                       00006640
C          BIT REVERSAL                                                 00006650
C                                                                       00006660
      J = 1                                                             00006670
      NV2 = NP/2                                                        00006680
      NPM1 = NP - 1                                                     00006690
      DO 50  I=1,NPM1                                                   00006700
      IF (I.GE.J) GO TO 30                                              00006710
      T1 = X(J)                                                         00006720
      T2 = Y(J)                                                         00006730
      X(J) = X(I)                                                       00006740
      Y(J) = Y(I)                                                       00006750
      X(I) = T1                                                         00006760
      Y(I) = T2                                                         00006770
   30 K = NV2                                                           00006780
   40 IF (K.GE.J) GO TO 50                                              00006790
      J = J - K                                                         00006800
      K = K/2                                                           00006810
      GO TO 40                                                          00006820
   50 J = J + K                                                         00006830
      RETURN                                                            00006840
      END                                                               00006850
C                                                                       00006860
C                                                                       00006870
      SUBROUTINE FBTR(RC,M,IP,IQ,ANS1,ANS2)                             00006880
C                                                                       00006890
      DIMENSION AREA(20),RC(1)                                          00006900
      AREA(M+1)=1.0                                                     00006910
      DO 10 J=1,M                                                       00006920
         I = M-J+1                                                      00006930
         AREA(I) = (1.+RC(I))*AREA(I+1)/(1.-RC(I))                      00006940
   10 CONTINUE                                                          00006950
C                                                                       00006960
C.........FIND BTR..........                                            00006970
C                                                                       00006980
      SUM=0.                                                            00006990
      SUM1=0.                                                           00007000
      MP=M-IP                                                           00007010
      DO 11 I=MP,M                                                      00007020
         SUM1=SUM1+AREA(I)                                              00007030
   11 CONTINUE                                                          00007040
      DO 12 I=1,M                                                       00007050
         SUM=SUM+AREA(I)                                                00007060
   12 CONTINUE                                                          00007070
C                                                                       00007080
      ANS1=SUM1/SUM                                                     00007090
C                                                                       00007100
C.........FIND FBR..........                                            00007110
C                                                                       00007120
      TEMP1 = AREA(1)                                                   00007130
      DO 13 I=1,IQ                                                      00007140
         IF (AREA(I) .GT. TEMP1) TEMP1=AREA(I)                          00007150
   13 CONTINUE                                                          00007160
      IR=IQ+1                                                           00007170
      IM=M-IR                                                           00007180
      TEMP2=AREA(IR)                                                    00007190
      DO 14 I=1,IM                                                      00007200
         J=I+IR                                                         00007210
         IF(AREA(J) .GT. TEMP2) TEMP2=AREA(J)                           00007220
   14 CONTINUE                                                          00007230
C                                                                       00007240
      FBR = TEMP1/TEMP2                                                 00007250
C                                                                       00007260
```

```
C                                                                       00007260
      IF(RC(1) .EQ. 0.) GO TO 19                                        00007270
         SGN=RC(1)/ABS(RC(1))                                           00007280
         GO TO 190                                                      00007290
 19        SGN = 1.0                                                    00007300
 190  ANS2=SGN*FBR                                                      00007310
C                                                                       00007320
      RETURN                                                            00007330
      END                                                               00007340
C                                                                       00007350
C                                                                       00007360
C                                                                       00007370
      SUBROUTINE SMOOTH (X,Y,NVAL)                                      00007380
C                                                                       00007390
C                                                                       00007400
C     THIS ROUTINE PERFORMS 3- AND 5-POINT MEDIAN SMOOTHING OF A        00007410
C     DISCRETIZED INPUT SIGNAL.  A 3-POINT HANNING WINDOW SMOOTHER      00007420
C     IS USED IN CONJUNCTION WITH THE MEDIAN SMOOTHERS.                 00007430
C                                                                       00007440
C          ARGUMENTS --                                                 00007450
C               X - ARRAY CONTAINING SIGNAL REPRESENTATION              00007460
C               NVAL - NUMBER OF VALUES STORED IN X                     00007470
C               Y - SMOOTHED ARRAY                                      00007480
C                                                                       00007490
C                                                                       00007500
C                                                                       00007510
C                                                                       00007520
      DIMENSION X(NVAL),Y(NVAL),ZMED(5),Z(402),W(400)                   00007530
      INTEGER POS                                                       00007540
C                                                                       00007550
C     PERFORM THE 3-POINT MEDIAN SMOOTHING                              00007560
C                                                                       00007570
      Y(NVAL)=X(NVAL)                                                   00007580
      Y(1)=X(1)                                                         00007590
      MAX=NVAL-1                                                        00007600
C                                                                       00007610
      DO 1000 J=2,MAX                                                   00007620
         X1=X(J-1)                                                      00007630
         X2=X(J)                                                        00007640
         X3=X(J+1)                                                      00007650
C                                                                       00007660
C           FIND THE MEDIAN OF THE VALUES ABOUT X(J)                    00007670
C                                                                       00007680
         IF(X1.LT.X2) GO TO 200                                         00007690
C                                                                       00007700
            IF(X3.LE.X2) GO TO 230                                      00007710
C                                                                       00007720
            IF(X3.GE.X1) GO TO 240                                      00007730
C                                                                       00007740
            GO TO 220                                                   00007750
C                                                                       00007760
C                                                                       00007770
C                                                                       00007780
200         IF(X3.LE.X1) GO TO 240                                      00007790
C                                                                       00007800
            IF(X3.GE.X2) GO TO 230                                      00007810
C                                                                       00007820
C                                                                       00007830
C                 X3 IS THE MEDIAN                                      00007840
220         Y(J)=X3                                                     00007850
            GO TO 1000                                                  00007860
C                                                                       00007870
C                 X2 IS THE MEDIAN                                      00007880
230         Y(J)=X2                                                     00007890
            GO TO 1000                                                  00007900
C                                                                       00007910
C                 X1 IS THE MEDIAN                                      00007920
240         Y(J)=X1                                                     00007930
```

```
240                     Y(J)=X1                                          00007930
C                                                                        00007940
1000    CONTINUE                                                         00007950
C                                                                        00007960
C                                                                        00007970
C                                                                        00007980
C       PERFORM THE 3-POINT RANNING SMOOTHING                            00007990
C                                                                        00008000
C                                                                        00008010
        Y2=Y(1)                                                          00008020
        Y3=Y2                                                            00008030
C                                                                        00008040
        DO 2000 J=1,MAX                                                  00008050
C                                                                        00008060
             K=J+1                                                       00008070
             Y1=Y2                                                       00008080
             Y2=Y3                                                       00008090
             Y3=Y(K)                                                     00008100
C                                                                        00008110
             Y(J)=.25*(Y1+Y3)+.5*Y2                                      00008120
C                                                                        00008130
C                 PERFORM THE SUBTRACTION OF X AND Y SIGNALS             00008140
C                                                                        00008150
             Z(J+1)=X(J)-Y(J)                                            00008160
2000    CONTINUE                                                         00008170
C                                                                        00008180
        Y(NVAL)=.25*(Y2+Y3)+.5*Y3                                        00008190
        Z(NVAL+1)=X(NVAL)-Y(NVAL)                                        00008200
C                                                                        00008210
C                                                                        00008220
C                                                                        00008230
C       PERFORM A 5-POINT MEDIAN SMOOTHING ON THE X-Y DIFFERENCE         00008240
C                                                                        00008250
C                                                                        00008260
        Z(NVAL+2)=Z(NVAL+1)                                              00008270
        Z(1)=Z(2)                                                        00008280
C                                                                        00008290
        DO 3300 J=2,MAX                                                  00008300
C                                                                        00008310
             K=J+1                                                       00008320
             ZMED(1)=Z(K-2)                                              00008330
        ZMED(2)=Z(K-1)                                                   00008340
                  ZMED(3)=Z(K)                                           00008350
             ZMED(4)=Z(K+1)                                              00008360
             ZMED(5)=Z(K+2)                                              00008370
C                                                                        00008380
             DO 3200 L=1,3                                               00008390
                  M=L+1                                                  00008400
                  POS=L                                                  00008410
C                                                                        00008420
                       DO 3150 N=M,5                                     00008430
3150                   IF(ZMED(POS).GT.ZMED(N)) POS=N                    00008440
C                                                                        00008450
                  IF(POS.EQ.L) GO TO 3200                                00008460
C                                                                        00008470
                       TEMP=ZMED(L)                                      00008480
                       ZMED(L)=ZMED(POS)                                 00008490
                       ZMED(POS)=TEMP                                    00008500
3200             CONTINUE                                                00008510
C                                                                        00008520
             W(J)=ZMED(3)                                                00008530
3300    CONTINUE                                                         00008540
C                                                                        00008550
        W(1)=Z(2)                                                        00008560
        W(NVAL)=Z(NVAL+1)                                                00008570
C                                                                        00008580
```

```
C
C          PERFORM A 3-POINT HANNING SMOOTHING ON THE DIFFERENCE SIGNAL   00008590
C                                                                         00008600
           W2=W(1)                                                        00008610
           W3=W2                                                          00008620
C                                                                         00008630
           DO 4000 J=1,MAX                                                00008640
C                                                                         00008650
                W1=W2                                                     00008660
                W2=W3                                                     00008670
                W3=W(J+1)                                                 00008680
C                                                                         00008690
                Y(J)=.25*(W1+W3)+.5*W2+Y(J)                               00008700
4000       CONTINUE                                                       00008710
C                                                                         00008720
C                                                                         00008730
           Y(NVAL)=.25*(W2+W3)+.5*W3+Y(NVAL)                              00008740
C                                                                         00008750
C                                                                         00008760
           RETURN                                                         00008770
           END                                                            00008780
C                                                                         00008790
C                                                                         00008800
C                                                                         00008810
C                                                                         00008820
C                                                                         00008830
           SUBROUTINE QUANTZ(X,Y,NVAL,XMAX,XMIN)                          00008840
C                                                                         00008850
C                                                                         00008860
C          THIS ROUTINE PERFORMS A QUANTIZATION OF THE VALUES             00008870
C          IN AN ARRAY.  THE RESULTING VALUES ARE IN THE RANGE            00008880
C          1 - 100.                                                       00008890
C                                                                         00008900
C          ARGUMENTS --                                                   00008910
C               X - ARRAY TO BE QUANTIZED                                 00008920
C               Y - QUANTIZED ARRAY                                       00008930
C               XMAX - LARGEST X                                          00008940
C               XMIN - SMALLEST X                                         00008950
C                                                                         00008960
C                                                                         00008970
C                                                                         00008980
           DIMENSION X(NVAL), Y(NVAL)                                     00008990
           XMAX=X(1)                                                      00009000
           XMIN=XMAX                                                      00009010
C                                                                         00009020
           DO 5000 J=2,NVAL                                               00009030
C                                                                         00009040
                IF (XMAX.LT.X(J)) XMAX=X(J)                               00009050
                IF (XMIN.GT.X(J)) XMIN=X(J)                               00009060
5000       CONTINUE                                                       00009070
C                                                                         00009080
           XDIFF=XMAX-XMIN                                                00009090
C                                                                         00009100
           DO 5100 J=1,NVAL                                               00009110
C                                                                         00009120
                Y(J)=1.+AINT((X(J)-XMIN)/XDIFF*99.+1.E-7)                 00009130
5100       CONTINUE                                                       00009140
C                                                                         00009150
C                                                                         00009160
           RETURN                                                         00009170
           END                                                            00009180
C                                                                         00009190
C                                                                         00009200
           SUBROUTINE SPECT(X,HLR)                                        00009210
C                                                                         00009220
           DIMENSION X(1)                                                 00009230
C                                                                         00009240
           RMS1=0.                                                        00009250
```

```
C                                                                       00009240
      RMS1=0.                                                           00009250
      DO 1 I=1,8                                                        00009260
1       RMS1=RMS1+X(I)                                                  00009270
C                                                                       00009280
      RMS2=0.                                                           00009290
      DO 2 I=49,56                                                      00009300
2       RMS2=X(I)+RMS2                                                  00009310
C                                                                       00009320
      HLR=RMS2/RMS1                                                     00009330
      RETURN                                                            00009340
      END                                                               00009350
C                                                                       00009360
C                                                                       00009370
C                                                                       00009380
C                                                                       00009390
C                                                                       00009400
C                                                                       00009410
C                                                                       00009420
C                                                                       00009430
      SUBROUTINE DIGIT(EV,BTR,STATE)                                    00009440
C                                                                       00009450
C     THIS ROUTINE EVALUATES THE SEQUENCE OF PHONEMES IN                00009460
C     A SPOKEN DIGIT IN ORDER TO DETERMINE WHICH DIGIT WAS              00009470
C     SPOKEN.  THE DECISION PROCESS IS TABLE DRIVEN.  IT WILL           00009480
C     IDENTIFY A DIGIT ONLY IF AN EXACT PHONEME SEQUENCE                00009490
C     IS GIVEN.  AN ADDITIONAL TEST IS REQUIRED TO DIFFERENTIATE        00009500
C     BETWEEN THE DIGITS 5 AND 9.                                       00009510
C                                                                       00009520
C     PARAMETERS --                                                     00009530
C                                                                       00009540
C          EV - CONTAINS THE SEQUENCE OF PHONEMES(1 - FV, 2 - MV,       00009550
C              3 - BV, 4 - VL, 5 - NVL, 6 - UNDEFINED,                  00009560
C              7 - END OF SEQUENCE)                                     00009570
C          BTR - USED TO DISTINGUISH BETWEEN 5 AND 9                    00009580
C          STATE - THE NODES OF THE DECISION TREE; THE FINAL            00009590
C               VALUE IS IN THE RANGE 0 - 9 IF A DIGIT WAS              00009600
C               RECOGNIZED OR IT IS 10, INDICATING THAT THE             00009610
C               DIGIT WAS UNRECOGNIZABLE.                               00009620
C                                                                       00009630
C                                                                       00009640
      INTEGER EV,DMATRX,START,STATE,PTR                                 00009650
      DIMENSION EV(7),DMATRX(13,6),BTR(1)                               00009660
C                                                                       00009670
     DATA START/13/,LP/6/, DMATRX/                                      00009680
     $  7,9,-99,11,8*-99,1,                                             00009690
     $  -99,12,10,9*-99,4,                                              00009700
     $  8,3*-99,6,7*-99,3,                                              00009710
     $  5,11*-99,2,                                                     00009720
     $  13*-99,                                                         00009730
     $  -6,4*-99,0,-7,-2,-3,-4,-5,-1,-99 /                              00009740
C                                                                       00009750
C                                                                       00009760
C                                                                       00009770
C                                                                       00009780
      WRITE(LP,10)((DMATRX(J,K),K=1,6),J=1,13)                          00009790
10    FORMAT(////(11X,6I10))                                            00009800
      PTR=1                                                             00009810
      STATE=START                                                       00009820
C                                                                       00009830
100   INPUT=EV(PTR)                                                     00009840
      KSTATE=STATE                                                      00009850
      STATE=DMATRX(STATE,INPUT)                                         00009860
      WRITE(6,999) KSTATE,INPUT,STATE                                   00009870
999   FORMAT(10X,I6,5X,I6,5X,E15.7)                                     00009880
      IF(STATE.LT.1) GO TO 200                                          00009890
          PTR=PTR+1                                                     00009900
          GO TO 100                                                     00009910
C
```

```
            PTR=PTR+1                                                    00009900
            GO TO 100                                                    00009910
C                                                                        00009920
C                                                                        00009930
200      STATE=-STATE                                                    00009940
C                                                                        00009950
C                                                                        00009960
C        DISTINGUISH BETWEEN 5 AND 9                                     00009970
C                                                                        00009980
C        AV=(BTR(1)+BTR(2)+BTR(3))/3.                                    00009990
C        IF(AV.GT. .15) STATE=9                                          00010000
C                                                                        00010010
C                                                                        00010020
         RETURN                                                          00010030
         END                                                             00010040
```

```
      SUBROUTINE PLOTER(NFRAME,ARRAY,ITITLE)                             00000820
      DIMENSION ARRAY(1),ITITLE(1)                                       00000830
      DIMENSION X(400)                                                   00000840
      DIMENSION B(400,1)                                                 00000850
      INTEGER IMAGE(5151),ICHAR(10)                                      00000860
      REAL RANGE(4)                                                      00000870
      DATA ICHAR(1)/1H0/,RANGE/4*0./                                     00000880
C                                                                        00000890
      CALL UGETIO(1,NIN,NOUT)                                            00000900
C                                                                        00000910
C                                                                        00000920
      DO 21  I=1 , NFRAME                                                00000930
      B(I,1) = ARRAY(I)                                                  00000940
      X(I) = FLOAT(I)                                                    00000950
   21 CONTINUE                                                           00000960
C                                                                        00000970
      INC=1                                                              00000980
      MM=1                                                               00000990
      IY = NFRAME                                                        00001000
      IOPT=1                                                             00001010
C                                                                        00001020
      CALL USPLT(X,B,IY,NFRAME,MM,INC,ITITLE,RANGE,ICHAR,IOPT,IMAGE,IER)00001030
C                                                                        00001040
      RETURN                                                             00001050
      END                                                                00001060
```

```
      DIMENSION IX(25000),S(2500)),SI(512),HAM(512),RC(20)                00000010
C     DIMENSION ITITLE(144),JTITLE(144),KTITLE(144)                       00000020
      DIMENSION ARRAY(10,40),BRRAY(10,40)                                 00000030
      DIMENSION FTR(400),CT2(400),BT2(400),WORK(400),OMT(400)             00000040
C     READ(5,105) ITITLE,JTITLE,KTITLE                                    00000050
  105 FORMAT(72A1)                                                        00000060
      READ(5,106) PRE                                                     00000070
  106 FORMAT(F6.2)                                                        00000080
      WRITE(6,107) PRE                                                    00000090
  107 FORMAT (//,10X,'SECOND ORDER PRE-EMPHASIS (GAIN = ',F6.2,')')       00000100
      NPS = 128                                                           00000110
      SCALE=.02                                                           00000120
      PI=3.141593                                                         00000130
      CONST=PI/75.0                                                       00000140
      DO 8 I=1,150                                                        00000150
      HAM(I)=.54-.46*COS((I-1)*CONST)                                     00000160
    8 CONTINUE                                                            00000170
      IDIGT=-1                                                            00000180
C.....READ THE INPUT DATA......                                           00000190
C                                                                         00000200
      IFL=2                                                               00000210
      ISTART=1                                                            00000220
    1 READ(IFL,101) IJUNK,ISTOP                                           00000230
  101     FORMAT(2I6)                                                     00000240
      IDIGT=IDIGT+1                                                       00000250
      WRITE(6,102) IDIGT,ISTOP                                            00000260
  102   FORMAT(10X,'DIGIT -- ',I4,10X,'NUMBER OF POINTS',I6)               00000270
      READ(IFL,104) (IX(K),K=1,ISTOP)                                     00000280
  104   FORMAT (20I4)                                                     00000290
      IDIF = ( ISTOP - ISTART + 1 )                                       00000300
      NF=IDIF/NPS                                                         00000310
      IF (NF*NPS.NE.IDIF) NF=NF+1                                         00000320
      NDIM=NF*NPS+ISTART                                                  00000330
      IK=ISTOP+1                                                          00000340
      DO 3 I=IK,NDIM                                                      00000350
    3 IX(I)=0                                                             00000360
      J=1                                                                 00000370
      DO 6 I=ISTART,NDIM                                                  00000380
           J=J+1                                                          00000390
           S(J)=FLOAT(IX(I))*SCALE                                        00000400
    6 CONTINUE                                                            00000410
      NPW=150                                                             00000420
      NPS=128                                                             00000430
      DO 5 I=1,NF                                                         00000440
      DO 4 J=1,NPW                                                        00000450
      K=J+(I-1)*NPS                                                       00000460
      SI(J)=S(K)                                                          00000470
    4 CONTINUE                                                            00000480
      SAVE=S(K)                                                           00000490
      M1=14                                                               00000500
      IP1=9                                                               00000510
      K1=0                                                                00000520
      K2=128                                                              00000530
      CALL DWINDF(SI,NPW,10,PRE,XPLE,HAM)                                 00000540
      DO 10 IM=1,128                                                      00000550
      SI(IM)=SI(IM+11)                                                    00000560
   10 CONTINUE                                                            00000570
      XPRE=SAVE                                                           00000580
```

```
C       CALL SLP(NPS,SI,M1,IP1,K1,K2,ALPHA,RC)                          00000590
        CALL AUTC(NPS,SI,M1,A,ALPHA,RC)                                 00000592
        CALL PRTR(RC,14,4,10,ANS1,ANS2,ANS3)                            00000600
        PTR(I)=ANS1                                                     00000610
        CTR(I)=ANS2                                                     00000620
        BTR(I)=ANS3                                                     00000630
  5     CONTINUE                                                        00000640
        CALL SMOOTH(PTR,WORK,NF)                                        00000650
        CALL QUANTZ(WORK,QNT,NF,XMAX,XMIN)                              00000660
        CALL PLOTER(NF,QNT,JTITLE)                                      00000670
        CALL SMOOTH(CTR,WORK,NF)                                        00000680
         CALL QUANTZ(WORK,QNT,NF,XMAX,XMIN)                             00000690
        CALL PLOTER(NF,QNT,JTITLE)                                      00000700
        CALL SMOOTH(BTR,WORK,NF)                                        00000710
        CALL QUANTZ(WORK,QNT,NF,XMAX,XMIN)                              00000720
        CALL PLOTER(NF,QNT,KTITLE)                                      00000730
        ARRAY(1,IGT+1,1)=FLOAT(NF)                                      00000740
        DO 70 I=1,NF                                                    00000750
  70    ARRAY(1,IGT+1,I+1)=QNT(I)                                       00000760
        IF(ICIGT .LT. 9) GO TO 1                                        00000770
        WRITE(3,150) ARRAY                                              00000780
 150    FORMAT(10E12.4)                                                 00000790
        STOP                                                            00000800
        END                                                             00000810


        SUBROUTINE AUTC(N,X,M1,A,ALPHA,RC)                              00005060
        DIMENSION  X(1),A(1),RC(1)                                      00005070
        DIMENSION  R(21),B(21)                                          00005080
        MP=M1+1                                                         00005090
        DO 100 K=1,MP                                                   00005100
        R(K)=0.0                                                        00005110
        NK=N-K+1                                                        00005120
        DO 100 NP=1,NK                                                  00005130
        NPK=NP+K-1                                                      00005140
 100    R(K)=R(K)+X(NP)*X(NPK)                                          00005150
        ALPHA=R(1)                                                      00005160
        RC(1)=-R(2)/R(1)                                                00005170
        A(1)=1.0                                                        00005180
        A(2)=RC(1)                                                      00005190
        ALPHA=ALPHA-RC(1)*RC(1)*ALPHA                                   00005200
        MF=M1                                                           00005210
        DO 400 MINC=2,MF                                                00005220
        M=MINC-1                                                        00005230
        DO 200 J=1,MINC                                                 00005240
        JB=MINC-J+1                                                     00005250
 200    B(J)=A(JB)                                                      00005260
        M=M+1                                                           00005270
        S=0.0                                                           00005280
        DO 300 IP=1,M                                                   00005290
        MIP=M-IP+2                                                      00005300
 300    S=S+B(MIP)*A(IP)                                                00005310
        RC(M)=-S/ALPHA                                                  00005320
        DO 350 IP=2,M                                                   00005330
 350    A(IP)=A(IP)+RC(M)*B(IP-1)                                       00005340
        A(M+1)=RC(M)                                                    00005350
        ALPHA=ALPHA-RC(M)*RC(M)*ALPHA                                   00005360
 400    CONTINUE                                                        00005370
        RETURN                                                          00005380
        END                                                             00005390
```

```
      DIMENSION ARRAY(10,40),BRRAY(10,40),TABLE(10,10)                  00000010
      COMMON ARRAY,BRRAY                                                00000020
C     READ(10,100) ARRAY                                                00000022
      IFL=10                                                            00000030
      DO 5 IP=1,5                                                       00000040
      READ(IFL,100) ARRAY                                               00000050
  100 FORMAT(40E12.4)                                                   00000060
      WRITE(6,160) IP                                                   00000080
  160 FORMAT(1H1,10X,'SET NUMBER : ',I4)                                00000120
      WRITE(6,150)                                                      00000130
      DO 1 ITER=1,10                                                    00000140
C     TABLE(ITER,ITER)=1.0                                              00000150
      NEXT=ITER                                                         00000160
      IF(NEXT .GT. 10) GO TO 2                                          00000165
      DO 2 JTER=NEXT,10                                                 00000170
      CALL CORR(ITER,JTER,COEFF)                                        00000180
      TABLE(ITER,JTER)=COEFF                                            00000190
      TABLE(JTER,ITER)=COEFF                                            00000200
    2 CONTINUE                                                          00000210
    1 CONTINUE                                                          00000220
      WRITE(6,200) TABLE                                                00000230
  200 FORMAT(1H0,10F10.3)                                               00000240
  150 FORMAT(40X,'RMS CORRELATION ',/,3X,                               00000250
     $   '   ZERO       ONE       TWO     THREE      FOUR      FIVE    00000260
     $   '    SIX     SEVEN     EIGHT      NINE')                       00000270
      IFL=IFL+1                                                         00000280
    9 CONTINUE                                                          00000290
      STOP                                                              00000300
      END                                                               00000310

      SUBROUTINE CORR(ITER,JTER,COEFF)                                  00000320
      DIMENSION ARRAY(10,40),BRRAY(10,40)                               00000330
      DIMENSION X(40),Y(40)                                             00000340
      COMMON ARRAY,BRRAY                                                00000350
      N1=INT(ARRAY(ITER,1))                                             00000360
      N2=INT(ARRAY(JTER,1))                                             00000370
      N=MAX0(N1,N2)                                                     00000380
      DO 10 I=1,N                                                       00000390
      X(I)=ARRAY(ITER,I+1)                                              00000400
      Y(I)=ARRAY(JTER,I+1)                                              00000410
   10 CONTINUE                                                          00000420
      XVAR=0.0                                                          00000430
      YVAR=0.0                                                          00000440
      XMEAN=0.0                                                         00000442
      YMEAN=0.0                                                         00000444
      XYVAR=0.0                                                         00000450
      DO 20 I=1,N                                                       00000460
      XMEAN=XMEAN+X(I)                                                  00000470
      YMEAN=YMEAN+Y(I)                                                  00000480
   20 CONTINUE                                                          00000490
      XMEAN=XMEAN/FLOAT(N)                                              00000500
      YMEAN=YMEAN/FLOAT(N)                                              00000510
      DO 30 I=1,N                                                       00000520
      RX=X(I)-XMEAN                                                     00000530
      RY=Y(I)-YMEAN                                                     00000540
      XVAR=XVAR+RX**2                                                   00000550
      YVAR=YVAR+RY**2                                                   00000560
      XYVAR=XYVAR+RY*RX                                                 00000570
   30 CONTINUE                                                          00000580
      VAR=XVAR*YVAR                                                     00000590
      COEFF=XYVAR/SQRT(VAR)                                             00000600
      RETURN                                                            00000610
      END                                                               00000620
```

VITA[2]

Ahmed Mahmoud Milyani

Candidate for the Degree of

Doctor of Philosophy

Thesis: A ROBUST PHONETIC DIGIT RECOGNITION SYSTEM FOR DIGITS SPOKEN IN AMERICAN ENGLISH AND ARABIC

Major Field: Electrical Engineering

Biographical:

Personal Data: Born in Mecca, Saudi Arabia, January 28, 1940, the son of Mr. Mahmoud M. Milyani and Mrs. Nakeia Y. Zawawi.

Education: Graduated from Ali Abdul-Latief High School, Cairo, Egypt, in August, 1958; received the H.N.C. degree in Electrical and Electronic Engineering from Cambridge College of Arts and Technology, Cambridge, England, in August, 1969; attended Part II courses for the Councel of Engineering Institution Fellowship at Cambridge College of Arts and Technology, England, from January, 1970 to July, 1970. received the Master of Science degree in Electrical Engineering from Oklahoma State University, Stillwater, Oklahoma, in May, 1976; completed requirements for the Doctor of Philosophy degree at Oklahoma State University, Stillwater, Oklahoma, in July, 1981.

Professional Experience: Electronics and test engineer, PYE TVT Ltd., Cambridge, England, September, 1967 to January, 1970; Electronics and Communication Technician, RAYTHEON Service Company, Saudi Arabia Branch, Jeddah, Saudi Arabia, September, 1970 to June, 1971; Instructor (full-time faculty member), University of Petroleum and Minerals, Dhahran, Saudi Arabia, July, 1971 to June, 1974.

Professional Organizations: Full Member of ETA KAPPA NU Association, and The Institute of Electrical and Electronics Engineers.