SPEECH SYNTHESIS IN REAL-TIME

BY MICROPROCESSOR CONTROL

by

John Peter Pinnell, B.Eng.

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of
Master of Engineering.

Department of Electrical Engineering,
McGill University,
Montreal, Canada.

March, 1979.

SPEECH SYNTHESIS IN REAL-TIME

BY MICROPROCESSOR CONTROL

by

John Peter Pinnell, B.Eng.

Department of Electrical Engineering

McGill University,

Montreal, Canada.

March, 1979.

ABSTRACT

A system for speech synthesis, based on microprocessor control in real-time, was developed for experimental work which is small enough to form the basis of a talking terminal. The equipment accepts as input a phonetic string composed of standard ASCII characters and converts these into intermittent, continuous or connected speech. Real-time operation permits the use of less than 3000 bytes of memory. Features are provided to vary the fundamental frequency during synthesis, initiate whispering, or obtain a parameter listing for analysis. Stressed speech is obtained through the use of lower case characters. The equipment is simple to use and produces intelligible speech.

Acoustic, numeric and visual methods used to evaluate performance are described. Perceptual confusion matrices are provided which illustrate areas where improvement in hardware could be made. The computer program for synthesis is given along with explanation and manner of use.

RESUME

Un système de synthèse de la parole, reposant sur une commande en temps réel par microprocesseur, a été conçu à des fins expérimentales et est de taille suffisamment petite pour servir de point de départ à un terminal parlant. En entrée, le dispositif accepte une séquence phonétique de caractères normalisés ASCII qu'il convertit en une lecture intermittente, continue ou cohérente. Son fonctionnement en temps réel permet l'emploi de moins de 3000 octects de mémoire. Il est possible de faire varier la fréquence fondamentale au cours de la synthèse, de réaliser une lecture à voix basse ou de produire un listage de paramètres en vue de leur analyse.

Il est possible d'obtenir un rendu vocal avec l'accent tonique grâce à l'usage, en entrée, de caractères en minuscules. L'équipment est simple d'emploi et fournit une lecture compréhensible.

On décrit également des méthods acoustiques, numériques et visuelles employées pour évaluer la qualité de fonctionnement du système. On présente des matrices reflètant les confusions de perception, permettant d'illustrer les domaines dans lesquels une amélioration du matériel serait possible. On décrit enfin le programme machine réalisant la synthèse ainsi que des explications relatives à son mode d'utilisation.

PREFACE

This thesis presents a description of the development of a self-contained speech synthesis system capable of use either as a talking peripheral or as a tool for perceptual studies. Work on this thesis commenced January 1978 while the author was enrolled in a speech communications course (304-689B) at McGill. Subsequently, Professor Douglas O'Shaughnessy agreed to become thesis director.

Software development was undertaken at the McGill University computer facilities on one of their IBM 370 systems. Acoustical analysis of synthetic speech was made at L'Institute National de la Recherche Scientifique-Télécommunications using specially designed programs for spectral analysis and plotting.

Sections of the synthesis strategy are based on Holmes (1964) and Nooteboom (1973). Formant frequencies used in synthesis were originally obtained from the works of Rabiner (1968) and Klatt (1977) but it was found that substantial modifications were required for optimum results with the synthesizer in use.

In the author's opinion, the configuration of the system and the software developed for it constitute new and original work. The scope of this investigation is primarily concerned with the development of hardware, system software, as well as presentation of some experimental results. Nevertheless, some background on speech synthesis techniques and brief mention of major contributions to this field seem desirable and

and are covered in the first three chapters. The last three chapters describe the hardware and software of the developmental system and the results of perceptual experiments to determine effectiveness of the synthesis strategy.

It is hoped that this thesis will encourage further interest in a subject which will undoubtedly become of major significance in man-machine interfaces of the future. A recent book by Rabiner and Schafer (1978) is an excellent introduction to this field.

The author wishes to acknowledge the very considerable support and assistance given by Professor Douglas O'Shaughnessy in the preparation of this work. Mention should be made of the various members of my family and friends who patiently took part in perceptual experiments. Financial support from a 1978 McGill Graduate Faculty Summer Research Fellowship was greatly appreciated.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER I

INTRODUCTION

1.1 Background

Originally speech related research was directed to narrow band encoding and decoding using vocoders. However, by the late 1960's or early 1970's, the telecommunications industry found it less expensive to expand bandwidth and interest in narrow band communications declined. Such use is now limited to a few satellite communications systems and methods of scrambling speech.

Despite a decline in the need for narrow band transmission equipment, there are other very important reasons for the study of speech synthesis. For example, speech synthesis has become a very valuable tool for research into phonology and perception. In the field of speech recognition, synthesis is used with great effect to determine those features which carry the greatest information. Currently, computer generated speech is forging powerful links for future man-machine communications.

Some of the more recent applications of speech synthesis are an automated weather bureau for general aviation (Thordarson 1977) and several military applications (Beek 1977). Earlier work includes a cockpit man-machine interface for air-ground communications (Hilborn 1972), a reading machine for the blind (Allen 1973), and computer-generated wiring instructions for telephone exchanges (Flanagan 1972).

The possibilities of man-machine interface are enormous and will eventually dictate the need for much work of a fundamental nature. At the present time, there seems to be a real need for an inexpensive system for speech synthesis which has inherent flexibility for use as a research tool. This is the reason and motivation in undertaking the development of a practical speech synthesis system.

It is appropriate at this time to review some of the characteristics, categories and perceptual concepts of speech. This will provide a useful introduction and background for developing subsequent chapters.

1.2 The Characteristics of Speech

Speech is composed of voiced and unvoiced sound in conjunction with periods of silence. Other important characteristics which are used in analysis are the fundamental frequency, the formant frequencies and loudness.

Voiced sounds are produced when the vocal cords at the opening of the larynx form a flexible obstruction to the air flow. Air forced through the larynx by the lungs cause these cords to vibrate and chop the air flow at a repetition rate of between 50 and 500 Hz. This rate is known as the fundamental frequency. Nominally, a male will have a fundamental frequency of 130 Hz and a female speaker, approximately 200 Hz.

The volume velocity of air above the vocal cords when plotted against time, produces a triangular shaped waveform (Figure 1) that decreases 12 dB/octave in amplitude (with frequency ω as $1/\omega^2$). These bursts of air escape through the vocal tract which acts as a resonator and determines their spectra. The type of resonances are dependent on the vocal tract cavity and constructions formed by the tongue and the lips. Concentrations of energy in the spectrum due to resonances are clearly visible as dark bands in spectrograms (Figure 2) and are called formants. The first three are typically located at 500, 1500 and 2500 Hz. Finally due to radiation from the mouth, there is a 6 dB/octave rise in level for frequencies up to 5000 Hz.

Unvoiced sounds are caused by air flow through an orifice such as the teeth or a narrowing in the vocal tract. A constriction of this nature produces a Bernoulli pressure that causes a hissing sound. Pops and clicks caused by the tongue, teeth or lips are also unvoiced sounds. Although the entire vocal tract shapes the spectrum of these sounds, effectively it is only that part of the tract after the point of narrowing. The sound spectrum can also be shaped by a protrusion of the lips especially when the narrowing is across the teeth. (Sounds as S in seat, or SH in sheet.) In general, the spectra of unvoiced sounds are above 1500 Hz and are non-periodic.

Formants play a very important role in the production of vowels. If vowels are plotted with respect to the first and second for-

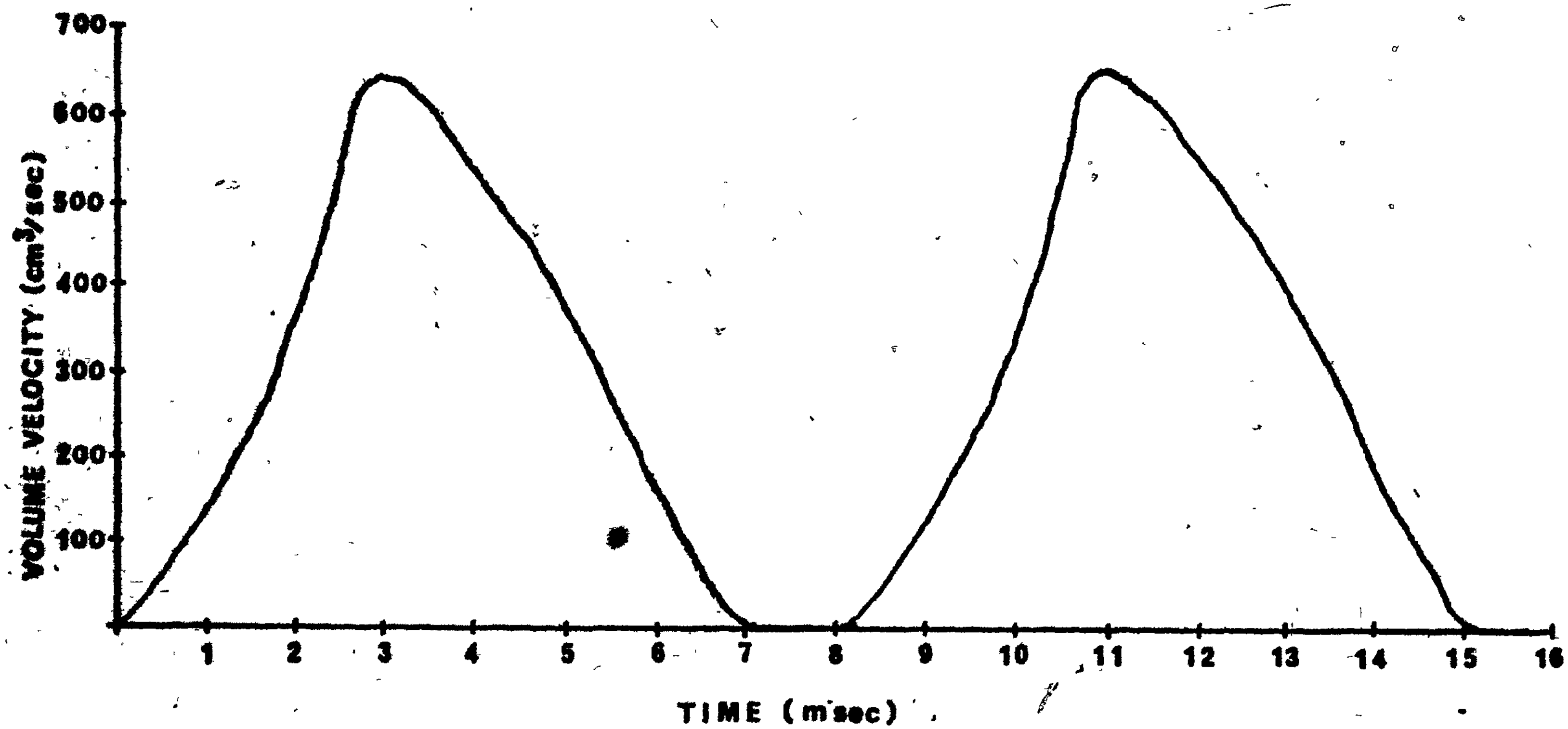


FIGURE 1 A TYPICAL GLOTTAL WAVEFORM

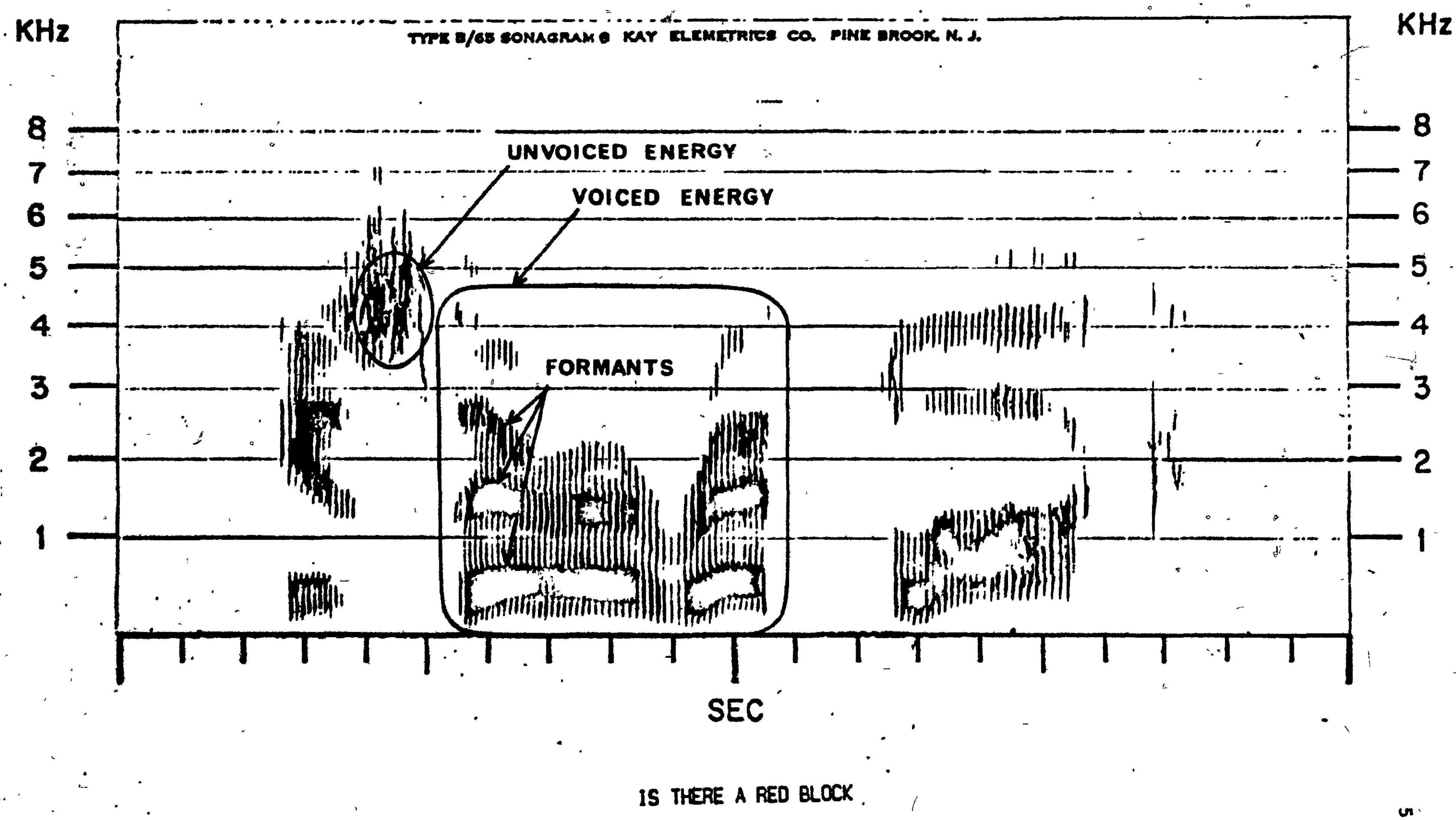


FIGURE 2 SPEECH SPECTROGRAM

mant frequencies, they will generate what is known as a vowel triangle (Figure 3). The lax or lenis vowels (ε, I, U) are characterized by little movement from the neutral position of the vocal tract and fall close to the center of the vowel triangle. Tense or fortis vowels occur further from the center. A rough correspondence exists between the first two formant frequencies and the positioning of the tongue (i.e., low F1 tongue high; high F1 tongue low, low F2 tongue back and high F2 tongue forward). Although formants are a characteristic of consonants as well as vowels, they are not generally as well defined. The first three formant frequencies are normally sufficient for individual recognition of either vowels or consonants.

Loudness of speech plays a part in determining intonation, rhythm and to some extent cues phonetic recognition. Two different sounds, for example /i/ and /a/ at the same intensity, will yield different loudness. For short periods of time (less than half a second) an increase in the duration of a sound causes an increase in loudness and is particularly noticeable when dealing with plosives. Unfortunately, the loudness of speech has not been well correlated with intensity and this results in difficulty of measurement.

1.3 Articulation of Linguistic Units

The basic linguistic unit with its own distinguishable sound is called a phoneme. These do not distinguish any concept or object by

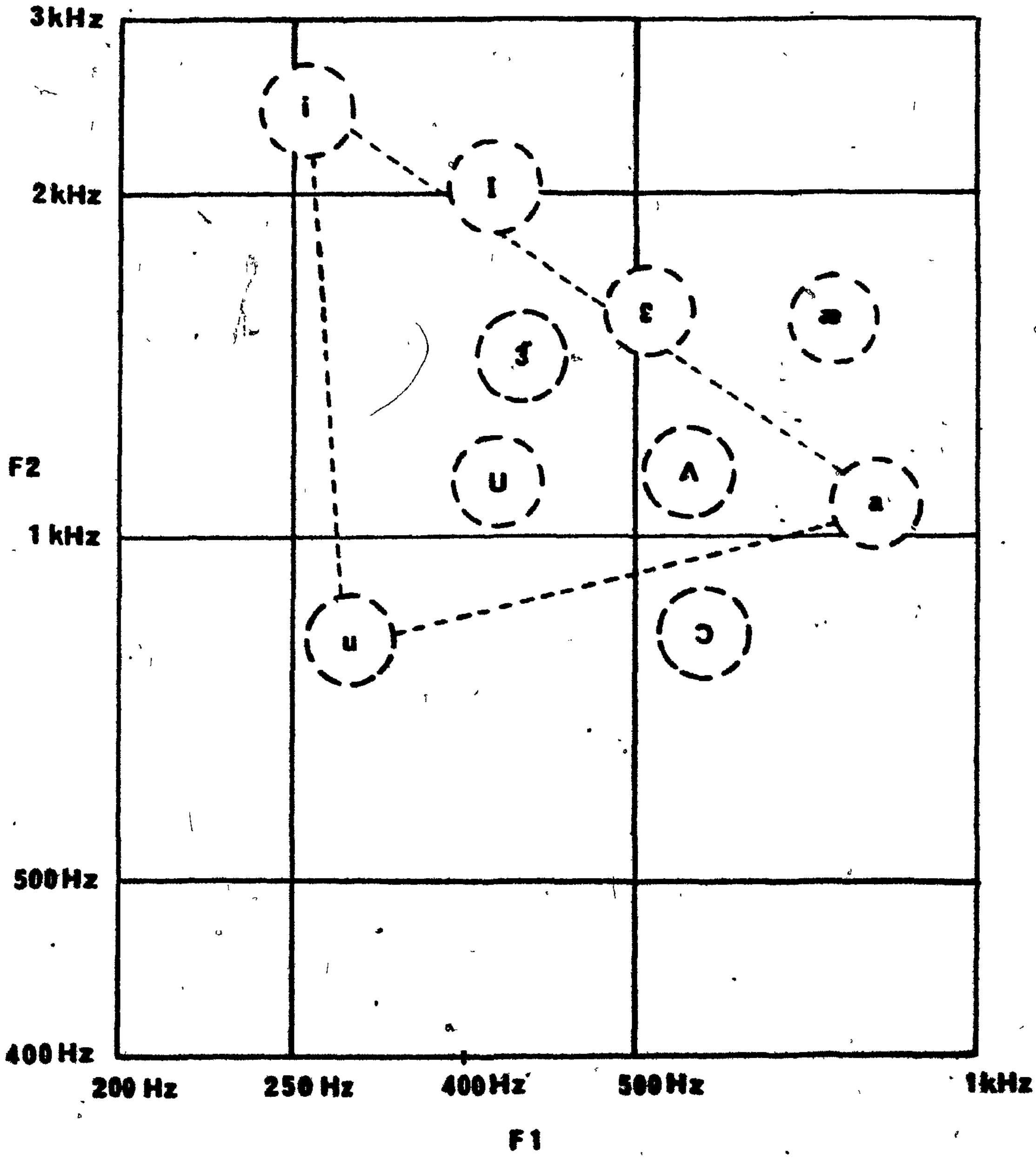


FIGURE 3 THE VOWEL TRIANGLE

themselves. Phonemes can be categorized as segmental as opposed to suprasegmentals (prosodemes) which carry prosodic information such as stress, pitch or pause. Table 1 lists English language segmentals and examples according to the International Phonetic Alphabet (IPA) and the Arpabet.

Speech sounds are generally classed in accordance to the extent the vocal tract is closed. Vowels, for example, are produced with little constriction in the vocal tract. Consonants are characterized by a definite constriction in the air stream.

Vowels can be described in terms of the shape of the tongue, the position of the highest part of the tongue (front, central or back), the height of the tongue, the tenseness of the muscles of the tongue, the position of the lips and the degree to which the nasal passages are open. Vowels are always voiced in the English language. Figure 4 illustrates the classification of vowels with respect to tongue position and tenseness. A diphthong is a special case of a vowel which involves the smooth but rapid transition from one vowel position to another.

Consonants can be related to laryngeal activity (voiced or voiceless sounds), amount of tension (tense or lax articulators), position of maximum constriction in the vocal tract (point of articulation), and the sound producing mechanism (manner of articulation).

Because of a constriction in the air stream, consonants always contain unvoiced energy but not necessarily voiced energy. The presence of voicing is in fact a main factor in differentiating /b, d or g /from /p, t or k /. Otherwise, these are identical in place and manner of articulation (Figure 5).

Plosives are recognizable for the use of tension although other consonants may have this type of articulation to a lesser degree. In general, consonants produced by strong articulation are classed as "fortis" whereas those with less articulation are designated "lenis". Fortis consonants tend to be voiceless and aspirated whereas lenis consonants tend to be voiced and unaspirated.

Points of articulation are defined in terms of the upper and lower articulators (Dresher 1972). It should be noted, however, that it is common to find intermediate points of articulation. The classifications given below are sufficiently precise to accurately describe this function.

LABIAL	**Constriction formed by**
Bilabial	upper and lower lips
Labiodental	upper teeth and lower lips
APICAL	
Dental	upper teeth and apex of tongue
Alveolar	alveolae and apex of tongue
Retroflex	apex of tongue turned back such that underside of tip is near the palate.

TABLE 1. ENGLISH SEGMENTALS

Symbol Used	ARPABET	IPA Symbol	Typical Word
Vowels			
IY	IY	i	beet
IH	IH	ɪ	bit
EH	EH	ɛ	bet
AE	AE	æ	bat
AA	AA	a	box
AH	AH	ʌ	but
AO	AO	ɔ	bought
UW	UW	u	boot
UH	UH	ʊ	book
ER	ER	ɝ	bird
Liquids			
WW	W	w	wet
WH	WH	ʍ	which
YY	Y	j	yet
RR	R	r	rent
LL	L	l	let
Fricatives			
FF	F	f	fin
VV	V	v	vat
TH	TH	θ	thin
TE	DH	ð	that
SS	S	s	sat
ZZ	Z	z	zoo
SH	SH	ʃ	shin
ZH	ZH	ʒ	azure
Nasals			
MM	M	m	mat
NN	N	n	nap
NG	NG	ŋ	sing
Aspirant			
HH	H	h	help
Stops			
BB	B	b	bat
DD	D	d	dog
GG	G	g	got
PP	P	p	pot
TT	T	t	tot
KK	K	k	cot
Diphthongs			
AOIY	OY	ɔɪ	boy
AAIY	AY	aɪ	bite
EHIY	EY	eɪ	bait
AOUW	OW	oʊ	boat
AAUW	AW	aʊ	bout
Affricates			
CH	CH	tʃ	chin
JJ	JH	dʒ	jin

Height of Tongue	Tension	Position of Tongue: Front	Center	Back
High	tense	i		u
	lax	I		U
Mid	tense	e		
	lax	ɛ	ʌ	ɔ
Low		æ		a

Figure 4. Classification of Vowels in terms of amount of tension, position, and height of tongue.

Consonants	Voiced / Unvoiced	Place of Articulation: Labial	Dental	Alveolar	Palatal	Velar
Fricatives	Voiced	v	ð	z	ž	
	Unvoiced	f	θ	s	š	h
Stops	Voiced	b		d	ǰ	g
	Unvoiced	p		t	č	k
Lateral				l		
Glides		w		r	j	
Nasals		m		n		ŋ

Figure 5. Classification of Consonants in terms of Voicing, Place and Manner of Articulation.

FRONTAL	**Constriction formed by**
Alveopalatal	**The alveolae in the far front of the palate with the front of the tongue.**
Prepalatal	**the front of the palate and the front of the tongue.**
DORSAL	
Palatal	**the back of palate with the back of the tongue**
Velar	**the velum and the back of the tongue**
Uvular	**the extreme back of the velum or uvula and the back of the tongue.**
GLOTTAL	**the vocal cords.**

The manner of articulation describes the extent of constriction in the vocal tract. Fricatives, plosives (or stops), laterals, glides and nasals form the various categories.

Fricatives are produced by forcing air through a narrow opening resulting in a rushing sound (frication). Fricatives can differ both in terms of place of articulation and in voicing.

Plosives are formed by completely blocking the air flow temporarily. This causes a short period of silence, roughly 100 msec followed by a burst of noise as the air rushes out from the narrow opening. Depending on the time it takes for the onset of voicing (VOT-voice onset time), the plosive is determined to be voiced or unvoiced. Generally, the burst of noise is of longer duration in unvoiced plosives.

Semi-vowels which consist of laterals and glides are consonants with vowel-like properties. These are always followed or preceded by a vowel and are produced by first positioning the vocal tract in a vowel-like manner and then rapidly changing it to the position required by the following vowel.

Nasals are produced in opening the nasal passages by lowering the soft palate and in closing the oral passages at different points of articulation.

1.4 Perception of Speech

Perception is the process by which the brain interprets audio information received through the aural process. Design of an appropriate set of rules for speech synthesis must include recognition of the influence of perception.

A number of models for speech perception have been reviewed by Cooper (1972). Perhaps the most eloquent is that developed by Stevens and Halle (1967). This model (Figure 6) postulates that acoustic information undergoes spectral analysis, pitch and acoustic feature extraction. Spectral and pitch information are temporarily stored over a period of several syllables. A preliminary analysis is made on extracted acoustic features and results in the production of phonetic segments and features used by a control section. The control function has access to the phone-

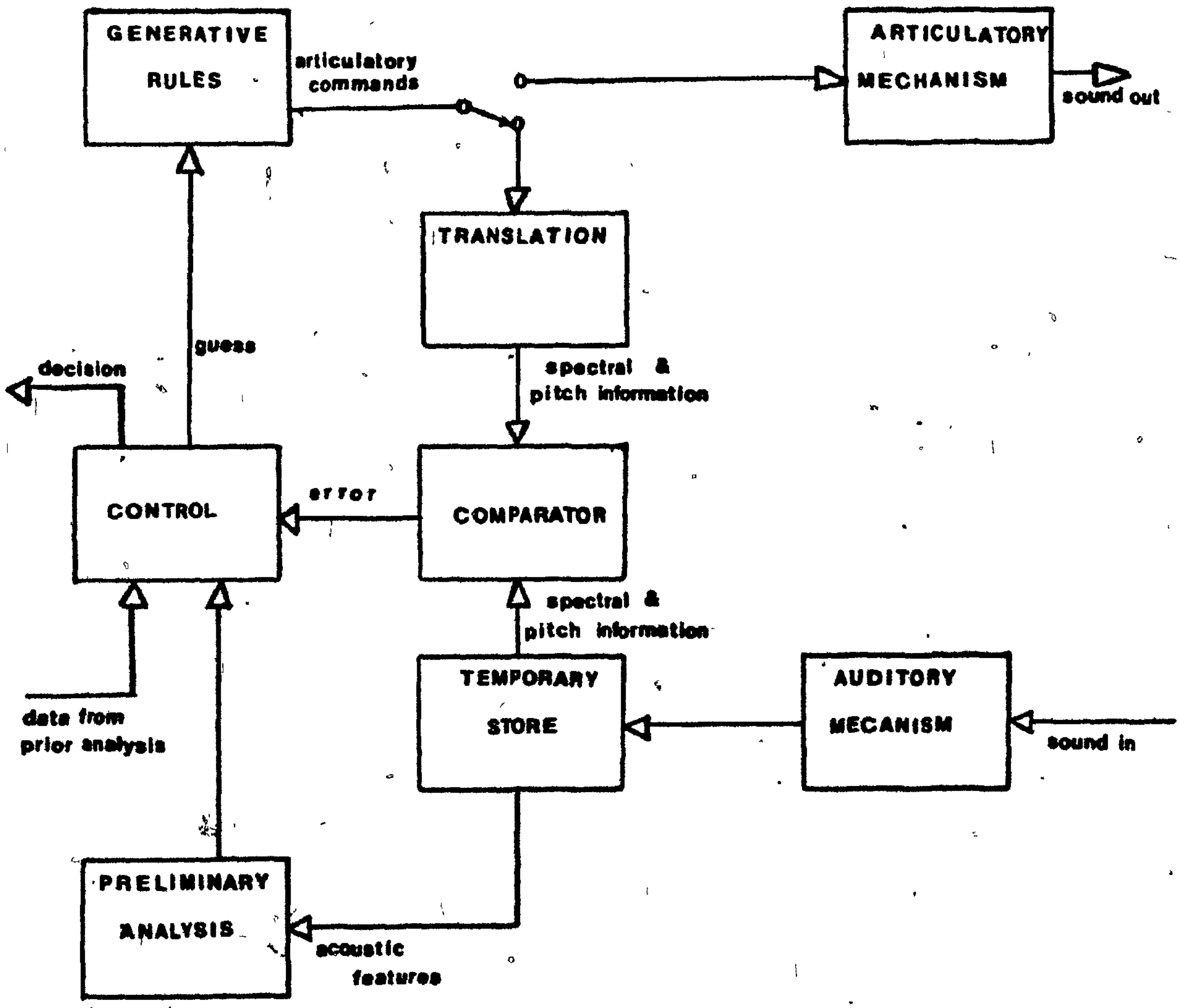


FIGURE 6 STEVENS AND HALLE MODEL FOR THE PERCEPTION AND GENERATION OF SPEECH

tic structure of past utterances. On the basis of these two inputs, the control section makes an educated guess at the phonetic segment. A generative rule system, normally used for speech production uses the guess to originate the necessary articulator movements for production of that phonetic segment. These movements are, however, short-circuited to another section which generates the spectral content of the guess. This information is compared with the stored spectral information and the result fed back to the control centre which can adjust its guess until the error is very small.

Perception can either be categorical or continuous. Categorical perception occurs when a small acoustic change can result in a large perceptual change. A small acoustic change in continuous perception, however, results in a small perceptual change. Typically stop consonants fall into the categorical classification whereas vowels are continuous.

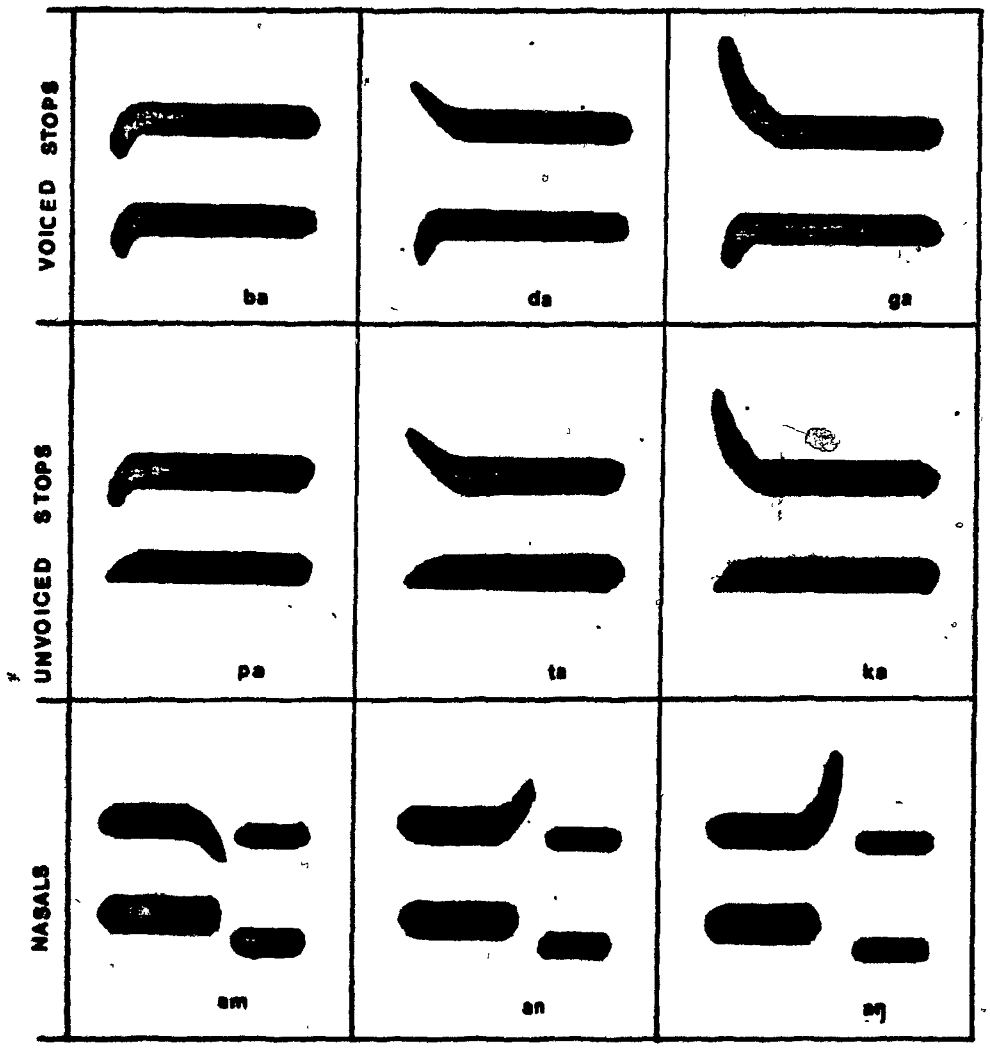
Continuous perception is subject to context effects. This is very apparent in vowels, especially when they are close to the categorical boundary. As an example, a sound close to the boundary between /i/ and /I/ is heard as /i/ if preceded by /I/ and will be perceived as /I/ if preceded by /i/ . Formant and fundamental frequencies vary widely for men, women and children. Lieberman (1973) postulated that a set of calibrating signals (the vowels /i/, /a/, /u/, or the glides /j/, /w/) determines the length and size of the speaker's vocal tract and are necessary to assign the acoustic signal to the correct phoneme.

Categorical perception is contingent on several acoustic cues. Cooper et al (1952), using the Haskin Laboratory's pattern-playback device, investigated stop and nasal consonants. Voiced-unvoiced pairs /b,p/ /d,t/ /g,k/ were discovered to differ systematically in VOT, and in the transitions of the first formant frequency. When the stops /b, d, g/ are followed by a vowel, they differ in second formant transitions. This also applies for unvoiced stops and nasals (Figure 7).

The duration of silence in stops aids in the perception of voicing/unvoicing. The word 'rabid', for example, becomes 'rapid' as the period of silence is increased from 20 msec to 60-80 msec. The duration of the formant transitions also aids perception. Short transitions result in hearing stops (/b/ or /d/) whereas medium transitions give /w/ or /j/ . Longer transitions appear as /u/ or /i/ .

In unvoiced stops, the relative frequency position of the noise burst cues perception. A high frequency noise burst results in /t/ whereas a low noise burst results in /p/ . When the noise burst is level with or slightly above the second formant frequency, a /k/ is heard.

(after Cooper et al 1952.)



--STYLIZED AFTER THE HASKINS LABS PATTERN-PLAYBACK FIGURES--

FIGURE 7 FIRST AND SECOND FORMANT TRANSITIONS IN STOPS AND NASALS

CHAPTER II

METHODS OF SYNTHESIS

The earliest forms of speech synthesis were acoustical-mechanical in nature (von Kempelen 1791, Wheatstone 1830, Gabriel 1879, ref. Mattingly 1968). These machines modeled the human vocal tract and were true analog equipment. Typically the vocal and nasal tracts were represented by bellows and resonators of the correct size and shape. When used by a skilled operator, these machines could be made to produce vowels, nasals, various words and even connected speech (Mattingly 1968). This early work had a significant influence on modern research particularly on vocal tract analog synthesis and articulatory models.

The transition from mechanical analogues to electrical ones began in 1937-1938 with the development of the Voder. Just prior to this a vocoder was developed (Dudley 1939) which performed a crude spectral analysis. The vocoder used filters covering 250-3000 Hz and a circuit to measure the fundamental frequency (Figure 8). Output from the fundamental frequency detection circuit controlled a buzz circuit at the receiver or synthesizer section. If the amplitude was sufficiently small, it failed to activate that circuit and a hiss generator was substituted. The receiver section consisted of a set of filters corresponding to those of the transmitter-analyzer driven by the buzz or hiss generators. Output from the transmitter's filters controlled the amplitude of the outputs of the receiver's filters which were then summed to produce the speech.

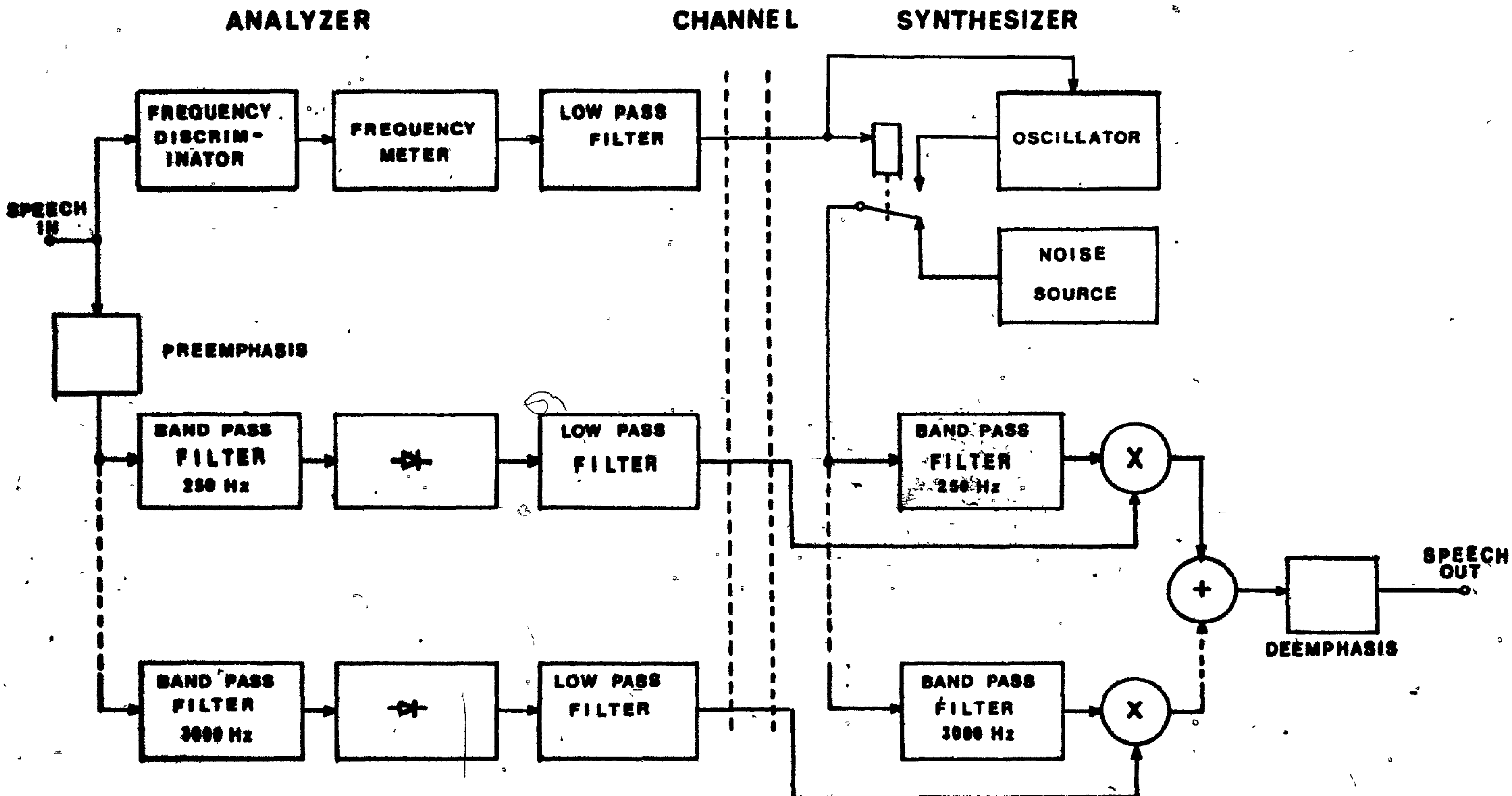


FIGURE 8 BLOCK DIAGRAM OF DUDLEY'S VOCODER

Dudley's Voder (demonstrated at the 1939 and 1940 World Fairs) was essentially the receiver section of the vocoder modified for manual operation. Dudley's vocoder differed from the previous work in two main aspects. Firstly, the model considered speech as an acoustical not articulatory function and secondly, synthesis was produced by electrical methods.

Development of the spectrogram and the Haskins pattern playback device in the 1940's, precipitated a more serious analysis of speech. Analysis-synthesis soon became an important tool in the understanding of the basic features of speech. Speech could now be broken down into a number of key elements and then re-synthesized according to those particular parameters.

Synthesis by rule is a method of production of artificial speech by formulation from a set of rules or algorithms. The ultimate objective of synthesis by rule is to generate natural sounding speech with a minimum input, and if possible, directly from a written text. However, it was not until the advent of the digital computer in the late 1950's that progress in this field was possible. Since then, the computer has become an essential tool in speech research.

2.1 Synthesis-by-Analysis

Synthesis of speech by analysis concerns the extraction of efficient parameters by which good synthetic speech can be produced. By modelling a particular set of parameters, speech can be reproduced with reasonable accuracy. The obvious limitation to this process is that any new vocabulary must be preceded by more analysis. Nonetheless, the early work in this field is extremely important because it provides understanding of the spectral contours of speech as well as electrical methods for reproduction.

2.1.1 Terminal Analog Synthesis

Terminal-analogs (Flanagan 1957) model the vocal tract in terms of its input and output characteristics. A source-filter decomposition (Fant 1960) takes place separating the sources of sound and the vocal tract. This decomposition, common to most synthesis models, assumes no source-vocal tract interaction and represents speech as a source spectrum, shaped by a vocal tract transfer function. Spectral characteristics are generated by resonant and antiresonant circuits arranged in series or parallel configurations. Early synthesizers were constructed along these lines using analog filters. However, once computers became readily available, filter simulation became the primary method of synthesizing speech.

Terminal analog synthesizers are often classified by the type of architecture used in the configuration of their formant filters. In a series configuration, the formant filters are connected in cascade, whereas a parallel model will have these filters connected in a shunt arrangement.

Both parallel and serial synthesizers have their advantages and disadvantages. Serial synthesis does not require the amplitudes of each resonance to be specified. This is highly desirable in synthesis by rule because of the simplification of the algorithms. Serial configurations also give much better production of vowel sounds. A parallel configuration, however, propagates noise additively resulting in a better signal to noise ratio. Since consonants frequently require emphasis on the higher frequencies, the ability to control the amplitudes of each formant individually in parallel synthesis is very useful. Errors in formant tracking occurring in parallel synthesis do not alter the amplitudes of those formants that are following a correct trajectory and are therefore less troublesome. One of the disadvantages of the parallel synthesis is the introduction of zeros falling between resonances. If perceptible, they distort the synthesis giving it a reverberant quality. The zeros do, however, provide low frequency emphasis.

Parameters for synthesis are obtained by spectral analysis of speech as derived by sonographs or fast Fourier transforms (FFT). The specifications generally used are formant frequencies, and their bandwidths or amplitudes.

2.1.2 Linear Prediction

The object of linear prediction coding (LPC) analysis is to predict the output signal solely on the basis of linear combinations of past input and output data. Insofar as analysis-synthesis systems are concerned, the linear predictive methodology appears to be the best available to date. Fundamentally, linear predictive coding models the vocal tract and then searches for appropriate parameters on the basis of least square error.

The coefficients used in LPC analysis are varied. They can be representative of the impulse response of a vocal tract filter, autocorrelation of the signal, spectrum, cepstrum, poles and zeros of the filter, or reflection coefficients (Makhoul 1975). Of these, the most frequently used are impulse response and reflection coefficients.

As an example of the processes involved in LPC analysis, consider the all pole model (Figure 9). The output signal is a linear combination of past output values and the present input:

$$x(n) = \sum_{k=1}^{p} a_k x(n-k) + A\,U(n)$$ where $x(n)$ is the present output, $U(n)$ the present input and a_k the LPC coefficients.

Taking transforms of both sides

$$X(z) = X(z)\left(\sum_{k=1}^{p} a_k z^{-k}\right) + A\,U(z)$$

The transfer function $H(z)$ then becomes:

$$H(z) = \frac{X(z)}{U(z)} = \frac{A}{1 - \sum_{k=1}^{p} a_k z^{-k}}$$ (an all pole transfer function.)

Frequently the input is unknown and the output is based solely on past samples:

$$\tilde{x}(n) = \sum_{k=1}^{p} a_k x(n-k)$$

The error between samples is then given by

$$e(n) = x(n) - \tilde{x}(n) \qquad \text{(also known as a residual)}$$

The a_k's are then chosen to minimize the total squared error E, i.e., to solve $\frac{\partial E}{\partial a_k} = 0$ where $E = \sum_{m=0}^{N-1} e^2(m)$.

The advantages of LPC analysis-synthesis lie in the very accurate estimates of the features of speech. It is also reasonably fast and robust, i.e., it is tolerant to noise and the distortions of speech typical in telephone line transmission. LPC frequently is used because it offers the capability of direct analysis and a means of obtaining accurate coefficients. For example, LPC is a precise way of plotting formant frequencies and trajectories. Until fairly recently, LPC was not used directly in synthesis-by-rule. The development of an integrated LPC synthesizer chip (TMC 0280) by Texas Instruments (Wiggins and Brantingham 1978) will undoubtedly have a significant influence on future applications. For the first time, it is now possible to obtain a low cost device which can be used in real time for speech synthesis.

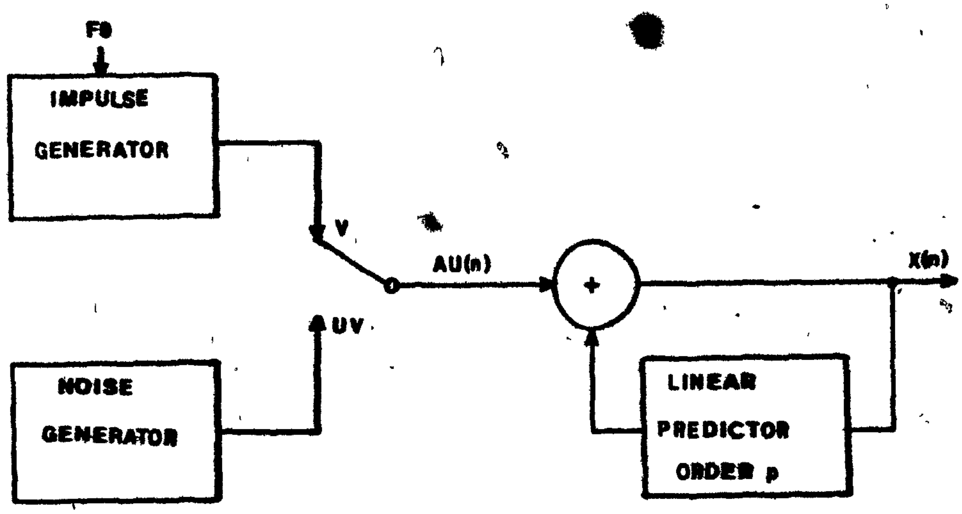


FIGURE 9 L.P.C. ALL-POLE MODEL

2.1.3 Vocal Tract Analog

The vocal tract can be modelled by a series of hard walled tubes, each connected end to end with differing cross sections in such a way as to create a quantized version of a vocal tract. Thus it is possible to describe the volume velocities or pressures within the vocal tract by Webster's horn equation with appropriate boundary conditions (Mermelstein 1973).

By utilizing the duality which exists between acoustical and electrical systems (i.e.,representing volume velocity by current, and pressure by voltage, etc.), it is possible to represent the vocal tract by a series of RLC networks. Early attempts to produce vocal tract analog synthesizers were constructed on these concepts. With the advent of computer simulation, it is now possible to synthesize connected speech (i.e., meaningful words).

The main problem with vocal tract analog synthesis is that of obtaining appropriate control data (i.e., cross sectional area, etc.). In the past, x-ray cinegraphy and palatography were used. The quality of speech produced by this methodology was intelligible and human-sounding but involved considerable effort in adjusting the model for optimum results. When LPC analysis became available, these methods were dropped because of obvious health hazards or inconvenience to the subject under test. The LPC technology enables analysis to be based directly upon speech rather than examination of vocal tract dimensions.

2.1.4 The Articulatory Model

The articulatory model for speech synthesis is one in which the parameters which determine output are based on tongue position, lip protrusion and other physiological factors.

Coker's model (1967) transformed the physiological parameters into formant frequencies and then used a formant synthesizer to complete the task. This was accomplished by classification of each sound in terms of the target configuration and velocities of the articulators. These parameters were then used in specifying an area function from which the formant frequencies could be extracted.

Ishizaka and Flanagan (1972) developed a laryngeal and vocal tract analog system. The laryngeal model was based on the symmetry of the vocal cords, representing them by two separate horizontally movable masses. This laryngeal model when used in conjunction with a vocal tract analog is one of the few that does not assume the excitation source to be independent of the vocal tract. Although this representation was useful for evaluation and understanding some physiological data, it tended to complicate other areas. The main problem seemed to be an inability to obtain proper control information. In general, articulatory movements are rather complex and are more appealing to a phonetician than anyone engaged in speech synthesis.

2.2 Synthesis-by-Rule

Speech synthesis by rule is a rather broad description of various approaches to synthetic generation of speech where rules of algorithms are used. It can be considered as the production of recognizable artificial speech by transforming a written representation of the utterance into a continuous acoustic output. Most synthesis systems are categorized by the manner in which the utterance to be generated is represented. Phonetic synthesizers, for example, operate on a phonetic representation of speech whereas text synthesizers rely on an orthographic representation of speech (i.e., printed text), translating it into phonetics and then utilizing phonetic synthesis to generate utterances. The two most important objectives of speech synthesis-by-rule lie first in attaining natural-sounding speech and second, in generation of this speech from a minimal input (ideally from written text or phonetic transcription).

Achievement of natural speech requires a high phonetic quality, voice quality and good prosodic content. Phonetic quality, to a large extent, determines intelligibility and is primarily dependent on formant and spectral composition. Most synthesis-by-rule systems are based upon description of formant trajectory. These descriptions range from simple linear interpolation between steady state sounds to complex fitting. Parameter transitions occur in any synthesis system and rules for generating them are quite complex. Reasonably good results have been obtained for phonetic quality by using various rules and synthesis models. The resulting speech, however, usually sounds very mechanical and cannot be improved without careful attention to voicing. Voice quality is a very important factor in improving the quality of synthetic speech, since it is the best method to eliminate what might be termed "mechanical speech" properties. Correct use of voicing can also generate speech characteristics of either sex and even age groupings. Several rules for improving naturalness and voice quality are presented by Sapozhkov (1972). One method involves modifying the hardware such that the aspiration source is bandpass filtered between 30 and 70 Hz. This is then used to modulate amplitude and frequency of the voicing source. At the present time rules governing voice quality are generally inadequate and suggest the need of a better understanding of the glottal waveforms.

Prosodic content is determined by the duration of formant transitions and timing in general. For example, the length of some consonants are dependent on the position occupied in a word (initial, medial or final). Stressed vowels are longer than unstressed vowels. Moreover, the context in which a sound is made determines its length as well as its position in the breath group. Thus the rules governing duration are complex. Considerable progress has been made in the development of rules governing prosodic quality by Umeda (1972). However, more work would seem to be required in the assignment of sound durations if the objectives of natural speech are to be met.

Phonetic synthesis systems rely on phonetic strings with special marks or modifiers as input. Between the written text and the phonetic string, some form of decision making must take place which interprets the whole sentence. A simple example of this problem is illustrated by the use of the word "lead". Obviously the process of sentence analysis is not simple and becomes increasingly complex as vocabulary is expanded. Analysis coupled with phonetic synthesis is termed text synthesis.

The English language is constantly changing and there is little hope of ever compiling a complete lexicon for analysis. Fortunately, most English words have an internal structure consisting of units called morphs which can be used to compile a lexicon one magnitude smaller than the number of words. Thus it appears reasonable that a morph lexicon would be an ideal basis for representing the majority of the words. Adjustments for morphophonemic and lexical stress would be required to synthesize speech from unrestricted text (Allen 1976).

Development of a text synthesizer capable of handling a reasonable vocabulary is a formidable task involving storage of a large amount of data. It is likely that use of text synthesis will be limited due to cost factors. A phonetic synthesizer, on the other hand, is much simpler, low cost and ideally suited for small systems.

CHAPTER III

MAJOR CONTRIBUTIONS TO SPEECH SYNTHESIS-BY-RULE

A review of some of the major contributions to the field of artificial speech generation is helpful in identifying and understanding earlier concepts which are incorporated in the synthesis strategy developed in this thesis. Many of the problems encountered in early work were due to the computational speed of the computers available at that time. In this respect, the use of a microprocessor and its inherent speed limitations are similar.

3.1 Kelly and Gerstman

The first attempt to use a computer in synthesizing speech by rule was by J.L. Kelly and L.J. Gerstman (1961). The computer (IBM 7090) was used to calculate the necessary parameters for synthesis from a phonetic input by using a set of relatively simple rules. A serial-synthesizer was used and controlled by 9 parameters, frication (the hiss amplitude), voicing (the buzz amplitude), the fundamental frequency (pitch) and the center frequencies and bandwidths of three formants. The program input consisted of a deck of punched cards. Each card contained a symbol corresponding to the required phoneme, a stress mark, or, modifications in stored values for circumstances that the rules were not equipped to handle. Each phoneme had thirteen associated parameters. These represented the duration of the initial transition, the duration of the steady-state, the nine required synthesis parameters during steady-state, the duration of the final transition and whether or not the phoneme was a vowel or a consonant.

The duration between the steady-state part of adjacent phonemes was the sum of the final and initial transitions. The parameters for consonant-to-vowel transitions followed a smooth convex path, vowel-to-consonant transitions a concave path, consonant-to-consonant and vowel-to-vowel transitions followed a straight line. During the steady-state, the parameters are held constant. If a stress mark is included in the input deck, another parameter table is used that incorporates the necessary durations and change to other parameters that are characteristic of stressed vowels.

The results of this synthesizing strategy are debatable. It is capable of generating clear and intelligible speech only after a great deal of ad-hoc changes to many of the stored values. The great contribution of Kelly and Gerstman was the use of a computer which permitted the rules of synthesis to be altered, tested and improved. This formed a basis for much of the more recent investigations.

3.2 Holmes

The second use of a computer in synthesis-by-rule was undertaken by J.W. Holmes et al (1964). In this system, the computer was used to prepare a paper tape of control parameters for the synthesizer. Input to the program consisted of phonetic symbols, fundamental frequency values, and auxiliary modifier characters. The program did not run in real-time and its only purpose was to prepare the punched tape for future use by the synthesizer.

The system is based on a parallel-terminal analog synthesizer consisting of five bandpass filters, a voicing source and a hiss source. Three of the formant filters have separate amplitude controls and may be driven from either source. The fourth filter (fixed at 3500 Hz) used only during voiced passages, shares a common amplitude control with the fifth filter (broadband from 3400-4000 Hz) used only during aspiration. The output from all five filters are summed together to form synthesized sound. The voicing source has a fundamental frequency (F0) ranging from 50 to 250 Hz in 31 levels arranged to be roughly logarithmic. The first formant (F1) consists of 30 levels each spread 30 Hz from 130 to 1030 Hz. The second formant frequency (F2) has 30 levels spaced 60 Hz varying from 760 to 2560 Hz. The third (F3) also has 30 levels spaced 60 Hz but ranges from 1540 to 3340 Hz. Amplitude controls are quantized into 31 levels. Thirty of these are spaced 1.75 dB. The other level is used to disconnect the filter. The synthesizer is controlled by a punched tape containing all the necessary parameters in 10 msec intervals.

The phonetic elements used by the program correspond roughly to the International Phonetic Alphabet (IPA). Because of difficulties in synthesizing stop consonants, several sub-phonetics were incorporated (i.e. silence, noise burst etc.). All the parameter values, except SW (voicing/aspiration) and F0, are determined for an initial transition, a steady state period, and subsequently the final transition. Steady state values are stored in a table along with corresponding phonemes. Both initial and final transitions are computed on the basis of adjacent phonemes. The table also contains a rank between 1 and 31 assigned to each phoneme in addition to three parameters governing transitions typical of the unstressed phoneme. These parameters (internal transitions, external transitions, and fixed contribution) represent the duration of the transition for the dominant phoneme, the duration of the adjacent phoneme and the steady-state duration respectively.

If the rank of the phoneme is higher than that of adjacent phonemes, then the transitions are characteristic of that phoneme. Should the rank be lower than either of adjacent phonemes, the phoneme with the highest rank determines the behaviour of the transitions. Generally, stop consonants have the highest rank, vowels the lowest, nasals and fricatives falling somewhere in between. If the ranks are equal, the first phoneme is dominant.

Transitions are based upon linear interpolation of parameters stored in the phoneme table. Fundamental frequency values are entered by hand from spectrographic data. Should the table parameters need to be adjusted on a temporary basis, a set of special modifier characters is incorporated.

The results obtained from this synthesizer and synthesis strategy are reported to be quite acceptable and capable of generating very realistic sounding speech but only when the text input is carefully edited. Problems associated with this technique stem from the lack of rules governing stress or intonation, which necessitates transcription of F0 values from spectrograms and alteration of the duration of stress phonemes.

3.3 Rabiner

The research work by Rabiner (1968) is of great importance to the development of speech synthesis by rule. As in previous examples, a computer is used to control a terminal analog synthesizer. The synthesizer is of the series type with one parallel side branch for the unvoiced component of fricatives. Static phoneme characteristics are determined as in other systems according to lookup table. The method involves considerable computation rendering it incapable of real-time processing.

Considerable effort went into obtaining an accurate representation of the formant transitions. A solution to a second order, critically damped differential equation was selected because it gave a good fit to experimental data and required only one time constant for solution. General motion for a formant with initial position Ai, to a formant with final position Af and an initial formant velocity Vi is described by the following equation,

$$x(t) = Af + (Ai - Af)\exp\left(-\frac{t}{\tau}\right) + \left[Vi + \frac{(Ai - Af)}{\tau}\right]t\exp\left(-\frac{t}{\tau}\right)$$

for $t \geq 0$.

Each formant can move from its present steady state value to the next at different rates. This necessitates the formation of a time constant (τ) for each formant per pair of phonemes. There are approximately 40 phonemes in English, yielding 1560 possible combinations of phoneme pairs. With three formants, this means some 4680 possible time constants need to be sorted and stored. By various approximations the number of time constants to be specified was reduced by one order of magnitude.

As each formant progresses towards its target value, its motion is specified by the differential equation. When all the formants are within frequency bands around their respective targets, the program determines if the phoneme to be generated is to be stressed. If it is to be stressed, then the duration of that phoneme is lengthened to correspond to the value stored in a table of stressed phonemes. As soon as the stressed duration has been generated, normal motion towards the next phoneme continues. If the phoneme were not stressed, then normal motion resumes immediately.

The remaining synthesizer control parameters are 'time locked' to the formant motion. The amplitude controls change linearly at predetermined rates, approximately one time constant (τ) after motion towards the new target value is initiated. Nasal and fricative poles start to move as soon as new formant target values are defined. The poles and zeros move in a linear fashion towards their targets and reach them as soon as the amplitude controls are switched. For nasals the first formant bandwidth is increased from 50 Hz to 100 Hz, 50 msec. before and after the nasal. For non-nasals, the nasal pole-zero pair are set to 1400 Hz where they will supposedly cancel. Unless the synthesizer is constructed using digital technology, it is doubtful that this claim would be met. For non-fricative sounds, the fricative pole-zero pair are set to 1500 Hz.

The fundamental frequency model is based on work done by

Lieberman. This work is founded upon the breath group. For the first 300 msecs of the breath group, the fundamental frequency rises quickly to about 125 Hz, whereupon it remains constant until the last 300 msec. when it falls off rapidly. If the breath group is a simple interrogative sentence (yes-no question) in the last 175 msec F0 rises 60 Hz. A stressed vowel would lead to a peak in F0 for roughly 500 msec.

The results of this synthesis strategy were excellent. The amount of calculations required, however, make it impractical for real-time synthesis. It is, nevertheless, one of the best attempts at accurate modelling of human speech to date.

3.4 Ainsworth

Further modifications of the Holmes synthesizer were made by W.A. Ainsworth (1972) at the University of Keele. In this concept the values of the first formant frequency (F1) were changed from 130-1030 Hz to 230-1030 Hz and another filter (FN) was incorporated with a range of 100-400 Hz for nasal resonances. Otherwise the synthesizer was identical.

The real improvement over the Holmes synthesizer was that Ainsworth designed the system to be controlled by a PDP-8 computer in real time.

Although the Holmes and Ainsworth synthesizers are virtually identical, the synthesis rules differ considerably. This is partially due to the constraints of operating in real-time. The concept of a phoneme having a 'rank' was eliminated by Ainsworth and only two parameters T1 and T2 determine the motion of transitions. These represent the duration of the steady-state and transitional parts of the corresponding phoneme respectively. An attempt at generating a set of rules for the fundamental frequency (F0) contours was made. This is based upon the isochronous foot theory (Ainsworth, 1972), where it is assumed that the duration of breath groups are constant. Stress marks are incorporated and positioned after each stressed syllable. The intervals of time between stress marks are then determined in the program. Should two marks appear within an interval of time, roughly 400 msec, then the duration (T1) of the last stressed phoneme is lengthened until the time between stress marks are equivalent to the threshold of 400 msec. Fundamental frequency was also varied in accordance to the stress marks. It can rise linearly to a peak during a stressed syllable and then fall away linearly to a minimum approximately halfway between stressed syllables. This methodology necessitates a buffer to store all the control parameters until such a time as the first stress mark is found (at which time the F0 values are inserted into the buffer). This results in a 250-500 msec delay before the system starts to synthesize speech. The synthesizer when controlled by a PDP-8 and 4 K core is capable of storing the controlled program, the phoneme lookup table and a buffer of sufficient length to store about 3.4 seconds of speech.

The results of this particular synthesizing scheme are relatively good and eliminate the need to search for fundamental frequencies from spectrograms as was the case with the Holmes synthesizer.

Although the speech is not natural sounding, it is reputed to be much easier to listen to than monotonic utterances.

3.5 Klatt

Several hybrid devices have been designed which take advantage of both series and parallel configurations (Kayto et al 1971, Ochiai et al 1972, and Klatt 1972). This synthesizer and its later modifications (Klatt 1976, 1977) are arranged in such a way that it can change from series to parallel configuration during synthesis. (A block diagram is shown in Figure 10). This means that vowels can be generated as a series network and consonants on a parallel configuration without difficulty. The synthesizer is simulated on a general purpose computer. Second order digital resonators are required with some twenty control parameters determining the output. Advantages of an entire software implementation are considerable. Calibration is not required, stability is assured and control of signal-to-noise ratio is available.

The synthesis strategy of the Klatt system is uncomplicated but the rules are quite complex. Each new phonetic symbol determines a target value for each parameter by table look-up. These target values are then modified according to the preceding or following phonetic symbols and stress or durational patterns. Transition time between target values is also determined by adjacent phonemes. Transitions are obtained from linear interpolation or half-cosine contours. The rules take into account

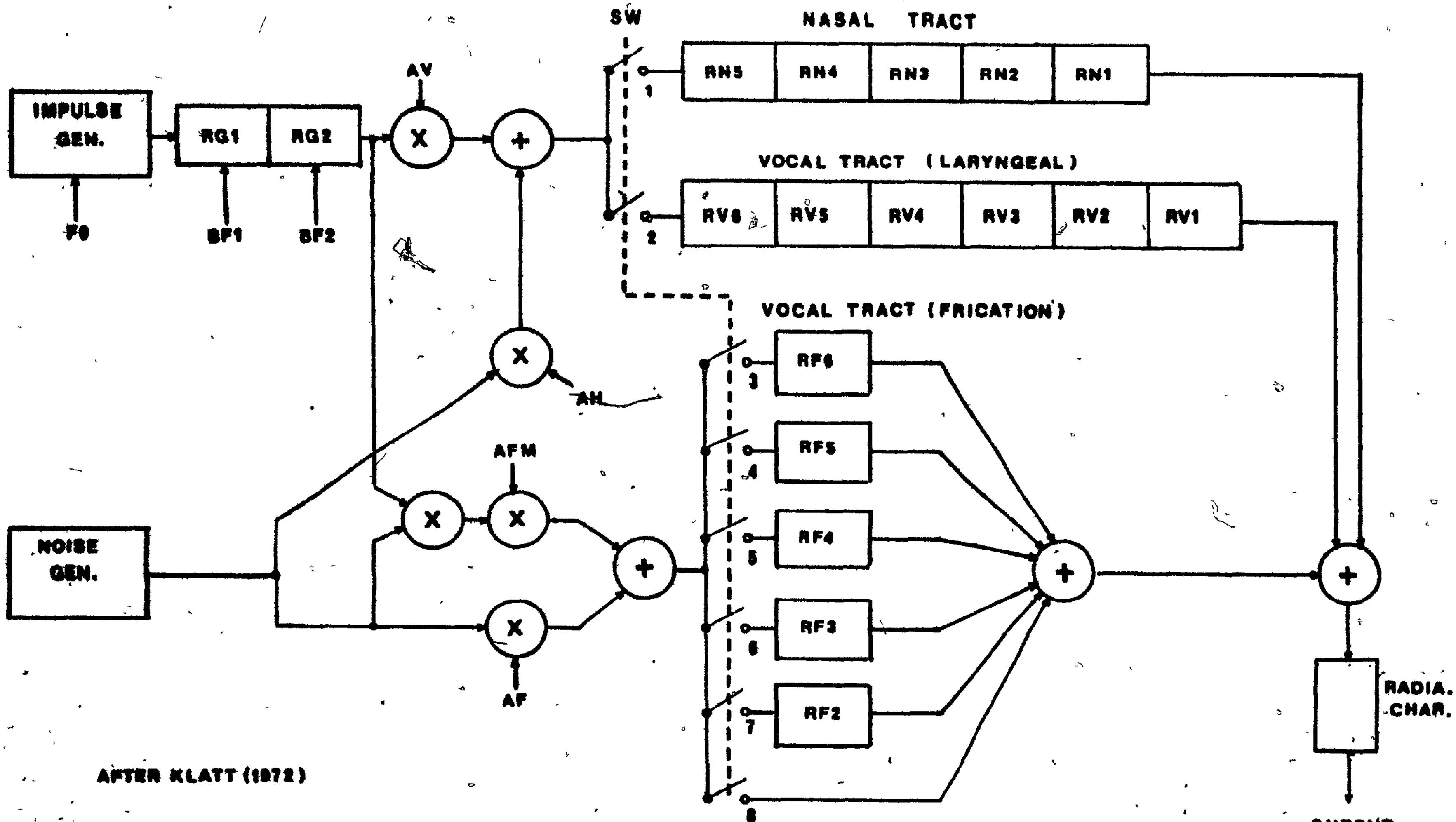


FIGURE 10 THE KLATT SYNTHESIZER

such factors as stress, segment duration, fundamental frequency variations, segmental insertions, deletions and substitutions. Although the results obtained from this system are good, the synthesizer simulation and complex rules prevent real-time synthesis.

3.6 The Keele System

A simple text synthesis word processing system was developed in England at the University of Keele (Ainsworth 1973). In this study, the orthographic text is transcribed onto paper tape and fed to an inexpensive minicomputer (PDP-8) where appropriate control parameters for a terminal analog synthesizer are generated.

The synthesis process involves four major states (Figure 11). These are breath group segmentation, phonemic translation, assignment of stress and parameter calculation.

Breath group segmentation boundaries are established by reading a character string into a buffer and assigning boundaries at one of the following identifiers (whichever comes first):

1) at a punctuation mark
2) preceding a conjunction
3) between a noun and verb phrase
4) before a prepositional phrase
5) before a noun phrase.

The character string up to this identifier is transferred

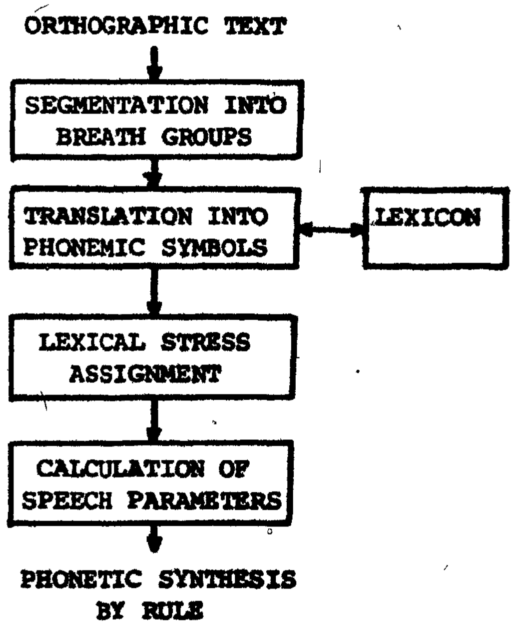


Figure 11. Processes of the Keele text synthesizer.

onward for further processing. The remaining characters are shifted down as part of the next breath group. If none of the above boundaries are found, the full buffer rounded to the next word boundary is designated as one breath group. Best results obtained are for a 50 character buffer, where boundary assignment is correct 80% of the time.

The letter-to-sound rules are based upon a table listing each letter and its common phonetic translation along with any conditions necessary for that phonemic pronunciation. e.g.,

(ough)t	=	/ɔ/
b(ough)	=	/a ʊ/
t(ough)	=	/ʌ f/
c(ough)	=	/ɔ f/

Common exceptions are stored in a lexicon. Where translation of a letter proves to be ambiguous and where context cannot be resolved, the most common or neutral phoneme is inserted.

According to Ainsworth, vowels are the most difficult to translate, with the letter 'O' the worst of all. Consonant errors are usually substitution of a voiced phoneme for unvoiced phonemes, e.g., confusion of /ð/ and /θ/ .

Any remaining errors are unstressed vowels which are caused primarily by letters with similar context, e.g.,

h(ea)rt	=	/a:/
f(ea)r	=	/i ɜ/

r(ea)lity = / iæ/

gr(ea)t = /ɛ i/

m(ea)t = / i /

It seems odd, therefore, that these words would not be stored in a lexicon along with other exceptions. Table 2 lists Ainsworth's error analysis of phonemic translation of several source words.

Words that are usually unstressed, i.e., articles, prepositions, conjuctions, etc. are stored in memory along with a list of prefixes.

Words stored in memory are left unstressed. Those words not stored in memory but with prefixes belonging to the stored list have the second syllable stressed. For words not in memory and with the prefix not stored, the first syllable only is stressed.

Table 3 lists errors incurred by mis-assignment of stress for text, bisyllabic words, trisyllabic and longer words.

Where a phoneme is preceded by an identical phomeme, the second is deleted and if a word ends in a vowel with the next word starting with a vowel, a glide is inserted. The remaining speech processing is performed by a synthesis by rule program as described earlier.

According to Ainsworth, the system worked well, especially the letter-to-sound rules employed. On the average, a seven word sentence will only contain one phonetic error. When tested on uninformed listeners, results of between 50 and 90 percent intelligibility were obtained. Problems arise, however, when a cluster of errors causes the listeners

TABLE 2

ANALYSIS OF ERRORS IN PHONEMIC TRANSLATION

Source	Error (percent)	Stressed vowels (percent)	Unstressed vowels (percent)	Consonants (percent)
Textbook	8%	4.5%	2.2%	1.3%
Novel	11%	6.8%	3.0%	1.2%
Newspaper	11%	6.9%	3.1%	1.0%

TABLE 3

ANALYSIS OF ERRORS IN ASSIGNMENT OF STRESS

Source	Error (percent)
Textbook	10%
Bisyllabic words only	17%
Trisyllabic words only	31%
all longer words	44%

*Note 90% of text was monosyllabic.

to lose track for a few sentences. Apparently the onus is placed on the listener to concentrate on the utterance to determine its meaning. In all, this system is very encouraging in the sense that synthesis was performed on a small computer with rather limited facilities. Improvements in the rules assigning stress and increasing the existing lexicon could lead to a highly acceptable system.

CHAPTER IV

HARDWARE

4.1 Systems Configuration

The development of the phonetic synthesizer described herein is based on the use of an MC 6800 microprocessor with MIKBUG firmware. The complete system has available 16 K RAM, 2 K ROM including monitor and floppy disc programs. Peripherals such as Model 33 Teletype, Volker Craig 303A CRT, Control Data Vucom 1, associated magnetic tape drive, PerSci disc and Computalker Consultants CT-1 synthesizer completed the system.

The program was initially assembled using a SWTPc editor-assembler but was later re-assembled using the Motorola M68SAM cross-assembler. This was executed on the development system described previously. Program de-bugging was undertaken using the Motorola decode-disassembler (C. R. Bilbe M6800 user group library # 56) with manual insertion and deletion of breakpoints.

Since the design objective for hardware was visualized as a self contained voice synthesizer and not as a minicomputer adaptation, memory and input-output facilities were minimized. The final system configuration can be constructed on one circuit card and if used in conjunction with the CT-1 synthesizer, the total system can be housed in an enclosure 30 x 20 x 5 cm. including the power supply. The systems configuration is shown in Figure 12.

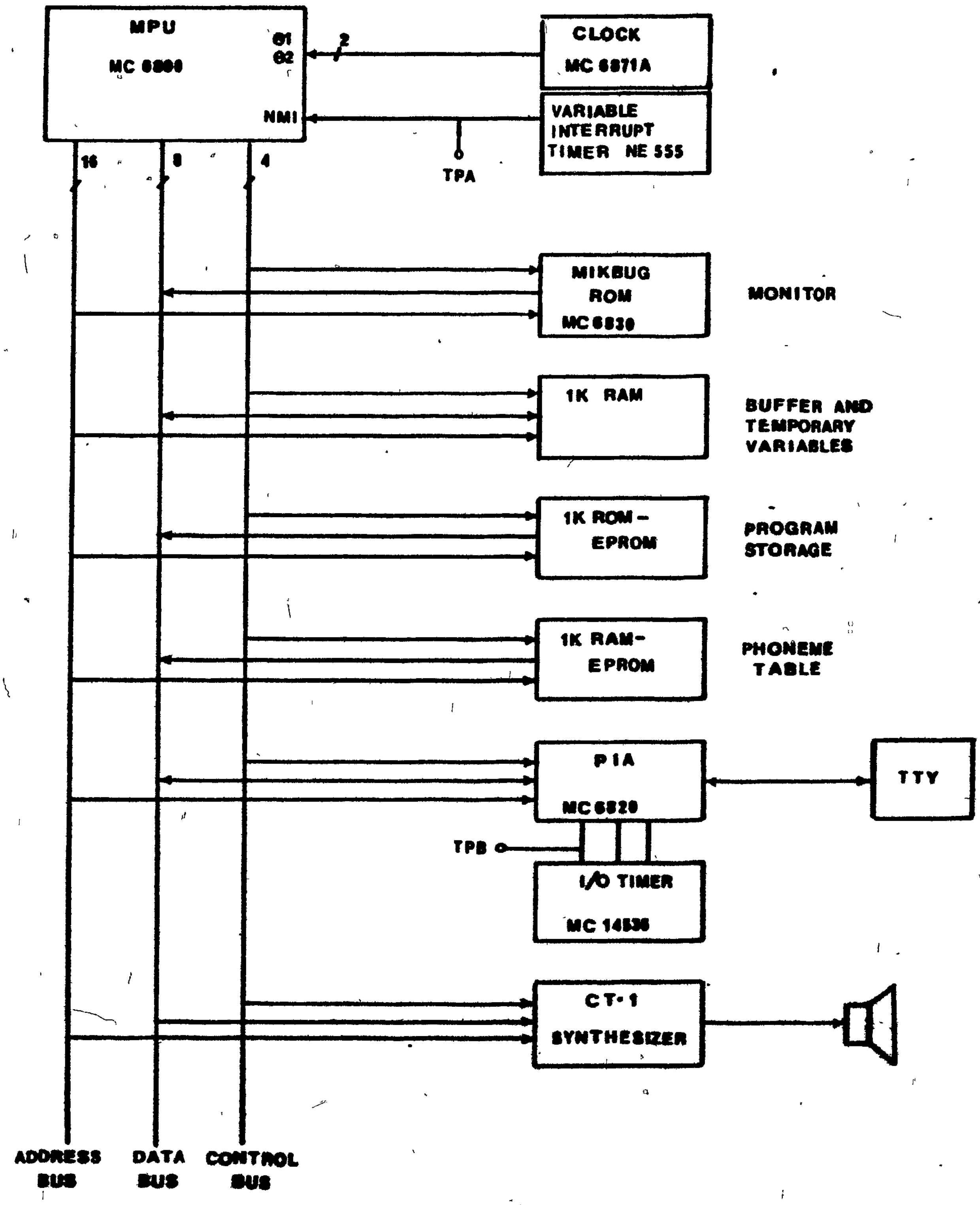


FIGURE 12 MINIMUM SYSTEM CONFIGURATION

4.2 The Synthesizer

4.2.1 General Description

A number of options are open in the selection of a synthesizer. For example, a synthesizer can either be designed and constructed with the attendant problems of stability of filter networks or a unit can be purchased and modified as necessary. These considerations are well covered by Cohen and Massaro (1976).

Currently there are several commercial synthesizers available. Of those studied, the Swedish equipment made by FONEMA (Model OVE 111d) has the most versatile features but is a relatively expensive device. The VOTRAX manufactured by the Federal Screw Works is a pre-programmed synthesizer with fixed phonemes and thus could not be used. A relatively new synthesizer has recently been introduced by Computalker Consultants (Model CT-1) which appears to have the desirable features and is attractively priced. It was, therefore, decided that the Model CT-1 would be used as the synthesizer and modified if necessary.

The synthesizer is similar in organization to that of Klatt (1972) in that series networks and parallel side branches are arranged into nasal, frication and formant networks. From an architectural standpoint, the CT-1 synthesizer closely resembles a simplified version of the OVE 111d by Fonema. The block diagram of the CT-1 synthesizer is shown in Figure 13.

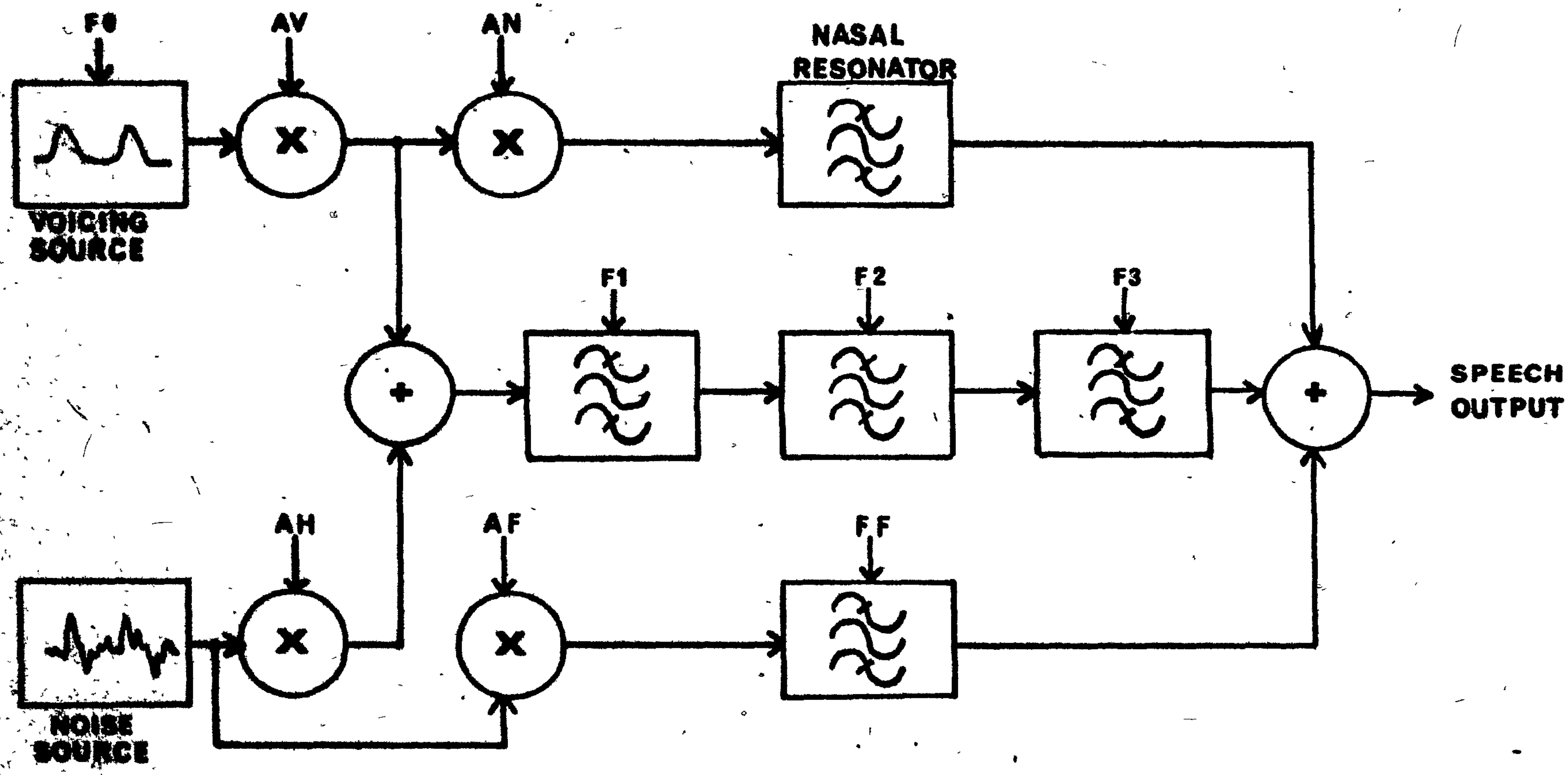


FIGURE 13 BLOCK DIAGRAM OF CT-1 SPEECH SYNTHESIZER

There are two sources of sound in the CT-1 synthesizer. These are the voicing and noise sources. The voicing source frequency is controlled by an input F0 and its level by a signal AV. The voicing waveform has to be carefully shaped to model the glottal pulse and associated spectral slope of -12 dB octave. The noise source level is controlled by AH and AF and its spectrum is essentially flat in the audio spectrum.

The formant network consists of three variable filters in series. These are excited by either voicing or noise sources, or both. Voicing level is controlled by AV and aspiration level by AH. Formant filters F1, F2, and F3 are controlled in frequency only.

The frication branch consists of a single variable resonator driven by the noise source with level control AF. Centre frequency of the filter is controlled by FF.

Nasal effects are created by a wide-band resonator with its centre frequency fixed at 1400 Hz. This side branch imparts a broad formant to the output which is common to nasal sounds. The resonator is driven by a voicing source in such a way that the input cannot exceed the voicing component of the formant network.

Output from the three branches (nasal, formant and frication) are added together to form speech output. Control of the various parameters is obtained by application of 8 bits of data to each of the 10 address positions. The received data is converted to the appropriate analogue signals within the unit. A list of control parameters, function, address and range is shown in Table 4.

4.2.2 Signals and Timing

The Model CT-1 is optimized for an S100 bus and operation with an 8080 microprocessor system and uses only 8 address lines. It was therefore necessary to modify the synthesizer to interface with the bus system used on the 6800 microprocessor. The following description gives a brief outline of signals required by the synthesizer and the methods of interface.

Of the eight address lines used by the synthesizer, the first four (A0-A3) select the specific parameter to be updated. The last four (A4-A7) are compared to a DIP switch and partially enable transfer of data from the computer to the synthesizer. Data transfer occurs when signals SOUT and $\overline{\text{PWR}}$ are in their active states (high and low respectively) and A4-A7 are valid. Two additional signals $\overline{\text{EXTCLR}}$ and $\overline{\text{POC}}$ are used to inhibit synthesis by disconnecting the audio output whenever either is low.

Because the processor treats the synthesizer as write only memory, the eight bit data bus can be either unidirectional or bidirectional. In the development system, the computer bidirectional bus is split into

TABLE 4

CONTROL PARAMETERS

Address A3---A0	Mnemonic	Name	Approx. Range
0 0 0 0	AV	Voicing Amplitude (dB)	40
0 0 0 1	F0	Voicing Frequency (Hz)	73.4-463
0 0 1 0	F1	First Formant Frequency (Hz)	174.9-1452
0 0 1 1	F2	Second Formant Frequency (Hz)	524.2-4356
0 1 0 0	F3	Third Formant Frequency (Hz)	1704-5508
0 1 0 1	AH	Aspiration Amplitude (dB)	40
0 1 1 0	AF	Frication Amplitude (dB)	12
0 1 1 1	FF	Frication Frequency (Hz)	1706-14160
1 0 0 0	AN	Nasal Amplitude (dB)	40
1 0 0 1		available as analogue voltages - not used by synthesizer	
1 0 1 0			
1 0 1 1			
1 1 0 0			
1 1 0 1		not used	
1 1 1 0		not used	
1 1 1 1	SW	Audio On / Off Switch	

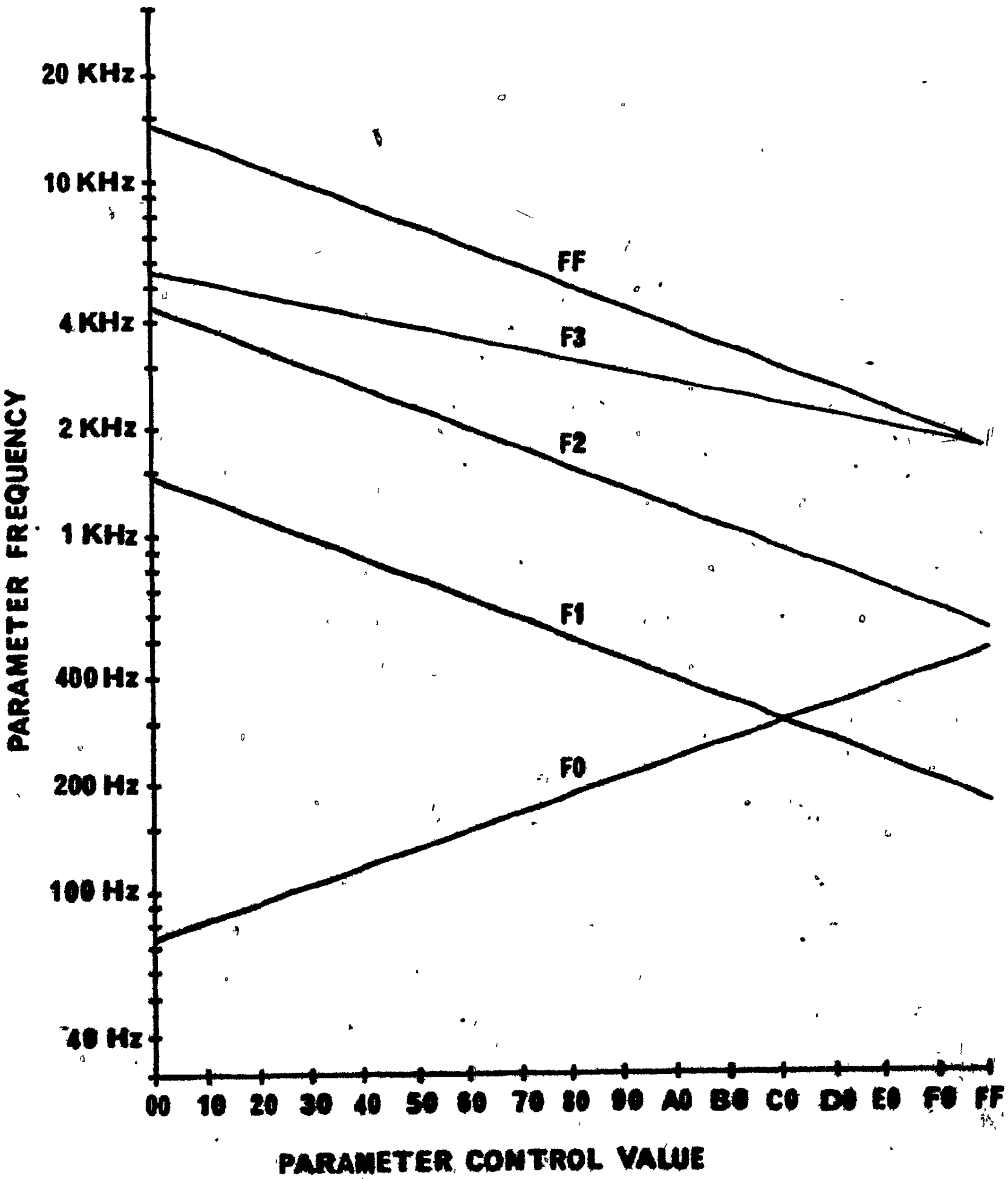


FIGURE 14 RESONATOR FREQUENCIES VS. CONTROL VALUES

two unidirectional buses to permit use of inexpensive RAM. The data-out unidirectional bus is used but only out of convenience. The signals, SOUT, $\overline{\text{EXTCLR}}$ and $\overline{\text{POC}}$ are tied high and $\overline{\text{PWR}}$ is generated from the NAND of $\overline{\text{R/W}}$, Ø2 and VMA signals from the MC 6800 bus. Address bits A4-A7 are selected such that the synthesizer is addressed above existing RAM, i.e., for location 2000 Hex. A4-A6, are connected to A4-A6 of the computer, while A7 is connected to A13.

On receipt of the correct address and write signals, the eight data bits and addresses A0-A3 presented to the synthesizer are stored in a latch. The data component is connected to an analogue voltage and directed toward the correct parameter channel according to address bits A0-A3. The analogue voltage is then retained by a sample-and-hold device for use by the analogue circuitry of the synthesizer. A minimum of 20 microseconds is allowed between individual updates in order to permit the sample-and-hold to stabilize under worst-case conditions. Furthermore, these updates should not exceed 50-100 milliseconds as the sample-and-hold will lose ability to maintain constant values.

CHAPTER V

SOFTWARE

5.1 Synthesis Strategy

The main objective in developing the program was to construct a set of rules for use on a 6800 microprocessor system that is capable of producing intelligible speech. It should also be flexible enough to permit basic research on speech synthesis or perception. A small scale system with limited memory is desirable because it can be developed into a self-contained voice synthesis peripheral and operate as a talking terminal. These requirements strongly suggest real-time operation to reduce the memory and eliminate the need for peripherals such as disk drives. Unfortunately, the slow speed of microprocessors limits computational capabilities and therefore imposes some constraints on the synthesis strategy. Consequently, rules must be developed which are essentially a compromise between those that minimize computation and yet still provide adequate intelligibility.

Natural sounding speech not only requires phonetic information but other information such as fundamental frequency contours, stress and duration patterns as well as syntactic and semantic factors. This requires a rather complex set of rules beyond the capabilities of the microprocessor system operating in real-time. Nevertheless, an attempt has been made to introduce some elements such as stress and duration which result in a more natural sounding speech. It is doubtful that this has improved intelligibility significantly, but it does make the output sound more natural.

The input to the program (Figure 15) consists of a sequence of phoneme representations and a group of secondary modifiers which are used to control pitch, duration and periods of silence. A phonetic coding scheme is used that is similar to the ARPABET (Table 1) but extended to two characters per phoneme. Other characters have been introduced to aid in editing phonetic strings or performing special functions (Table 5).

A look-up table is resident in the program and supplies information concerning each phoneme. This information is in the form of 22 bytes organized into three fixed format fields (Table 6):

The first field contains two ASCII characters which represent the phoneme. A second field contains nine parameters used to represent steady state sounds of each phoneme in isolation. The third field with the remaining eleven parameters, represents the duration of onset, transitional and steady-state parts of the phoneme.

Grouping of the timing parameters in the third field may on first glance seem arbitrary. However, changes in F1, F2, F3 and FF occur as the articulators move from one position (and sound) to another and consequently share common timing. The parameters AV, AH, AF and AN not only rely on the positioning of the articulators but also depend on other factors such as sub-glottal pressure and volume velocity. In addition, these four parameters change independently of each other. AF

>MMLHEHIYCH = "Mary"

>%MMEHEHIYQ0%

Time ↓

Frame Time: 10 msec.

>20	35	BF	E9	99	00	00	80	7F
>40	35	FE	52	B2	00	00	80	FE
>60	35	FF	53	B3	00	00	80	FF
>60	35	FF	53	B3	00	00	80	FF
>60	35	FF	53	B3	00	00	80	FF
>60	35	FF	53	B3	00	00	80	FF
>60	35	FF	53	B3	00	00	80	FF
>60	35	FF	53	B3	00	00	80	FF
>60	35	FF	53	B3	00	00	80	FF
>6A	35	29	5A	06	00	00	80	00
>74	35	53	61	59	00	00	80	00
>7E	35	7D	68	AC	00	00	80	00
>80	35	7E	68	AD	00	00	80	00
>80	35	7E	68	AD	00	00	80	00
>80	35	7E	68	AD	00	00	80	00
>80	35	7E	68	AD	00	00	80	00
>80	35	7E	68	AD	00	00	80	00
>80	35	7E	68	AD	00	00	80	00
>80	35	80	77	C8	00	00	80	00
>80	35	82	86	E3	00	00	80	00
>80	35	84	95	FE	00	00	80	00
>80	35	85	96	FF	00	00	80	00
>80	35	85	96	FF	00	00	80	00
>80	35	85	96	FF	00	00	80	00
>80	35	85	96	FF	00	00	80	00
>80	35	85	96	FF	00	00	80	00
>80	35	85	96	FF	00	00	80	00
>85	35	94	BA	18	00	00	80	00
>8A	35	A3	DE	31	00	00	80	00
>8F	35	B2	02	4A	00	00	80	00
>90	35	C1	26	63	00	00	80	00
>90	35	D0	4A	7C	00	00	80	00
>90	35	D0	4E	80	00	00	80	00
>90	35	D0	4E	80	00	00	80	00
>90	35	D0	4E	80	00	00	80	00
>90	35	D0	4E	80	00	00	80	00
>00	35	80	80	80	00	00	80	00
>00	35	80	80	80	00	00	80	00
>00	35	80	80	80	00	00	80	00
>AV	F0	F1	F2	F3	AH	AF	FF	AH

(HEXADECIMAL NOTATION)

FIGURE 15 SAMPLE OF INPUTS FOR NORMAL OPERATION AND NUMERIC FEEDBACK

TABLE 5

COMMAND SUMMARY OF SOFTWARE

'CNTRL O'	BACK SPACE/CURSOR LEFT, DELETE LAST CHAR
'='	REPEAT LAST STRING
'+'	END OF LINE, GENERATES CR/LF
'@'	PUNCH A TAPE OF LAST STRING
'CR'	END OF STRING
'#'<CR>	INITIATE WHISPERING
'$'<CR>	INITIATE VOICING (DEFAULT)
'ε'<CR>	INITIATE PRINT OPTION
'%'<CR>	INITIATE SYNTHESIS (DEFAULT)
'ESC'<CR>	RETURN TO MONITOR (MIKBUG)
'*'<CR>	CLEARS SYNTHESIZERS PARAMETERS
'/'<CR>	CAUSES THE FUNDAMENTAL FREQUENCY TO INCREASE
'\'<CR>	CAUSES THE FUNDAMENTAL FREQUENCY TO DECREASE
' '<CR>	SETS UP A TIME DELAY BETWEEN WORDS

NOTES:

<CR> INDICATES A CARRIAGE RETURN

ALL COMMANDS SHOWN FOLLOWED BY A <CR> MAY BE USED

IN THE STRING AND EXECUTED AS SYNTHESIS PROGRESSES

UPPER CASE CHARACTERS ARE UNSTRESSED

LOWER CASE CHARACTERS ARE STRESSED

TABLE 6

STRUCTURE OF PHONEME TABLE

Byte	Field	Mnemonic	Function
1	1	L1	First Character of Phoneme Label
2	1	L2	Second Character of Phoneme Label
3	2	AV	Voicing Amplitude
4	2	F0	Voicing Frequency
5	2	F1	First Formant Frequency
6	2	F2	Second Formant Frequency
7	2	F3	Third Formant Frequency
8	2	AH	Aspiration Amplitude
9	2	AF	Frication Amplitude
10	2	FF	Frication Frequency
11	2	AN	Nasal Amplitude
12	3	T1	Total Duration of Phoneme
13	3	T2	Onset Time for F1, F2, F3, and FF
14	3	T3	Transition Time for F1, F2, F3, and FF
15	3	T4	Onset Time for AV
16	3	T5	Transition Time for AV
17	3	T6	Onset Time for AH
18	3	T7	Transition Time for AH
19	3	T8	Onset Time for F0
20	3	T9	Transition Time for F0
21	3	T10	Onset Time for AF, and AN
22	3	T11	Transition Time for AF, and AN

and AN, however, can be grouped together since frication is excluded in the synthesis of nasals and vice-versa.

Basically, the program operates as follows: A phonetic command string is deposited in a buffer. The phonetic information is later sequentially read from this buffer as synthesis progresses. Editing of the phonetic string is provided for by backspacing from an input terminal and correcting the error. Provisions for accepting paper or magnetic tape input are also incorporated in the software. The carriage return is interpreted as the end of the string, therefore the character '+' is used to generate a carriage return and line feed for those machines that do not generate them automatically.

Receipt of carriage return causes the synthesis to begin by reading the first phoneme in the buffer and comparing it with field 1 of each entry in the data table. If the phoneme is not found in the table, an error message to this effect is generated and synthesis resumes on the next phoneme. Should a match occur, the steady state and timing parameters of fields two and three are sorted and stored in several buffers. These are arranged such that each steady state parameter is assigned three time intervals. These are the onset time, transition time, and total phoneme duration (Figure 16). The onset is that period of time at the beginning of the phoneme that the parameter sent to the synthesizer remains at its previous value. The transition time is the duration of the linear change from the past value to the new steady state value. Transitions

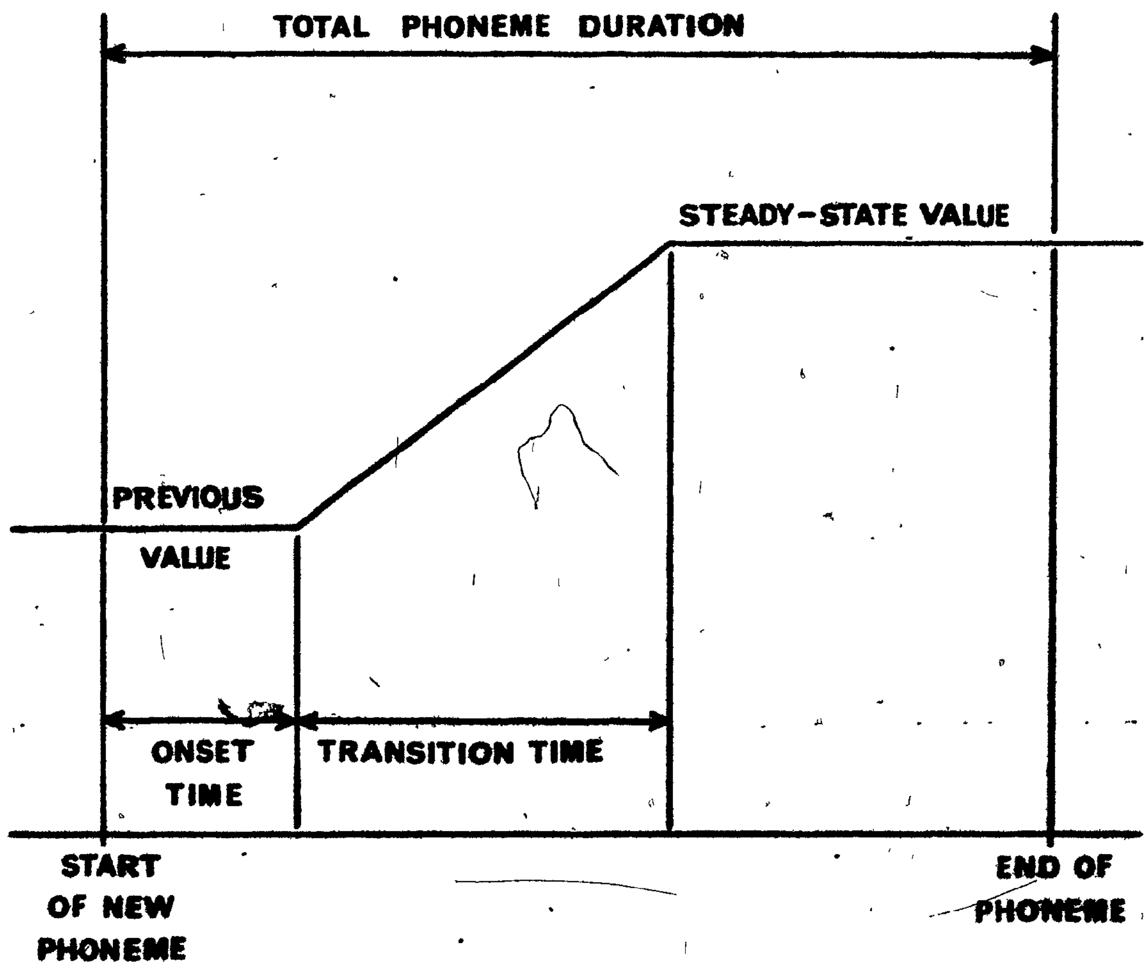


FIGURE 16 PROGRESSION OF A PARAMETER THROUGH TIME

are generated by linear interpolation. The phoneme duration is the total time allotted the phoneme and in effect, determines the duration of the steady state value of the parameter in question.

In this manner, it is possible to produce most of the sounds of speech. The rules of synthesis appear in the data tables and are explained for the following categories:

5.1.1 Vowels and Liquids

Vowels and liquids are the easiest sounds to synthesize since both are always voiced with clear, sharply defined formant structures. Vowels contain higher acoustic power than consonants because during their utterance, there are no constrictions in the vocal tract. Both vowels and liquids can be identified according to their formant frequency location and length of formant transitions. Vowels have short formant transitions and a lengthy steady state. Liquids, on the other hand, have lengthy formant transitions and a relatively short steady state.

These sounds are produced by the voicing source and formant network of the synthesizer. Liquids such as /WW/ /YY/ terminate as soon as the formants finish their transitions whereas /RR/ and /LL/ are completed only after remaining in the steady state for some time. Although vowels have similar characteristics to the latter liquids, they are divided into categories of long and short vowels. Fortis vowels, /AA/, /AE/, /AW/,

/AO/, /ER/, /IY/ and /UW/ dwell in the steady state longer than lenis vowels, /EH/,/IH/ and /UH/. Information re formant frequencies and timing is given in the Phoneme Table in Appendix C.

5.1.2 Fricatives

Fricatives are characterized by the occurrence of sustained noise. They are produced by forcing air through a constriction in the vocal tract which produces a turbulence that acts as the noise source. The vocal tract ahead of this constriction forms resonances in the noise spectra while the cavity behind causes anti-resonances. The latter restricts the noise component of the fricative spectra to above 2 KHz.

The specific cut-off frequency of the fricative spectra is largely determined by the location of the constriction and consequently is important in identification of phonemes. In voiced fricatives, the vocal cords are set in vibration and the resulting spectra contains weak formant structures in addition to noise. Unvoiced fricatives contain only noise.

A synthesized fricative is produced by a combination of the noise source, fricative resonator, voicing source and formant network. Unvoiced fricatives use the resonator and noise source only. Formant transitions for fricatives are much more rapid than for vowels and amplitude (AF) varies rapidly. Resonance frequencies and relative frication amplitudes are given below.

Phoneme Voiced	Phoneme Unvoiced	Fricative Resonator Frequency (FF)	Relative Amplitude (AF)
ZH	SH	2520 Hz	1.0
ZZ	SS	4894 Hz	0.8
VV	FF	7289 Hz	0.6
TE	TH	14160Hz	0.6

5.1.3 Nasals

Nasals are the result of a complete closure of the vocal tract and lowering of the velum which force sound through the nasal passages. The cavity behind the constriction of the vocal tract acts as a resonator; absorption at its natural frequencies causes anti-resonances in the spectra. The nasal passages cause resonances to occur which are broader than vowels due to the increased surface area and convolutions of the nasal tract.

Spectrally speaking, nasals differ from vowels by the presence of a low frequency "voice bar" formant. Transitions of the second and third formant help determine the point of articulation and the type of nasal involved.

Nasals are synthesized by using a nasal resonator, a voicing source and a formant network. Formant transitions are relatively smooth and similar to vowels. Nasal amplitude (AN) however, changes very abruptly from fully off to fully on causing the nasal formant to appear very suddenly. Voicing amplitude has to be reduced slightly to accommodate the effect of the parallel configuration of the nasal branch and the additional sound energy that it carries.

5.1.4 Aspirant

The aspirant /HH/ is synthesized with the voicing and noise sources in conjunction with the formant network. Relative to the following vowel, the amplitude of the aspirant is subdued. During transitions, the voicing amplitude rises to meet that of the vowel and aspiration decreases to zero.

5.1.5 Stops

Stops are characterized by a period of silence of about 100 ms, followed by a burst of noise; for voiced stops voicing accompanies the noise burst whereas in unvoiced stops a period of about 50 ms of aspiration precedes the onset of voicing. Following the noise burst the formants change towards their new target values. The principal characteristic of stops is the rapid change in amplitudes. Timing is critical for, if the onset of voicing is too long, the formant transitions are not heard clearly and perception of the stop becomes difficult.

Synthesis of stops involve both voicing and noise generators as well as the frication resonator and formant network. The period of silence is synthesized with a separate 'phoneme' /QQ/ which turns off both the voicing and noise sources and sets the other parameters to neutral values. The noise burst voice onset and transitions are generated by the following phoneme. For example, Sat becomes /SSAEQQTTQQ/. Figure 17 shows the six stops and their characteristics.

5.1.6 Dipthongs and Affricates

The dipthongs /OY, AY, EY, OW, AW/ are synthesized as two successive vowels, i.e., /OY = AOIY, AY = AAIY, EY = EHIY, OW = AOUW, AW = AAUW/. Their characteristics and rules, therefore, follow those of the vowels and liquids.

The affricates /CH/ and /JJ/ can be generated by /TTSH/ and /DDZH/ respectively. It was discovered, however, that the preceding stop is not critical to perception so long as it is voiceless and voiced respectively. For this reason, the parameters of /SH/ and /ZH/ are altered so as to have rapid transitions and changes in amplitudes. These new phonemes, /CH/ and /JJ/, give stop like qualities to /SH/ and /ZH/ and when preceded by "QQ" give better results than previous methods.

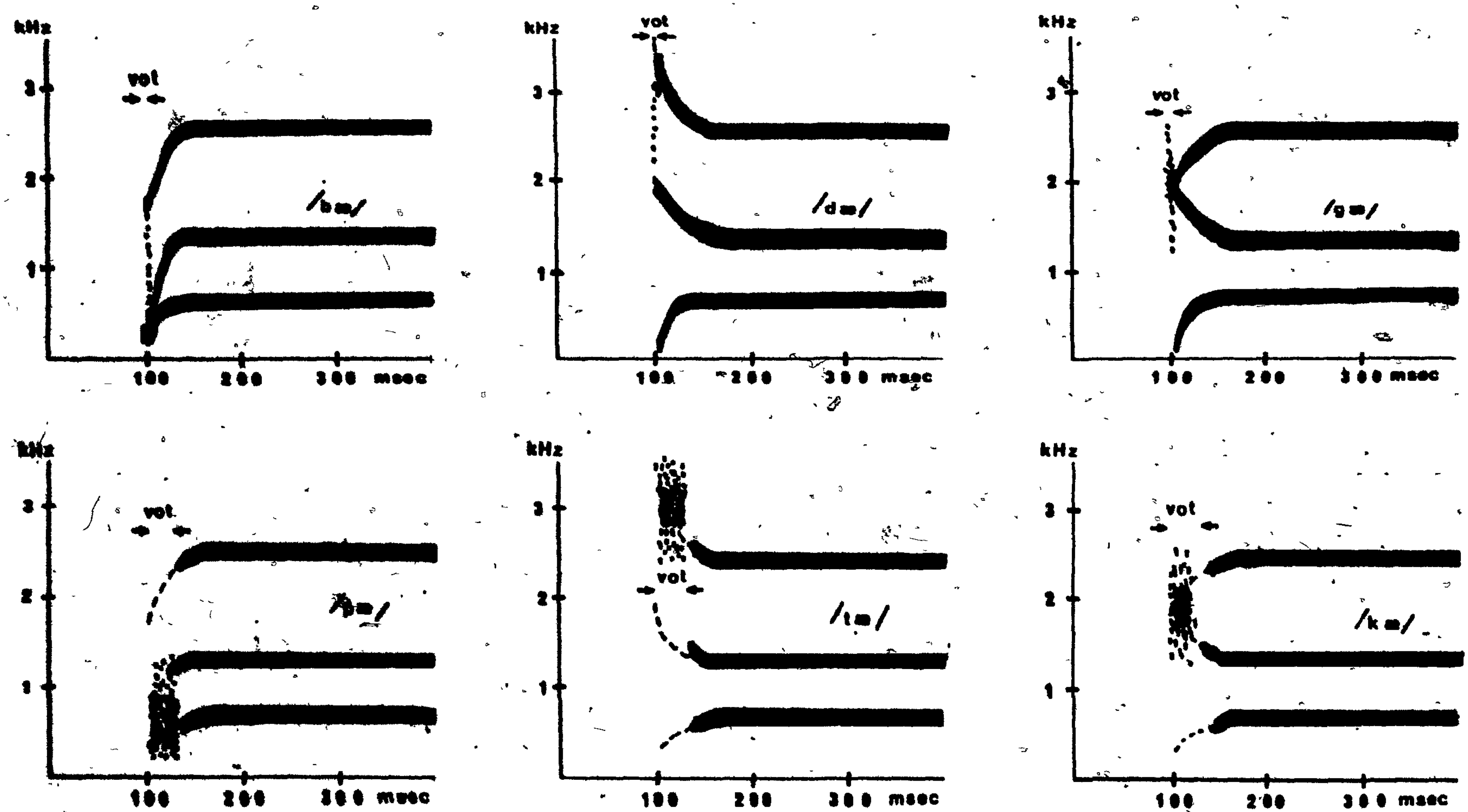


FIGURE 17 STOPS AND THEIR CHARACTERISTICS

5.2 Software Description

The starting location for the program is at 0100 hexadecimal (Reference line 94 of the listing, Appendix A) which initializes commands for clearing variables VOICE and HRDCPY, disabling the whispering and dump options. OLD DATA buffer amplitude parameters are then cleared and resonance parameter values are set to their mid-ranges. Reference Figure 18a.

Following the initialization procedure, the program proceeds to read a line from the console character-by-character and stores this in a RAM buffer. At the same time, back spacing, tape preparation, TTY formatting and re-synthesis of the last string entered are processed if special control characters are present. (The RAM buffer starts at 0500 hexadecimal). A character string prompt is printed by carriage return (C/R), a line feed (L/F), three nulls and a '>' symbol. The X register is now set to the beginning of the buffer and accumulator A is loaded with the ASCII value of any character introduced from the console. Exceptions are:-

If the character introduced by the console is a C/R then an EOT is stored in the location directed by the X register. Automatically a LF is returned to the console to show line termination (end of buffer). The program then proceeds to LINK 8 which initiates synthesis of the buffer content.

If the input character is CNTRL O and the X register does not correspond to the location of the start of the buffer (0500 hex.) then the X register is decremented and a BS (back space or cursor left) is transmitted to the console and the program continues by reading another character from the console. In the situation where the X register is at 0500, the CNTRL O command is ignored.

If the input character is a "+", then the prompt (C/R ; LF ; Nulls ; >) is printed and the process returns to read another character from the console. This feature is added as a facility for those terminals which do not possess an automatic C/R and LF when the end of a line is reached. Thus entering a "+" symbol will cause a CR, LF without terminating the buffer. This permits the development of long strings.

If the character is a '"' then the program proceeds directly to LINK 8. Since this leaves the contents of the buffer undisturbed it provides a convenient means of repeating synthesis of a string. Care must be taken in the use of this control as it relies in an embedded EOT in the string.

If the input character is a '@' and the X register corresponds to the start of the buffer, then a DC2 ASCII character is sent to the console to activate any automatic punch or tape facility. This is followed by 25 nulls, the string for synthesis, another 25 nulls, and finally a DC3 ASCII character to deactivate the punch or tape facility.

Should a '@' be received and X register location not equal the start of the buffer, then an EOT is stored to terminate the string and then activate the punch or tape facilities.

If the input character is a 'NUL' it is ignored and the program returns to read another character. NULLS will only be used as leaders on tape functions.

All the other characters received are interpreted as part of the string and stored in the buffer as directed by the X register. After each character, the X register is automatically incremented and another character read from the console. Thus character-by-character, the phonetic string is assembled until either a C/R or a '"' control is received. Either condition will transfer the program to LINK 8 where the string is prepared for processing.

At LINK 8 (Reference Figure 18b) the X register is reset to the start of the buffer (0500 Hex.) and the first character is read from the buffer. This character could either represent control data or a phoneme. These control characters change the parameters used for synthesis of phonemes and are separated from incoming information.

If the incoming character is a "/", then the variable F0 is increased by 5, (i.e., F0 = F0 + 5). This increases the fundamental frequency of all subsequent synthesis by a factor of 1.03676. The X register is then incremented and another character read from the buffer. (Refer to Figure 18c for details of LINK 13).

If a "\" is read, the variable F0 would be decreased by 5, reducing the fundamental frequency by the factor 0.96453 (reciprocal of 1.03676). The X register is incremented and another character read from the buffer. (Refer Figure 18c Link 14).

The variable F0 provides a means of altering the fundamental frequency to something other than that stored in the look-up table. Thus by using F0 as an offset of the tabled value controls / and \ will result in changes of $\pm$ 125 which will remain in effect until either is changed or a power-down situation occurs.

If the incoming character is an asterisk "*", then the parameters stored in the old data buffer are cleared. Effectively this causes a period of silence equal to one interrupt cycle and is used in the synthesis of stops and occasional initialization of synthesis. The X register is again incremented to read the next character.

If the character read is "#", then the VOICE parameter is incremented as well as the X register. Another character is then read from the buffer.

If the character is a "$" then the VOICE parameter is cleared. Again, the X register is incremented and another character read.

The VOICE parameter is an indicator that the passage following is to be voiced or to be whispered. Whispering is achieved by equating the aspiration and voicing amplitudes and then reducing the voicing to zero.

If the character "&" is read from the buffer, then the parameter HRDCPY is incremented as is the X register and another character is read.

If the character read is a "%" then the HRDCPY is set to zero before incrementing the X register and return to read another character.

The HRDCPY variable determines if a dump of synthesis parameters will take place. The dump will list all the parameters used by the synthesizer for each interrupt cycle. When this feature is used synthesis is slowed down to accommodate the printer or terminal used. This option has proven to be extremely valuable in developing look-up tables.

If the character "ESC" is read, then the starting location of the program is saved as the program counter such that it can be restarted easily by the MIKBUG control G and the program returns to the monitor (MIKBUG).

Finally, if the character read from the buffer is an EOT, the end of the buffer has been reached and the program returns to its starting point (LINK 1).

Apart from the above control symbols, all other data will be processed in pairs of characters representing phonemes. The first is stored as INP1 and the X register is incremented and the next character of the buffer is read and stored as INP2. The X register which is being used as the buffer pointer is then incremented to the next location and stored in the variable THERE.

The first character read (INP1) is then compared with the first character in the look-up table. If these are not equal, then the look-up table pointer is advanced to the next table entry where INP1 is again compared with the first characters in that entry. This continues until either a match is found or until the end of the table is reached (indicated by an asterisk). Should the latter occur an error message is generated, the X register loaded with the stored value in THERE and returned to LINK 9 where the next character will be read from the buffer.

If a match between INP1 and the first character of field one of a particular entry in the look-up table is found, then the second character read (INP2) is compared with the second character of field. If these are not equal, then the look-up table pointer is advanced to the next entry and INP1 is again compared to the first character of field 1 in that entry. If INP2 is equal to the second character of field one, however, the table contents are sorted and stored into the buffers NEW DATA, ONSET, and TRANSITION. The deltas $\left(\text{delta} = \frac{\text{new data - old data}}{\text{transition}}\right)$ are then formed for each synthesis parameter and stored in DELTA.

Five buffers contain all the necessary information for synthesis. The buffer OLD DATA contains the values of the synthesis parameters for the last phoneme when its frame time was exceeded. The NEW DATA buffer contains the steady-state values of synthesis parameters for the present phoneme to be synthesized. The buffer ONSET contains the onset times for each synthesis parameter whereas TRANSITION contains the transition times for each synthesis parameter. Finally the DELTA buffer contains the step height necessary to change linearly from OLD DATA to NEW DATA within the transition time.

Once all buffers are readied, the program checks the variable VOICE. If it is non-zero, then whispering is desired and the parameter AH is made equivalent to AV then AV is cleared. If VOICE is zero, AV and AH are not changed. The offset value F0 is then added to the fundamental frequency parameter. Frame time T1 is then decremented and compared to zero. If zero, the input buffer pointer is restored and the program reads the next phoneme. If T1 is non-zero, the program stops and awaits an interrupt of the first parameter.

The interrupt routine transfers the information in the OLD DATA buffer to the synthesizer. Experiments with a variable interrupt determined that a good quality of speech generation can be achieved providing the interrupt time is less than 20 ms.

On completion of the interrupt, the onset time of the first parameter is compared to zero. If non-zero, it is decremented and the pointer moved to the next parameter and its onset time is again compared to zero. If any onset time is less than or equal to zero, its transition time is compared to zero. If the transition time is less than or equal to zero, then that parameter of the buffer OLD DATA is replaced by the one contained in NEW DATA. The pointer is then moved to the next synthesis parameter and if there are remaining parameters the next parameter onset time is again compared to zero.

In the case where the transition time is greater than zero, the appropriate delta is added to the corresponding parameters in OLD DATA and the result is compared to the value in NEW DATA (the target). The purpose of this routine is to determine if an overshoot occurs. If it doesn't, the result is stored in OLD DATA. If it does overshoot, the value of NEW DATA is stored in OLD DATA instead of the result. Determination of the overshoot criterion is complicated by the microprocessor's interpretation of the data as being signed (i.e., -128 to 128) whereas the synthesizer assumes it is unsigned (0-256). Consequently the signs of the result, DELTA and NEW DATA are determined and thus form the basis of the decision. This routine is best described by the flow chart, reference Figure 18f.

The process continues until all the synthesis parameters have been converted. The value of the variable HRDCPY is then compared to zero. If non-zero, the parameters of the buffer OLD DATA are displayed or printed on the system console and then the frame time T1 is decremented and continues the loop. On the other hand, if HRDCPY were zero, then the frame time will be decremented immediately and then continue its loop. These cycles will continue until the frame time is zero at which time the program reads another character from the input buffer until an "EOT" is reached. This returns the program to its starting address (0100 HEX).

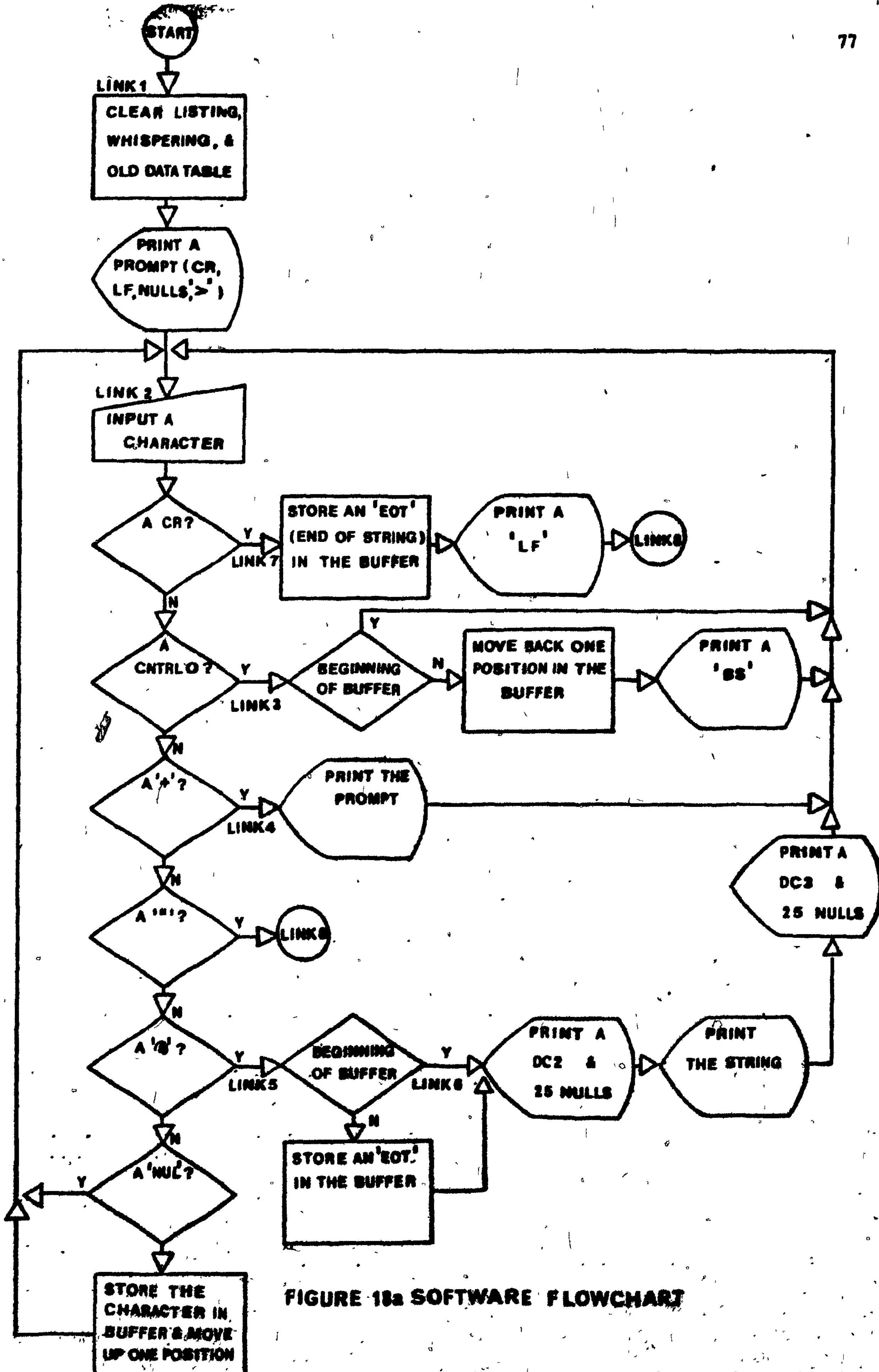


FIGURE 18a SOFTWARE FLOWCHART

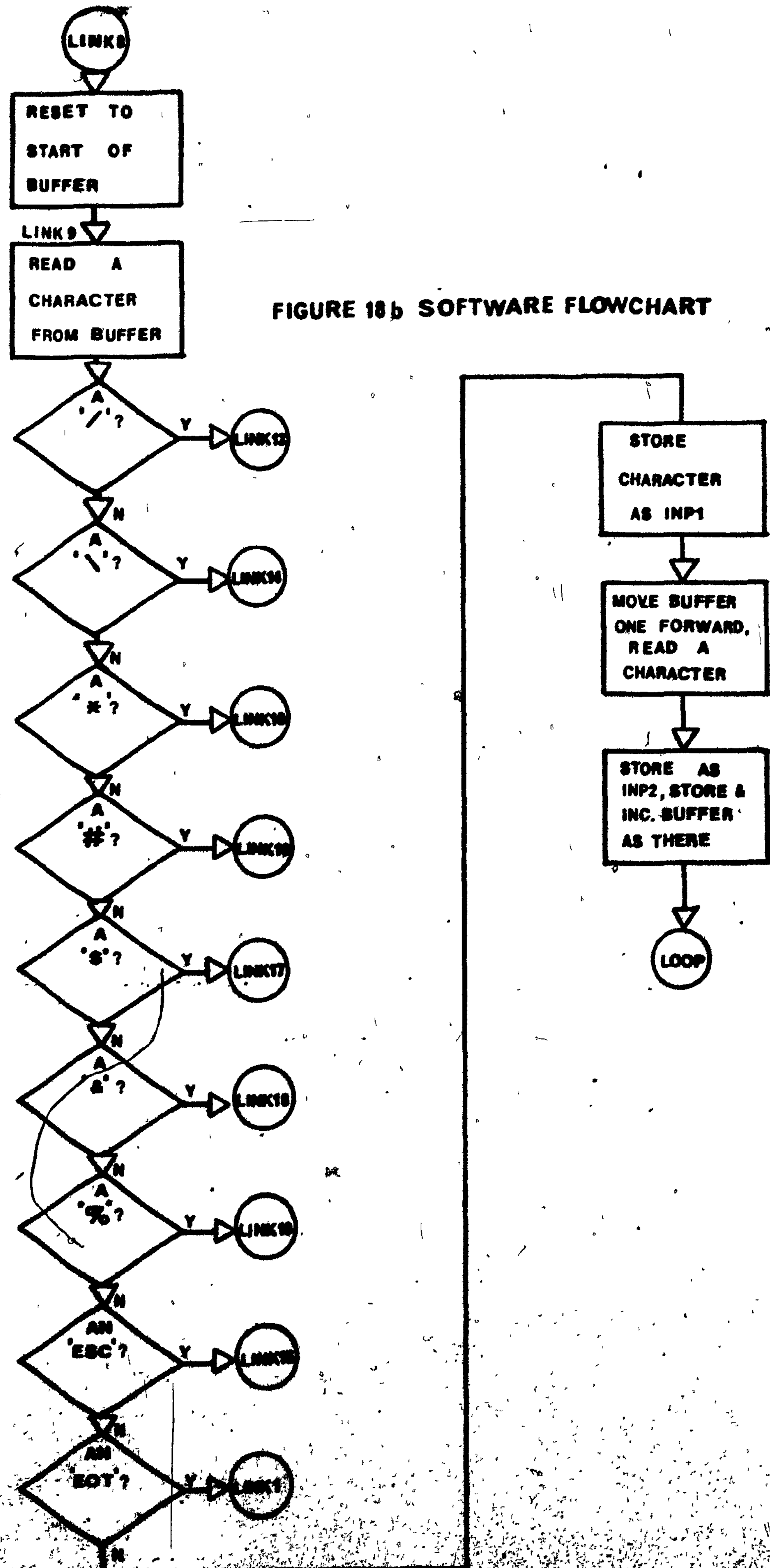


FIGURE 18 b SOFTWARE FLOWCHART

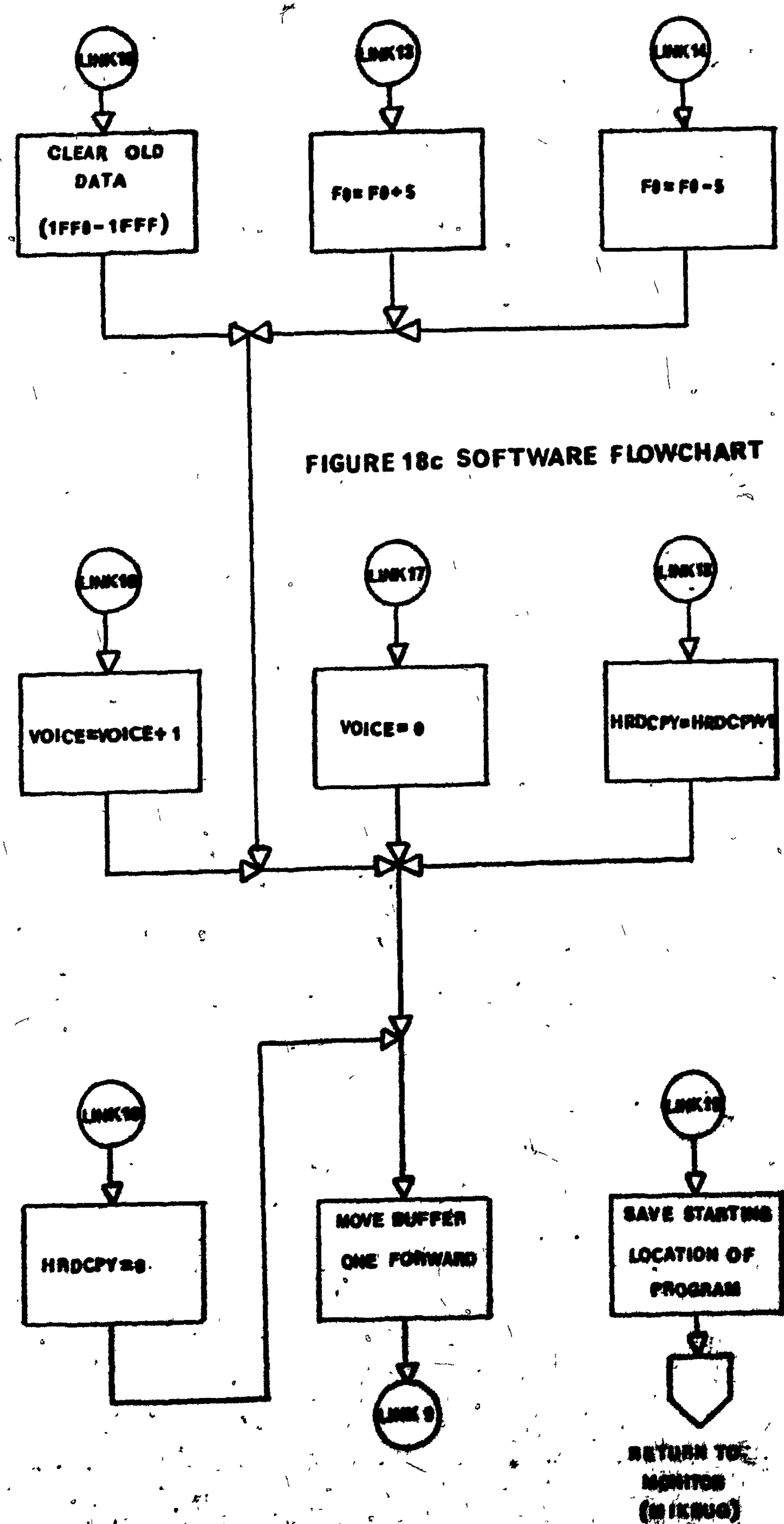


FIGURE 18c SOFTWARE FLOWCHART

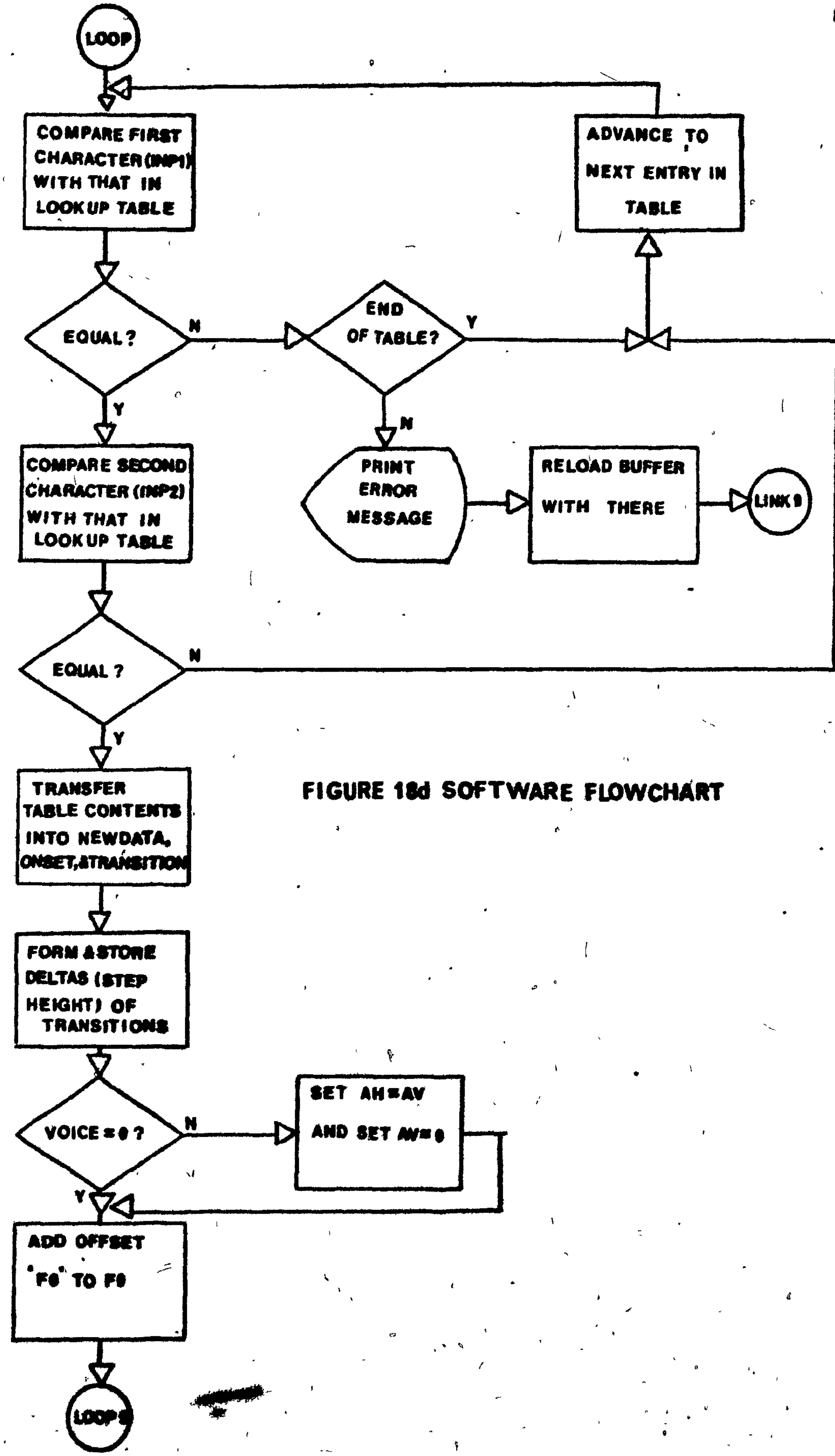


FIGURE 18d SOFTWARE FLOWCHART

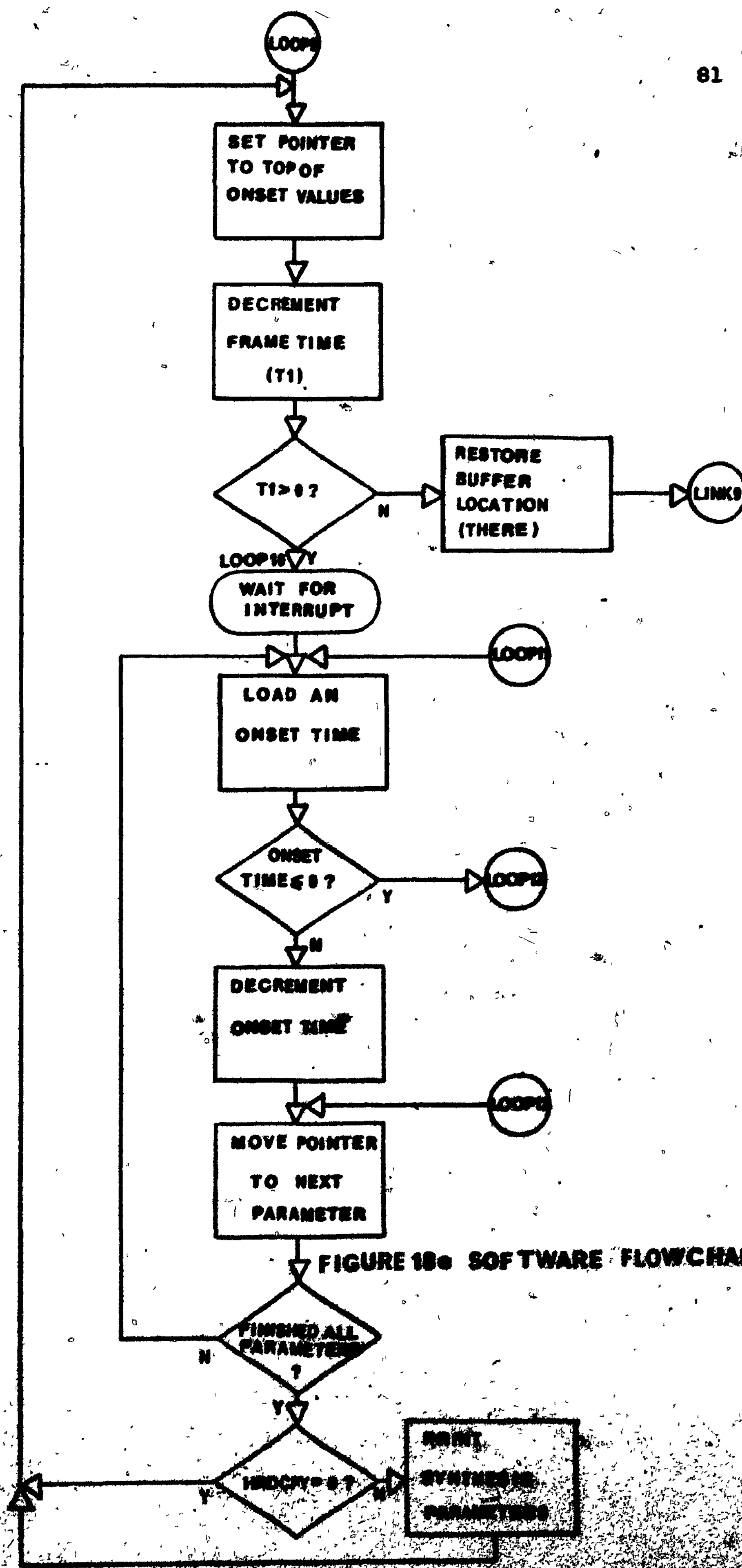


FIGURE 18a SOFTWARE FLOWCHART

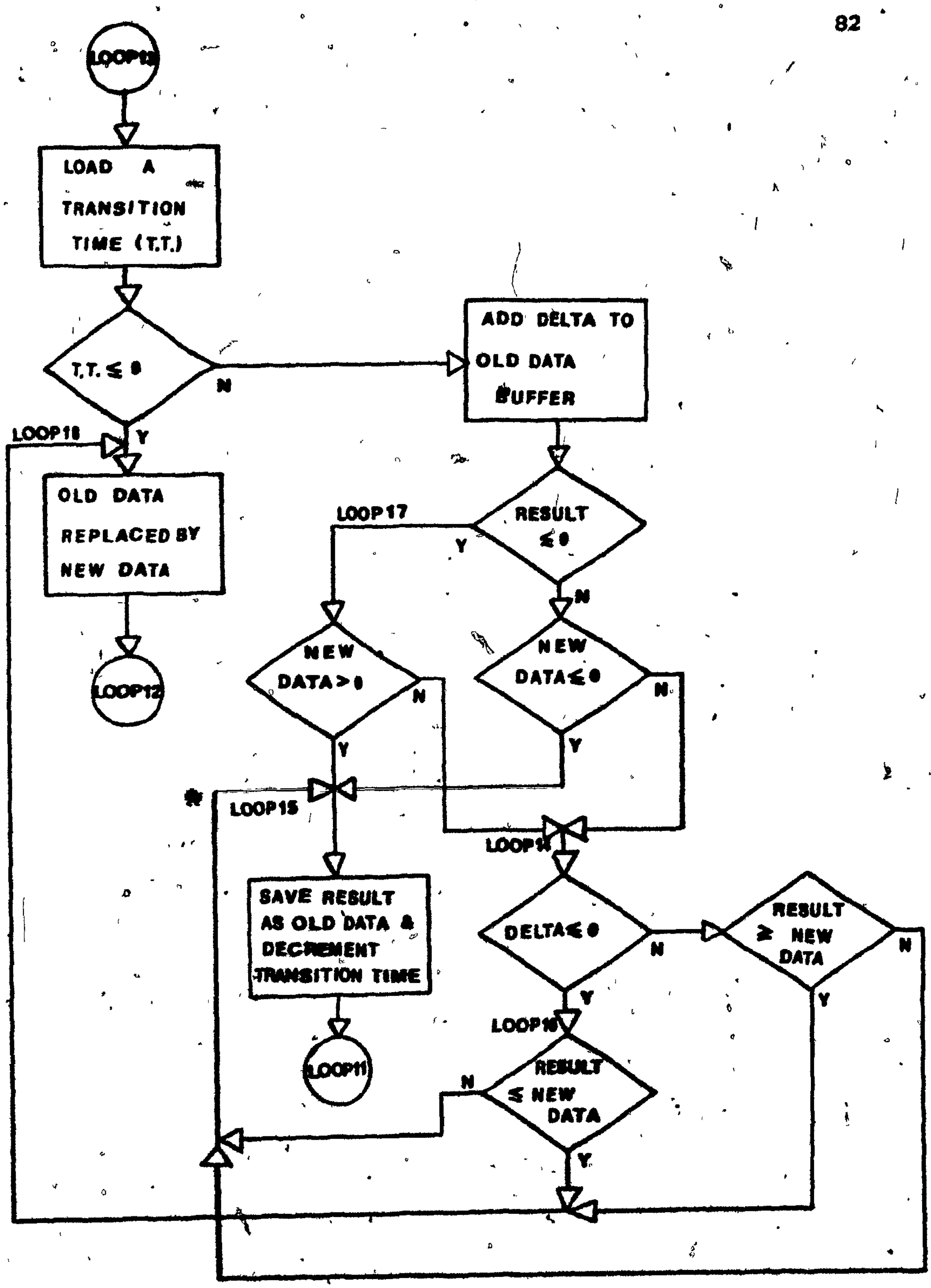


FIGURE 18f SOFTWARE FLOWCHART

CHAPTER VI

RESULTS AND CONCLUSIONS

6.1 Evaluation Criteria

There are a number of requirements that should be met by any synthesis system. The equipment should be versatile and easy to operate. Ideally, the apparatus should incorporate features which enable acoustic, numeric and visual feedback. Facilities for adjusting and testing the effects of one or more parameters, independent of the synthesis strategy are highly desirable. The system should be flexible enough to permit experimentation with various components of speech. It should produce intelligible and if possible, natural speech. The synthesizer should have the capability of a large vocabulary and incorporate economical (minimal) message storage.

6.1.1 Acoustic Feedback

One of the methods of evaluation is what may be termed the "listening test". Acoustic feedback is the process of listening to the results and then making modifications to improve the output. This is useful for improving vowel sounds but has very limited application where categorical perception is involved. With increased exposure to synthetic speech, comprehension increases to such an extent that the objectivity of the listener can be questioned. Therefore, in developing data for synthesis, the subjects were given very short tests and the sounds or words were rotated to prevent both listener fatigue, and eliminate the phenomenon of comprehension by exposure.

6.1.2 Numeric Feedback

The parameter dump feature referred to in the software description will activate a print-out of all parameters sent to the synthesizer each interrupt cycle for all inputs between & and % symbols. This is a very valuable aid in establishing values for all variables and a basis for observing change from one sound to a new sound. Numeric feedback was found to be particularly useful for interpolation between sounds and in establishing timing.

6.1.3 Visual Feedback

Spectrographic and acoustic waveforms give accurate information on frequencies, formants and amplitudes. This graphic information is very useful in verifying transitions, timing and amplitudes. Although visual observations are a powerful method of getting feedback to improve speech output, measurements tend to be time consuming and have to be restricted to specific problem areas.

Waveforms were plotted at L'Institut National de la Recherche Scientifique Télécommunications (INRS) using specialized software. Additional spectrograms were also made on an analog machine at the McGill Department of Linguistics (Vocieprint's Sound Spectrograph Model 4691C).

6.2 Experimental Results

6.2.1 Intelligibility Tests

A number of tests were performed to determine the intelligibility of the speech output of the system. These tests took the form of exposing a group of listeners to a synthesized sound and recording their responses as to what each perceived the sound to be. Listeners' responses were then plotted on what is known as a 'confusion matrix'. These display not only those sounds which are correctly understood but also sounds that are nearly correct. The position in the matrix determines the degree of comprehension of various sounds and the type of errors which are occurring. All the phonemes were tested except /WH/ which is virtually interchangeable with /WW/.

Steady-state vowel sounds used in intelligibility tests were lengthened to approximately one second so as to give sufficient dwell time for the listener. Consonants were generated at their conversational rates. Stops, affricates and sonorants were followed by the vowel /AH/. Nasals were preceded by /IH/ and fricatives produced in isolation.

Five people were tested with vowels and diphthongs, two of them taking the tests twice. Four people took part in testing consonants

RECEIVED

SENT	IY	IH	EH	AE	AA	AH	AO	UW	UH	ER	OY	AY	EY	OW	AW
IY	19	1	1												
IH	5	14	2												
EH		3	16			1	1								
AE				21											
AA				2	10	2	7								
AH				2	1	12	6								
AO					6	1	13		1						
UW								18	3						
UH							2	5	13	1					
ER										21					
OY											17	4			
AY												21			
EY													21		
OW														13	8
AW														1	20

TABLE 7 VOWEL AND DIPHTHONG CONFUSION MATRIX

RECEIVED

SENT	BB	DD	GG	PP	TT	KK	MM	NN	NG	FF	TH	SS	SH	VV	TE	ZZ	ZH	CH	JJ	WW	LL	YY	HH	RR
BB	4			4																				
DD	2	3	2		1																			
GG	3	1	2		2																			
PP	2			6																				
TT			2		5	1																		
KK			1	1	3	3																		
MM							11	1																
NN								7	5															
NG							1	4	7															
FF										9	2	1												
TH										3	8	1												
SS												10	2											
SH													12											
VV													1	6	3	2								
TE														8	3	1								
ZZ												1		5		6								
ZH											1				3	1	7							
CH																		2	4	1		1		
JJ																		1	7					
WW																		1		6			1	
LL																					7	1		
YY																						8		
HH																							8	
RR																								8

TABLE 8 CONSONANT CONFUSION MATRIX

each taking the test once. Results of intelligibility tests are shown in Tables 7 and 8. Confusion with vowels can be attributed to the uniform one second duration of the sound. Fortis and lenis vowels are often differentiated by the duration of the sound. This indicates that a better test could be based on consonant-vowel-consonant groupings at conversational rates.

A second item which is believed to have influence is that most of the subjects used in these experiments have no prior experience in phonetics or with synthetic speech. Although they were supplied with a list of phonemes and several examples, there may have been some confusion. This could result in the selection of the first phoneme in the list that sounds similar. A supporting example would be to consider the vowels /AO/, /AA/ and /AH/ which are in the sequence of the list of phonemes supplied. It was found that /AO/ was incorrectly chosen for /AA/ 33% of the time and for /AH/ 28% of the time. Selection of /AO/ constituted the majority of errors in both cases.

The consonant confusion matrix indicates that voiced stops are a problem. The sounds /BB/, /DD/ and /GG/ are heard as /BB/ 50% of the time for /BB/, 25% for /DD/ and 38% for /GG/. The most probable cause for this is too rapid a transition time for the formants. The sound /BB/ is perceived 50% of the time as /BB/ and 50% of the time as /PP/. This indicates a confusion between voiced and unvoiced sounds and is most likely caused by the voice onset time. There is a problem between the nasals /NN/ and /NG/ . The sound /NN/ is chosen incorrectly as /NG/ 42% of the time and /NG/ chosen as /NN/ 33% of the time. A potential cause would be too rapid a transition of formants or possible masking of the transitions. Voiced fricatives in general are chosen correctly 45% of the time whereas unvoiced fricatives are correct 80% of the time. The errors tend to be scattered among the fricatives and do not indicate any specific cause.

In summary, the overall merits of the system can be represented by the articulation scores. An articulation score is the percentage of sounds heard correctly. The articulation score for consonants was determined to be 65.7%, for vowels and diphthongs 79.0% .

6.2.2 Spectrographic Analysis

A series of spectrograms made on a Voiceprint analog machine are presented (Figures 19a to 19q inclusive). When interpreting this information, it should be remembered that this particular measurement apparatus has a 12 dB dynamic range. This narrow dynamic range means that low energy sounds are not visible, in particular the third formant, and makes some sounds appear better than they actually are.

Apart from lack of the third formant due to dynamic range, the vowels contain the correct spectral content (Figures 19a and 19b). Existence of the third formant and other low energy sounds were verified by sectional spectrograms (Figures 20a and 20b).

Liquids and diphthongs (Figure 19c) have the required characteristics although the transitions in the diphthongs would have been better if made slightly longer.

Unvoiced fricatives (Figures 19d and 19e) appear correct for /ʃ/ and /s/ but /f/ and /θ/ have too much low frequency noise. More accurate analysis at INRS confirms excessive low frequency noise in all unvoiced fricatives with no indication of anti-resonances (Figure 20c). A sharp cutoff causing a definite spectral gap is essential to the correct identification of the phonemes. The source of this problem was determined to be the CT-1 synthesizer and is more fully described in Limitations of the System. Voiced fricatives (Figures 19f and 19g) have the same problems associated with low frequency frication, but are not as much of a problem because of the voicing which fills in the spectral gap. The voiced fricatives, however, have a tendency toward excessive voicing, causing almost vowel-like qualities to be observed. While these fricatives are readily identifiable in an intelligibility test, in connected speech the frication has a tendency to be ignored by the listener as static and consequently intelligibility is lessened.

Unvoiced stops are correct from a spectral standpoint (Figure 19b), whereas voiced stops (Figure 19i) have a voice onset time which is

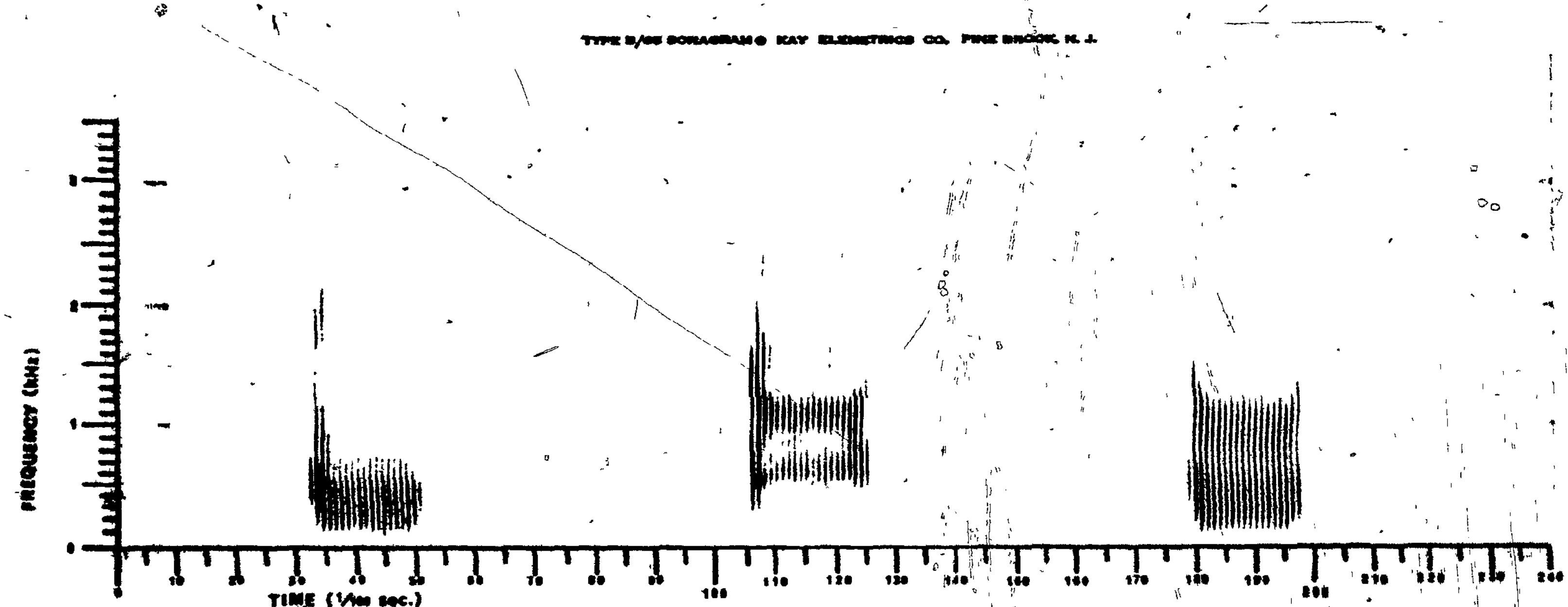


FIGURE 19 a SPECTROGRAM OF SYNTHETIC SPEECH

a

u

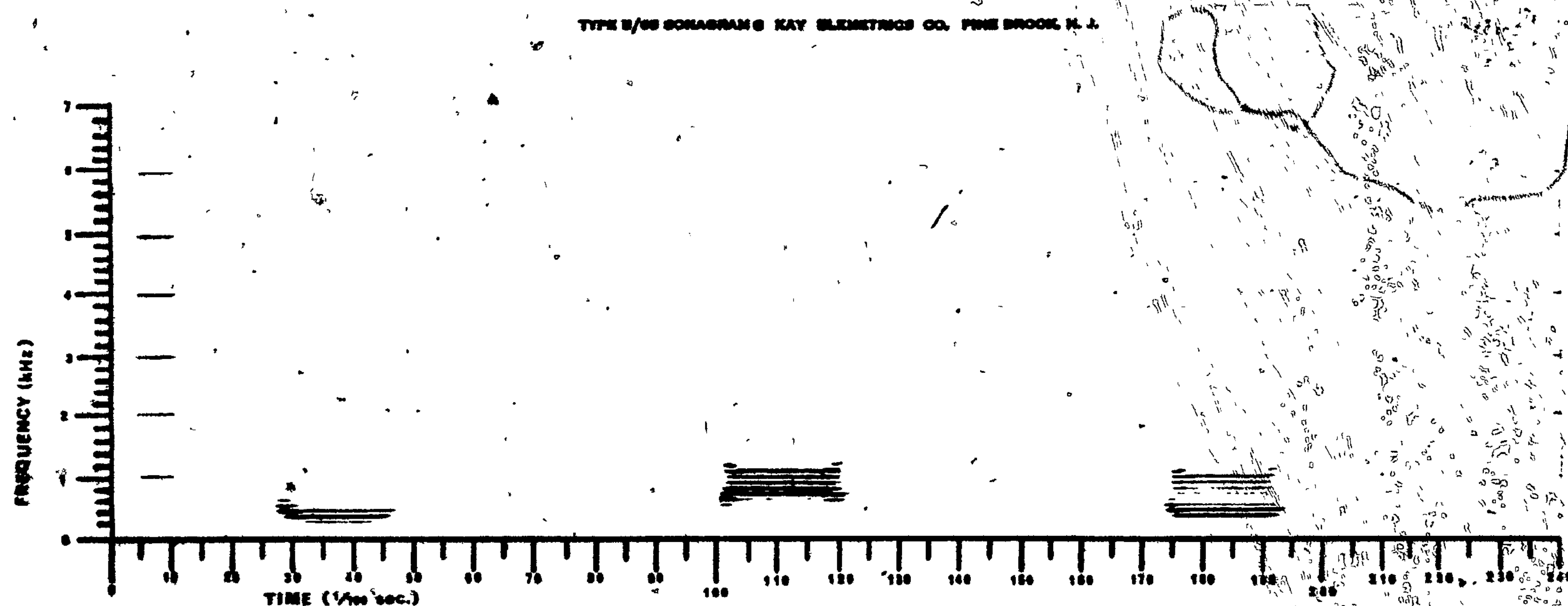


FIGURE 19 b SPECTROGRAM OF SYNTHETIC SPEECH

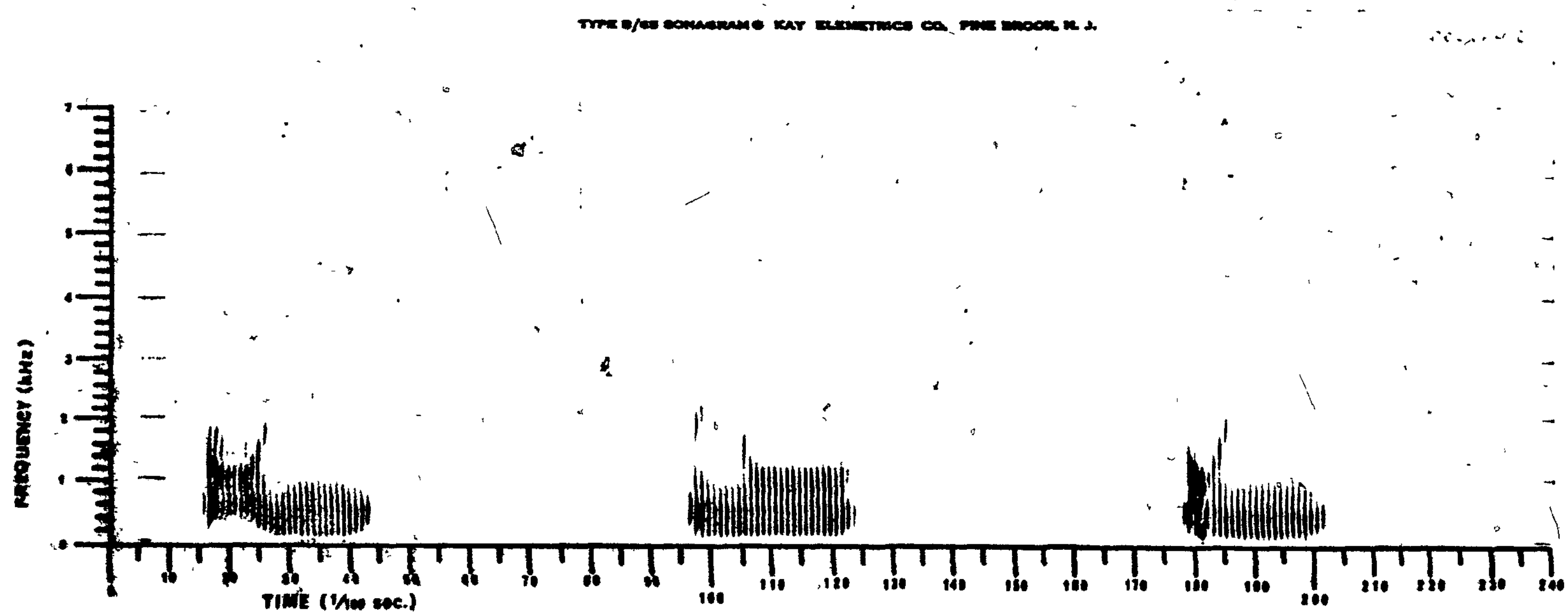


FIGURE 19 c SPECTROGRAM OF SYNTHETIC SPEECH

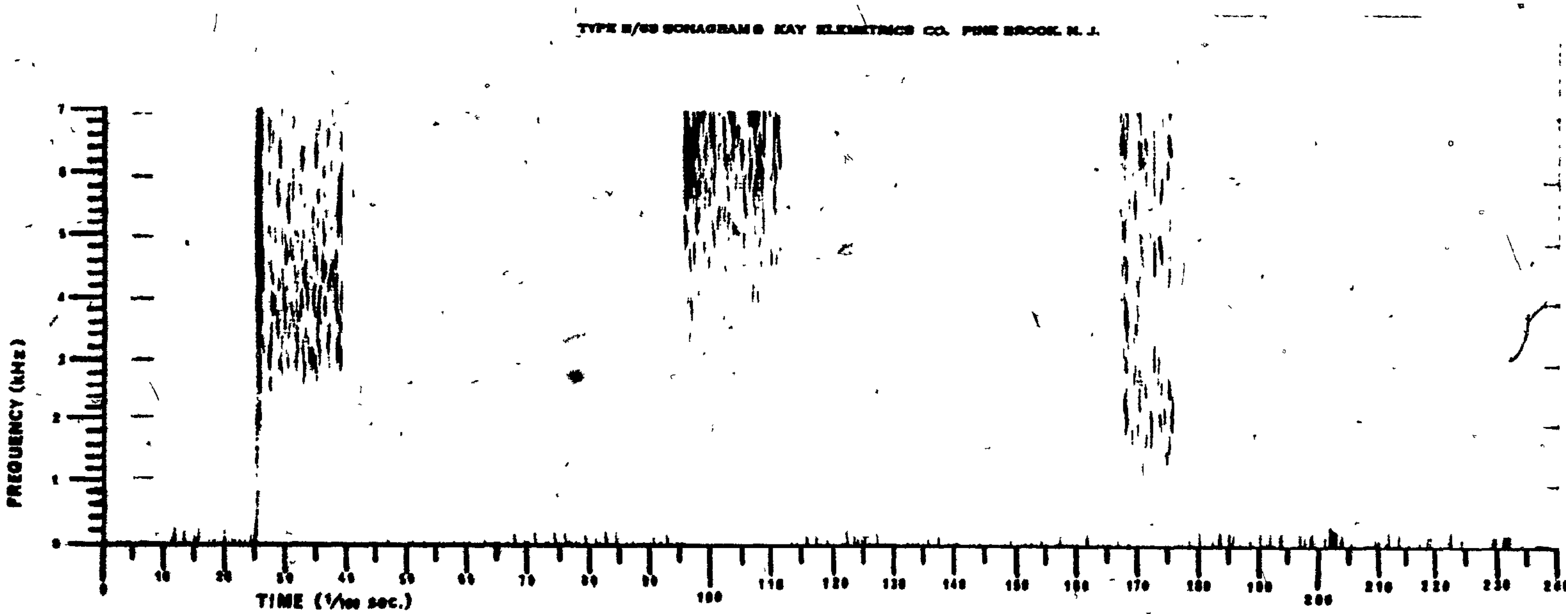


FIGURE 19 d SPECTROGRAM OF SYNTHETIC SPEECH

ʃ s f

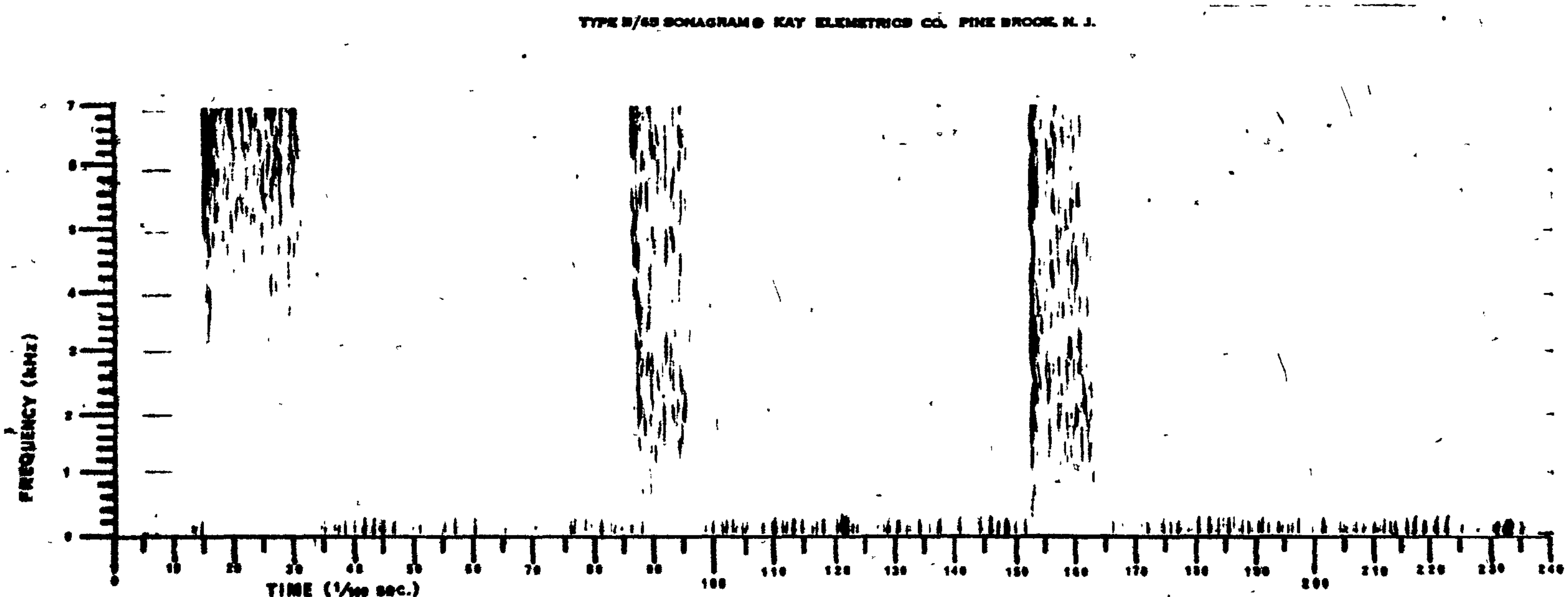


FIGURE 19 a SPECTROGRAM OF SYNTHETIC SPEECH

s f θ

Rec. 4

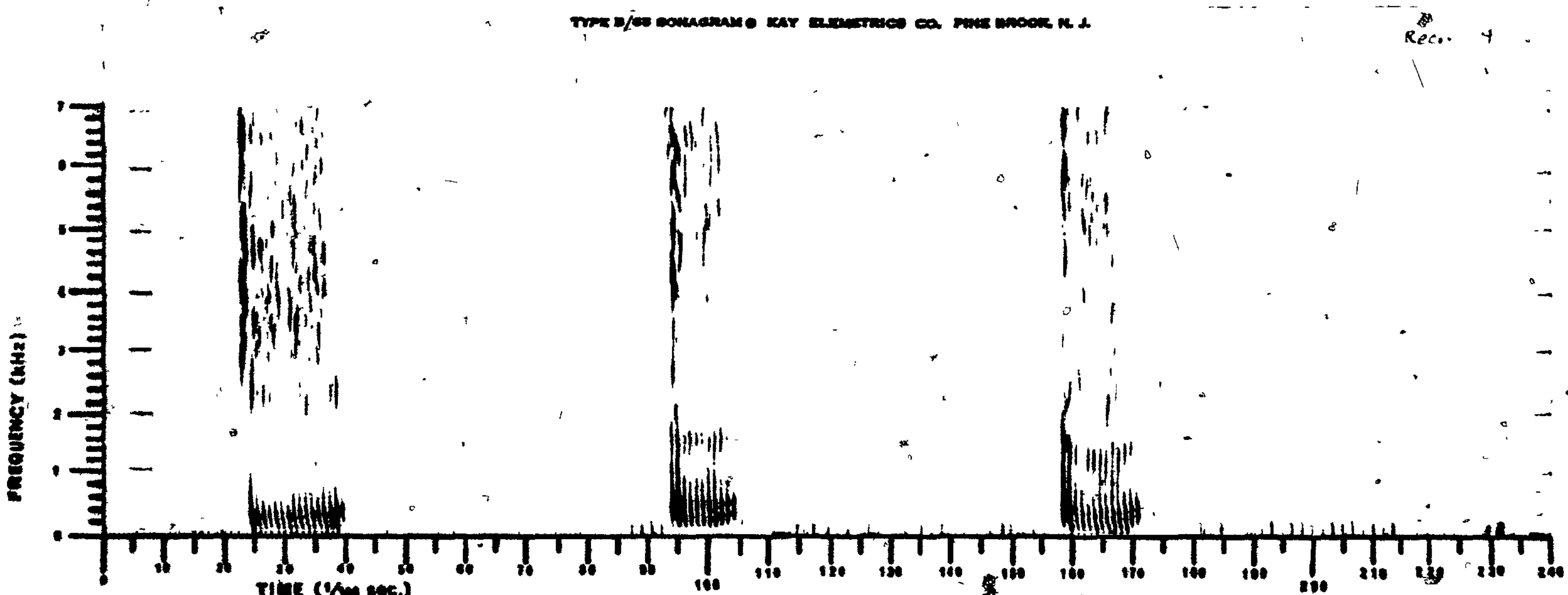


FIGURE 19 (SPECTROGRAM OF SYNTHETIC SPEECH

3 z v

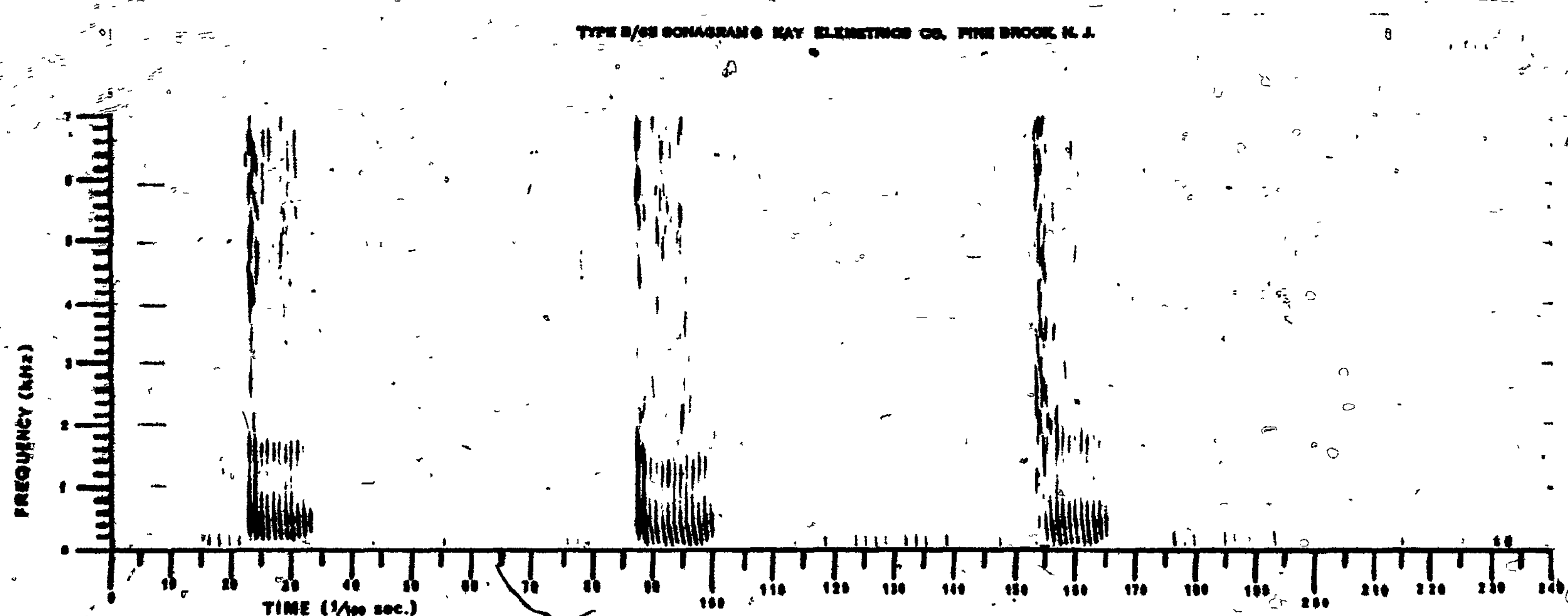


FIGURE 10 g SPECTROGRAM OF SYNTHETIC SPEECH

z v ð

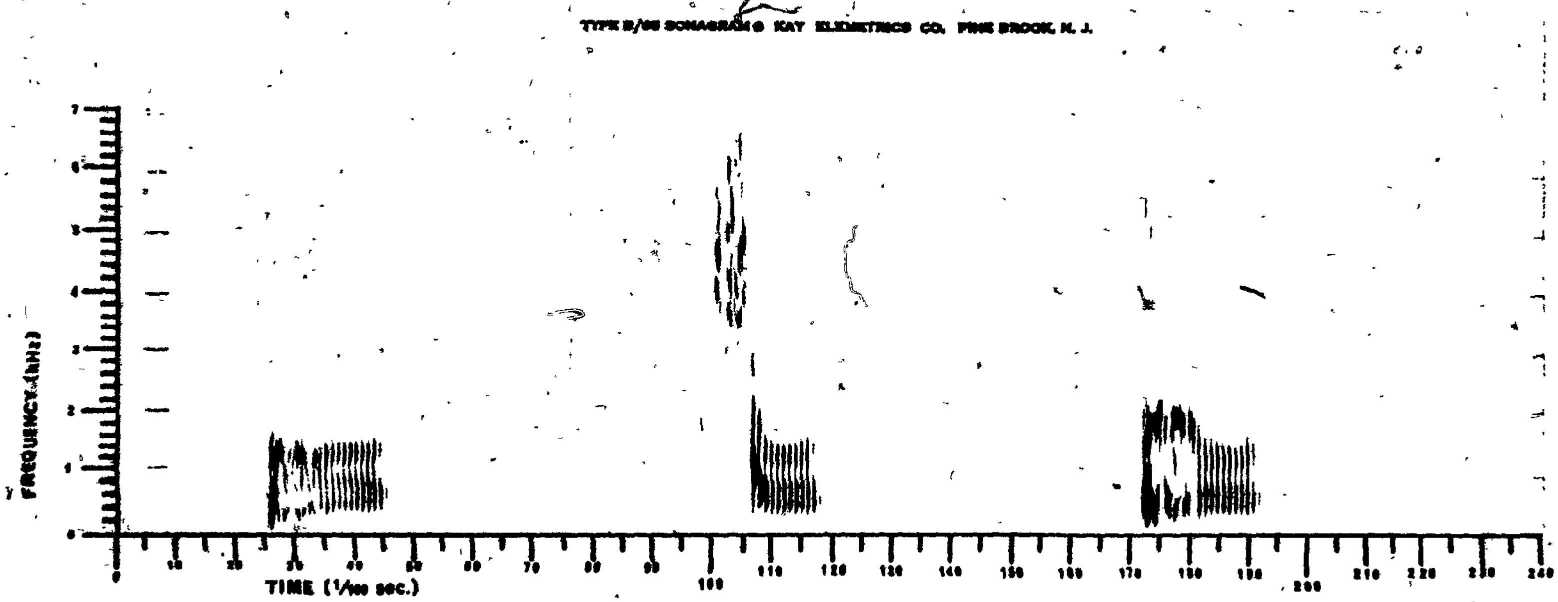


FIGURE 19 h SPECTROGRAM OF SYNTHETIC SPEECH

pʌ tʌ kʌ

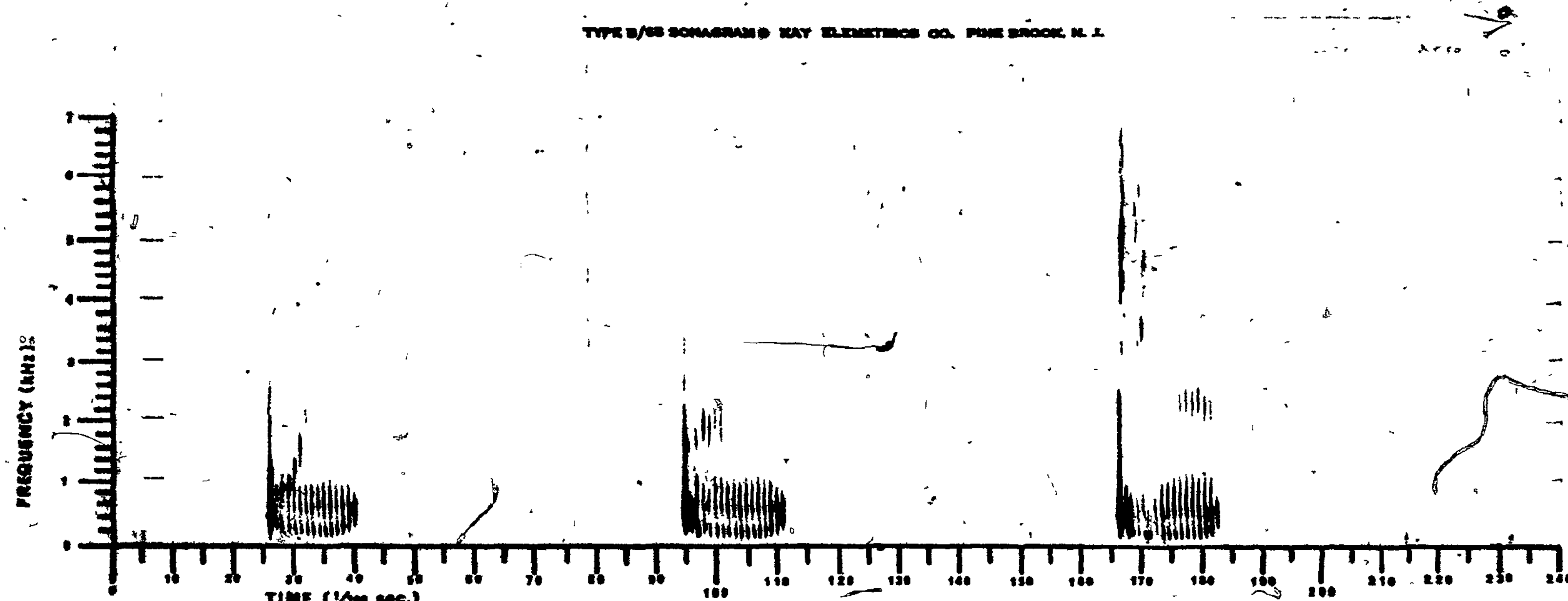


FIGURE 10.1 SPECTROGRAM OF SYNTHETIC SPEECH

bi di gi

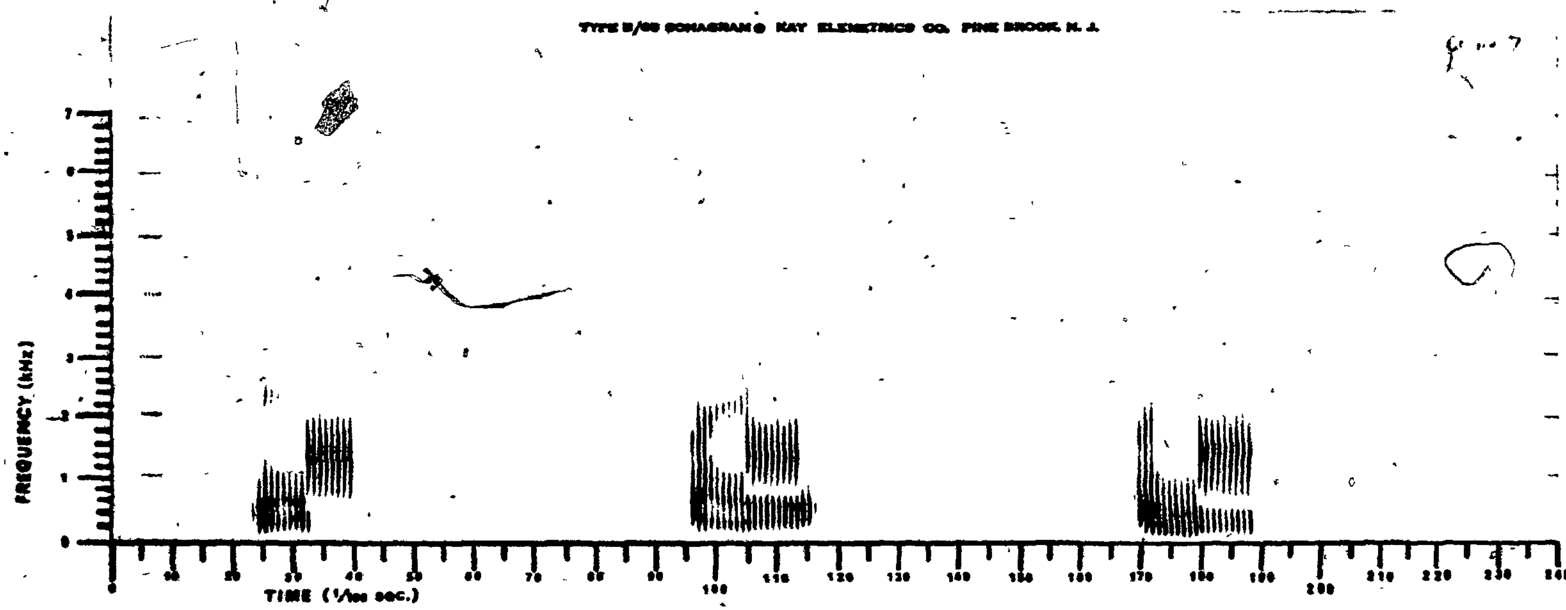


FIGURE 10 | SPECTROGRAM OF SYNTHETIC SPEECH

Im In Iŋ

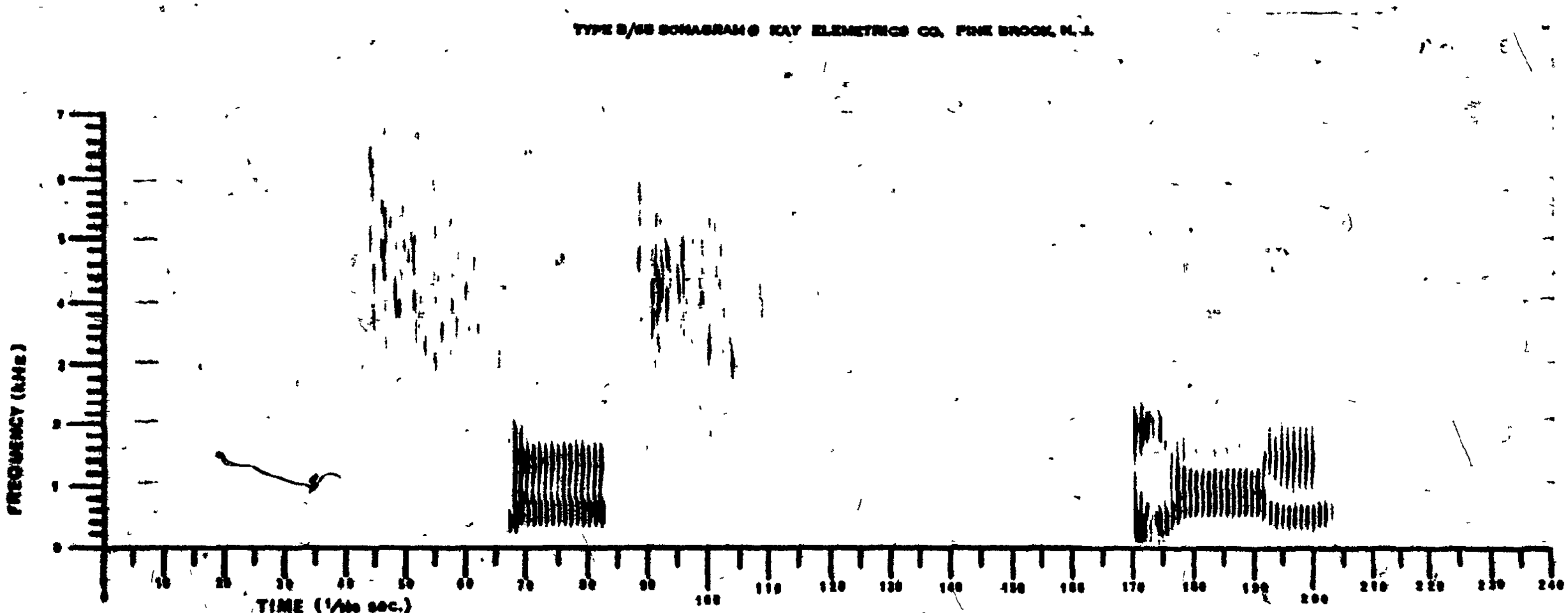


FIGURE 10 k SPECTROGRAM OF SYNTHETIC SPEECH

tʃ ʒ tʃ dʒ ɔ n

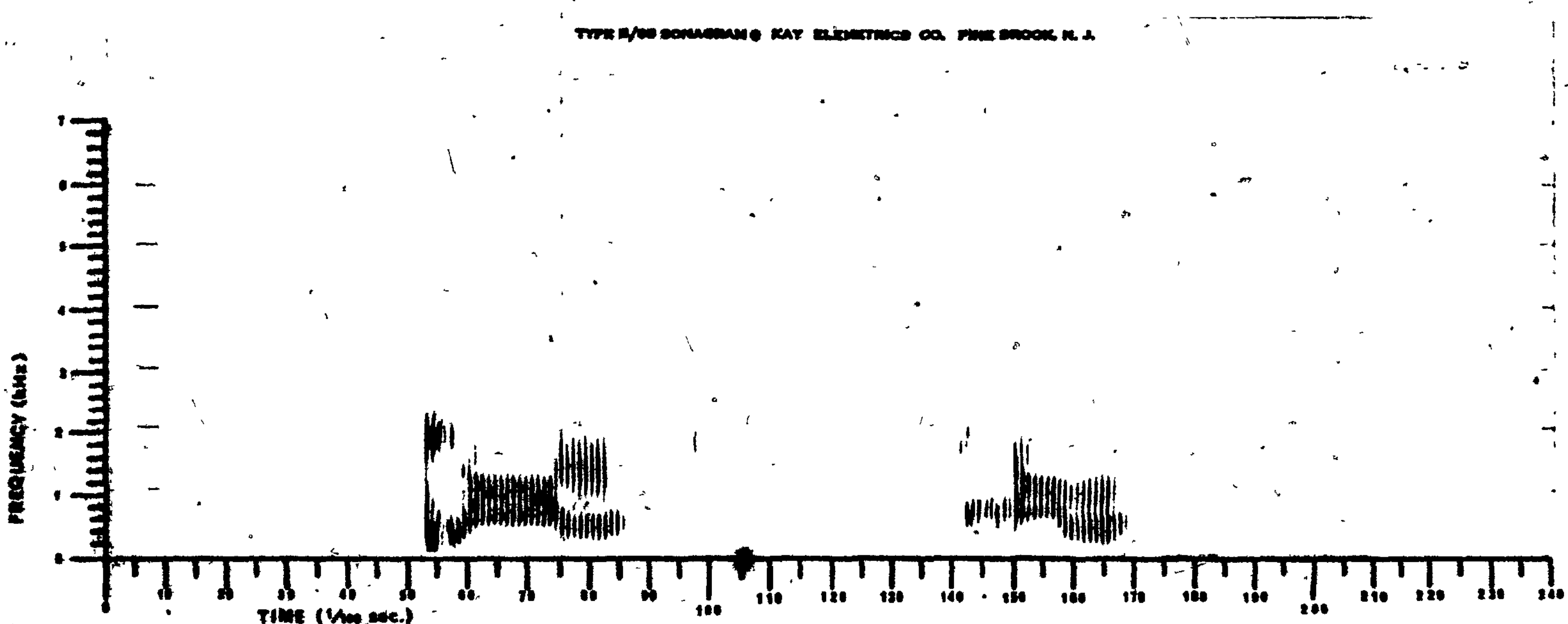


FIGURE 10.1 SPECTROGRAM OF SYNTHETIC SPEECH

dʒ ɔ n   h a ʊ

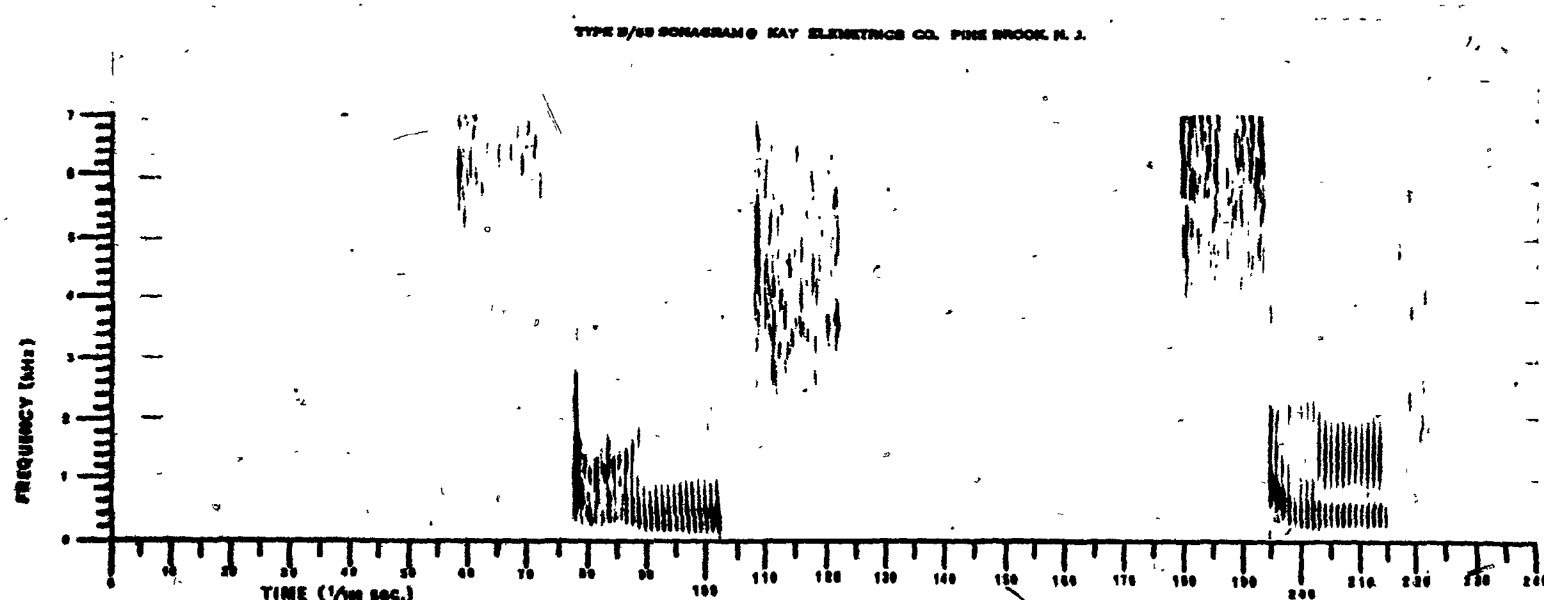


FIGURE 18 — SPECTROGRAM OF SYNTHETIC SPEECH

s p i tʃ s ɪ n θ

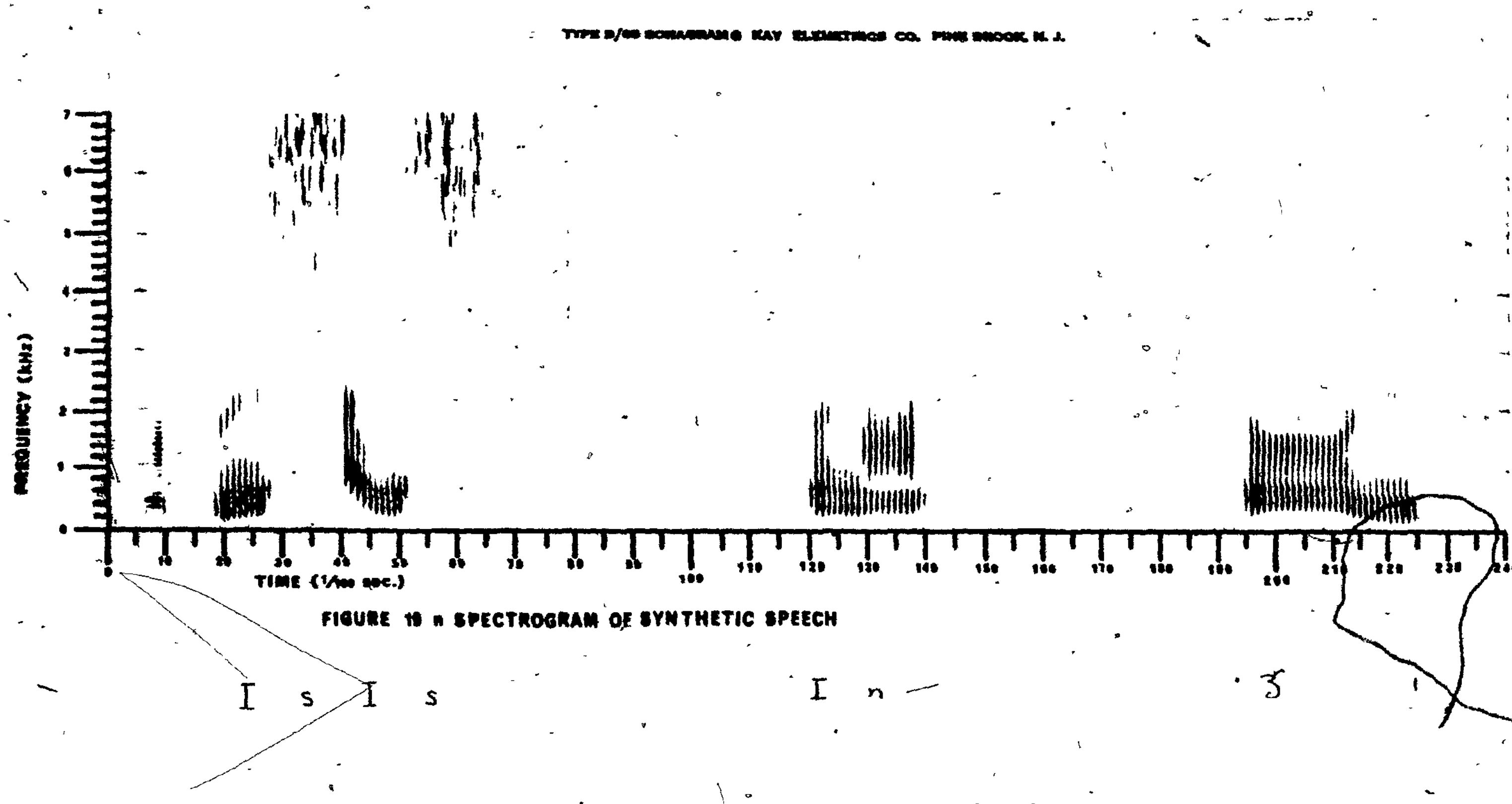


FIGURE 19 n SPECTROGRAM OF SYNTHETIC SPEECH

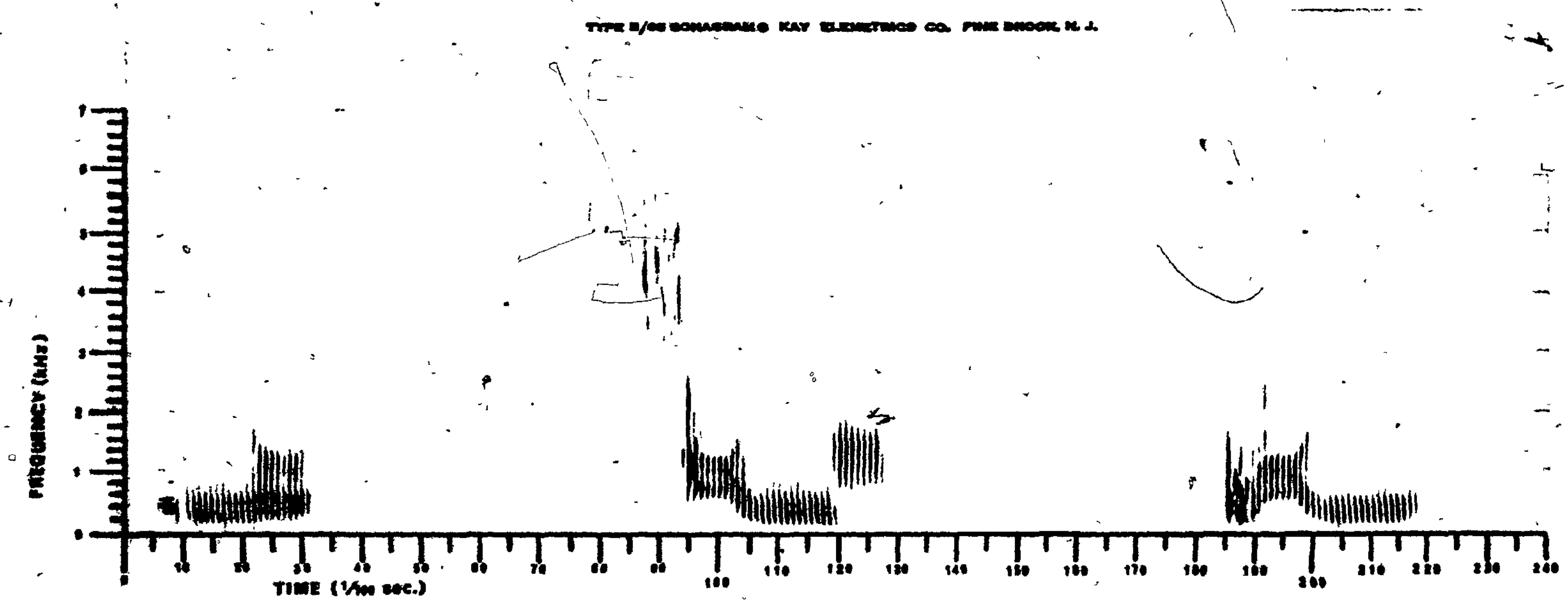


FIGURE 19 • SPECTROGRAM OF SYNTHETIC SPEECH

i l t aI m b aI

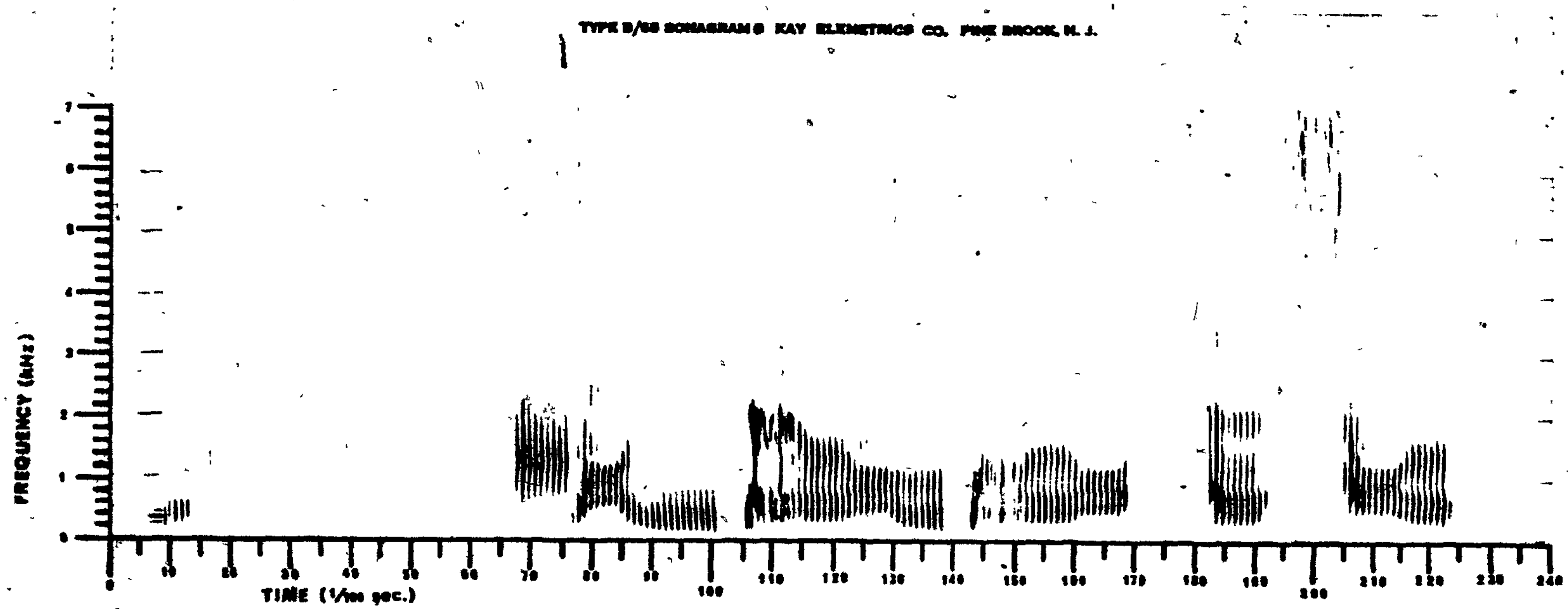


FIGURE 15 p SPECTROGRAM OF SYNTHETIC SPEECH

m aI k ʒ oU p ʒ ɔ s ɛ s oUʒ

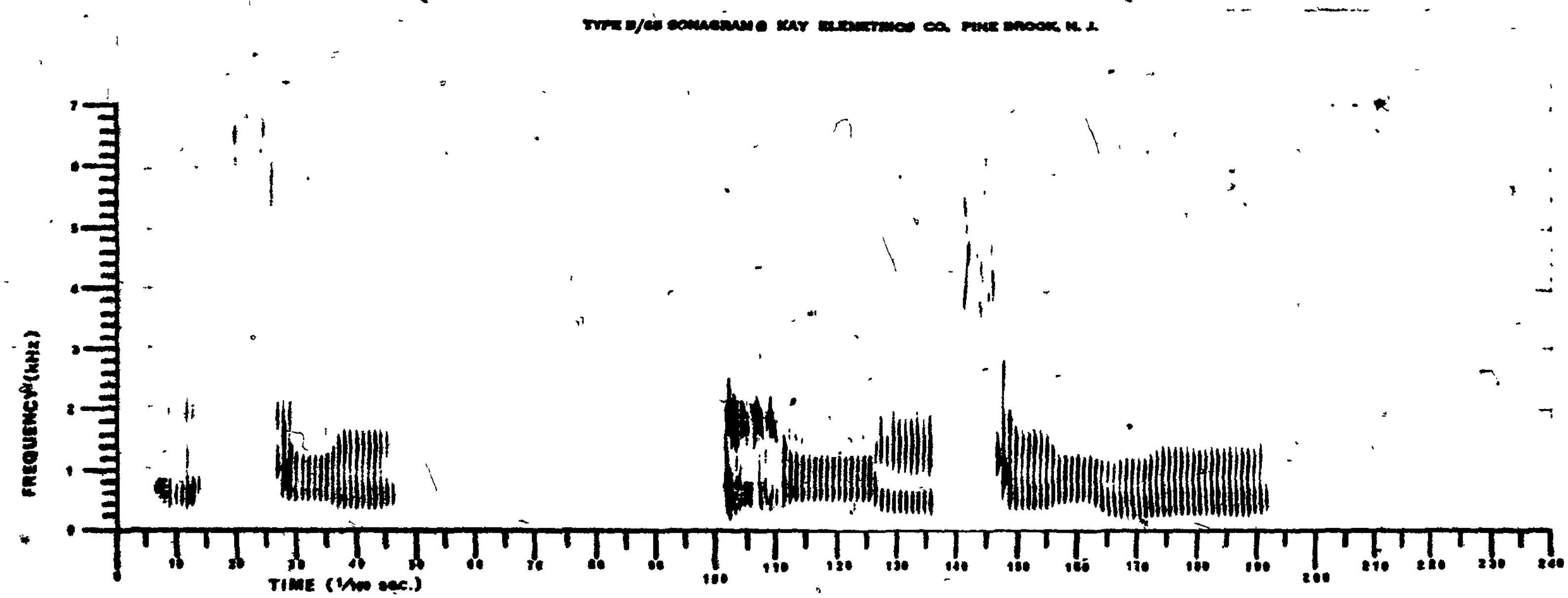


FIGURE 19 SPECTROGRAM OF SYNTHETIC SPEECH

s oU ʒ k ɔ n t ʒ oU əl

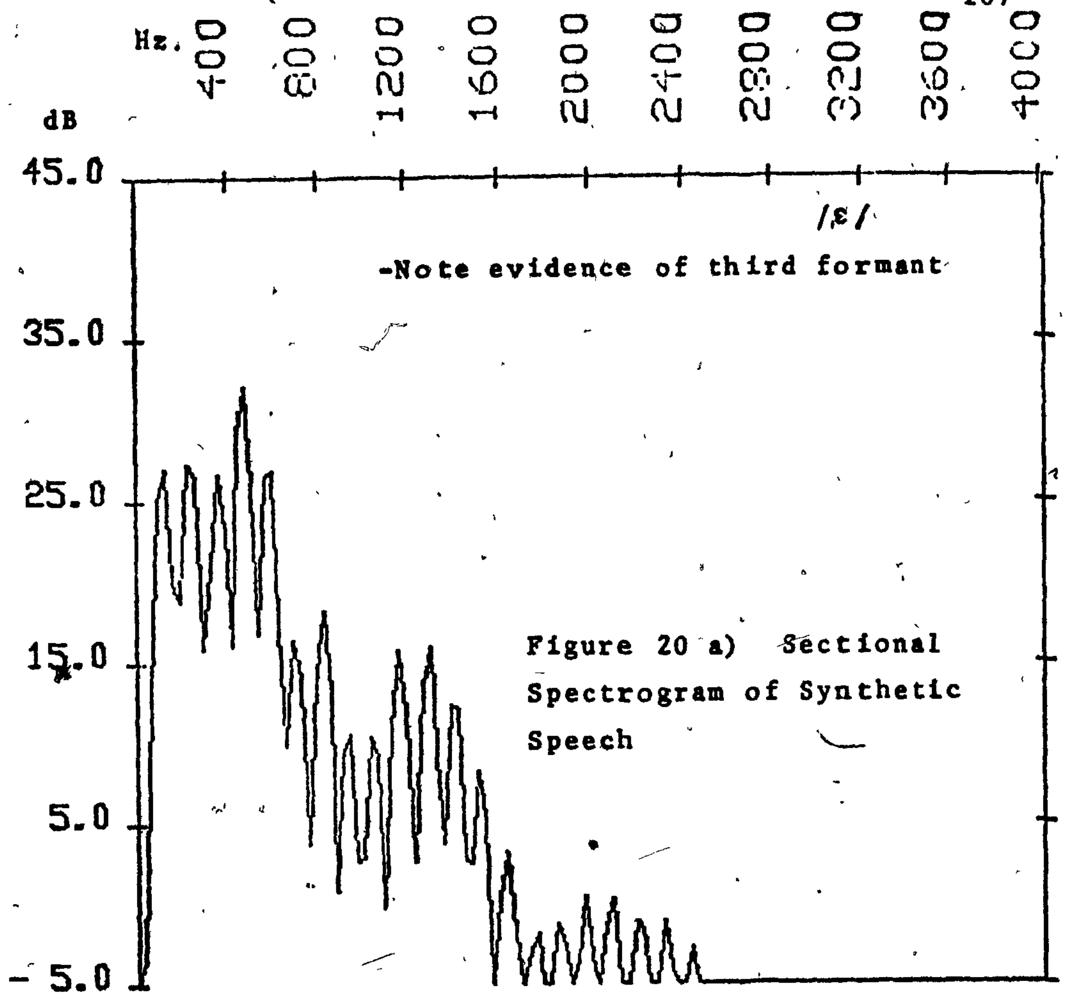


Figure 20 a) Sectional Spectrogram of Synthetic Speech

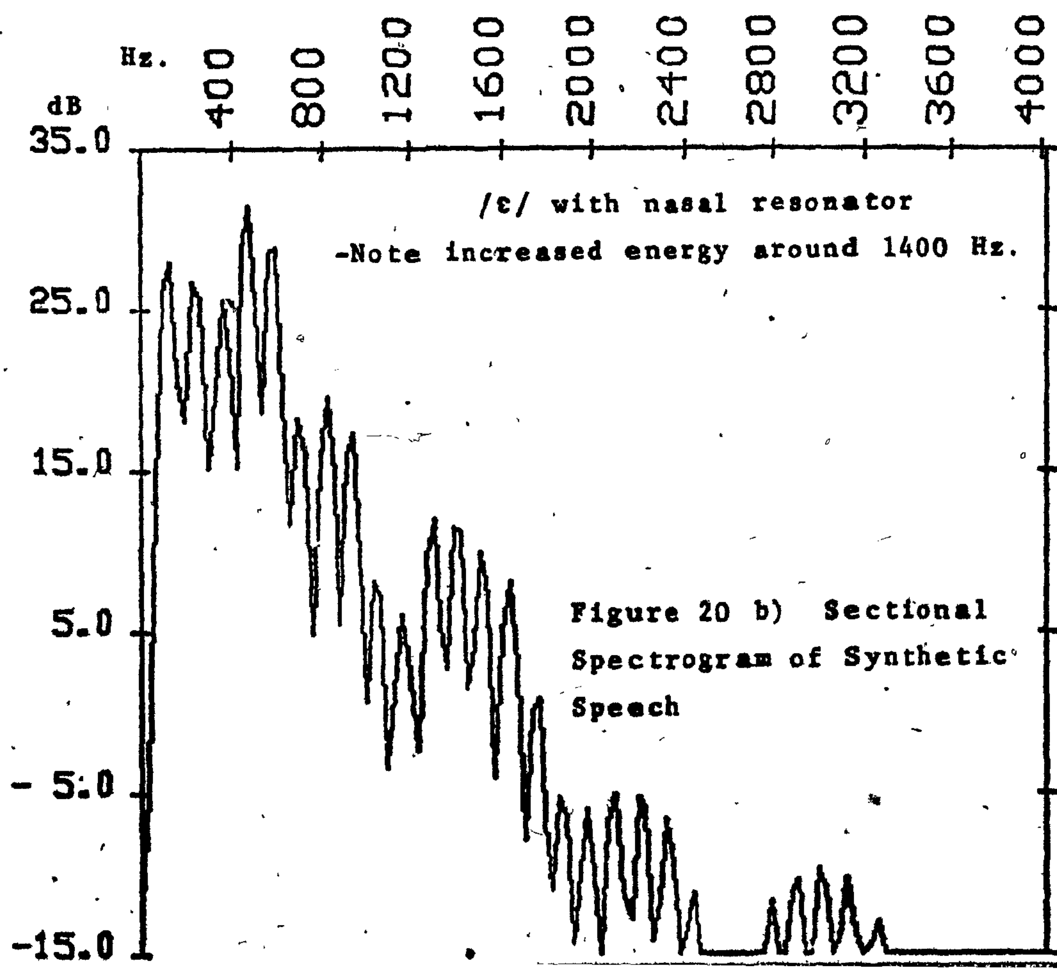
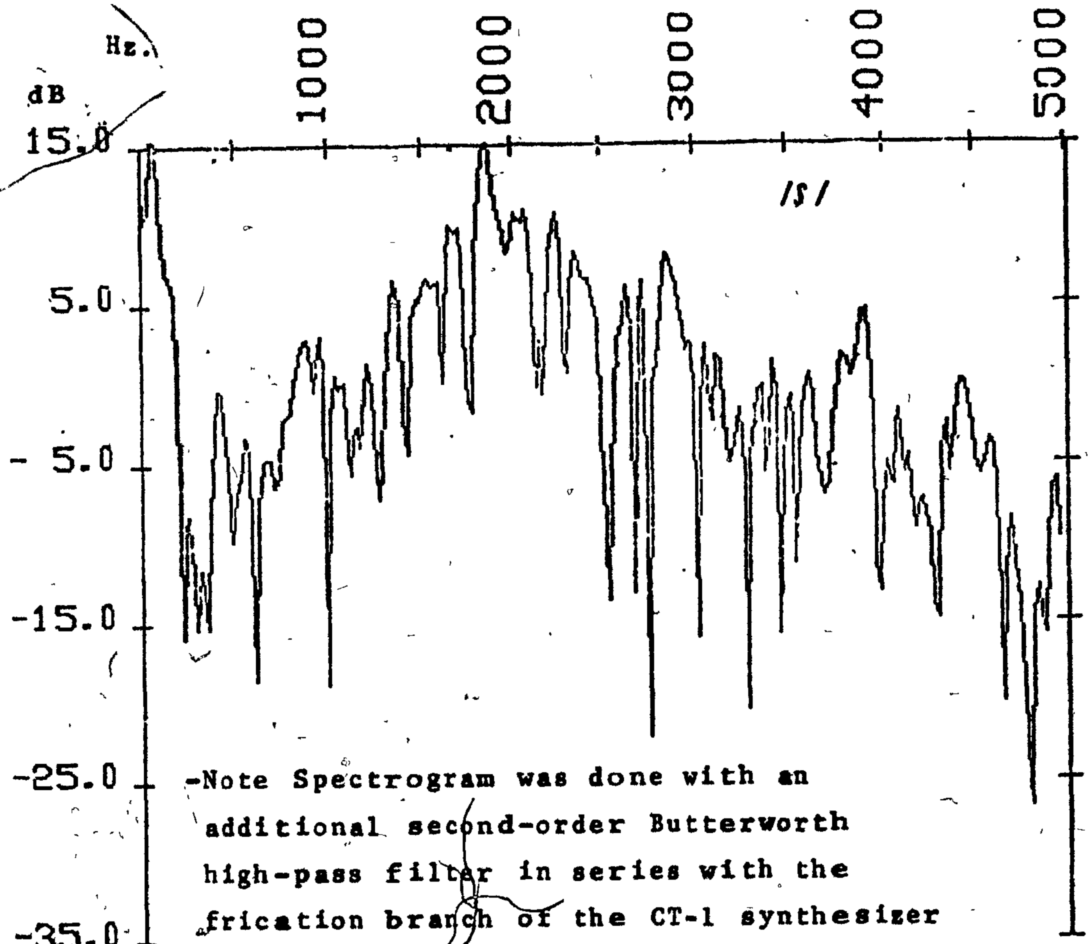


Figure 20 b) Sectional Spectrogram of Synthetic Speech



-Note Spectrogram was done with an additional second-order Butterworth high-pass filter in series with the frication branch of the CT-1 synthesizer (cutoff 1800 Hz.)

Figure 20 c) Sectional Spectrogram of Synthetic Speech too long (especially /gi/) and could cause these sounds to be perceived as unvoiced stops. This could be corrected by decreasing the voicing onset parameter in the phoneme tables.

The nasal formant appears as a fixed formant centered at 1400 Hz, bandwidth 1 kHz. This masks all second formant transitions which are primarily responsible for distinguishing /NN/ from /NG/. Ideally the nasals should have a concentration of low frequency energy with mid ranges subdued and not exhibiting major resonances. Spectrally this would result in an intense low frequency formant (a voice bar) and weaker, broad formant structures.

Affricates (Figure 19k and 19ℓ) have the correct spectral content. The aspirant also exhibits the correct characteristics. The vowel /ɜ/ in church could be slightly shorter, because of adjacent phonemes but it does not detract from intelligibility.

Following the above figures, a spectrogram of connected speech (the title of the thesis) is shown (Figures 19m to 19q).

6.2.3 Acoustic Waveforms

Amplitude vs time (acoustic) waveforms for the corresponding spectrograms are shown (Figure 21a to Figure 21h). Those acoustic waveforms representing vowels, diphthongs, liquids and unvoiced fricatives

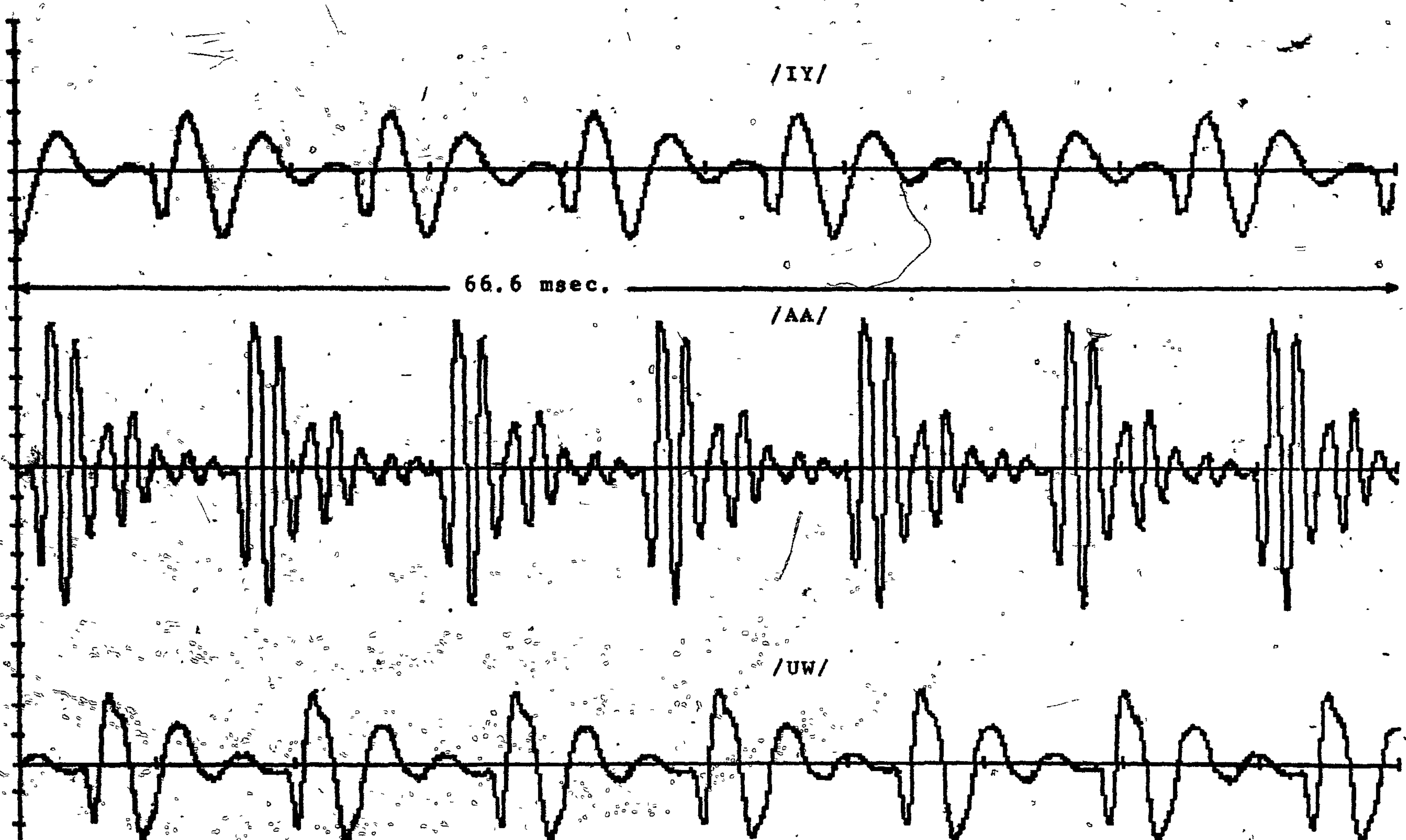


Figure 21 a) Acoustic Waveforms of Synthetic Speech

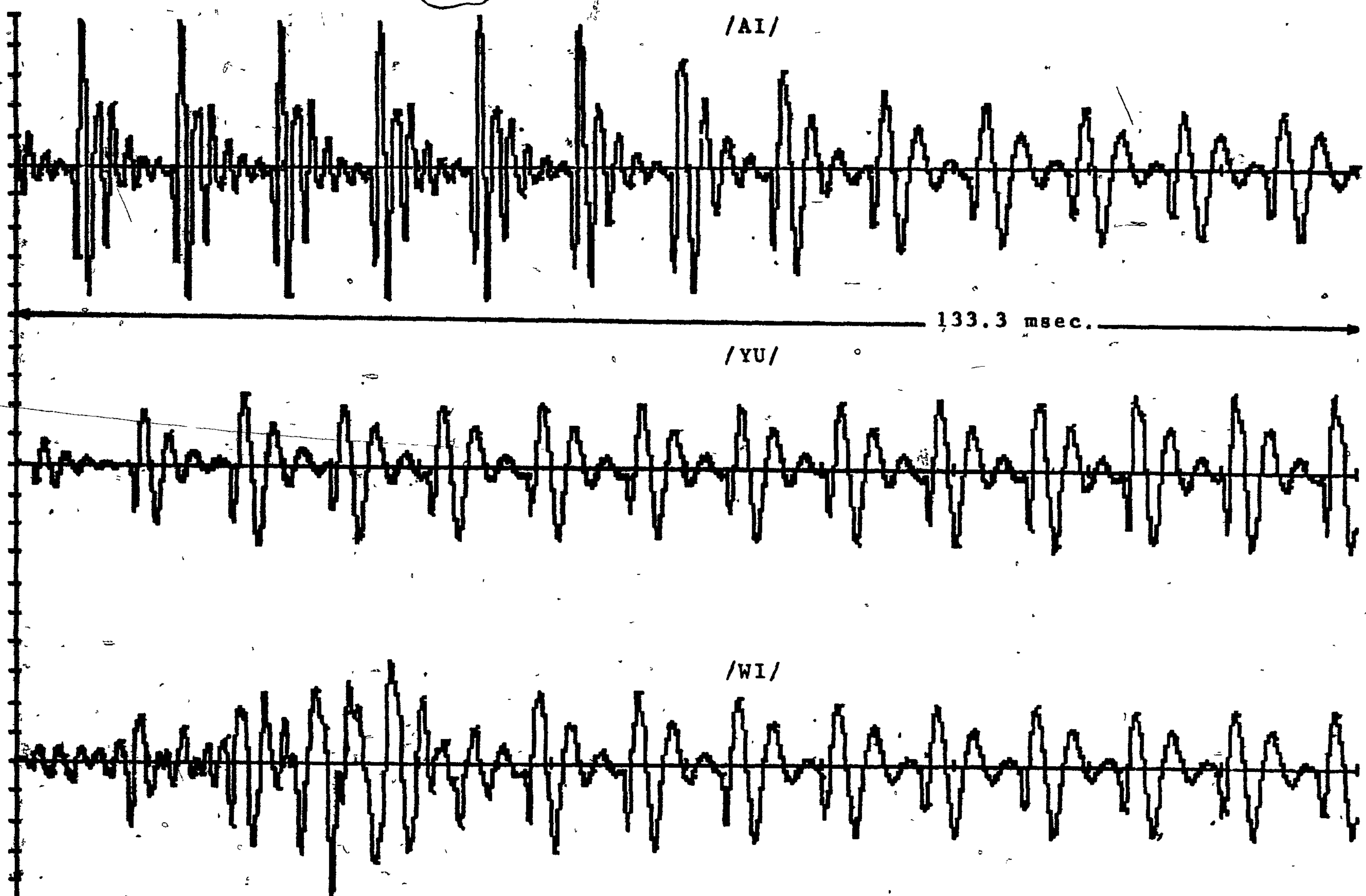


Figure 21 b) Acoustic Waveforms of Synthetic Speech

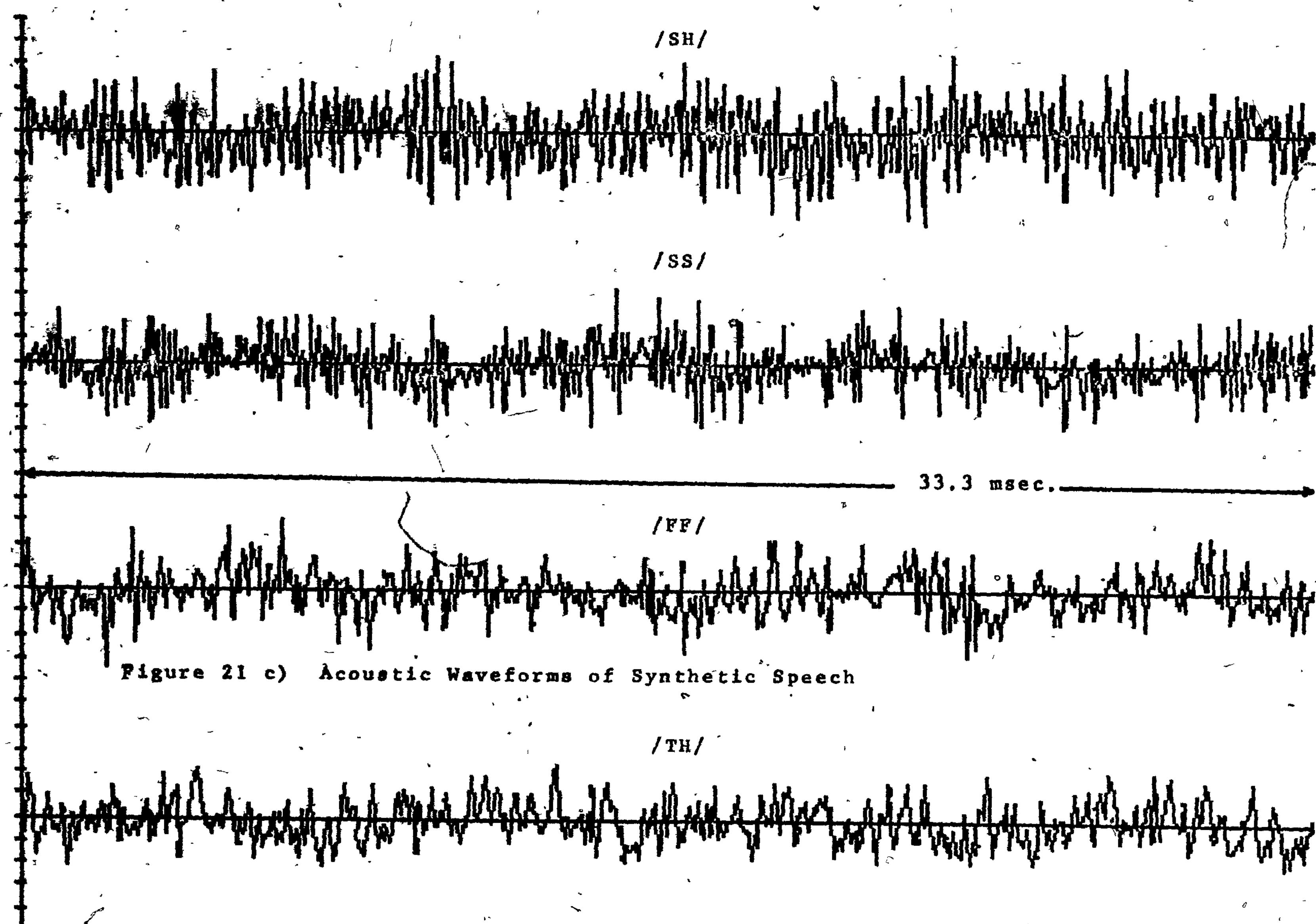


Figure 21 c) Acoustic Waveforms of Synthetic Speech

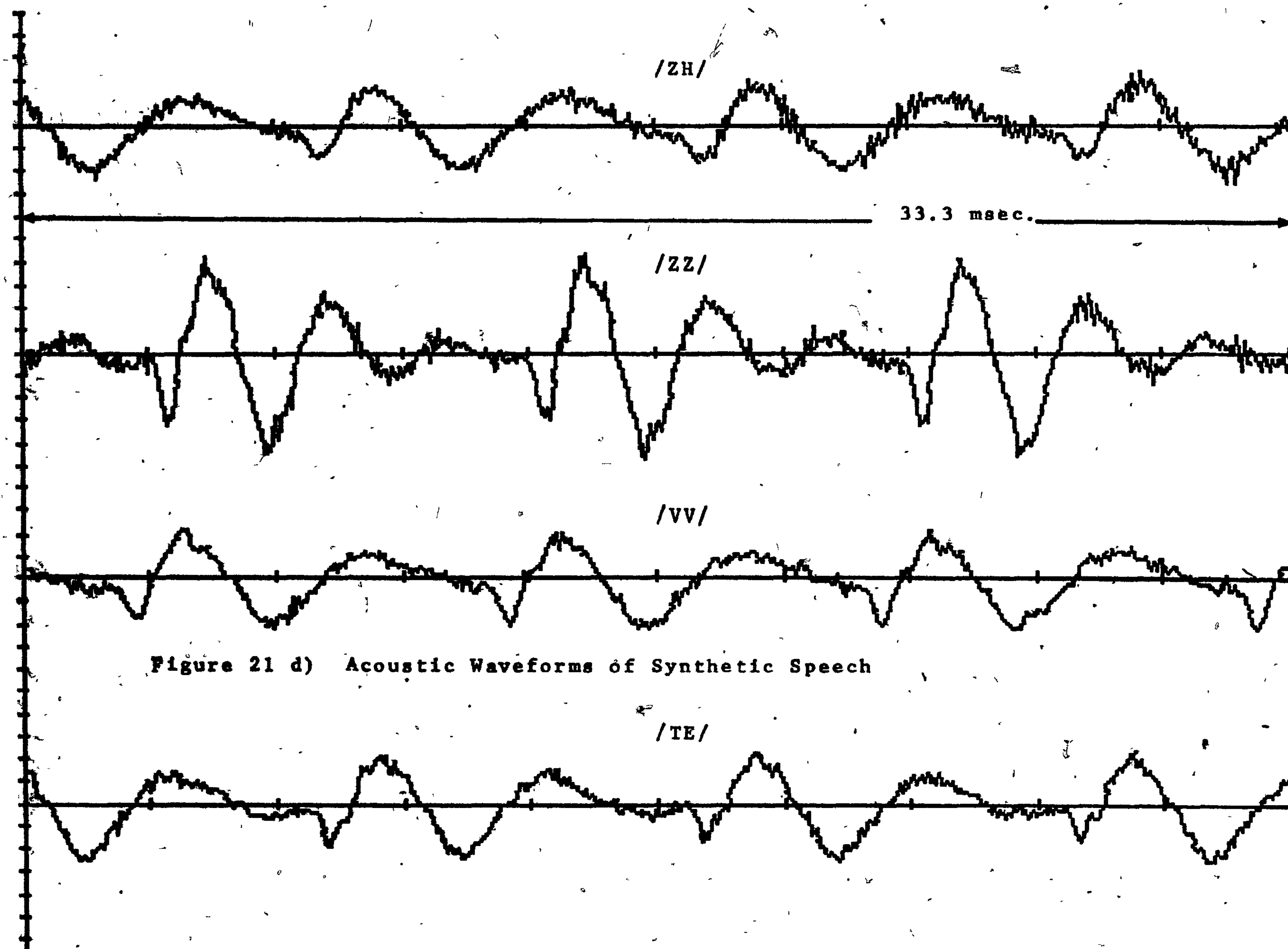


Figure 21 d) Acoustic Waveforms of Synthetic Speech

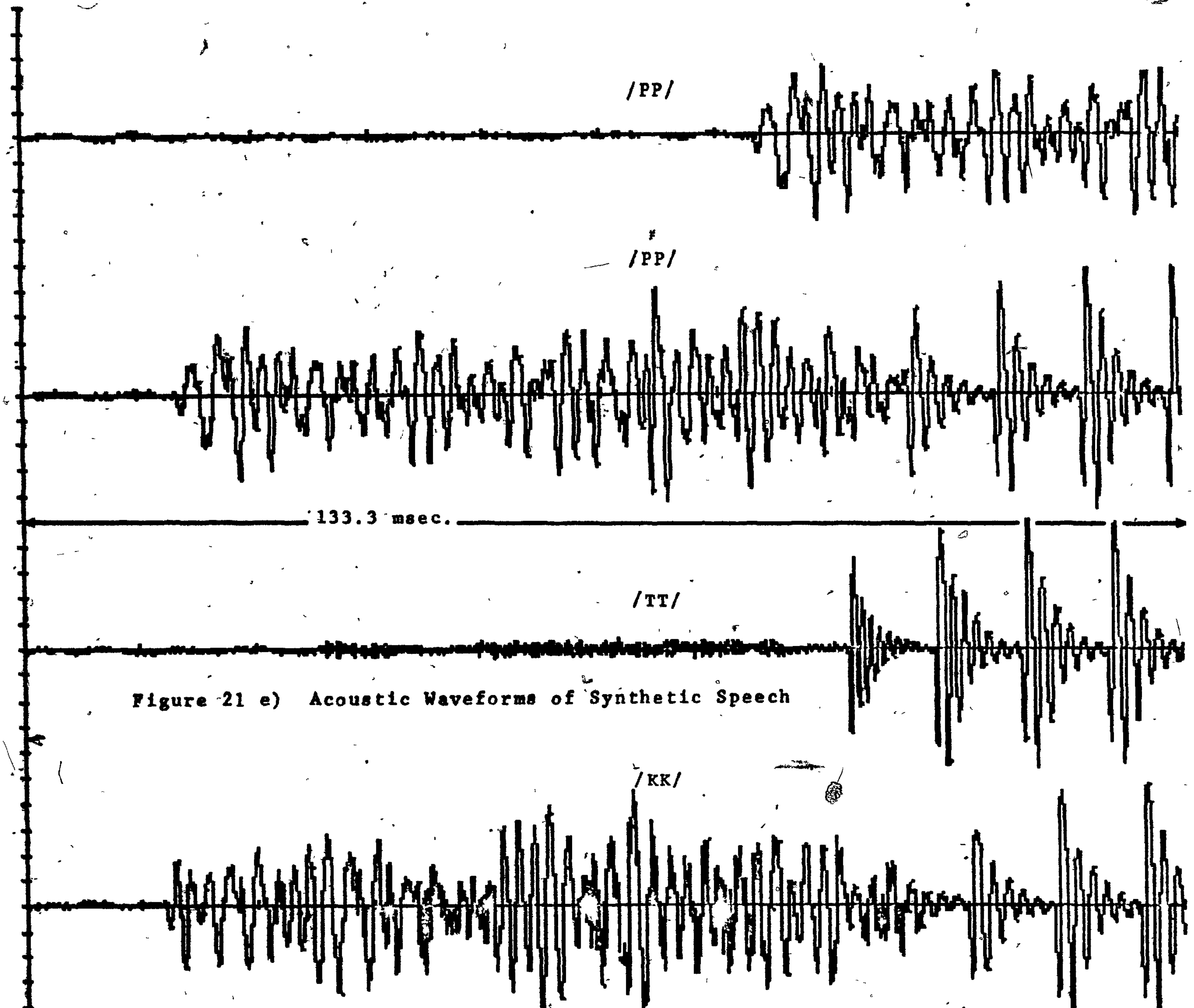


Figure 21 e) Acoustic Waveforms of Synthetic Speech

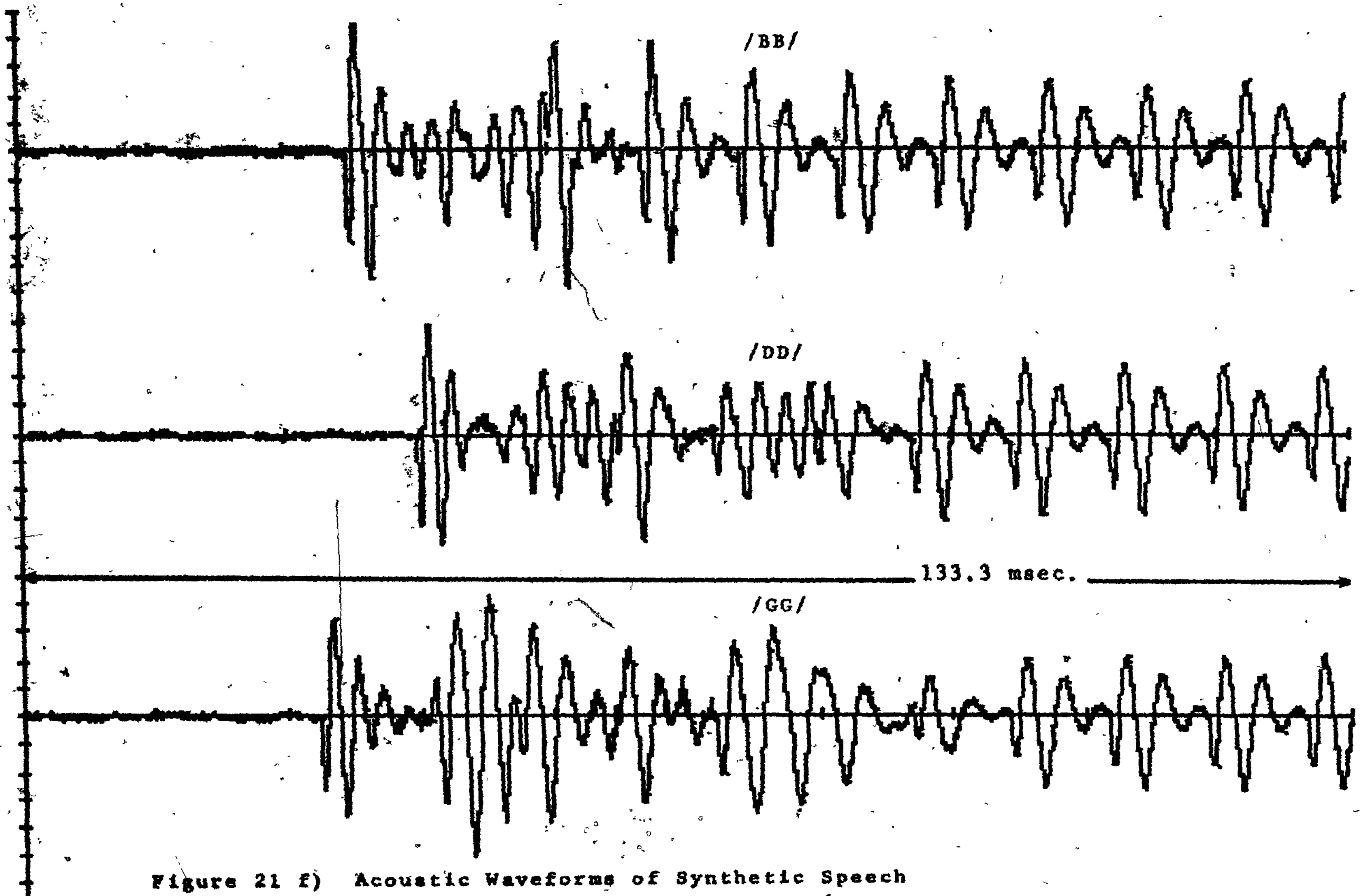


Figure 21 f) Acoustic Waveforms of Synthetic Speech

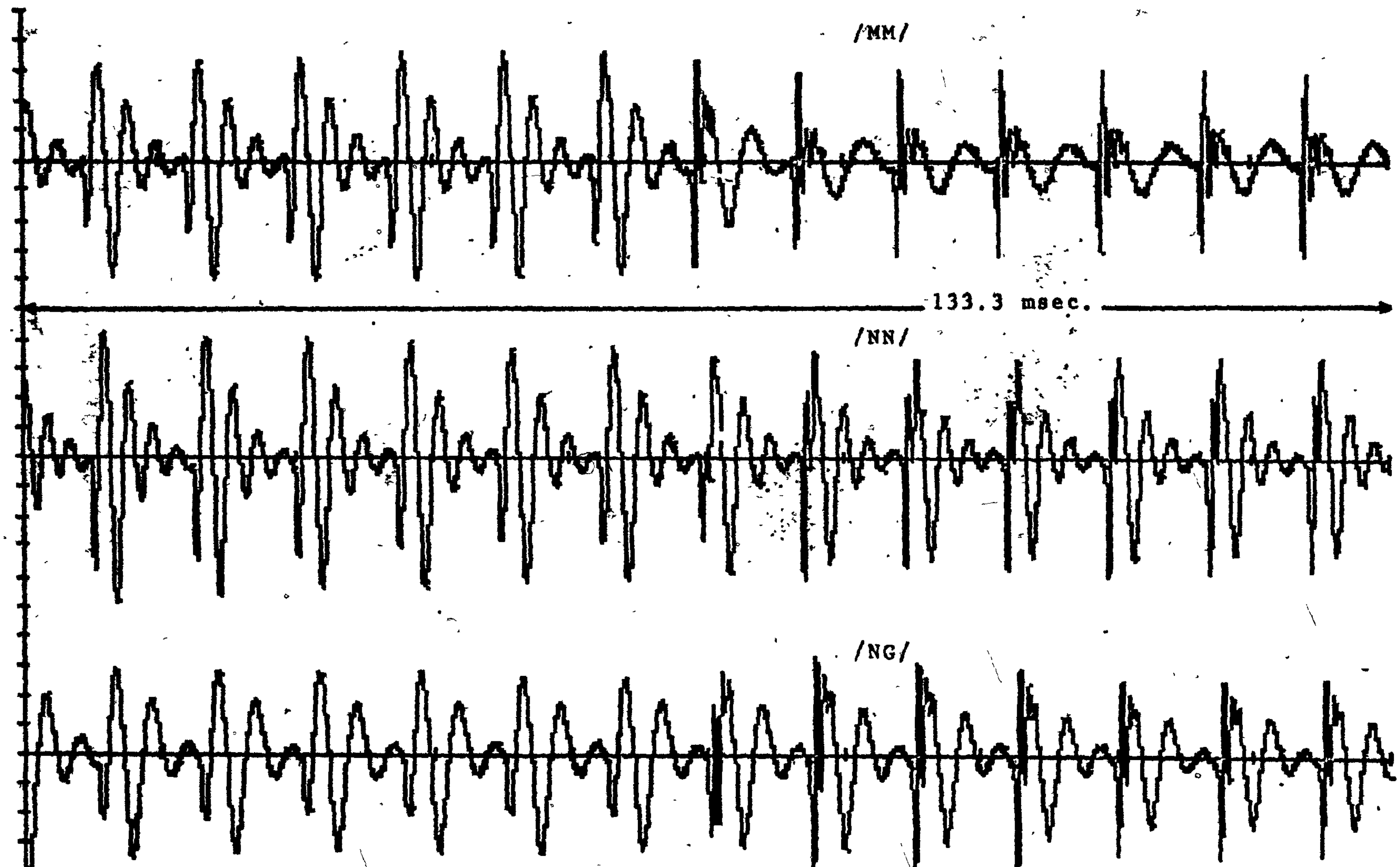


Figure 21 g) Acoustic Waveforms of Synthetic Speech

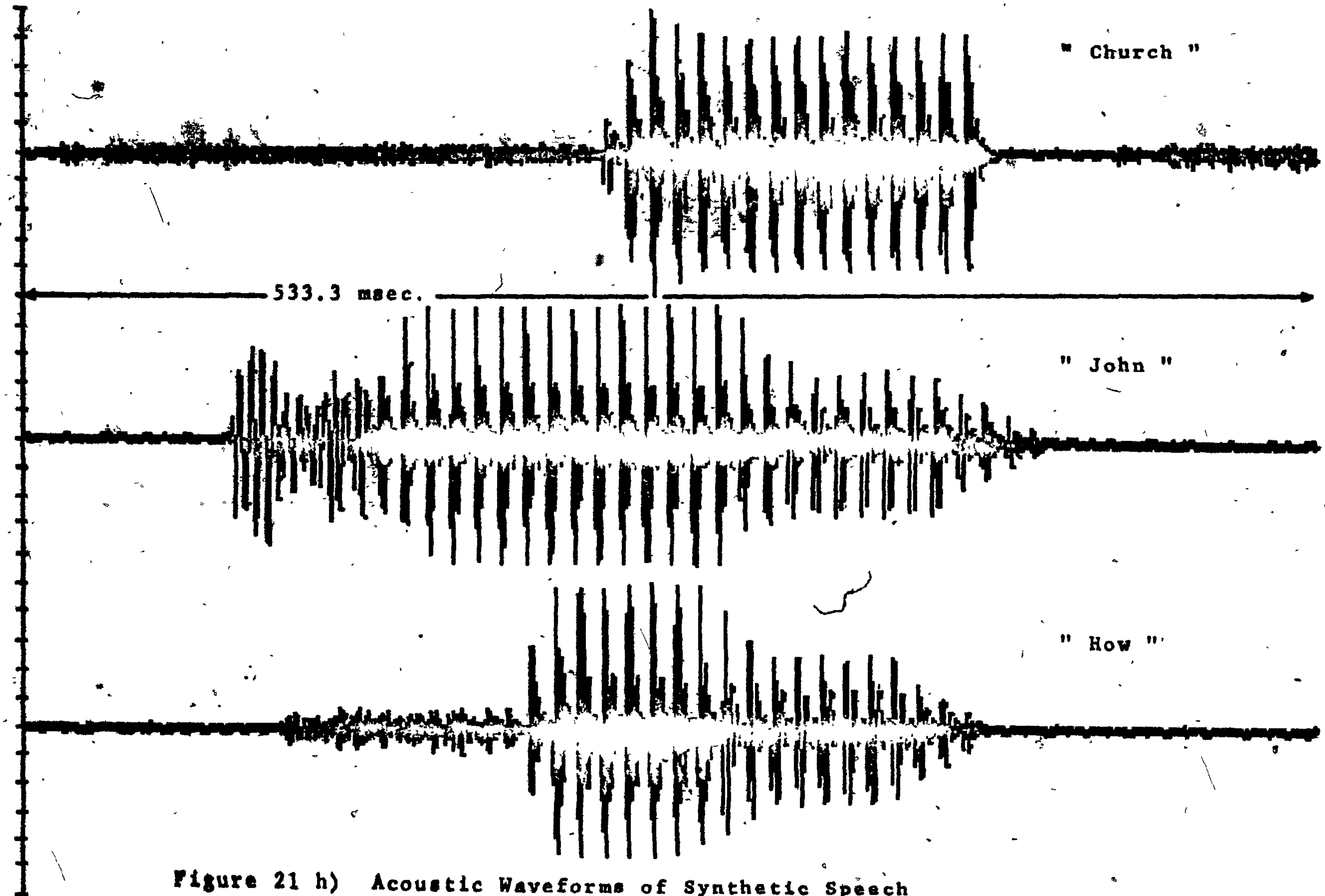


Figure 21 h) Acoustic Waveforms of Synthetic Speech agree with accepted values. The voiced fricatives (Figure 21d) are again shown as having too much voicing amplitude. Unvoiced stops (Figure 21e) have somewhat more frication than is necessary whereas the voiced stops are characterized by an excessive voice onset time (VOT).

The nasals (Figure 21g) have amplitudes that are slightly high but are otherwise satisfactory.

6.2.4 Limitations of the System

The quality of the speech produced by the synthesis equipment described in this thesis is satisfactory but has room for improvement in a number of areas. The defects in the fricatives and nasals described earlier are quite apparent to listeners. It is unfortunate that much time was spent in trying to solve what was originally believed to be a software problem, but was in fact, a hardware defect. The problems associated with fricatives and nasals are directly related to inadequacies in the CT-1 synthesizer.

The frication branch in the synthesizer should have incorporated a sharp variable high-pass filter and not a band-pass filter as used. A sharp filter would eliminate the low frequency frication and improve intelligibility. Ideally a high-pass filter and a low-pass filter pair could have been used creating a band-pass network. Both filters, however, should be of high order to ensure the low frequency frication at least 30 dB down below the pass band. The nasal branch resonator of the CT-1 synthesizer should not have been fixed and broadband. This causes interference with second formant trajectories and results in confusion of the sounds /NN/ and /NG/ . The nasal resonator should have been constructed as a variable band-pass filter with a bandwidth of 600 Hz. Another solution would be to eliminate the nasal branch altogether and introduce control of the bandwidths of the formant filters. The formant network could then be used effectively in the synthesis of nasals.

An attempt at improving the fricatives was made by adding a fixed second-order Butterworth high-pass filter (cutoff at 1800 Hz) in series with the frication band pass filter. This gave some improvement but the problem still remained. Higher order filters are necessary to achieve the impression of the spectral gap typical of fricatives. Even this filter is at best a partial solution.

In order to remedy synthesizer faults, a major redesigning is required. Given a redesign based on the above comments, a substantial improvement can be made to the quality of the synthesized speech.

6.3 Recommendations and Comments

The synthesis system presented is relatively easy to operate once the modified Arpabet is learned. Supra-segmentals are handled by a simple set of commands. As experimental equipment, the numeric feedback feature provided by the parameter dump is a very powerful tool in verifying synthesis strategy or detecting software errors. The apparatus can be used to experiment with transitional or steady state sounds. Like the speech from many other synthesizers, the synthesized speech here is intelligible but does not sound natural.

The equipment memory requirement is low due to real-time operation. A scratch pad memory of 1K bytes is required for temporary storage of variables and the input buffer. The program requires only 1K bytes RAM or EPROM. The phoneme table requires an additional 1K bytes.

The software for the system is quite adaptable and functions well. Other features could have been incorporated but were found unnecessary. For example, a form of visual feedback can be provided by utilizing the four surplus analog outputs available on the CT-1 synthesizer. These signals can be fed to an oscilloscope and displayed with respect to time. A slight modification of the interrupt routine would enable simultaneous display of the F1, F2, F3, and FF or F0, AV, AH, and AF contours.

In order to operate in real-time, a number of compromises had to be made which limit the scope of the synthesis strategy. However, from the standpoint of the strategy developed, the equipment performed very well. The major limitations of the system were in the production of nasals and fricatives, caused by inadequacies in the design of the synthesizer.

Some attempt was made at rectifying this but it became apparent that a complete re-design was necessary. Two courses of action are suggested. The synthesizer can be redesigned using modified and improved filters, or a synthesizer can be constructed around a recently developed LPC chip. The latter course might result in the converse situation where it is not hardware but software that limits performance of the system.

BIBLIOGRAPHY

Ainsworth, W.A. "A Real-Time Speech Synthesis System", Correspondence, IEEE Transactions on Audio and Electroacoustics, December 1972, pp. 397-399.

Ainsworth, W.A. "A System for Converting English Text into Speech", IEEE Transactions on Audio and Electroacoustics, Vol. AU-21, No. 3, June 1973, pp. 288-290.

Allen, J. "Reading Machines for the Blind : The Technical Problems and the Methods Adopted for their Solution", IEEE Transactions, Vol. AU-21, No. 3, June 1973, pp. 259-264.

Allen, J. "Synthesis of Speech from Unrestricted Text", Proceedings IEEE, Vol. 64, No. 4, April 1976, pp. 433-442.

Beek, B., Neuberg, E.P. and Hodge, D.C. "An Assessment of the Technology of Automatic Speech Recognition for Military Applications", IEEE Transactions, ASSP-25, August 1977, pp. 310-322.

Clark, J.E. "A Real-Time Speech Synthesis System", Monitor - Proceedings of the IREE Aust., March 1977, pp. 56-67.

Cohen, M.M. and Massaro, D.W. "Real-Time Speech Synthesis", Behavior Research Methods and Instrumentation, Vol. 8 (2), 1976, pp. 189-196.

Coker, C.H. "Synthesis by Rule from Articulatory Parameters", Proceedings of the 1967 Conference on Speech Communication and Processing, Cambridge, Massachusetts; IEEE 1967, A9, pp. 52-53.

Cooper, F.S. "How is Language Conveyed by Speech?" In J.F. Kavanagh and I.G. Mattingly (Editors), Language by EAR and by EYE : The Relationships Between Speech and Reading. Cambridge, Massachusetts : M.I.T. Press, 1972, pp. 25-45.

Cooper, F.S., Delattre, P.C., Lieberman, A.M., Borst, J.M. and Gerstman, L.J. "Some Experiments on the Perception of Synthetic Speech of Sounds", The Journal of the Acoustical Society of America, Vol. 24, No. 6, November 1952, pp. 597-606.

Crichton, R.G. and Fallside, F. "Speech Analysis-Synthesis on a Small Computer", Spring Meeting British Acoustical Society, paper 73SHB7, April 1973.

Dreshner, E. Sound 1, a module on Sound developed under the supervision of the Departments of Linguistics and English, McGill University, Montreal, 1972.

Dudley, H. "The Vocoder", Bell Labs. Record, 18 (1939), pp. 122-126.

Fant, C.G.M. Acoustic Theory of Speech Production. The Hague, 1960.

Ferrero, F.E. "SPAR - A Terminal Analog Speech Synthesizer", ACUSTICA, Vol. 22 (1969/70), pp. 357-362.

Flanagan, J.L. "Note on the Design of 'Terminal Analog' Speech Synthesizers", Journal of the Acoustical Society of America, 25 (1957), pp. 306-310.

Flanagan, J.L. "Computers that Talk and Listen : Man-Machine Communications by Voice", Proceedings of the IEEE, Vol. 64, No. 4, April 1976, pp. 405-415.

Flanagan, J.L., Coker, C.H., Rabiner, L.R., Schater, R.W. and Umeda, N. "Synthetic Voices for Computers", IEEE Spectrum, October 1970, pp. 22-45.

Flanagan, J.L., Rabiner, L.R., Schafer, R.W. and Denman, J. "Wiring Telephone Apparatus from Computer-Generated Speech", The Bell System Technical Journal, Vol. 51, 1972, pp. 391-397.

Hilborn, E.H. "Preliminary Evaluation of Synthetic Speech", U. S. Department of Aviation, Federal Aviation Administration, Interim Report No. FAA-RD-72-109, Washington, August 1972.

Holmes, J.N. "The Influence of Glottal Waveform on the Naturalness of Speech from a Parallel Formant Synthesizer", IEEE Transactions, Vol. AU-21, No. 3, June 1973, pp. 298-305.

Holmes, J.N., Mattingly, I.G. and Shearme, J.N. "Speech Synthesis by Rule", Language and Speech, 7, (1964), pp. 127-143.

Ishizaka, K. and Flanagan, J.L. "Synthesis of Voiced Sounds from a Two-Mass Model of the Vocal Cords", The Bell System Technical Journal, 51, (1972), pp. 1233-1268.

Kato, Y., Ochiai, K. and Araseki, T. "A Terminal Analog Speech Synthesizer in a Small Computer", 1971 IEEE International Convention Digest, Paper 2F.4 (1971), pp. 102-103.

Kelly, J.L. and Gerstman, L.J. "An Artificial Talker Driven from a Phonetic Input", (abstract), Journal of the Acoustical Society of America, 33, (1961), p. 835.

Klatt, D.H. "Acoustic Theory of Terminal Analog Speech Synthesis", 1972 International Conference on Speech Communications and Processing, Boston, April 1972, pp. 131-135.

Klatt. D.H. "Structure of a Phonological Rule Component for a Synthesis-by-Rule Program", IEEE Transactions, Vol. ASSP-24, No. 5, October 1976, pp. 391-398.

Klatt, D.H. "A Cascade / Parallel Terminal Analog Speech Synthesizer and a Strategy for Consonant-Vowel Synthesis", Handout for Spring Meeting of the Acoustical Society of America, June 1977.

Lieberman, P. "On the Evolution of Language : A Unified View", Cognition, 2, (1973), pp. 59-94.

Makhoul, J. "Linear Prediction : A Tutorial Review", Proceedings of the IEEE, Vol. 63, No. 4, April 1975, pp. 561-580.

Mattingly, I.G. "Synthesis by Rule of General American English", Ph.D. Thesis, Yale University, New York, 1968.

Mermelstein, P. "Articulatory Model for the Study of Speech Production", The Journal of the Acoustical Society of America, Vol.53, No. 4, 1973, pp. 1070-1082.

Nooteboom, S.G., Slis, I.H. and Willems, L.F., "Speech Synthesis by Rule; WHY, WHAT AND HOW?", I.P.O. Annual Progress Report, 8, (1973), pp. 3-13.

Ochiai, K. and Araseki, T. "A Terminal Analog Speech Synthesizer", 1972 International Conference on Speech Communications and Processing, Boston, April 1972, pp. 427-430.

Rabiner, L.R. "Speech Synthesis by Rule : An Acoustic Domain Approach", The Bell System Technical Journal, January 1978, pp. 17-37.

Rabiner, L.R. and Schafer, R.W., Digital Processing of Speech Signals, Prentice-Hall, New Jersey, (1978).

Rao, P.V.S. and Thosar, R.B. "A Programming System for Studies in Speech Synthesis", IEEE Transactions, Vol. ASSP-22, No. 3, June 1974, pp. 217-225.

Sapozhkov, M.A. "Improving the Quality of Synthesized Speech", Soviet Physics - Acoustics, Vol. 17, No. 4, April-June 1972, pp. 510-513.

Stevens, K.N. and Halle, M., "Remarks on Analysis by Synthesis and Distinctive Features"; In W. Wathen-Dunn (Editor), Models for the Perception of Speech and Visual Form, Cambridge, Massachusetts; M.I.T. Press, 1967, pp. 88-102.

Umenda, N. "Vowel Duration in Polysyllabic Words in American English", Journal of the Acoustical Society of America, 1972, pp. 52-133.

Warmuth, D.B., Mundie, J.R. and Vaughn, G.L. "Voice Communication by Speech Synthesis", National Aerospace and Electronics Conference Record, 1976, pp. 933-937.

Wiggins, R. and Brantingham, L. "Three-Chip System Synthesizes Human Speech", Electronics, Vol. 51, No. 18, August 1978, pp. 109-116.

APPENDIX A

PROGRAM LISTING

```
*PATCH M68SAM JJ23000 SPEECH (20.20)
*********************************************************************************

M68SAM IS THE PROPERTY OF MOTOROLA SPD. INC.
       COPYRIGHT 1974 BY MOTOROLA INC

MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.1
```

```
        NAM     SPEECH
* A GENERAL PROGRAM FOR SPEECH SYNTHESIS
* BY RULE FOR THE MC6800

* PROGRAM BY JOHN P PINNELL
*        NOV 3 1978

* --- COMMAND SUMMARY FOR VERSION CT-1.78
*
*
* 'CNTRL O' BACK SPACE/CURSOR LEFT.DELETE LAST CHAR
*
* '"'       REPEAT LAST STRING
*
* '+'       END OF LINE.GENERATES CR/LF
*
* '@'       PUNCH A TAPE OF LAST STRING
*
* 'CR'      END OF STRING
*
*
* '#'<CR>   INITIATE WHISPERING
*
* '$'<CR>   INITIATE VOICING (DEFAULT)
*
* '&'<CR>   INITIATE PRINT OPTION
*
* '%'<CR>   INITIATE SYNTHESIS (DEFAULT)
*
*'ESC'<CR>  RETURN TO MONITOR (MIKBUG)
*
* '*'<CR>   CLEARS SYNTHESIZERS PARAMETERS
*
* '/'<CR>   CAUSES THE FUNDAMENTAL FREQUENCY TO INCR
*
* ' '<CR>   CAUSES THE FUNDAMENTAL FREQUENCY TO DECR
*
* '  '<CR>  SETS UP A TIME DELAY BETWEEN WORDS
*
*
* NOTES;
*
* <CR> INDICATES A CARRIAGE RETURN
*
* ALL COMMANDS SHOWN FOLLOWED BY A <CR> MAY BE USED
*
* IN THE STRING AND EXECUTED AS SYNTHESIS PROGRESSES
*
* UPPER CASE CHARACTERS ARE UNSTRESSED
*
* LOWERCASE CHARACTERS ARE STRESSED
*
```

```
0000                ORG    $0000
           E1AC  INEEE  EQU    $E1AC      (MIKBUG INPUT ROUTINE)
           E1D1  OUTEEF EQU    $E1D1      (MIKBUG OUTPUT ROUTINE)
           E07E  PDATA1 EQU    $E07E      (MIKBUG PRINT STRING)
           E0D0  START  EQU    $E0D0      (START OF MIKBUG)
           E0CA  OUT2HS EQU    $E0CA      (MIKBUG PRINT HEX VALUE+SPACE
           0500  BUFFER EQU    $0500      START OF INPUT BUFFER
           1000  TABLE  EQU    $1000      START OF DATA TABLE
0000 0001  T1     RMB    1          OVERALL PHONEME TIMING
0001 0001  T2     RMB    1          ONSET TIME F1 F2 F3 FF
0002 0001  T3     RMB    1          TRANSITION TIME F1 F2 F3 FF
0003 0001  T4     RMB    1          ONSET TIME AV
0004 0001  T5     RMB    1          TRANSITION TIME AV
0005 0001  T6     RMB    1          ONSET TIME AH
0006 0001  T7     RMB    1          TRANSITION TIME AH
0007 0001  T8     RMB    1          ONSET TIME F0
0008 0001  T9     RMB    1          TRANSITION TIME F0
0009 0001  T10    RMB    1          ONSET TIME AF AN
000A 0001  T11    RMB    1          TRANSITION TIME AF AN
000B 0001  INP1   RMB    1          STORAGE 1 ST CHAR
000C 0001  INP2   RMB    1          STORAGE 2 ND CHAR
000D 0001  F0     RMB    1          FUNDAMENTAL FREQ UP/DOWN
000E 0001  VOICE  RMB    1          WHISPERING OR VOICED
000F 0001  HRDCPY RMB    1          LIST SYNTHESIS PARAMETERS
0010 0001  TEMP   RMB    1
0011 0001  CNTR   RMB    1
0012 0001  CNTX   RMB    1
0013 0002  THERE  RMB    2
0015 0002  HERE   RMB    2
0017 0002  CENTR  RMB    2
0019 0002  CENTX  RMB    2
0019 0002  REMB   RMB    2
0030              ORG    $0030
0030 DD    STRING FCB    $D,$A,0,0,0,$3E,$04    C/R,LF,>
0031 DA
0032 00
0033 00
0034 00
0035 3E
0035 04
0037 2A    ERROR  FCB    '*,'*,'*,'E,'R,'R,'O,'R
0038 2A
0039 2A
003A 45
003B 52
003C 52
003D 4F
003E 52
003F 20           FCB    $20,'P,'H,'O,'N,'E,'M,'E,$20
0040 50
0041 48
0042 4F
```

```
0043 4E
0044 45
0045 4D
0046 45
0047 20
0048 4E                 FCB    'N,'D,'T,$20,'I,'N,$20,'T,'A,'B,'L,'E
0049 4F
004A 54
004B 20
004C 49
004D 4E
004E 20
004F 54
0050 41
0051 42
0052 4C
0053 45
0054 0D                 FCB    $D,$A,0,0,0,$04
0055 0A
0056 00
0057 00
0058 00
0059 04
                   *
                   * PROGRAM STARTS AT LOCATION 0100 HEX
                   *
0100                    CRG    $0100
                   *
                   * INITIALIZATION
                   *
0100 7F 000E LINK1      CLR    VOICE      CLEAR WHISPERING
0103 7F 000F            CLR    HRDCPY     CLEAR PARAMETER LISTING
0106 8D 29              BSR    CLEAR      GOTO SUBROUTINE CLEAR
                   *
                   * THIS SECTION INPUTS A LINE FROM THE
                   * CONSOLE INTO A RAM BUFFER, PROCESSES
                   * BACK SPACE FUNCTIONS, TAPE PREPARATION,
                   * CR/LF GENERATION, AND PERMITS THE LAST
                   * STRING ENTERED TO BE RESYNTHESIZED
                   *
0108 CE 0030            LDX    #STRING     X='STRING'
010B BD E07E            JSR    PDATA1     PRINT STRING
010E CE 0500            LDX    #BUFFER     X= START OF BUFFER
0111 BD E1AC LINK2      JSR    INEEE      INPUT A
0114 81 0D              CMP A  #$0D       IF A='CR'
0116 27 79              BEQ    LINK7      THEN LINK7
0118 81 0F              CMP A  #$0F       ELSE IF A='CNTRL O'
011A 27 31              BEQ    LINK3      THEN LINK3
011C 81 2B              CMP A  #$2B       ELSE IF A='+'
011E 27 3A              BEQ    LINK4      THEN LINK4
0120 81 22              CMP A  #$22       ELSE IF A='"'
0122 27 76              BEQ    LINK8      THEN LINK8
```

```
0124 81 40               CMP A  #$40      ELSE IF A='@'
0126 27 3A               BEQ    LINK5     THEN LINK5
0128 81 00               CMP A  #$00      ELSE IF A='NUL'
012A 27 E5               BEQ    LINK2     THEN LINK2
012C A7 00               STA A  0,X       ELSE STORE A @ X
012E 08                  INX              X=X+1
012F 20 E0               BRA    LINK2     GOTO LINK2
                   *
                   * SUBROUTINE CLEARS OLD DATA TABLE (INITIALIZATION)
                   *
0131 86 80         CLEAR LDA A  #$80      A=128
0133 B7 1FF2             STA A  $1FF2     STORE FOR F1
0136 B7 1FF3             STA A  1FF3      STORE FOR F2
0139 B7 1FF4             STA A  $1FF4     STORE FOR F3
013C B7 1FF7             STA A  $1FF7     STORE FOR FF
013F 4F                  CLR A            A=0
0140 B7 1FF0             STA A  $1FF0     STORE FOR AV
0143 B7 1FF5             STA A  $1FF5     STORE FOR AH
0146 B7 1FF6             STA A  $1FF6     STORE FOR AF
0149 B7 1FF8             STA A  $1FF8     STORE FOR AN
014C 39                  RTS              RETURN FROM SUBROUTINE
                   *
014D 8C 0500 LINK3       CPX    #BUFFER    IF X=START OF BUFFER
0150 27 BF               BEQ    LINK2     THEN LINK2
0152 09                  DEX              ELSE X=X-1
0153 86 08               LDA A  #$08      A='BS'
0155 BD E1D1             JSR    OUTEEE    PRINT A
0158 20 B7               BRA    LINK2     GOTO LINK2
                   *
015A CE 0030 LINK4       LDX    #STRING    X='STRING'
015D BD E07E             JSR    PDATA1    PRINT STRING
0160 20 AF               BRA    LINK2     GOTO LINK2
                   *
                   *
0162 8C 0500 LINK5       CPX    #BUFFER    IF X=START OF BUFFER
0165 27 04               BEQ    LINK6     THEN LINK6
0167 86 04               LDA A  #$04      A='EOT'
0169 A7 00               STA A  0,X       STORE A @ X
016B 86 12         LINK6 LDA A  #$12      A='DC2'
016D BD E1D1             JSR    OUTEEE    PRINT A
0170 8D 10               BSR    NULL      PRINT 25 NULLS
0172 CE 0500             LDX    #BUFFER    X=START OF BUFFER
0175 BD E07E             JSR    PDATA1    PRINT STRING
0178 8D 08               BSR    NULL      PRINT 25 NULLS
017A 86 13               LDA A  #$13      A='DC3'
017C BD E1D1             JSR    OUTEEE    PRINT A
017F 7E 0100 LINKEA      JMP    LINK1     GOTO LINK1
                   *
                   * SUBROUTINE PRINTS 25 NULLS
                   *
0182 C6 1A         NULL  LDA B  #$1A      B=25
0184 5A            NULL1 DEC B            B=B-1
```

```
0185 C1 00              CMP B   #$00      IF B=0
0187 27 07              BEQ     NULOUT    THEN NULOUT
0189 86 00              LDA A   #$00      ELSE A='NUL'
018B BD E1D1            JSR     OUTEEE    PRINT A
018E 20 F4              BRA     NULL1     GOTO NULL1
0190 39         NULOUT  RTS               RETURN FROM SUBROUTINE
                *
0191 86 04      LINK7   LDA A   #$04      A='EOT'
0193 A7 00              STA A   0,X       STORE A @ X
0195 86 0A              LDA A   #$0A      A='LF'
0197 BD E1D1            JSR     OUTEEE    PRINT A
019A CE 0500    LINK8   LDX     #BUFFER    X=START OF BUFFER
019D A6 00      LINK9   LDA A   0,X       LOAD A @ X
019F 81 2F              CMP A   #$2F      IF A='/'
01A1 27 41              BEQ     LINK13    THEN LINK13
01A3 81 5C              CMP A   #$5C      ELSE IF A=' '
01A5 27 45              BEQ     LINK14    THEN LINK14
01A7 81 2A              CMP A   #$2A      ELSE IF A='*'
01A9 27 25              BEQ     LINK10    THEN LINK10
01AB 81 23              CMP A   #$23      ELSE IF A='#'
01AD 27 4E              BEQ     LINK16    THEN LINK16
01AF 81 24              CMP A   #$24      ELSE IF A='$'
01B1 27 4F              BEQ     LINK17    THEN LINK17
01B3 81 26              CMP A   #$26      ELSE IF A='&'
01B5 27 50              BEQ     LINK18    THEN LINK18
01B7 81 25              CMP A   #$25      ELSE IF A='%'
01B9 27 51              BEQ     LINK19    THEN LINK19
01BB 81 1B              CMP A   #$1B      ELSE IF A='ESC'
01BD 27 35              BEQ     LINK15    THEN LINK15
01BF 81 04              CMP A   #$04      ELSE IF A='EOT'
01C1 27 BC              BEQ     LINK6A    THEN LINK1
01C3 97 0B              STA A   INP1      INP1=A
01C5 08                 INX               X=X+1
01C6 A6 00              LDA A   0,X       LOAD A @ X
01C8 97 0C              STA A   INP2      INP2=A
01CA 08                 INX               X=X+1
01CB DF 13              STX     THERE     THERE=X
01CD 7E 0211            JMP     LOOP      GOTO LOOP
                *
                * CLEARS OUT OLD DATA
                *
01D0 DF 15      LINK10  STX     HERE      HERE=X
01D2 CE 1FF0            LDX     #$1FF0    X=START OF OLD DATA
01D5 6F 00      LIMP10  CLR     0,X       STORE 0 @ X
01D7 08                 INX               X=X+1
01D8 8C 2000            CPX     #$2000    IF X= END OF OLD DATA
01DB 27 02              BEQ     LINK11    THEN LINK11
01DD 20 F6              BRA     LIMP10    ELSE LIMP10
01DF DE 15      LINK11  LDX     HERE      X=HERE
01E1 08         LINK12  INX               X=X+1
01E2 20 B9              BRA     LINK9     GOTO LINK9
                *
```

```
                      * CAUSES F0 TO BE INCREASED
                      *
01E4 96 0D            LINK13 LDA A  F0        A=F0
01E6 8B 05                   ADD A  #$05      A=A+5
01E8 97 0D                   STA A  F0        F0=A
01EA 20 F5                   BRA    LINK12    X=X+1 & GOTO LINK9
                      *
                      * CAUSES F0 TO BE DECREASED
                      *
01EC 96 0D            LINK14 LDA A  F0        A=F0
01EE 80 05                   SUB A  #$05      A=A-5
01F0 97 0D                   STA A  F0        F0=A
01F2 20 ED                   BRA    LINK12    X=X+1 & GOTOLINK9
                      *
                      * RETURN TO MIKBUG SAVING STARTING ADDRESS
                      *
01F4 CE 0100          LINK15 LDX    #LINK1    X=LINK1
01F7 FF A048                 STX    $A048     STORE X AS P.C. ON STACK
01FA 7E E0D0                 JMP    START     GOTO START OF MIKBUG
                      *
                      * INITIATE WHISPERING
                      *
01FD 7C 000E          LINK16 INC    VOICE     VOICE=VCICE+1
0200 20 DF                   BRA    LINK12    X=X+1 & GOTO LINK9
                      *
                      * INITIATE VOICING
                      *
0202 7F 000E          LINK17 CLR    VOICE     VOICE=0
0205 20 DA                   BRA    LINK12    X=X+1 & GOTO LINK9
                      *
                      * INITIATE PARAMETER DUMP
                      *
0207 7C 000F          LINK18 INC    HRDCPY    HRDCPY=HRDCPY+1
020A 20 D5                   BRA    LINK12    X=X+1 & GOTO LINK9
                      *
                      * INITIATE NORMAL SYNTHESIS
                      *
020C 7F 000F          LINK19 CLR    HRDCPY    HRDCPY=0
020F 20 D0                   BRA    LINK12    X=X+1 & GOTO LINK9
                      *
                      *
                      * START OF SYNTHESIS PROGRAM
                      *
                      *
0211 CE 1000          LOOP   LDX    #TABLE    SET X TO TOP OF TABLE
0214 A6 00            LOOP1  LDA A  0,X       LOAD FIRST CHAR
0216 81 2A                   CMP A  #$2A      IF A='*'
0218 27 09                   BEQ    EOT       THEN GOTO EOT
021A 91 0B                   CMP A  INP1      ELSE IF A=INP1
021C 27 17                   BEQ    LOOP3     THEN LOOP3
021E 08               LOOP2  INX              X=X+1
021F 8D 0B                   BSR    IINX      X=X+21
```

```
0221 20 F1              BRA    LOOP1      GOTO LOOP1
0223 CE 0037 EOT        LDX    #ERROR     X=ERROR
0226 BD E07E            JSR    PDATA1     PRINT ERROR
0229 7E 0321            JMP    EOTA       X=THERE & GOTO LINK9
                   *
                   * SUBROUTINE IINX X=X+21 B=B A=0
                   *
022C 86 15   IINX       LDA A  #$15       A=21
022E 08      AGAIN      INX               X=X+1
022F 4A                 DEC A             A=A-1
0230 81 00              CMP A  #$00       IF A#0
0232 26 FA              BNE    AGAIN      THEN AGAIN
0234 39                 RTS               RETURN FROM SUBPCUTINE
                   *
                   *
0235 A6 01   LOOP3      LDA A  1,X        LOAD SECOND CHAR
0237 91 0C              CMP A  INP2       IF A#INP2
0239 26 E3              BNE    LOOP2      THEN LOOP2
023B DF 17              STX    CENTR      ELSE X=TOP OF SELECTED SECTIO
                   *
                   *DATA TRANSFER
                   *
023D CE 1FE0            LDX    #1FE0      X=START OF NEW DATA
0240 DF 19              STX    CENTX      CENTX=X
0242 DE 17   LOOP4      LDX    CENTR      X=CENTR
0244 A6 02              LDA A  2,X        LOAD DATA @ X+2
0246 08                 INX               X=X+1
0247 DF 17              STX    CENTR      CENTR=X
0249 DE 19              LDX    CENTX      X=CENTX
024B A7 00              STA A  0,X        STORE A @ X
024D 08                 INX               X=X+1
024E DF 19              STX    CENTX      CENTX=X
0250 8C 1FE9            CPX    #$1FE9     IF X=END OF NEW DATA
0253 27 02              BEQ    LOOP4A     THEN LOOP4A
0255 20 EB              BRA    LOOP4
                   *
0257 CE 0000 LOOP4A     LDX    #$0000     X=T1
025A DF 19              STX    CENTX      CENTX=X
025C DE 17   LOOP4B     LDX    CENTR      X=CENTR
025E A6 02              LDA A  2,X        LOAD A @ X+2
0260 08                 INX               X=X+1
0261 DF 17              STX    CENTR      CENTR=X
0263 DE 19              LDX    CENTX      X=CENTX
0265 A7 00              STA A  0,X        STORE TIMING IN ITS PROPER LO
0267 08                 INX               X=X+1
0268 DF 19              STX    CENTX      CENTX=X
026A 8C 000B            CPX    #$000B     IF X=END OF TIMES
026D 27 02              BEQ    LOOP5      THEN LOOP5
026F 20 EB              BRA    LOOP4B     ELSE LOOP4B
                   *
                   * FORM DELTAS
                   *
```

```
0271 96 02    LOOP5  LDA A  T3        LOAD TRANSITION TIME T3
0273 B7 1FC2         STA A  $1FC2     STORE FOR F1
0276 B7 1FC3         STA A  $1FC3     STORE FOR F2
0279 B7 1FC4         STA A  $1FC4     STORE FOR F3
027C B7 1FC7         STA A  $1FC7     STORE FOR FF
027F 96 04           LDA A  T5        LOAD TRANSITION TIME T5
0281 B7 1FC0         STA A  $1FC0     STORE FOR AV
0284 96 06           LDA A  T7        LOAD TRANSITION TIME T7
0286 B7 1FC5         STA A  $1FC5     STORE FOR AH
0289 96 08           LDA A  T9        LOAD TRANSITION TIME T9
028B B7 1FC1         STA A  $1FC1     STORE FOR F0
028E 96 0A           LDA A  T11       LOAD TRANSITION TIME T11
0290 B7 1FC6         STA A  $1FC6     STORE FOR AF
0293 B7 1FC8         STA A  $1FC8     STORE FOR AN
              *
0296 CE 1FC0         LDX    #$1FC0    X=START OF TRANSITION TIMES
0299 A6 20    LOOP6  LDA A  $20,X     A=NEW DATA
029B E6 30           LDA B  $30,X     B=OLD DATA
029D DF 1B           STX    REMB      REMB=X
029F D7 10           STA B  TEMP      TEMP=B
02A1 5F              CLR B            B=0
02A2 90 10           SUB A  TEMP      A=A-TEMP
02A4 C2 FF           SBC B  #$FF      B=SIGN
02A6 C1 FF           CMP B  #$FF      IF RESULT NEGATIVE
02A8 27 12           BEQ    LOOP6B    THEN LOOP6B
02AA DE 1B           LDX    REMB      ELSE X=REMB
02AC E6 00           LDA B  0,X       B=ASSOCIATED TRANSITION TIME
02AE 8D 16           BSR    DIV       FORM A=A/B
02B0 DE 1B    LOOP6A LDX    REMB      X=REMB
02B2 A7 10           STA A  $10,X     STORE A AS DELTA
02B4 08              INX              X=X+1
02B5 8C 1FC9         CPX    #$1FC9    IF X=END OF TRANSITION TIMES
02B8 27 21           BEQ    LOOP7     THEN LOOP7
02BA 20 DD           BRA    LOOP6     ELSE LOOP6
02BC 43       LOOP6B COM A            A=-A
02BD DE 1B           LDX    REMB      X=REMB
02BF E6 00           LDA B  0,X       B=ASSOCIATED TRANSITION TIME
02C1 8D 03           BSR    DIV       FORM A=A/B
02C3 43              COM A            A=-A
02C4 20 EA           BRA    LOOP6A
              *
              *DIVISION SUBROUTINE A=A/B B=$FF X=X
              *
02C6 7F 0011  DIV    CLR    CNTR      CNTR=0
02C9 D7 12           STA B  CNTX      CNTX=B
02CB 7C 0011  DOT    INC    CNTR      CNTR=CNTR+1
02CE 5F              CLR B            B=0
02CF 90 12           SUB A  CNTX      A=A-CNTX
02D1 C2 00           SBC B  #$00      B=SIGN OF SUBTRACTION
02D3 C1 00           CMP B  #$00      IF RESULT POSITIVE
02D5 27 FA           BEQ    DOT       THEN DOT
02D7 96 11           LDA A  CNTR      ELSE A=CNTR
```

```
0209 4A                 DEC A             A=A-1
020A 39                 RTS               RETURN FROM SUBROUTINE
                  *
                  *
020B 96 01     LOOP7    LDA A   T2        LOAD ONSET TIME T2
020D B7 1FB2            STA A   $1FB2     STORE FOR F1
02E0 B7 1FB3            STA A   $1FB3     STORE FOR F2
02E3 B7 1FB4            STA A   $1FB4     STORE FOR F3
02E6 B7 1FB7            STA A   $1FB7     STORE FOR FF
02E9 96 03              LDA A   T4        LOAD ONSET TIME T4
02EB B7 1FB0            STA A   $1FB0     STORE FOR AV
02EE 96 05              LDA A   T6        LOAD ONSET TIME T6
02F0 B7 1FB5            STA A   $1FB5     STORE FOR AH
02F3 96 07              LDA A   T8        LOAD ONSET TIME T8
02F5 B7 1FB1            STA A   $1FB1     STORE FOR F0
02F8 96 09              LDA A   T10       LOAD ONSET TIME T10
02FA B7 1FB6            STA A   $1FB6     STORE FOR [illegible]
02FD B7 1FB8            STA A   $1FB8     STORE FOR [illegible]
0300 96 0E              LDA A   VOICE     A=VOICE
0302 81 00              CMP A   #$00      IF A=0
0304 27 09              BEQ     LOOP8     THEN LOOP8
0306 B6 1FE0            LDA A   $1FE0     ELSE A=AV
0309 B7 1FE5            STA A   $1FE5     AH=A
030C 7F 1FE0            CLR     $1FE0     AV=0
030F B6 1FE1   LOOP8    LDA A   $1FE1     A='F0'
0312 9B 0D              ADD A   F0        A=A+F0
0314 B7 1FE1            STA A   $1FE1     'F0'=A
0317 CE 1FB0   LOOP9    LDX     #$1FB0    X=TOP OF ONSET TIMES
031A 96 00              LDA A   T1        A=T1
031C 4A                 DEC A             A=A-1
031D 97 00              STA A   T1        T1=A
031F 2E 05              BGT     LOOP10    IF A>=0 THEN LOOP10
0321 DE 13     EOTA     LDX     THERE     ELSE X=THERE
0323 7E 019D            JMP     LINK9     GOTO LINK9
0326 3E        LOOP10   WAI               WAIT FOR INTERRUPT
0327 A6 00     LOOP11   LDA A   0,X       LOAD A @ X
0329 2F 0A              BLE     LOOP13    IF A<=0 THEN LOOP13
032B 6A 00              DEC     0,X       ELSE ONSET TIME DECRIMENTED
032D 08        LOOP12   INX               X=X+1
032E 8C 1FB9            CPX     #$1FB9    IF X=$1FB9
0331 27 30              BEQ     LOOP19    THEN LOOP19
0333 20 F2              BRA     LOOP11
0335 A6 10     LOOP13   LDA A   $10,X     LOAD A @ X+10
0337 2F 24              BLE     LOOP18    IF A<=0 THEN LOOP18
0339 A6 40              LDA A   $40,X     ELSE A=OLD DATA
033B AB 20              ADD A   $20,X     A=A+DELTA
033D 2F 18              BLE     LOOP17    IF A<=0 THEN LOOP17
033F E6 30              LDA B   $30,X     ELSE B=NEW DATA
0341 2F 08              BLE     LOOP15    IF A<=0 THEN LOOP15
0343 E6 20     LOOP14   LDA B   $20,X     ELSE B=DELTA
0345 2F 0A              BLE     LOOP16    IF B<=0 THEN LOOP16
0347 A1 30              CMP A   $30,X     ELSE IF A>=NEW DATA
```

```
0349 2E 12                BGT    LOOP18     THEN LOOP18
034B A7 40       LOOP15   STA A  $40,X      ELSE OLD DATA=A
034D 6A 10                DEC    $10,X      TRANSITION TIME DECREMENTED
034F 20 DC                BRA    LOOP12     GOTO LOOP12
0351 A1 30       LOOP16   CMP A  $30,X      IF A<=NEW DATA
0353 2F 0B                BLE    LOOP18     THEN LOOP18
0355 20 F4                BRA    LOOP15     ELSE LOOP15
0357 E6 30       LOOP17   LDA B  $30,X      B=NEW DATA
0359 2E F0                BGT    LOOP15     IF B>=0 THEN LOOP15
035B 20 E6                BRA    LOOP14     ELSE LOOP14
035D A6 30       LOOP18   LDA A  $30,X      A=NEW DATA
035F A7 40                STA A  $40,X      OLD DATA=A
0361 20 CA                BRA    LOOP12     GOTO LOOP12
0363 96 0F       LOOP19   LDA A  HRDCPY     A=HRDCPY
0365 27 B0                BEQ    LOOP9      IF A=0 THEN LOOP9
0367 CE 0030              LDX    #STRING     ELSE X=STRING
036A BD E07E              JSR    PDATA1     PRINT STRING
036D CE 1FF0              LDX    #$1FF0     X=START OF OLD DATA
0370 BD E0CA     LOOP20   JSR    OUT2HS     PRINT PARAMETER+SPACE
0373 8C 1FF9              CPX    #$1FF9     IF X=$1FF9
0376 27 9F                BEQ    LOOP9      THEN LOOP9
0378 20 F6                BRA    LOOP20     ELSE LOOP20
                          END
```

SYMBOL TABLE

```
INEEE    E1AC   OUTEEE   E1D1   PDATA1   E07E   START    E000   OUT2HS   E0CA
BUFFER   0500   TABLE    1000   T1       0000   T2       0001   T3       0002
T4       0003   T5       0004   T6       0005   T7       0006   T8       0007
T9       0008   T10      0009   T11      000A   INP1     000B   INP2     000C
F0       000D   VOICE    000E   HRDCPY   000F   TEMP     0010   CNTR     0011
CNTX     0012   THERE    0013   HERE     0015   CENTR    0017   CENTX    0019
REMB     001B   STRING   0030   ERROR    0037   LINK1    0100   LINK2    0111
CLEAR    013F   LINK3    014D   LINK4    015A   LINK5    0162   LINK6    016B
LINK6A   017F   NULL     0182   NULL1    0184   NULOUT   0190   LINK7    0191
LINK8    019A   LINK9    019D   LINK10   01D0   LIMP10   01D5   LINK11   01DF
LINK12   01E1   LINK13   01E4   LINK14   01EC   LINK15   01F4   LINK16   01FD
LINK17   0202   LINK18   0207   LINK19   020C   LOOP     0211   LOOP1    0214
LOOP2    021E   EOT      0223   IINX     022C   AGAIN    022E   LOOP3    0235
LOOP4    0242   LOOP4A   0257   LOOP4B   025C   LOOP5    0271   LOOP6    0299
LOOP6A   02B0   LOOP6B   02BC   DIV      02C6   DOT      02CB   LOOP7    02DB
LOOP8    030F   LOOP9    0317   EOTA     0321   LOOP10   0326   LOOP11   0327
LOOP12   032D   LOOP13   0335   LOOP14   0343   LOOP15   034B   LOOP16   0351
LOOP17   0357   LOOP18   035D   LOOP19   0363   LOOP20   0370
```

McGill University Computing Centre


```
S00600004844521B
S1130030000A0000003E042A2A2A4552524F522038
S113004050484F4E454D45204E4F5420494F205464
S10D005041424C450D0A0000000473
S11301007F000E7F000F8D29CE0030BDE07ECE052E
S11301100005DE1AC810D2779810F2731812B2[illegible]3A6E
S1130120812227768140273A810027E5A70004200D
S1130130E08680B71FF2B71FF3B71FF4B71FF74F5E
S1130140B71FF0B71FF5B71FF6B71FF8398C0500B6
S1130150275F098608BDE1D120B7CE0030BDE07EBF
S113016020AF8C05002704A604A7008612BDE1D1C8
S11301708D10CE0500BDE07E8D088613BDE1D17ED5
S11301800100C61A5AC10027078600BDE1D120F438
S113019039860 4A700860ABDE1D1CE0500A60081F8
S11301A02F2741815C2745812A27258123274E81DA
S11301B024274F8126275081252751811B273581EC
S11301C004278C970BC8A600970C08DF137E0211C6
S11301D00F15CE1FF06F00088C2000270220F6DE0A
S11301E0150820B9960D8805970D20F5960D800501
S11301F0970D20EDCE0100FFA0487EE0D07C000EDC
S113020020DF7F000E20DA7C000F20D57F000F2036
S1130210D0CE1000A600812A2709910B2717088D3C
S11302200820F1CE0037BDE07E7E03218615084AFF
S1130230810026FA39A601910C26E3DF17CE1FE0D0
S1130240DF19DE17A60208DF17DE19A70008DF1979
S11302508C1FE9270220EBCE0000DF19DE17A6026F
S113026008DF17DE19A70008DF198C0008270220 0E
S1130270EF9602B71FC2B71FC3B71FC4B71FC796F9
S113028004B71FC09606B71FC59608B71FC1960AC4
S1130290B71FC6B71FC8CE1FC0A620E630DF1BD7C6
S11302A0105F9010C2FFC1FF2712DE1BE6008D16FF
S11302B0DE1BA710088C1FC9272120DD43DE1BE6A7
S11302C0008D034320EA7F001107127C00115F9058
S11302D012C200C10027F496114A399601B71FB221
S11302E0B71FB3B71FB4B71FB79603B71FB09605B0
S11302F0B71FB59607B71FB19609B71FB6B71FB892
S1130300960EA1002709B61FE0B71FE57F1FE086F0
S11303101FF19B0DB71FE1CE1FB096004A97002E38
S113032005DE137E019D3EA6002F0A6400088C1F7D
S1130330B9273020F2A6102F24A640AB202F18E6B0
S1130340302F08E6202F0AA1302E12A7406A102071
S1130350DCA1302F0820F4E6302EF020E6A630A7EA
S1130360402 0CA960F27B0CE0030BDE07ECE1FF0ED
S10D0370BDE0CABC1FF9279F20F698
S9030000FC
```


```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ A C C O U N T I N G $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$                                                                                    $
$    15.21  CPU SECS.        501 I/O REQS.          615 LINES.          454 CARDS    $
$                                                                                    $
$                             CPU CHARGE        2.53                                 $
$                             I/O CHARGE        0.83                                 $
$                             U/R CHARGE        1.15                                 $
$                                SERVICE        0.40                                 $
$                                         ----------                                 $
$                             TOTAL             4.91                                 $
$                                                                                    $
$          YOU HAVE       73.20 REMAINING                                            $
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

APPENDIX B

INTERRUPT ROUTINE LISTING

```
0000                 NAM NMI
0000           *
0000           * SYNTHESIZER INTERRUPT-SCAN ROUTINE
0000           * NOTE MIKBUG USES A006, A007 TO CONTAIN
0000           * THE INTERRUPT ROUTINE LOCATION (1500)
0000           *
1500                 ORG $1500
1500 CE 1FF0         LDX #$1FF0    X=$1FF0
1503 C6 03     THEN  LDAB #$03     B=3
1505 5A        HERE  DECB          B=B-1
1506 A6 00           LDAA 0,X      A=VALUE @ X (TRANSFER CONTENTS FROM
1508 A7 10           STAA 10,X     STORE A @ X+16   OLD DATA TO SYNTHESIZER)
150A C1 00           CMPB #$00     IF B=0
150C 27 02           BEQ THIS      THEN GOTO THIS
150E 20 F5           BRA HERE      ELSE GOTO HERE
1510 08        THIS  INX           X=X+1
1511 8C 1FF9         CPX #$1FF9    IF X=$1FF9
1514 27 02           BEQ THEM      THEN GOTO THEM
1516 20 EB           BRA THEN      ELSE GOTO THEN
1518 86 FF     THEM  LDAA #$FF     A=$FF
151A B7 200F         STAA $200F    STORE A @ $200F  (TURN ON AUDIO)
151D 3B              RTI           RETURN FROM INTERRUPT
```

APPENDIX C

PHONEME LOOK-UP TABLE

PHONEME TABLE

1000

1000	II	00	30	00	00	00	00	80	BB	00	08	00	01	00	01	02	01	00	01	02	01
1016	OO	00	80	80	80	80	00	00	80	00	04	00	01	00	01	00	01	00	01	00	01
102C	IY	90	30	D0	4E	80	00	00	80	00	0A	00	05	00	03	00	03	00	06	00	01
1042	[illegible]	80	30	BE	C2	C4	00	00	80	00	0A	00	03	00	03	00	03	00	06	00	01
1058	AE	80	30	53	7E	C8	00	00	80	00	0A	00	03	00	03	00	03	00	06	00	01
106E	AJ	50	30	65	BA	9A	00	00	80	00	0A	00	03	00	03	00	03	00	06	00	01
1084	IH	80	30	AC	58	90	00	00	80	00	0A	00	06	00	03	00	03	00	06	00	01
109A	EH	80	30	7E	68	AD	00	00	80	00	0A	00	03	00	03	00	03	00	06	00	01
10B0	AH	80	30	6C	A0	C8	00	00	80	00	0A	00	03	00	03	00	03	00	06	00	01
10C6	UH	80	30	A6	A8	C4	00	00	80	00	0A	00	03	00	03	00	03	00	06	00	01
10DC	ER	80	30	35	96	FF	00	00	80	00	0A	00	03	00	03	00	03	00	06	00	01
10F2	EB	80	30	C0	FF	FF	50	30	FF	00	07	00	05	01	01	02	01	00	01	03	01
1108	SS	00	30	25	64	9C	00	80	80	00	10	00	02	00	03	00	03	00	06	00	01
111E	WW	67	30	BE	EB	C8	80	00	80	00	06	00	06	00	06	00	06	00	06	00	02
1134	NN	60	30	A0	20	20	00	00	80	FF	0A	00	02	00	03	00	03	00	06	00	02
114A	YY	80	30	BE	52	83	00	00	80	00	0A	00	03	00	03	00	03	00	06	00	01
1160	LL	80	30	A2	B1	A5	00	00	80	00	0A	00	03	00	03	00	03	00	06	00	01
1176	MM	60	30	FF	53	B3	00	00	80	FF	0A	00	02	00	03	00	03	00	06	00	02
118C	AA	60	30	53	BA	A2	00	00	80	00	0A	00	03	00	03	00	03	00	06	00	01
11A2	CH	A0	30	98	54	88	30	FF	96	00	08	00	01	01	02	00	01	00	06	00	01
11B8	KK	90	30	80	70	FF	60	60	A0	00	09	00	01	06	01	00	01	00	01	00	01
11CE	PP	A0	30	80	B0	E0	50	50	FF	00	09	00	01	06	01	00	01	00	01	00	01
11E4	DD	80	30	C0	6A	80	50	18	B0	00	07	00	05	01	01	02	01	00	01	03	01
11FA	HH	06	30	68	69	92	28	00	80	00	0A	00	01	00	01	00	01	00	03	00	01
1210	ZZ	60	30	C1	87	A7	00	70	80	00	09	00	01	00	03	00	03	00	06	00	02
1226	TK	F0	30	8E	7A	C0	46	56	FA	00	02	00	03	00	01	00	01	00	01	00	01
123C	KX	00	30	8E	7A	C0	00	00	FA	00	06	00	03	00	01	00	06	00	06	00	04
1252	GG	80	30	D6	00	00	A0	A0	C0	00	07	00	05	01	01	03	01	00	01	02	01
1268		00	30	80	80	80	00	00	80	00	40	00	06	00	06	00	06	00	06	00	06
127E	FF	00	30	FF	C0	B4	00	60	50	00	0A	00	02	00	06	00	06	00	01	00	01
1294	VV	60	30	FF	A8	C4	00	50	50	00	0A	00	02	00	02	00	02	00	01	00	01
12AA	TH	00	30	F0	8A	C8	00	60	00	00	0A	00	02	00	06	00	06	00	01	00	01
12C0	TE	60	30	F0	78	C8	00	30	40	00	0A	00	02	00	06	00	06	00	01	00	01
12D6	NG	60	30	C6	4E	37	00	00	80	FF	0A	00	02	00	06	00	06	00	02	00	02
12EC	IY	FF	40	CC	50	8C	00	00	80	00	1A	00	05	00	03	00	03	00	06	00	01
1302	[illegible]	C7	40	BE	C2	C4	00	00	80	00	1A	00	03	00	03	00	03	00	06	00	01
1318	AE	C7	40	5F	70	B4	00	00	80	00	1A	00	03	00	03	00	03	00	06	00	01
132E	AJ	C7	40	71	C6	B4	00	00	80	00	1A	00	03	00	03	00	03	00	06	00	01
1344	IH	C7	40	9E	5F	A7	00	00	80	00	1A	00	06	00	03	00	03	00	06	00	01
135A	EH	C7	40	79	68	AD	00	00	80	00	1A	00	09	00	05	00	06	00	06	00	01
1370	AH	B7	40	6E	9C	B6	00	00	80	00	1A	00	03	00	03	00	03	00	06	00	01
1386	UH	C7	40	70	AA	B0	00	00	80	00	1A	00	03	00	03	00	03	00	06	00	01
139C	ER	C7	40	96	A3	FF	00	00	80	00	1A	00	03	00	03	00	03	00	06	00	01
13B2	WW	67	40	BE	EB	C8	80	00	80	00	08	00	02	00	01	00	01	00	01	00	01
13C8	NN	C4	40	A0	20	20	00	00	80	FF	1A	00	03	00	03	00	03	00	06	00	01
13DE	YY	C7	40	BE	52	83	00	00	80	00	1A	00	03	00	03	00	03	00	06	00	01
13F4	LL	C7	40	A3	B1	A5	00	00	80	00	1A	00	03	00	03	00	03	00	06	00	01
140A	MM	C7	40	FF	53	B3	00	00	80	FF	1A	00	03	00	03	00	03	00	06	00	01
1420	AA	C7	40	53	A7	B1	00	00	80	00	1A	00	03	00	03	00	03	00	06	00	01
1436	CH	60	40	98	54	88	00	40	96	00	10	00	01	00	01	00	01	00	06	00	01
144C	ZZ	80	40	C1	87	A7	00	40	64	00	1A	00	10	00	10	00	10	00	06	00	01
1462	NG	C7	40	C6	4E	37	00	00	80	FF	1A	00	06	00	06	00	06	00	06	00	02
1478	SH	00	30	FF	70	E0	00	A0	D0	00	10	00	02	00	01	01	02	00	06	00	01
148E	ZH	60	30	FF	70	E0	00	A0	D0	00	10	00	01	01	01	01	01	00	01	00	01
14A4	JJ	D0	30	FF	70	E0	D0	30	FF	00	06	00	01	01	01	01	01	00	01	00	01
14BA	RR	80	30	C0	96	FF	00	00	80	00	0A	00	03	00	03	00	03	00	06	00	01
14D0	WH	00	30	BE	EB	C8	80	00	80	00	06	00	02	00	02	00	02	00	02	00	02
14E6	**	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
*Addr	L1 L2	AV	F0	F1	F2	F3	AH	AF	FF	AN	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11

LOWER CASE CHARACTERS

(HEXADECIMAL NOTATION) --NOTE FIGURE 15