Report NASA--4
Technical Report No. 22
N71-35305

PULSE INTERVAL REPRESENTATIONS OF SPEECH EVENTS

J. W. Atwood

Grant NGR 14-005-111
National Aeronautics and Space Administration
and
Grant AFOSR 70-1865
Air Force Office of Scientific Research

Biological Computer Laboratory
Electrical Engineering Research Laboratory
Engineering Experiment Station
University of Illinois
Urbana, Illinois

BCL

Report NASA-4
(Technical Report No. 22)

PULSE INTERVAL REPRESENTATIONS OF SPEECH EVENTS

by
JOHN WILLIAM ATWOOD

15 October 1970

Jointly Sponsored By

National Aeronautics and Space Administration
Grant NGR 14-005-111

Air Force Office of Scientific Research
Grant AFOSR 70-1865

BIOLOGICAL COMPUTER LABORATORY
ELECTRICAL ENGINEERING RESEARCH LABORATORY
ENGINEERING EXPERIMENT STATION
UNIVERSITY OF ILLINOIS
URBANA, ILLINOIS

UILU-ENG-70-294

ABSTRACT

Analysis, synthesis, and transmission of human speech is approached by a new technique which is based on the computation of time intervals between threshold transitions of the derivative in the speech waveform.

In contrast to most present speech research which postulates that measurements on the amplitude spectrum of the Fourier Transform of the speech waveform are sufficient for complete analysis of this signal, the position taken here is that the sequence in which significant speech events occur, and their temporal relationships to each other, are for the description of speech as important as its frequency-domain properties, and that the dynamic properties and the sustained properties of the speech waveform each contribute to our understanding of human speech.

The proposed method of coding speech, in which only the intervals between transitions are recorded, is shown to have advantages in certain areas of speech analysis, synthesis, and transmission, over other methods which rely essentially on phenomena in the frequency domain.

In particular, it is shown that this coding method provides a basis for transmitting speech with high intelligibility at low transmission rates; permits the measurement of many properties of the speech waveform which are difficult or impossible to evaluate using conventional techniques; and allows synthesis, by rule, of many speech sounds.

ACKNOWLEDGEMENT

The author is indebted to the many individuals who have contributed to the completion of this work. In particular, he would like to express his sincere appreciation to Professor Murray L. Babcock for his day-to-day interest, advice, and encouragement, and to his thesis advisor Professor Heinz Von Foerster for his advice and support of this work.

I would also like to thank:

My colleagues Jordan R. Cohen and Michael P. Hoffman, for many stimulating and rewarding discussions.

Professor H. George Friedman, Jr. and Mr. Richard A. Wells, of the Digital Computer Laboratory, for their advice and assistance in many areas of computer programming.

My financée Noreen, who has been a constant source of encouragement and support.

This research was supported in part by the National Aeronautics and Space Administration, Grant NGR 14-005-111; and in part by the Air Force Office of Scientific Research, Grant 70-1865.

My thanks are extended to Miss Janet Ficken, who typed this report, and to the Electrical Engineering Publications Office, who printed it.

TABLE OF CONTENTS

1. INTRODUCTION

This paper presents a new technique for the analysis, synthesis, and transmission of human speech. This technique has as its basis the computation of time intervals between threshold transitions of the derivative in the speech waveform.

Previous investigators of speech have placed major emphasis on the applicability of Fourier Transform techniques(1) in consequence of an historical development contributed to by Fourier, the mechanical models of Von Kempelen, Wheatstone, and Willis, Ohm's Law of Acoustics, and Helmholtz' resonators (2). The invention of the sound spectograph in the early 1930's provided an instrument which has served to further inbreed this line of research.

The basic assumption which characterizes most present speech research is that measurements on the amplitude spectrum of the Fourier Transform of the speech waveform are sufficient for complete analysis of this signal. However, the position taken here is that the sequence in which significant speech events occur, and their temporal relationships to each other, are for the description of speech as important as its frequency-domain properties. The dynamic properties and the sustained properties of the speech waveform each contribute to our understanding of human speech.

It has been discovered that speech is still highly intelligible, even when it has been converted to a pulse train, whose times of occurrence correspond to transitions of the waveform (or one of its derivatives) through a threshold. This new method of coding speech, in which only the intervals between transitions are recorded, will be shown to have advantages in certain areas of speech analysis, synthesis, and transmission, over other methods which rely essentially on phenomena in the frequency domain.

In particular, it will be shown that this coding method provides a basis for transmitting speech with high intelligibility at low transmission rates; permits the measurement of many properties of the speech waveform which are difficult or impossible to evaluate using conventional techniques; and allows synthesis, by rule, of many speech sounds.

The techniques which have been developed in this study represent a new and basic measurement system, and their successful applications along with future possibilities will be presented.

2. THRESHOLD SAMPLING OF SPEECH

Babcock *et al.* (3) have proposed a novel method of separating the forced response of a measurement system (due to its input signal), from the free (transient) response of that system (due to its stored energy or initial conditions). The assumption is made that at certain times in the production of speech the derivative of the acoustical waveform is sufficiently large as to be considered a discontinuity. The time of occurrence of this discontinuity is defined as the time at which the derivative of the waveform exceeds a certain threshold. This discontinuity will then give rise to a free response, which is considered to be a form of distortion in the measurement system.

In order to test this proposal, the system shown in Figure 1 has been constructed (3). The speech signal S(t) is applied to a "Discontinuity Analyzer", which consists of a differentiator, a threshold senser, and a pulser. The resulting pulse train P(t) is applied through the "Initial Condition" generator to the first measurement system. The output of this system is the free response component $R_1(t)$. This is then subtracted from the complete response $R_2(t)$ of the second measurement system to obtain the forced response O(t).

When a speech event S(t) is applied as input to the discontinuity analyzer, and the pulse train P(t) which is thus produced is monitored by the experimenter, the output from the loudspeaker is still intelligible as speech (3). The intelligibility is dependent on the threshold level, the width of the pulse, and the order of the derivative (3-5).

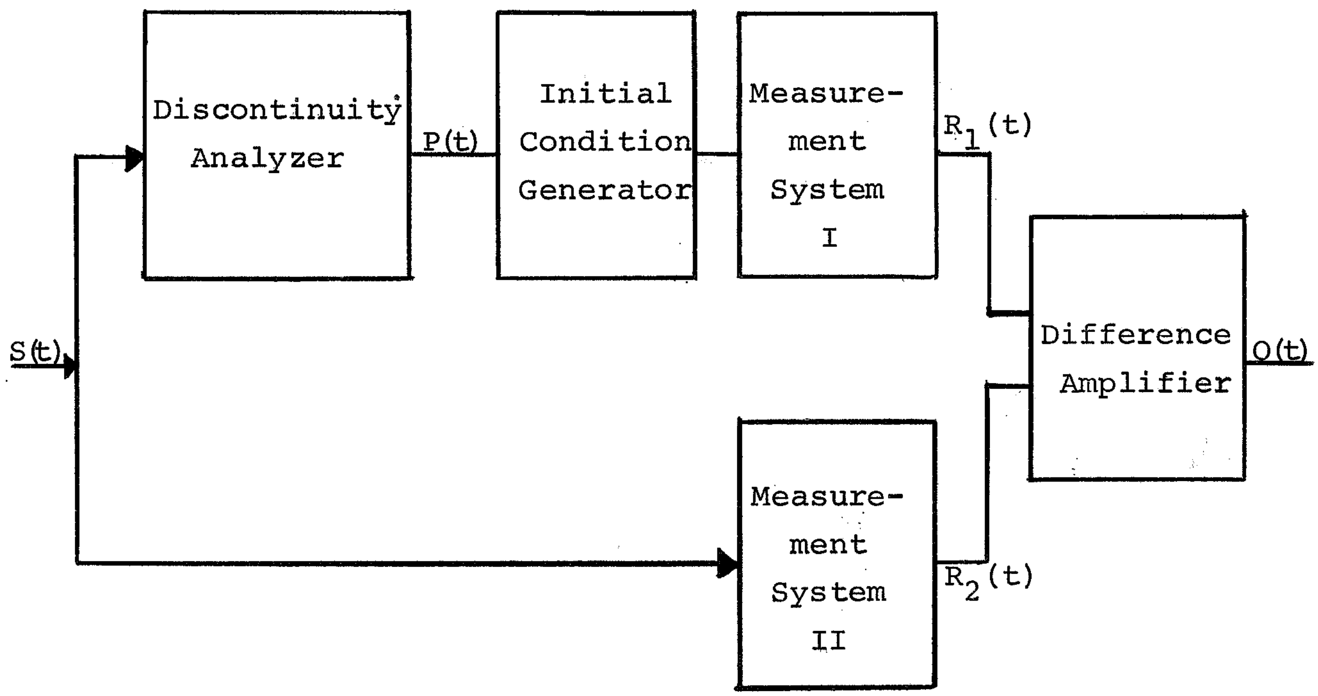

Figure 1. Free Response Distortion Corrector.

The described effect is also observed if the thresholding process is applied to the original speech waveform, without differentiation. The system consisting of the differentiator, the threshold detector, the pulser, and the loudspeaker will be called the Threshold Sampling System in the sequel.

The Threshold Sampling System may be regarded as a new means of coding the speech waveform. The questions which are to be asked now are: is this coding of the speech waveform useful? Does it have any properties which will result in a superior method for transmitting, measuring, synthesizing, or recognizing speech events?

Some answers to these questions will be given below.

2.1 A Model of the Threshold Sampling System

In order to permit a more precise evaluation of the properties of the Threshold Sampling System (TSS), portions of it have been modelled on a digital computer.

The TSS consists of three portions--the differentiator (if used)*, the threshold senser, and the pulse generating components.

The threshold senser has been modelled† using FORTRAN IV on the IBM System/360 computer, model 75, in the Ditigal Computer Laboratory (DCL) of the University of Illinois, Urbana. (This system, and the IBM 1800 mentioned later, are described more fully in Appendix A.) The output of this model is a table of intervals between pulses.

The digitized signal (normal speech or first derivative) is examined point by point, and each time the threshold level

---

*Due to the difficulty of obtaining reliable numerical derivatives (6), it was decided not to attempt to model the differentiator. Another reason was that any numerical derivative would necessarily be different from the analog derivative, and it was not desirable that this be so.

†A listing of the program PIT is found in Appendix B.

is exceeded in the positive direction, the interval between it and the preceding pulse is recorded in the pulse interval table. If the speech was digitized at one sample rate, and it is desired to reproduce the pulses at a different sample rate, the interval is muliptlied by their ratio before being recorded. Due to the use of integer arithmetic in FORTRAN IV, this may or may not be equivalent to sampling the original speech at the second rate. After 100 intervals have been recorded, they are converted to an 8-bit format, and written onto digital tape.

The numbers 0 through 255 can be represented in an 8-bit format ($2^8-1=255$), so the following convention is used: at whatever output sample rate is being used, if the interval is between 1 and 254 samples, that number is used directly as an entry in the pulse interval table. If the interval exceeds 254 samples, then the interval is represented by successive entries of 255 (signifying 254 units of silence), followed by an entry of less than 255. In this way all intervals may be correctly represented. An entry of zero is used to signify end-of-table. For ease of auditing the events, approximately one half second of silence is inserted between each event and the next.

This program also gives statistics concerning the number of seconds of speech analyzed, (including the inserted silences), and the number of entries generated for the tables. From these data the average transmission rate is calculated.

The pulse generating components have been modelled* on the IBM 1800 computer at DCL. The digital tape containing the pulse interval tables is read, and an image of the pulse train is built in memory. This image is then converted to an actual pulse train using the Digital-to-Analog converter described in Appendix A. The program also provides for certain convenient operations, such as changing input tapes, stopping synthesis, etc.

The test sample (described in the next section) has been processed using PIT and PTSYN. It has been shown that the digital model has the same properties as the TSS. Also some statistics have been gathered on the transmission rates to be expected if this model were to be used as a vocoder. However, no attempt has been made to minimize the transmission rate. For example, it is possible that the total number of bits required to specify an event will be less if a 7-bit format is used. Long intervals would then require more entries to represent them. The 8-bit format was chosen entirely for programming convenience.

The statistics and some suggestions for further use of this model appear in Chapter 4.

2.2 The Experimental Sample

A test vocabulary has been prepared, which has been used to evaluate and develop the model of the threshold Sampling

*A listing of program PTSYN is found in Appendix B.

System. This test vocabulary consists of twenty consonant-vowel pairs, chosen at random, i.e., a consonant is selected at random, and then its companion vowel is selected, also at random. The twenty pairs are listed in Table I.

1	zi	11	tæ
2	mi	12	fo
3	ri	13	lo
4	kI	14	dzU
5	ðI	15	dzu
6	fI	16	θʌ
7	vI	17	hʌ
8	gε	18	nʌ
9	hε	19	wʌ
10	sε	20	wɝ

Table I. Word list for Experimental Sample. (IPA Symbols)

Five speakers were chosen, two women and three men. These subjects have a fundamental frequency (glottal excitation rate) ranging from about 250 Hz to about 80 Hz. Each subject was requested to read the list (Table I) five times. One reading for each subject was selected to be in the experimental sample, the criterion for selection being that the reading contain as few mistakes as possible. The resulting master tape contains 100 events.

The original speech waveform for these 100 events, and the first derivative waveform, also for 100 events, were converted to digital samples using the IBM 1800. These two groups of 100 events form the experimental sample which is used for all subsequent development.

2.3 Automatic Pitch Extraction

Cohen (7), working with the output from the model described in Section 2.1, and with oscillographs of the output from the TSS, has shown that it is possible to automatically extract the pitch period (time between successive openings of the glottis) of a voiced speech waveform. The existence of this automatic extraction algorithm is basic to the further development of this study, so it is described here.

The effect which makes the extraction possible is the fact that, for speech input, the output of the TSS at certain threshold levels consists of bunches of pulses followed by a relatively longer interval where no pulses occur. The algorithm is as follows:

1. A number, called a continuing sum, is defined, and initially set to zero.
2. A second number, called a stop number is defined. This number is dependent on the pitch period, but can vary over a certain range.
3. The interval between the present pulse and the

preceding one is determined, and added to the continuing sum.

4a. If the interval in step 3 is less than the stop number, go to step 3 and repeat.

4b. If the interval in step 3 is equal to or greater than the stop number, the contents of the continuing sum is the length of the pitch period.

5. Reset the continuing sum to zero, and return to step 3.

The output of this algorithm is the length of time from the end of one long interval to the end of the next long interval. The distinction long/not-long is made by reference to the stop number.

2.4 The Pulse Position Listings*

The pulse interval representation is an efficient method for transmitting speech. In order to evaluate the usefulness of this representation for measurements on the speech waveform, it is necessary to search for useful features in the output from the model. Many of these features will be dependent on the pitch period. A program has been written to present the output of the model of the threshold senser as printed listings. Whenever the pitch extraction algorithm detects the end of a pitch period, a new line is begun on

*A listing of program PPL is found in Appendix B.

the printed output. Both the pulse intervals and the pulse positions within a pitch period (defined by the value of the continuing sum at each pulse time) are displayed.

An example of the pulse position listing for the event /kI/ is shown in Figure 2. This is for a female speaker with a glottal excitation frequency of approximately 230 Hz. The pitch period, time of occurrence of the pulse, and pulse count are listed, followed by pulse positions and pulse intervals. The symbol "#" is used in column 4 to indicate that 20 pulses have been recorded without detection of a pitch period. The symbol "*" is used to indicate that a pitch period has been detected, but more than 10 pulses (one line) were recorded in the interim.

This example displays most of the features found in the pulse position listings. For example:

1. The interval of length 1781 (103 ms) on the first line is the length of the silence preceding the event.
2. No pitch periods are detected in the next 12 lines. This corresponds to the unvoiced phoneme /k/.
3. The interval of length 560 (32 ms) on the next line corresponds to the silent interval between /k/ and /I/.
4. Voicing then begins, and six pitch periods are detected. Then twenty periods are missed. Eight

EVENT 4, COMP= -5000, STEP= 42

P PER	PTIME	CNT		PULSE POSITIONS										PULSE INTERVALS									
1781	1781	2		1781										1781									
180	1961	22		25	28	32	38	50	63	93	104	108	130	25	3	4	6	12	13	30	11	4	22
			#	136	140	147	151	157	160	163	168	174	180	6	4	7	4	6	3	3	5	6	6
166	2127	42		14	18	31	42	48	51	62	69	73	75	14	4	13	11	6	3	11	7	4	2
			#	80	96	100	130	138	141	154	158	162	166	5	16	4	30	8	3	13	4	4	4
119	2246	62		4	7	11	14	17	21	28	30	34	37	4	3	4	3	3	4	7	2	4	3
			#	46	59	62	69	76	81	85	105	110	119	9	13	3	7	7	5	4	20	5	9
134	2380	82		4	15	24	28	32	37	41	47	59	76	4	11	9	4	4	5	4	6	12	17
			#	85	89	94	98	105	109	117	126	129	134	9	4	5	4	7	4	8	9	3	5
130	2510	102		4	8	11	14	46	49	57	74	81	85	4	4	3	3	32	3	8	17	7	4
			#	87	93	97	104	110	112	116	120	122	130	2	6	4	7	6	2	4	4	2	8
344	2854	122		5	8	17	20	24	28	33	37	46	49	5	3	9	3	4	4	5	4	9	3
			*	55	67	79	108	143	160	178	198	224	344	6	12	12	29	35	17	18	20	26	120
560	3414	123		560										560									
79	3493	125		6	79									6	73								
77	3570	128		8	13	77								8	5	64							
78	3648	131		8	13	78								8	5	65							
78	3726	134		8	12	78								8	4	66							
77	3803	137		8	12	77								8	4	65							
77	3880	139		7	77									7	70								
407	4287	159		7	11	38	76	82	115	151	158	191	226	7	4	27	38	6	33	36	7	33	35
			#	233	266	268	302	309	331	341	376	383	407	7	33	2	34	7	22	10	35	7	24
348	4635	179		2	4	9	45	52	85	120	127	154	160	2	2	5	36	7	33	35	7	27	6
			#	195	202	225	230	234	271	278	301	309	343	35	7	23	5	4	37	7	23	8	39
307	4942	199		7	30	38	69	76	83	107	109	115	153	7	23	8	31	7	7	24	2	6	38
			#	160	186	192	216	230	237	263	270	294	307	7	26	6	24	14	7	26	7	24	13
268	5210	219		7	34	40	64	77	85	103	112	117	142	7	27	6	24	13	8	18	9	5	25
			#	155	162	181	190	196	220	233	240	259	268	13	7	19	9	6	24	13	7	19	9
198	5408	230		30	43	50	68	77	84	121	128	147	155	30	13	7	18	9	7	37	7	19	8
			*	198										43									
78	5486	234		8	26	35	78							8	18	9	43						
78	5564	238		7	26	35	78							7	19	9	43						
77	5641	242		7	26	34	77							7	19	8	43						
78	5719	246		8	27	35	78							8	19	8	43						
77	5796	250		7	26	34	77							7	19	8	43						
76	5872	254		7	26	34	76							7	19	8	42						
76	5948	258		7	26	34	76							7	19	8	42						
76	6024	262		7	27	34	76							7	20	7	42						
373	6397	282		7	26	34	75	82	103	109	150	157	177	7	19	8	41	7	21	6	41	7	20
			#	184	225	232	252	260	299	306	327	333	373	7	41	7	20	8	39	7	21	6	40
146	6543	289		7	28	35	73	80	101	146				7	21	7	38	7	21	45			
71	6614	293		6	19	27	71							6	13	8	44						
71	6685	297		6	19	27	71							6	13	8	44						
71	6756	301		6	19	27	71							6	13	8	44						
70	6826	305		5	19	27	70							5	14	8	43						
69	6895	309		4	18	25	69							4	14	7	44						
67	6962	313		5	18	25	67							5	13	7	42						
69	7031	316		19	27	69								19	8	42							
67	7098	318		17	67									17	50								
68	7166	320		18	68									18	50								
67	7233	322		17	67									17	50								
67	7300	323		67										67									
67	7367	324		67										67									
66	7433	325		66										66									
66	7499	326		66										66									
66	7565	327		66										66									
68	7633	328		68										68									

Figure 2. Pulse position listing for /kI/.

periods are detected, seven are missed, and the remainder are detected.

5. The number of pulses in a pitch period varies from two to four to one.

The listings for the 100 events in the experimental sample have been computed. The differentiated waveform is used for this computation, since it has been shown to produce higher intelligibility scores (5).

Each event in the experimental sample exhibits one or more of the characteristics displayed in the example of Figure 2, viz:

1. Silence.
2. Voicing not present.
3. Voicing present, pitch extraction works.
4. Voicing present, pitch extraction does not work.

The pitch period extraction algorithm appears to fail when the number of pulses per pitch period increases.

In addition, several characteristics of the events are evident, some of which appear to be independent of speaker. However, these characteristics are made clearer by the presentation techniques developed later, so discussion of them will be deferred.

2.5 The Variable Threshold Sampling System*

The major difficulties with the pulse position listings

*A listing of program PPLV is found in Appendix B.

are: (1) The pitch period extraction algorithm often misses one or several periods; and (2) The number of pulses within a pitch period changes with the intensity of the sound. These two problems are related to the fact that if the threshold is kept constant, it is changing relative to the peak value of the event. Also, it appears from examination of the output from the experimental sample that the best level for extraction of the pitch period is different from the best level for measurement of the other properties of the event. The pulse position listings are useful only if the pitch period is determined accurately and consistently.

Cohen, in the meantime, had observed (7), by comparing the output of his algorithm for several threshold levels, that in order to extract pitch reliably over entire speech events, it is necessary to adjust the threshold level, depending on the instantaneous peak amplitude of the waveform. This adjustment has to take place at least every few pitch periods.

Therefore the pulse position listing program was modified to have the threshold level dependent on the peak value of the waveform in previous pitch periods. Two methods of computing the new threshold have been tried. In the first, the threshold is based on the peak value in the just-completed pitch period. In the second, the peak value of the next pitch period is predicted, using the two previous pitch periods, and the threshold is computed from this predicted value.

The first method tends to miss pitch periods at the beginning and ending of voicing, when the peak value is changing very rapidly. The second method does well in these regions, but tends to miss pitch periods in the central portion of an event. By examination of the peak and threshold levels used, it has been shown that this effect is due to local variations in the peak value of the waveform. The derivative of the change in peak value changes sign from one pitch period to the next, causing the predicted peak value to be erroneous.

Since method one misses pitch periods only when the peak amplitude is changing rapidly, the following correction method is used: the continuing sum is compared against an auxilliary stop number which is arbitrarily defined as 2.5 times the stop number. If the continuing sum is greater than the auxilliary stop number, the threshold is continuously recomputed, as each point is examined, until a pitch period is recognized. This addition makes method one superior to method two, so it is used in subsequent development.

An example of the output of this program is shown in Figure 3, for the same event (/kI/) as is shown in Figure 2. All pitch periods in the voiced portion of the waveform, which were missed by the first version of the model, are detected. Slight differences in the length of pitch periods compared with Figure 2 are due to the variable threshold. The extra pulses listed on the last line are not detected as

EVENT 4, STOP= 42, GLOTP= 70, RECP= 20

PTIME COMPG	P PER		PULSE POSITIONS										PULSE INTERVALS									
1343 -3000	1343		1343										1343									
1585 -3000	242		242										242									
1675 -3000	90		6	13	18	90							6	7	5	72						
1868 -5753			28	44	72	78	101	105	120	123	126	130	28	16	28	6	23	4	15	3	3	4
		#	132	137	143	150	155	159	168	172	183	193	3	4	6	7	5	4	9	4	11	10
1974-11943			198	203	209	213	224	229	234	237	241	244	5	5	6	4	11	5	5	3	4	3
		#	247	251	256	262	265	268	273	279	285	299	3	4	5	6	3	3	5	6	6	14
2008-11943	333		303	316	320	327	330	333					4	13	4	7	3	3				
2101 -8803	93		3	8	14	18	21	25	27	31	37	43	3	5	6	4	3	4	2	4	6	6
		*	48	52	56	60	67	76	82	87	89	93	5	4	4	4	7	9	6	5	2	4
2188 -9249			4	13	17	21	25	29	32	35	39	42	4	9	4	4	4	4	3	3	4	3
		#	46	53	55	59	62	71	79	82	84	87	4	7	2	4	3	9	8	3	2	3
2235 -9571	134		90	94	98	101	105	110	115	124	129	134	3	4	4	3	4	5	5	9	5	5
2308 -7739			5	7	10	12	14	20	22	25	34	38	5	2	3	2	2	6	2	3	9	4
		#	42	44	47	49	51	53	56	63	69	73	4	2	3	2	2	2	3	7	6	4
2397 -7495			86	91	95	99	103	108	115	119	123	125	13	5	4	4	4	5	7	4	4	2
		#	127	132	136	139	144	148	151	155	158	162	2	5	4	3	5	4	3	4	3	4
2436 -8982	201		177	180	183	190	193	201					15	3	3	7	3	8				
2513 -6308			6	11	13	16	21	24	27	30	35	40	6	5	2	3	5	3	3	3	5	5
		#	47	53	55	58	63	65	67	69	73	77	7	6	2	3	5	2	2	2	4	4
2604 -8867			81	85	88	90	93	97	100	106	110	119	4	4	3	2	3	4	3	6	4	9
		#	122	128	132	140	149	152	156	159	165	168	3	6	4	8	9	3	4	3	6	3
2707 -6907			172	178	181	188	191	199	203	208	211	216	4	6	3	7	3	8	4	5	3	5
		#	222	224	226	233	237	244	251	255	267	271	6	2	2	7	4	7	7	4	12	4
2733 -4743	297		277	297									6	20								
2847 -3074	114		12	39	44	46	53	61	65	75	78	83	12	27	5	2	7	8	4	10	3	5
		*	94	101	114								11	7	13							
3412 -4967	565		6	18	23	132	231	473	496	502	565		6	12	5	109	99	242	23	6	63	
3490 -4967	78		6	15	19	24	78						6	9	4	5	54					
3566 -4176	76		6	14	76								6	8	62							
3644 -6115	78		13	78									13	65								
3719 -7926	75		14	75									14	61								
3798 -8391	79		18	43	79								18	25	36							
3874 -9809	76		16	41	46	74	76						16	25	5	28	2					
3949-10523	75		7	16	40	47	75						7	9	24	7	28					
4025-11823	76		8	41	50	76							8	33	9	26						
4099-13210	74		8	17	38	41	49	74					8	9	21	3	8	25				
4174-15169	75		9	18	35	39	41	48	75				9	9	17	4	2	7	27			
4250-16941	76		9	36	42	76							9	27	6	34						
4324-16174	74		8	35	38	40	74						8	27	3	2	34					
4400-18053	76		9	37	44	76							9	28	7	32						
4476-18429	76		9	38	43	76							9	29	5	33						
4550-15960	74		8	34	38	42	46	74					8	26	4	4	4	28				
4630-19671	80		10	36	40	42	77	80					10	26	4	2	35	3				
4705-18466	75		7	32	41	73	75						7	25	9	32	2					
4780-15174	75		9	34	42	75							9	25	8	33						
4860-18688	80		11	37	44	80							11	26	7	36						
4936-17747	76		7	34	42	67	76						7	27	8	25	9					
5014-18502	78		9	35	44	69	78						9	26	9	25	9					
5093-19013	79		8	35	44	69	79						8	27	9	25	10					
5169-18527	76		7	35	43	68	76						7	28	8	25	8					
5248-14021	79		9	30	36	46	69	79					9	21	6	10	23	10				
5326-17654	78		9	28	36	45	78						9	19	8	9	33					
5404-17248	78		9	28	36	78							9	19	8	42						
5482-17823	78		9	28	36	78							9	19	8	42						
5559-13710	77		9	28	35	77							9	19	7	42						
5637-18202	78		10	29	36	78							10	19	7	42						
5714-17929	77		9	28	36	77							9	19	8	41						
5791-16896	77		9	28	36	77							9	19	8	41						
5867-16567	76		9	28	36	76							9	19	8	40						
5943-14186	76		10	28	37	76							10	18	9	39						
6018-15248	75		10	29	37	75							10	19	8	38						
6094-17624	76		11	29	38	76							11	18	9	38						
6168-15499	74		10	29	37	74							10	19	8	37						
6243-16661	75		11	30	38	67	75						11	19	8	29	8					
6317-17987	74		11	29	38	74							11	18	9	36						
6392-18152	75		11	29	39	75							11	18	10	36						
6465-18377	73		10	28	38	73							10	18	10	35						
6538-19409	73		10	29	73								10	19	44							
6609-16877	71		10	22	28	71							10	12	6	43						
6680-19317	71		10	23	29	71							10	13	6	42						
6750-18915	70		10	23	29	70							10	13	6	41						
6820-18369	70		10	23	30	70							10	13	7	40						
6889-15096	69		9	22	29	69							9	13	7	40						
6957-16774	68		10	22	29	68							10	12	7	39						
7025-16622	68		10	22	29	68							10	12	7	39						
7093-16515	68		22	68									22	46								
7159-14667	66		21	66									21	45								
7227-12682	68		10	22	68								10	12	46							
7294-13356	67		22	67									22	45								
7361-11527	67		9	23	31	67							9	14	8	36						
7428-11282	67		22	67									22	45								
7494 -9325	66		22	61	66								22	39	5							
7561 -8019	67		24	67									24	43								
7629 -6620	68		61	68									61	7								
7905 -4657	-3000		67	136	144	208	210	276					67	69	8	64	2	66				

Figure 3. Variable threshold pulse position listing for /kI/.

pitch periods, but are printed by the program for informational purposes. There are four pitch periods on this line. Their level is just above the minimum threshold value (-3000) for this program, but below the fixed threshold level (-5000) used in Figure 2.

The listings for the 100 events in the experimental sample have been computed, using the differentiated waveform. The pitch extraction algorithm, with corrections, works very well for female voices, but still fails to detect some pitch periods for male voices.

The pulse positions in adjacent pitch periods (see Figure 3) are subject to some variation. Pulses appear and disappear, and pulses often shift one or two places (sample times).

Comparison of phonemes spoken by the same subject in different contexts (e.g.,/i/ in /zi/, /mi/, /ri/) demonstrate that they are very similar. In addition, events spoken by different speakers show some of the same similarities (and some differences). The pulse position specification of speech thus appears to have usefulness as an analysis and measurement tool. The development of a superior method for presenting the information is the subject of the next chapter.

3. PULSE POSITION SPECIFICATION OF SPEECH

The existence of relationships between the pulse positions in successive pitch periods is now clear. The automatic extraction of pitch has been improved to the point where few pitch periods are missed, so that the pulse position listings are easy to read. It is evident that the pulse positions within a pitch period, and the pitch period itself, are subject to some variation from period to period.

The evaluation of these variations is difficult, however. Pulses are occasionally missed, (not due to any fault of the processing programs, but to some local anomaly in the waveform), so it is not possible to design a simple algorithm for grouping the pulses together. It is now necessary to develop a method of presentation better suited to be interpreted by the experimenter. (Not because the experimenter is going to ultimately recognize the events, but because he has to understand what is going on in order to instruct the machine.)

3.1 The Pulse Position Charts*

One method of making temporal relationships clearer is to present them spatially. Therefore a program has been written which prints the pulse positions for each pitch period as asterisks on a line. Thus pulse position is presented

*A listing of program PPC is found in Appendix B.

against period number. Time increases with period number, but not linearly unless the pitch period is constant.

The relationship between the pulse positions in adjacent pitch periods, presented in Figure 3, is much easier to interpret in the pulse position chart presentation of Figure 4. From such a chart it is possible to determine such useful parameters as pitch period, changes in pitch period, (both short and long term), and similar parameters for the other pulse positions as will be discussed later.

3.2 Pitch Period Correction

The pulse position charts for the 100 events in the experimental sample have been computed, using the differentiated waveform. Examination of these charts shows the possibility of improving the operation of the pitch extraction algorithm. Once the pitch extraction algorithm has detected a pitch period, pulses may be moved between adjacent pitch periods, in order to make the presentation of the pulse positions more accurate and consistent. The decision to change the pitch period boundaries is based on three tests.

The first made is for voicing. In order for voicing to be present, the length of the present pitch period must be within a certain percentage η of the length of the previous pitch period. This percentage may be chosen by the experimenter. If this test is failed, the next test is not applied.

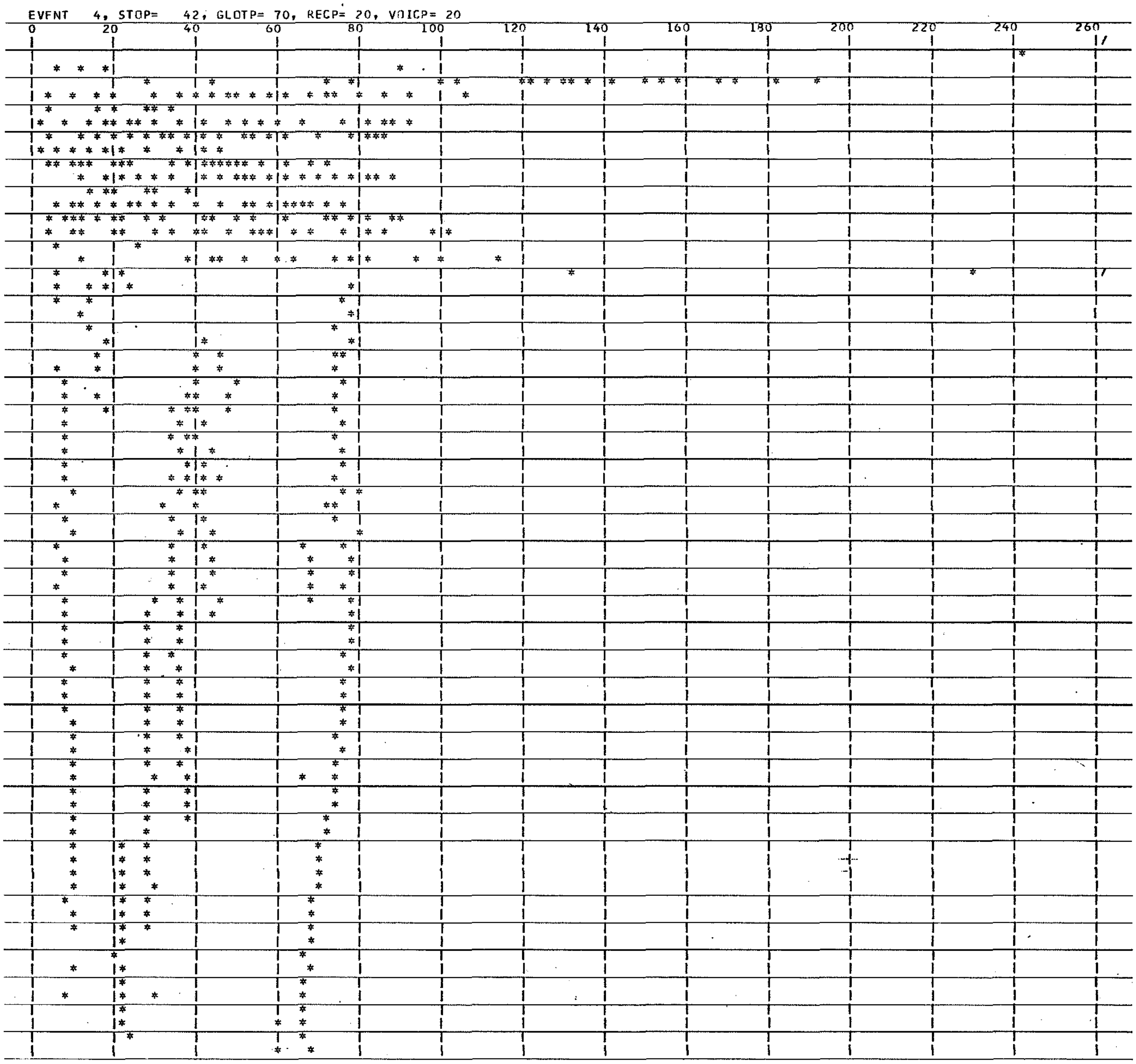

Figure 4. Pulse position chart for /kI/.

The next test is made for skipping. Figure 5 illustrates this effect. At the point labeled A on Figure 5, the pitch period is longer than expected. At the point labeled B, a short period occurs, as the pitch period algorithm skips back. This is a consequence of the recomputation of the threshold level at the end of each pitch period, and of the period-to-period variation in peak amplitude. Figure 6 illustrates the derivative waveform from a portion of the phoneme /i/. The horizontal lines drawn are the threshold levels used for that particular pitch period. It is seen that the extra length of segment S in Figure 6 is due to an increase in the threshold level, relative to the peaks P and Q. This is corrected in the pulse position chart program in the following way: if the previous pitch period has pulses at positions

$$p_1, p_2, p_3, \ldots, p_n,$$

and the present pitch period has pulses at positions

$$q_1, q_2, q_3, \ldots, q_{m-1}, q_m,$$

and if q_{m-1} is closer to p_n in value than q_m is, then skipping is said to have occurred. Then q_{m-1} is taken as the pitch period, and

$$r_1 = q_m - q_{m-1}$$

becomes the first pulse position in the next pitch period.

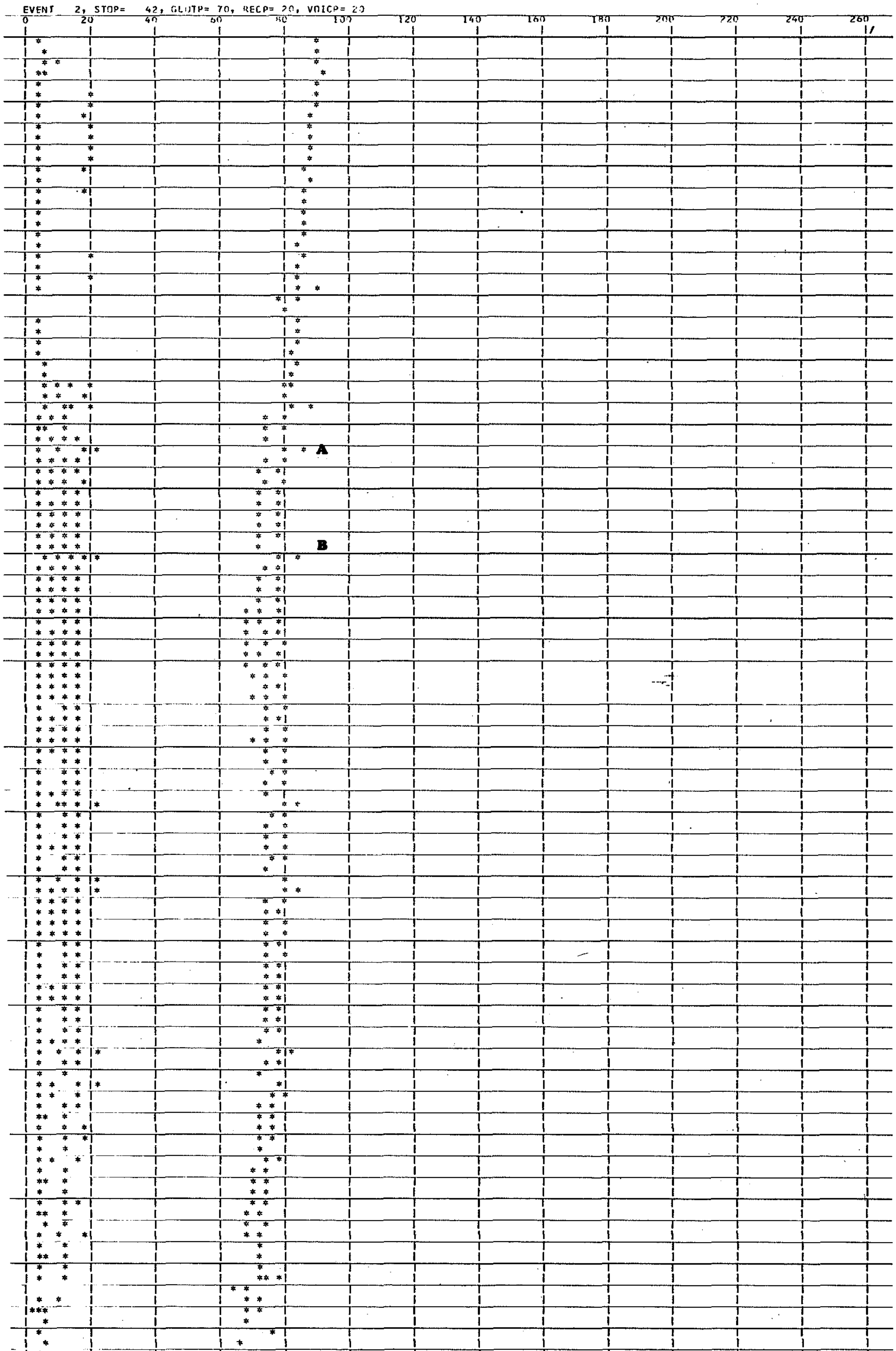

Figure 5. Pulse position chart for /mi/.

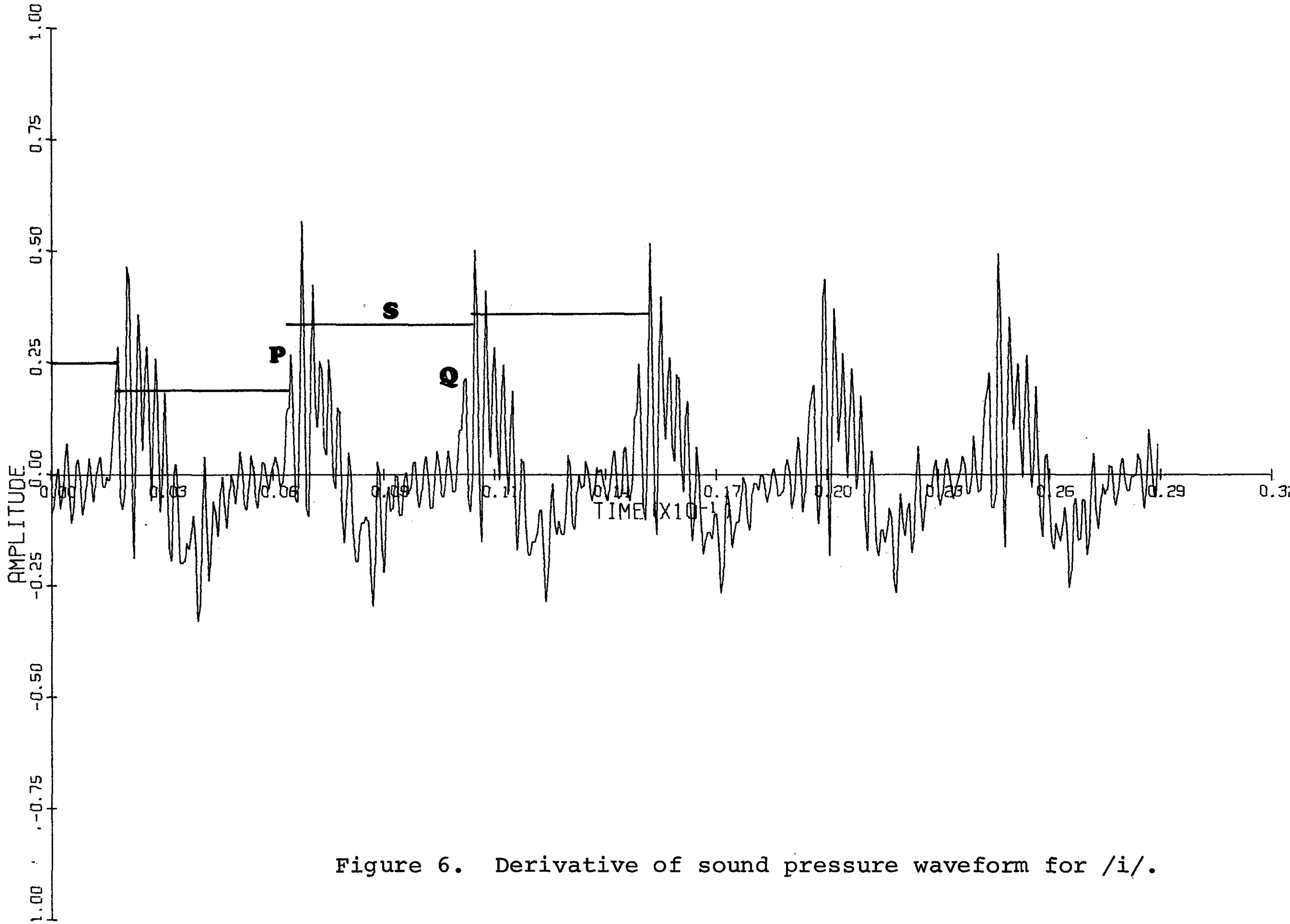

Figure 6. Derivative of sound pressure waveform for /i/.

Double skipping also occurs in the experimental sample. In this case q_{m-2} is closer to p_n than either q_{m-1} or q_m. Two pulses must then be moved to the next pitch period.

If the voicing test is failed, a test is made for double pitch periods. Two criteria are used: the length of the new pitch period must be within η percent of twice the length of the previous pitch period; and there must exist a pulse position which is within η percent of the length of the previous pitch period. If these two conditions are satisfied, the recorded pitch period is split in two. The first half is printed, and the test for skipping is applied to the second half. No test for skipping is necessary on the first half, because the pulse position closest to the length of the previous pitch period is chosen as the length of the first half.

Correction for skipping and double pitch periods results in much more accurate and consistent presentation of the pulse positions. Only when three or more pitch periods are grouped together does the presentation suffer. This rarely happens in the experimental sample.

3.3 Pulse Position Tracking

As can be seen from Figures 4 and 5, the pulse positions for the vowels /i/ and /I/ remain relatively constant through the vowel. This is true for most vowels for all five speakers. One example of an exception noted in the experimental sample is the vowel /o/ which, as observed by Flanagan (8) usually exists as a dipthong in General American dialect. The pulse position chart for the event /lo/ is given in Figure 7 for the same female speaker who produced the samples given in the earlier figures. It is seen that the tracks move about in relation to the ends of the pitch period. However, it has been discovered that it is possible to apply a continuity criterion to the positions, which enables the connecting of the points in a track. The exact nature of this continuity criterion has not been fully investigated, and more will be said about this in Chapter 4.

3.4 Pulse Position Synthesis*

The regularities of the pulse positions indicate the possibility of synthesis of vowels and continuants. A program has been written which produces pulse intervals, given a set of control cards.

No attempt has been made at this time to synthesize events using this program. However, the synthesis program has been used in conjunction with part of the pulse position chart program (which produces as intermediate output card images suitable for controlling the synthesis program) to

*A listing of program PITG is found in Appendix B.

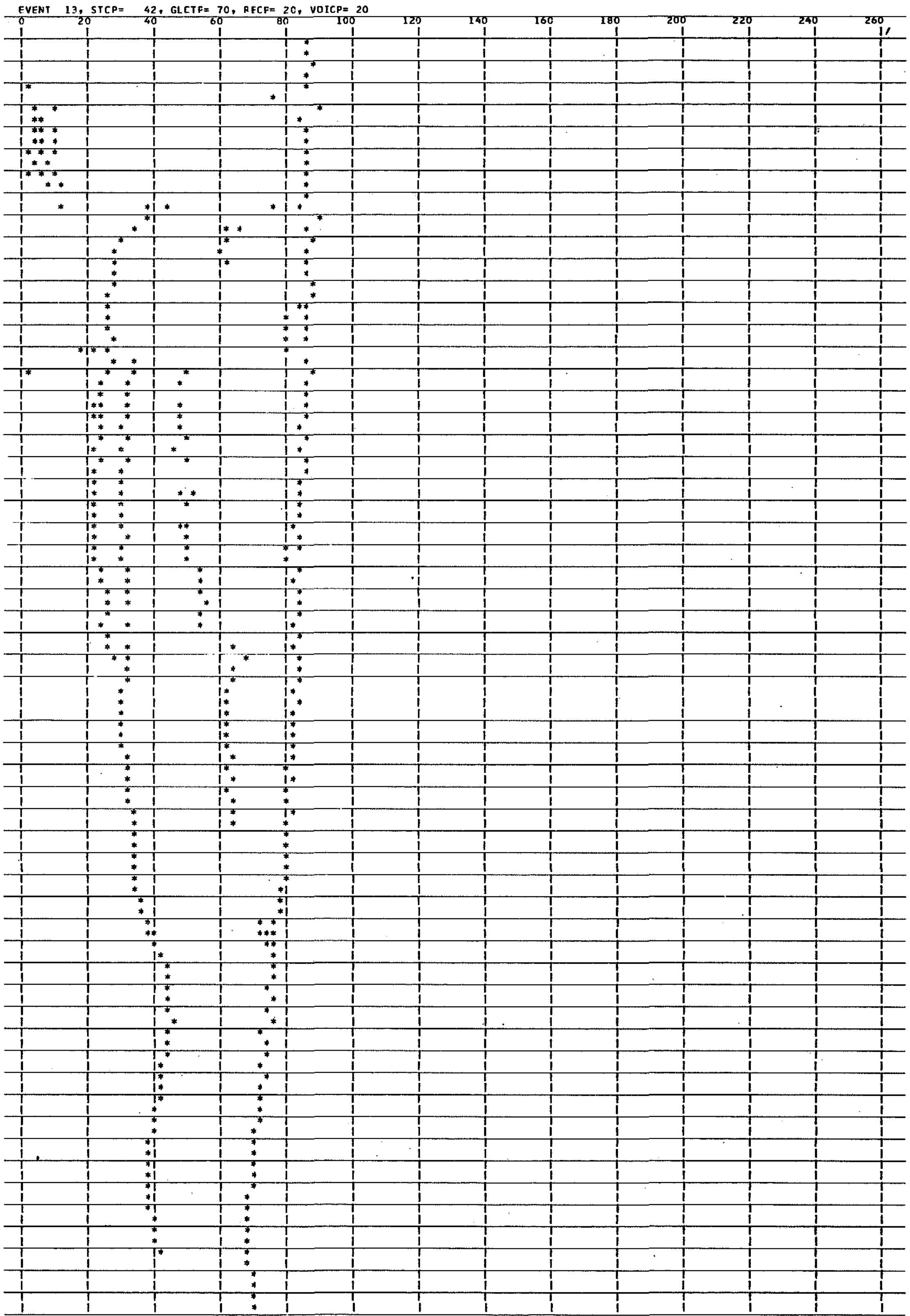

Figure 7. Pulse position chart for /lo/.

test the desirability of varying the threshold in the pulse interval table program (Section 2.1). The results of informal testing indicate that there is an improvement.

4. RESULTS

This study into the temporal structure of speech waveforms has resulted in the development of two useful representations:

1. The pulse interval representation, as a new method of coding speech, provides a basis for the transmitting of speech at low transmission rates, and for the specification of certain unvoiced speech events.
2. The pulse position representation, as a new method of specifying the relationships between pitch periods in speech, provides a basis for measurements on the speech waveform, which has applications primarily in the area of analysis.

The application of these representations to the transmission, analysis, recognition, and synthesis of speech will be discussed in the following four sections.

4.1 Speech Transmission

The Threshold Sampling System has been modelled, and the model has been shown to have the same properties. This model is also a model for a digital vocoder. With a sampling rate of 20,000 samples per second, the average transmission rate for the 100 events in the experimental sample is approximately 3200 bits per second (bps). This figure, however, must be treated only as a preliminary estimate of the requirements for a vocoder, for the following reasons:

1. The transmission quality, although good, is not as good as that of current vocoders at 1600 bps or higher. In particular, low energy unvoiced phonemes are not reproduced properly.
2. It may be possible to lower the transmission rate through the use of a 7-bit or 6-bit format for encoding.
3. The use of a variable threshold system may improve the intelligibility and/or decrease the necessary transmission rate. This could be achieved either by digital techniques, as in the model, or by analog techniques prior to thresholding. For example, a peak rectifier might be used to obtain the current peak value, or the signal might be compressed.
4. The figure of 3200 bps is an average over 100 events. The peak transmission rate could be as high as 80,000 bps for a sampling rate of 20,000 samples per second. A digital memory sufficient to store one second of encoded speech should be sufficient to achieve the indicated average rate.

Two phenomena should be investigated before a vocoder is designed using the pulse interval representation. The first is the effect of the reproducing system on the intelligibility. It has been noted (9) that reproduction appears better if a poor quality loudspeaker is used. The possibility of using L-C circuits in a reproducing system, in order to make the reproduction independent of the loudspeaker, should be examined. The second is the dependence of the intelligibility on the order of the derivative. It may be possible to design a superior vocoder using some combination of the zeroth, first and second order derivatives.

4.2 Speech Analysis

The pulse interval and pulse position representations of speech are new tools for the evaluation of the speech waveform. These tools are not meant, however, to replace any of the existing techniques of speech analysis, but rather to complement them. In particular, much research, from the acoustical, psychological and linguistic points of view, has gone into the development of a "distinctive features"(1) description of speech. It has been the aim of most of this research to determine the time dependence of the frequency-domain parameters of speech events.

The pulse interval and pulse position representations are techniques for the determination of the time dependence of the time-domain parameters of speech events. Therefore, much of the experimentation which has been done must be repeated in the context of this new measurement system. Only in this way will it be possible to evaluate the ultimate usefulness of this method of speech analysis.

The pitch extraction algorithm, which is basic to the presentation of the pulse positions, has been improved over the original proposal by Cohen. Each improvement is subject to one constraint: that it be causal. No decision is made based on future information not yet available to the processing programs. In this way it is always possible to implement in hardware each of the algorithms discussed.

Based on the results from the analysis of the experimental sample, the threshold level for pitch extraction should be set at 70 percent of the current peak value of the waveform.

The stop number used for each speaker was picked somewhat carefully at the beginning of the experiments, in order to minimize errors. The effect of stop number variation on the ability of the algorithm to extract pitch has not been investigated. It is suspected that the tolerance of the algorithm to variation of stop number is improved by the other correction algorithms.

The number of pulses per pitch period increases rapidly as the threshold level is dropped to 30 percent of the current peak value, and then increases more slowly. This is in accord with the findings of Babcock (5) that intelligibility scores improve as the threshold level is lowered to 30 percent of peak level, and remain essentially constant below that point. This indicates that a threshold level of 20 to 30 percent of the current peak value is optimum for specification of speech events.

4.3 Speech Recognition

Automatic recognition of speech events has not been attempted. The pulse interval and pulse position specifications of speech are useful for automatic recognition of many distinctive features of speech. For example, it is possible to detect voicing, when it occurs. However, it is not always possible to detect that a speech event is taking place which is unvoiced (/s/ has a very definite pulse interval representation; /h/ is not detected). The constancy of the pulse position tracks will permit the simple automatic recognition of vowels and liquids, but dipthongs will present considerably greater difficulty. These representations need to be carefully evaluated, and compared with other methods, before any attempt is made to build a general recognizer.

In this study, attention has been concentrated on the first derivative waveform, due to the fact that it has been shown to produce higher intelligibility scores (5). However, it is likely that the undifferentiated waveform, and the higher-order derivative waveforms, will permit the detection of significant parameters in simpler ways.

For example, the extraction of pitch is probably most easily done using the undifferentiated waveform. In a hardware implementation of a general recognizer, this would be a simple approach. However, in a digital model, the necessity of examining twice as many data, which must be in time synchronism, would make this a less attractive alternative.

On the other hand, those features or events which have been characterized in the frequency domain as having high energy in the high frequency portion of the spectrum will probably be easier to evaluate using the second or higher derivatives.

4.4 Speech Synthesis

Once the pulse position specifications of speech events have been determined, it will be possible to synthesize speech events by rule. It is felt, from examination of the pulse position charts for the experimental sample, that these rules will have to specify not only the features which are constant over several pitch periods, but also the nature of the period-to-period fluctuations of the pulse positions.

5. CONCLUSIONS AND OUTLOOK

The pulse interval and pulse position representations developed in this paper represent a major departure from the frequency-domain analyses traditionally used in investigations of the properties of the speech waveform.

The precise specification offered by these representations, which are inherently (rather than artificially) digital, is an obvious advantage in the context of transmission and recognition of speech events by automatic devices. In addition, these representations will provide a tool for the better understanding of the processes which are taking place in the production (and perhaps the analysis) of the speech waveform.

The construction of automatic devices based on these representations will, in turn, provide additional insight into the important features of the pulse interval and pulse position specification of speech events.

The application of these techniques to other areas of investigation will follow from the availability of the devices and from an understanding of the significance of the output of these devices.

APPENDIX A. THE DIGITAL PROCESSING SYSTEM

The preparation and analysis of the data for this study have been carried out on computers which are a part of the Service Computing Facility provided by the Digital Computer Laboratory at the University of Illinois, Urbana. The conversion of acoustical signals to digital samples (and vice versa) is done on an IBM 1800 computer. The analysis of the data is done on an IBM System/360 computer, model 75.

The 1800 is a medium-speed processor-controller with 16,384 16-bit words, and a 2 microsecond cycle time. Several special input-output devices are attached to it: (10)

1. An analog-to-digital converter, with the following features: 16 multiplexed input ports; conversion accuracy of 14-, 11-, or 8-bits, plus sign; maximum sampling rate of 18,000, 21,000 or 24,000 samples per second (nominal), depending on conversion accuracy; internal or external synchronization of data conversion.
2. A digital-to-analog converter, with: two output ports, selected by the least significant bit; conversion accuracy of 12 bits, plus sign; crystal controlled conversion rate of 10,000, 20,000, 30,000 or 40,000 samples per second, or free running at channel rate (about 100,000 samples per second).

3. Access to up to five high-speed digital tape drives, which are shared with the 360/75.
4. A direct connection (channel-to-channel adapter) to the 360/75. All programs, input data (e.g., control cards), output data (e.g., printed lines), etc., are received from or sent to a supervisory program (which runs on the 360/75) via this direct connection. The 1800 has no card reader or printer.
5. Process interrupt.
6. Contact operate.
7. Contact sense.
8. A typewriter/console.

In addition there is a high-quality tape recorder/reproducer attached to the system, which can be controlled (start/stop/record) either manually or by the computer through contact operate.

The operating system for the 1800 is called AMØS/1800, and was written by a member of the DCL staff (11)(12). It specifically provides for the real-time nature of the processing, for the existence of the shared tape drives, and for the utilization of the direct connection to the 360/75. It also provides communication paths to data-processing programs running concurrently on the 360/75. Further information on this system is provided in the named reports (10-12).

The 1800 is a hands-on (open shop) machine, and programs for it must be written in 1800 assembler language. (No other higher-level language is available.) However, two supervisory programs are available for use in controlling analog-to-digital and digital-to-analog conversion. ADSUP (12) and DASUP(12) are written with a command structure such that an user need know nothing of the operation of the 1800, except where to connect his wires, and how to operate the typewriter.

External voltages in the range ± 5 volts are converted to internal integers in the range ± 32766. There is a reversal in sign during A/D conversion due to the multiplexor.

The 360/75 is a very large, high-speed, general purpose computer with many peripherals, which is operated as the "service computing" facility of the University (13). The supervisory program for the 1800 (AMØS/360) runs on the 360/75 in the same manner as any other user program. The 360/75 is multiprogrammed, i.e., more than one user is utilizing the central processor and memory at one time, and only a small fraction of the capabilities of the 360/75 are used to supervise the 1800. However, AMØS/360 permits the attaching of a subtask, which can use the full capabilities of the 360/75. In this manner it is possible to process data as soon as it is prepared by the 1800, or, conversely, to prepare data, examine the output via the 1800, and then prepare more data, after this examination.

All processing programs have been written in FORTRAN IV. The only non-standard feature of these programs is the use of a set of special routines to read and write tapes. These are used because the 1800 does not prepare tapes in a format acceptable to FORTRAN. In addition they are faster in operation than the standard FORTRAN input/output routines.

APPENDIX B. COMPUTER PROGRAMS

This appendix contains the computer programs which implement the techniques described in this paper. In most cases a complete working program with all control cards is presented. A few notes are given below on the subroutines which are used. Further information on these subroutines is contained in the current AMØS Users Guide (12).

PIT and PTSYN. The un-named FORTRAN program at the beginning, and the 1800 program SPEAK provide a real-time communications path between the FORTRAN program PIT and the 1800 program PTSYN, using the subroutines MAIN, PT360, WT360, TM360, PT1800, WT1800. Subroutines with entry points of the form TPXXXZ or TPXXX are tape handling routines. The subroutines of the form CAXXX communicate with the statement READ (R, 89, ...) in program PIT. The speech samples are input from tape Ø25858 (data set reference number 25), and the pulse interval table is output to tape NØLBL (dsrn 26). Control of the operation of the programs is from the typewriter on the 1800.

PPL and PPLV. The speech samples are input from tape Ø25858. The pulse position listing is printed.

PPC. This is written as a short main program and two subroutines. The pitch period correction algorithms are all grouped together. The speech samples are input from tape

Ø25858 and the pulse positions are written on the temporary dataset FT18F001. Subroutine PRPPC then prints the asterisks.

PITG. This is a subroutine without a controlling program. It may be used with a slightly modified version of PPC, or with other routines.

```
/*ID PS=2335,DEPT=EE,CODE=DIFEQU,NAME=ATWOOD,MSGLEVEL=(1,1),
/* IOREQ=5000
/*SETUP UNIT=TAPE,ID=(025858,NORING,NL)
/*SETUP UNIT=TAPE,ID=(NOLBL,NL)
// EXEC FORTLKED
      INTEGER RTNC,R,P
      P=6
      SRATE=17400.
      ORATE=20000.
      RTNC=63
      CALL TPOPIZ(25)
      CALL TPOPIZ(26)
    1 CALL WT1800(R)
      IF (R.GE.100) GO TO 2
      CALL PIT(R,P,SRATE,ORATE)
      REWIND R
      CALL PT1800(RTNC)
      GO TO 1
C     TERMINATE THIS TASK
    2 CALL TPCLSZ(25)
      CALL TPCLSZ(26)
      RETURN
      END
      SUBROUTINE PIT(R,P,SRATE,ORATE)
C     ALL LOGIC IS REVERSED DUE TO SIGN REVERSAL BY MULTIPLEXOR
C     ON THE 1800
C     THE SPEECH IS ON DATASET INSET, IN RECORDS OF LENGTH 2048
      INTEGER*2 XY(2048)
C     THE PULSE INTERVAL TABLES ARE OUTPUT
C     TO DATASET OUTSET, 100 INTERVALS AT A TIME
      INTEGER OUTSET,PEST4(100),SKIP
      INTEGER*2 PEST(50)
C     R IS THE READ DATASET NUMBER FOR CONTROL CARDS
C     P IS THE PRINT DATASET NUMBER FOR STATISTICS
      INTEGER COMP,EOF,XX,EVENT,R,P,OPEN
      ASSIGN 101 TO EOF
      OPEN=0
C     SRATE IS THE SAMPLE RATE FOR THE INPUT DATA
C     ORATE IS THE SAMPLE RATE FOR THE OUTPUT DATA
      DELTA=2048./SRATE
      FACTOR=ORATE/SRATE
C     THIS SUBROUTINE PRODUCES OUTPUT FOR THE 1800 PROGRAM PTSYN
C
C             ***** GET DATASET NUMBERS *****
   10 READ (R,89,END=90) INSET,OUTSET,SKIP
   89 FORMAT(2I2,I6)
      IF (INSET.EQ.0) GO TO 10
      EVENT=1
      IFLAG=0
      IF (OUTSET.EQ.OPEN) GO TO 11
      IF (OPEN.EQ.0) GO TO 13
      CALL TPREWZ(OPEN)
      CALL TPWTEZ(OPEN)
   13 OPEN=OUTSET
      IF (SKIP.LE.0) GO TO 11
      DO 12 I=1,SKIP
```

```
   12 CALL TPFSFZ(OPEN)
   11 CONTINUE
C
C              ***** GET GROUP PARAMETERS *****
    4 READ (R,91) NFILES,COMP
   91 FORMAT(2X,6I6)
      IF (NFILES) 6,3,5
C     NEGATIVE NFILES WILL SKIP NFILES EVENTS
    6 NFILES=-NFILES
      DO 7 I=1,NFILES
      EVENT=EVENT+1
    7 CALL TPFSFZ(INSET)
      IFLAG=1
      GO TO 4
C
    5 NFILE=1
      WRITE (P,86) INSET,OPEN,COMP,NFILES
   86 FORMAT(' IN='I2,', OUT='I2,', LEVEL='I6,', NFILES='I6)
      IPEST=1
      IBYTES=0
      TIME=0.
C
C              ***** PROCESS ONE FILE *****
C     INITIALIZATION
    1 LEVEL=1
      INTER=0
C
C              ***** PROCESS ONE RECORD *****
    2 CALL TPGETZ(INSET,XY)
      TIME=TIME+DELTA
      CALL TPCHKZ(INSET,NB,EOF)
      IFLAG=0
      DO 444 IT=1,2048
      XX=XY(IT)
C
C              ***** PRODUCE AN INTERVAL *****
      INTER=INTER+1
      IF(XX.GT.COMP) GO TO 443
      IF(LEVEL.EQ.1) GO TO 444
C
      LEVEL=1
      INVER=INTER*FACTOR
C     EXAMINE INTERVAL
   73 IF (INVER.LT.255) GO TO 84
C     ***** INVER.GE.255 *****
      PEST4(IPEST)=255
      IF (IPEST.LT.100) GO TO 71
      DO 72 M=1,50
      I=2*M
   72 PEST(M)=256*PEST4(I-1)+PEST4(I)
      CALL TPPUTZ(OUTSET,PEST,100)
      IBYTES=IBYTES+100
      CALL TPWTEZ(OUTSET)
      IPEST=0
   71 IPEST=IPEST+1
      INVER=INVER-254
```

```
      GO TO 73
   84 IF (INVER.LT.1) INVER=1
C     ***** INVER.LT.255 *****
      PEST4(IPEST)=INVER
      IF (IPEST.LT.100) GO TO 81
      DO 82 M=1,50
      I=2*M
   82 PEST(M)=256*PEST4(I-1)+PEST4(I)
      CALL TPPUTZ(OUTSET,PEST,100)
      IBYTES=IBYTES+100
      CALL TPWTEZ(OUTSET)
      IPEST=0
   81 IPEST=IPEST+1
C     END OF EXAMINATION
      INTER=0
      GO TO 444
  443 LEVEL=0
C
C     END OF ONE-RECORD LOOP
  444 CONTINUE
      GO TO 2
C
C     END OF FILE EXIT
  101 IF(IFLAG.EQ.1) GO TO 3
      IFLAG=1
      EVENT=EVENT+1
      DO 61 J=1,40
      PEST4(IPEST)=255
      IF (IPEST.LT.100) GO TO 61
      DO 62 M=1,50
      I=2*M
   62 PEST(M)=256*PEST4(I-1)+PEST4(I)
      CALL TPPUTZ(OUTSET,PEST,100)
      IBYTES=IBYTES+100
      CALL TPWTEZ(OUTSET)
      IPEST=0
   61 IPEST=IPEST+1
      TIME=TIME+.508
      IF (NFILES.EQ.NFILE) GO TO 41
      NFILE=NFILE+1
      GO TO 1
   41 CONTINUE
      IBSEC=IBYTES/TIME
      WRITE (P,42) IBYTES,TIME,IBSEC
   42 FORMAT(' IBYTES='I10,', TIME='F7.2,' SEC, RATE='
     *I10,' BYTES PER SECOND')
      GO TO 4
C     TWO END OF FILES IN A ROW
C
    3 CALL TPREWZ(INSET)
      CALL TPWTEZ(INSET)
      PEST4(IPEST)=0
      DO 83 M=1,50
      I=2*M
   83 PEST(M)=256*PEST4(I-1)+PEST4(I)
      CALL TPPUTZ(OUTSET,PEST,100)
```

```
      CALL TPWTEZ(OUTSET)
      GO TO 10
C
C     END OF FILE ON SYSIN
   90 CONTINUE
      CALL TPEOFZ(OUTSET)
      CALL TPEOFZ(OUTSET)
      CALL TPREWZ(OUTSET)
      CALL TPWTEZ(OUTSET)
      RETURN
      END
// EXEC ASM1800
*CTL 029
*ASM SPEAK--AN 1800 CONTROL PROGRAM
                    *      SPEAK LISTENS TO THE 1816 TYPEWRITER AND SENDS
                    *      HEARS TO THE 75 ON DATASET FT10F001
                    *      INPUT OF THE MESSAGE 'R,END' RESULTS IN THE PO
                    *      THE 75
                    *      INPUT OF THE MESSAGE 'R,QUIT' CAUSES ALL PROCE
                    *      TO BE TERMINATED
                           ENT      SPEAK
                    SPEAK  DC       *-*
                           CALL     MAP
                           CALL     MAIN        FIRE UP THE 75
                           ZAC
                           STO      RTNC
                    AKWAK  CALL     CWRIT       WE'RE HERE
                           DC       MSG1              *
                    LISEN  CALL     REPLY       LOOK FOR OPERATOR RESPONSE
                           DC       INPUT             *
                           DC       ECW               *
                           CALL     WAIT              *
                           DC       ECW               *
                           LD       INPUT       IS THERE A MESSAGE?
                           A        ONE
                           SRA      1           YES.
                           STO      INPUT       BYTES/2=WORDS
                           LD       INPUT+1     GET FIRST TWO CHAR
                           CMP   L  END         IS IT END?
                           B        QUIT        NO,
                           B        QUIT        NO,
                           B        RUN         GO POST 360
                    QUIT   CMP   L  QU          IS IT QUIT?
                           B        PUSH        NO,
                           B        PUSH        NO,
                           CALL     PT360       YES, TERMINATE SUBTASK
                           DC       TERM
                           CALL     TM360
                           DC       CCODE
                           B     I  SPEAK       GO AWAY NICELY
                    PUSH   CALL     CAPUT       WRITE MSG TO ADAPTER
                           DC       FILEN       *
                           DC       INPUT       *
                           CALL     CACHK       *
                           CALL     CWRIT
                           DC       MSG2
```

```
             B         LISEN        LOOK FOR MORE
       RUN   CALL      CAEOF
             DC        FILEN
             CALL      CAREW
             DC        FILEN
             CALL      PT360
             DC        DSN
             CALL      CWRIT
             DC        MSG3
             CALL      WT360
             DC        RTNC
             ZAC
             STO       RTNC
             CALL      CAREW
             DC        FILEN
             CALL      PTSYN
             B         AKWAK
       MSG1  DC        5
             EBC       .SPEAK.
       MSG2  DC        1
             EBC       .?.
       MSG3  DC        10
             EBC       .360 POSTED.
       ECW   DC        0
       INPUT BSS       40
       FILEN EQU       10
       DSN   DC        FILEN
       RTNC  DC        0
       ONE   DC        1
       QU    EBC       .QU.
       END   EBC       .EN.
       TERM  DC        101
       CCODE BSS       2
             END
*CTL 029
*ASM PTSYN--A PULSE TRAIN SYNTHESIZER
             ENT       PTSYN
       PTSYN DC        *-*          ENTRY POINT
             CALL      CWRIT        SAY WE ARE IN CONTROL
             DC        MSG1              *
       LABEL CALL      CWRIT        TAPE LABEL=?
             DC        MSG2              *
             CALL      REPLY        LOOK FOR REPLY
             DC        INPUT             *
             DC        RECW              *
             CALL      WAIT              *
             DC        RECW              *
             LD    L   INPUT        CHECK LENGTH
             CMP   L   SIX               *
             B         LABEL        L .NE. 6
             B         LABEL        L .NE. 6
             LDD   L   INPUT+1      STORE LABEL
             STD   L   TCT          IN TCT
             LD    L   INPUT+3           *
             STO   L   TCT+2             *
             CALL      TPOPI        OPEN TAPE
```

```
      DC      TCT               *
      HDNG    THE SYSTHESIZER
*****************************************************
*     INITIALIZATION                                *
*****************************************************
FPEST LDD  L  PESTI        SET PEST POINTERS
      STD  L  PESTP             *
      CALL    TPGET        FILL FIRST PEST
      DC      TCT               *
      DC      PEST1
*     THE SPECIFICATION OF THE PULSE TRAIN IS IN
*     PEST -- THE PULSE EVENT SPECIFICATION TABLE
*     PEST IS OF LENGTH PESTL
STSYN EQU     *
      CALL    CWRIT        READY.
      DC      MSG8              *
      CALL    REPLY        LOOK FOR ANY MESSAGE
      DC      INPUT             *
      DC      RECW              *
      CALL    WAIT              *
      DC      RECW              *
      LDD  L  DBUFI        SET DABUF BASE ADDRESSES
      STO     BUFA1             *
      STO     BUFA2             *
      STO     BUFA3             *
      STO     BUFA4             *
      XCH                  SET CURRENT DABUF POINTER
      STD  L  DBUFP             *
      ZAC                  SET FIRST BUFFER
      STO  L  FIRST        POINTER=0
*     XR2  =  CURRENT DABUF POINTER
      LDX  L2 -DASIZ
      EJCT
*****************************************************
*     THE DAO SUPERVISORY LOOP                      *
*****************************************************
*     THE DAO SUPERVISORY LOOP KEEPS AHEAD OF DAO
*     BY FILLING BUF1 AND BUF2 ALTERNATELY, BUT
*     PAUSING AT THE END OF EACH BUFFER UNTIL THE
*     OTHER IS EMPTY.
*     POINT TO THE NEXT PEST AND BEGIN REFILLING
*     THE OLD ONE.
RPEST LDX  I1 PESTP        SET PEST POINTER
      MDX   1 PESTL+1           *
      STX  L1 LCTAD             *
      LDD  L  PESTP        FLIP POINTERS
      XCH                       *
      STD  L  PESTP             *
      STO     PESTA        POINT TO OLD PEST
      CALL    TPCHK        CHECK LAST I/O COMPLETE
      DC      TCT               *
      DC      EOFXT             *
      BZ      CKERR        I/O ERROR
      CALL    TPGET        REFILL IT
      DC      TCT               *
PESTA DC      *-*               *
```

```
*       XR3   =  CURRENT PEST POINTER
        LDX    3 -PESTL
LGETC   ZAQ                   ZERO THE A-Q
        LD    L3 *-*          GET THE NEXT TWO COUNTS
LCTAD   EQU      *-1
        RTE      8            SEPARATE THEM
        SLA      8                 *
        RTE      8                 *
        STD   L  CNTR1        PUT THEM AWAY
        S     L  CCLV         SET '255' POINTER
        STO   L  CCLV1
        XCH
        S     L  CCLV
        STO   L  CCLV2
        MDM      CNTR1,0      CHECK COUNTER 1
        B        LPUL1        OK, GO ON
        B        ENSYN        IF COUNT IS ZERO, GO AWAY
LPUL1   LDD   L  ZERO         SET A-Q
        LDX   I1 CNTR1        LOAD COUNT
        MDX    1 -1           DECREMENT
        B        LSTZE        GO ON
        XCH                   SKIP, COUNT WAS ONE
        B        LF1
LSTZE   STO   L2 *-*          STORE ZERO DEFLECTION
BUFA2   EQU      *-1
        MDX    2 1            END OF D/A BUFFER?
        B        LZ2          NO, CHECK COUNT
        BSI      DWAIT        YES, WAIT ON INTERRUPT
LZ2     MDX    1 -1           END OF COUNT?
        B        LSTZE        NO, GO BACK
        XCH                   FLIP A-Q
        MDM      CCLV1,0      WAS COUNT=255?
        B        LF1          NO, GO ON
        B        CKCN2        YES, SKIP DOWN
LF1     STO   L2 *-*          STORE FULLSCALE DEFLECTION
BUFA1   EQU      *-1
        MDX    2 1            END OF D/A BUFFER?
        B        CKCN2        NO, GO ON
        BSI      DWAIT        YES, WAIT ON INTERRUPT
CKCN2   MDM      CNTR2,0      YES, CHECK COUNTER 2
        B        LZ3          OK, GO ON
        B        ENSYN        IF COUNT IS ZERO, GO AWAY
LZ3     XCH                   FLIP A-Q
        LDX   I1 CNTR2        LOAD COUNT
        MDX    1 -1           DECREMENT
        B        LSTZR        GO ON
        XCH                   SKIP, COUNT WAS ONE
        B        LF2
LSTZR   STO   L2 *-*          STORE ZERO DEFLECTION
BUFA4   EQU      *-1
        MDX    2 1            END OF D/A BUFFER?
        B        LZ4          NO, CHECK COUNT
        BSI      DWAIT        YES, WAIT ON INTERRUPT
LZ4     MDX    1 -1           END OF COUNT?
        B        LSTZR        NO, GO BACK
        XCH                   FLIP A-Q
```

```
      MDM      CCLV2,0     WAS COUNT=255?
      B        LF2         NO, GO ON
      B        CKNEW       YES,SKIP DOWN
LF2   STO  L2 *-*          STORE FULLSCALE DEFLECTION
BUFA3 EQU      *-1
      MDX    2 1           END OF D/A BUFFER?
      B        CKNEW       NO, GO ON
      BSI      DWAIT       YES, WAIT ON INTERRUPT
CKNEW MDX    3 1           THEN, NEED NEW COUNT
      B        LGETC       MORE YET IN TABLE
      B        RPEST       NEED NEW TABLE POINTERS
      EJCT
***************************************************
*     THE INTERRUPT WAIT ROUTINE                  *
***************************************************
DWAIT DC       *-*         ENTRY POINT, RETURN ADR
      STD  L   AQ          SAVE A-Q
      MDM      FIRST,0     IS THIS THE FIRST TIME?
      B        CKDAO       NO,
      STX  L0 FIRST        YES, SET 'FIRST' NON-ZERO
*     THIS IS THE FIRST PASS THRU DWAIT SINCE
*     STARTING (OR RESTARTING) SYNTHESIS, AND
*     THE FIRST TABLE IS NOW FULL.
*     WE MUST NOW START THE DAO, AND THEN GO BACK
*     TO THE DAO SUPERVISORY LOOP, BYPASSING THE
*     WAIT ON DAECB.
STDAC EQU      *
      CALL     COPUT       SET D/A RATE
      DC       DRATE           *
      STX    1 XR1         IOSUP CLOBBERS
      STX    3 XR3         EVERYTHING
      CALL     IOSUP       BEGIN D/A OUTPUT
      DC       SDAO            *
      DC       12              *
      DC       DAECB           *
      LDX  L1 *-*          RESTORE EVERYTHING
XR1   EQU      *-1
      LDX  L3 *-*              *
XR3   EQU      *-1
      B        DASET       GO RESET DABUF POINTER
*
*     THIS PATH IS USED FOR ALL PASSES AFTER
*     THE FIRST ONE.
CKDAO MDM      DAECB,0     HAS TABLE-END OCCURRED?
      B        AAAGH       YES, WE BLEW IT
      CALL     WAIT        NO, WAIT ON IT
      DC       DAECB           *
      STX  L2  DAECB       ZERO THE ECB
DASET LDD  L   DBUFP       RESET DABUF POINTER
      STO      BUFA1           *
      STO      BUFA2           *
      STO      BUFA3           *
      STO      BUFA4           *
      XCH                      *
      STD      DBUFP           *
      LDD  L   AQ          RESTORE A-Q
```

```
      LDX  L2 -DASIZ     RESET DABUF POINTER
      B    I  DWAIT      RETURN TO SUPERVISORY LOOP
      HDNG    ERROR HANDLING
***************************************************
*     ERRORS AND OTHER STRANGE THINGS             *
***************************************************
AAAGH CALL    CWRIT      OVERRUN
      DC      MSG3           *
      B       ENSYN      GO AWAY
CKERR CALL    CWRIT      I/O ERROR
      DC      MSG7           *
      B       ENSYN      GO AWAY
EOFXT CALL    CWRIT      END OF FILE
      DC      MSG4           *
      LD   L  MINUS      SET 'FIRST' NEGATIVE
      STO  L  FIRST      TO SHOW 'EOF'
ENSYN CALL    CWRIT      END OF SYNTHESIS
      DC      MSG5           *
      CALL    IOSUP      BLAST DAO
      DC      BDAO           *
      DC      12             *
      DC      DAECB          *
LOOK  CALL    REPLY      LOOK FOR OPERATOR RESPONSE
      DC      INPUT          *
      DC      RECW           *
      CALL    WAIT           *
      DC      RECW           *
***** PERMITTED RESPONSES *****
*     GO   =  GO ON          *
*     QUIT =  GO AWAY        *
*     TAPE =  START AGAIN    *
*******************************
      MDM     INPUT,0    IS THERE A MESSAGE?
      B       CHECK      YES, LOOK AT IT
      CALL    DUMP       NO, GO AWAY MAD
      DC      PTSYN
      DC      16383
      B       QUIT
CHECK LD      INPUT+1    LOOK AT MESSAGE
C1    CMP     CGO        IS IT 'GO'?
      B       C2         NO,
      B       C2         NO,
GOON  LD   L  FIRST      RESUME SYNTHESIS...
      BN      FPEST      AT FPEST, IF EOF
      B    L  STSYN      AT STSYN, OTHERWISE
C2    CMP     CTAPE      IS IT 'TAPE'?
      B       C3         NO,
      B       C3         NO,
TAPE  CALL    TPCLS      YES, CLOSE TAPE
      DC      TCT            *
      B    L  LABEL      LOOK FOR NEW LABEL
C3    CMP     CQUIT      IS IT 'QUIT'?
      B       C4         NO,
      B       C4         NO,
QUIT  CALL    TPCLS      YES,GO AWAY
      DC      TCT
```

```
      B     I  PTSYN       GO BACK TO AMOS
C4    EQU      *           WE DONT UNDERSTAND
      CALL     CWRIT       HUH?
      DC       MSG6           *
      B        LOOK        TRY AGAIN
      HDNG     POINTERS,CONSTANTS AND EQUATES
*     EQUATES
DASIZ EQU      2000        D/A BUFFER SIZE
PESTL EQU      50          LENGTH OF PEST
*     POINTERS AND SAVE AREAS
AQ    BSS   E  2           AQ SAVE AREA
CNTR1 BSS   E  1           COUNT 1
CNTR2 BSS      1           COUNT 2
PESTP BSS   E  2           CURRENT PEST POINTER
DBUFP BSS   E  2           CURRENT DABUF POINTERS
CCLV1 BSS      1
CCLV2 BSS      1
DAECB BSS      1
FIRST BSS      1           FIRST PASS POINTER
*     CONSOLE INPUT AREA
RECW  BSS   E  1           FORCE ODD BOUNDARY
INPUT BSS      21          FOR INPUT BUFFER
*     CONSTANTS
      BSS   E  0           FORCE EVEN BOUNDARY
ZERO  DC       0           ZERO DEFL, CH 1
FULL  DC       /3FF8       FULL SCALE DEFL, CH 1
PESTI DC       PEST1       INITIAL PEST POINTERS
      DC       PEST2          *
DBUFI DC       BUF1+DASIZ  INITIAL DABUF POINTERS
      DC       BUF2+DASIZ     *
SDAO  DC       DABUF       IOCC TO START DAO
      DC       /65C0
BDAO  DC       0           IOCC TO STOP DAO
      DC       /6420
SIX   DC       6
CCLV  DC       255
MINUS DC       -1
DRATE DC       1024        20 KS/S
CGO   EBC      .GO.
CQUIT EBC      .QUIT.
CTAPE EBC      .TAPE.
      HDNG     TABLES AND STUFF
*     DIGITAL OUTPUT BUFFERS
      DC       *-1         CHAINING POINTER
DABUF DC       /8000+DASIZ+1  SCAN CONTROL+BUFR SIZE
      DC       /7E            DEVICE ADDRESS
BUF1  BSS      DASIZ          BUFFER
      DC       *-1
      DC       /8000+DASIZ+1
      DC       /7E
BUF2  BSS      DASIZ
      DC       DABUF-1
*     PULSE EVENT SPECIFICATION TABLES
PEST1 DC       PESTL
      BSS      PESTL
PEST2 DC       PESTL
```

```
                BSS      PESTL
        TCT     BSS   E  3          TAPE CONTROL TABLE
        *       MESSAGES
        MSG1    DC       13
                EBC      .PROGRAM PTSYN.
        MSG2    DC       12
                EBC      .TAPE LABEL=?.
        MSG3    DC       7
                EBC      .OVERRUN.
        MSG4    DC       11
                EBC      .END OF FILE.
        MSG5    DC       16
                EBC      .END OF SYNTHESIS.
        MSG6    DC       4
                EBC      .HUH?.
        MSG7    DC       9
                EBC      .I/O ERROR.
        MSG8    DC       6
                EBC      .READY..
                END      PTSYN
// EXEC AMOS,MINUTES=60,REGION=(45K,116K)
//FT06F001 DD SYSOUT=A
//FT10F001 DD UNIT=2314,DSN=&&SYSUT1,SPACE=(80,(25,10)),
// DCB=(LRECL=80,BLKSIZE=80,RECFM=F)
//FT25F001 DD UNIT=TAPE,LABEL=(,BLP),VOL=SER=025858
//FT26F001 DD UNIT=TAPE,LABEL=(,BLP),VOL=SER=NOLBL
//AMOSIN DD *
$$EXC,*
/*
```

```
/*ID PS=2335,DEPT=EE,NAME=ATWOOD,CODE=DIFEQU,360=1.,LINES=6000,
/* IOREQ=1000,MSGLEVEL=(1,1)
/*SETUP UNIT=TAPE,ID=(025858,NORING,NL)
// EXEC FORTLDGO,TIME.GO=4
//FORT.SYSIN DD *
C     PROGRAM PPL
      INTEGER*2 XY(2048)
      INTEGER EVENT,COMP,STOP,EOF,XX,DUR(20),DIFF(20)
      ASSIGN 101 TO EOF
   10 READ (5,89,END=90) INSET
   89 FORMAT(I2)
      IF (INSET.EQ.0) GO TO 10
      CALL TPOPIZ(INSET)
      EVENT=1
      IFLAG=0
    4 READ (5,91) NFILES,COMP,STOP
   91 FORMAT(2X,3I6)
      IF (NFILES.LE.0) GO TO 3
      NFILE=1
    1 LEVEL=1
      NUM=1
      IOUT=1
      NGLOT=0
      NPLS=0
      INTER=0
      WRITE (6,77) EVENT,COMP,STOP
   77 FORMAT('1EVENT 'I3,', COMP= 'I6,', STOP='I4)
      WRITE (6,78)
   78 FORMAT(' P PER PTIME CNT'20X,'PULSE POSITIONS'
     1,33X,'PULSE INTERVALS')
    2 CALL TPGETZ(INSET,XY)
      CALL TPCHKZ(INSET,NB,EOF)
      IFLAG=0
      DO 444 IT=1,2048
      INTER=INTER+1
      XX=XY(IT)
      IF (XX.LT.COMP) GO TO 443
      IF (LEVEL.EQ.1) GO TO 444
      LEVEL=1
C     EXAMINE INTERVAL
      NUM=NUM+1
      DIFF(IOUT)=INTER
      NPLS=NPLS+INTER
      NGLOT=NGLOT+INTER
      DUR(IOUT)=NGLOT
      IF (INTER.LT.STOP) GO TO 60
      IF (IOUT.GT.10) GO TO 63
      WRITE (6,66) NGLOT,NPLS,NUM,(DUR(M),DIFF(M),M=1,IOUT)
      GO TO 64
   63 WRITE (6,65) NGLOT,NPLS,NUM,(DUR(M),M=1,10),
     1 (DIFF(M),M=1,10)
      WRITE (6,67) (DUR(M),DIFF(M),M=11,IOUT)
      GO TO 64
   60 IF (IOUT.GE.20) GO TO 62
      IOUT=IOUT+1
      GO TO 69
```

```
   62 WRITE (6,65) NGLOT,NPLS,NUM,(DUR(M),M=1,10),
     1 (DIFF(M),M=1,10)
      WRITE (6,68) (DUR(M),M=11,20),(DIFF(M),M=11,20)
   64 NGLOT=0
      IOUT=1
   69 CONTINUE
C     END OF EXAMINATION
      INTER=0
      GO TO 444
  443 LEVEL=0
  444 CONTINUE
      GO TO 2
C     END OF FILE EXIT
  101 IF (IFLAG.EQ.1) GO TO 3
      IFLAG=1
      EVENT=EVENT+1
      IF (NFILES.EQ.NFILE) GO TO 4
      NFILE=NFILE+1
      GO TO 1
C     TWO END OF FILES IN A ROW
    3 CALL TPCLSZ(INSET)
      GO TO 10
C     END OF FILE ON SYSIN
   90 STOP
   66 FORMAT(1X,I5,I6,I5,3X,I5,T74,I5,T26,I5,T79,I5,T31,I5,T84,
     1I5,T36,I5,T89,I5,T41,I5,T94,I5,T46,I5,T99,I5,T51,I5,T104,
     2I5,T56,I5,T109,I5,T61,I5,T114,I5,T66,I5,T119,I5)
   67 FORMAT(19X,'*'I5,T74,I5,T26,I5,T79,I5,T31,I5,T84,
     1I5,T36,I5,T89,I5,T41,I5,T94,I5,T46,I5,T99,I5,T51,I5,T104,
     2I5,T56,I5,T109,I5,T61,I5,T114,I5,T66,I5,T119,I5)
   68 FORMAT(19X,'#'10I5,3X,10I5)
   65 FORMAT(1X,I5,I6,I5,3X,10I5,3X,10I5)
      END
/*
//GO.FT25F001 DD UNIT=TAPE,LABEL=(,NL),VOL=(,RETAIN,SER=025858)
//GO.SYSIN DD *
25
      20 -5000    42
      20 -5000    89
      20 -5000   110
      20 -5000    51
      20 -5000    66

/*
```

```
/*ID PS=2335,DEPT=EE,NAME=ATWOOD,CODE=DIFEQU,360=2.,LINES=10000,
/* IOREQ=2000,MSGLEVEL=(1,1)
/*SETUP UNIT=TAPE,ID=(025858,NORING,NL)
// EXEC FORTLDGO,TIME.GO=5
//FORT.SYSIN DD *
C      PROGRAM PPLV
C      ALL LOGIC IS REVERSED DUE TO SIGN REVERSAL BY MULTIPLEXOR
C      ON 1800
C      THE BEGINNING OF A GLOTTAL FRAME IS DEFINED BY THE FIRST
C      TRANSITION THROUGH COMPG, WITH NEGATIVE SLOPE, WHICH
C      CORRESPONDS TO THE LEADING EDGE OF THE FRAME.
       INTEGER*2 XY(2048)
       INTEGER EVENT,STOP,EOF,XX,DUR(20),DIFF(20),COMPR
       INTEGER COMPG,GLOTP,RECP,R,P,CK1,CK2
       ASSIGN 101 TO EOF
       R=5
       P=8
C
C              ***** GET DATASET NUMBER AND OPEN TAPE *****
   10 READ (R,89,END=90) INSET
   89 FORMAT(I2)
       IF (INSET.EQ.0) GO TO 10
       CALL TPOPIZ(INSET)
       EVENT=1
       IFLAG=0
C
C              ***** GET GROUP PARAMETERS              *****
    4 READ (R,91) NFILES,STOP,GLOTP,RECP,MINLG,MINLR
   91 FORMAT(2X,6I6)
       IF (NFILES) 6,3,5
C      NEGATIVE NFILES WILL SKIP NFILES EVENTS
    6 NFILES=-NFILES
       DO 7 I=1,NFILES
       EVENT=EVENT+1
    7 CALL TPFSFZ(INSET)
       IFLAG=1
       GO TO 4
C
    5 NFILE=1
       CK1=(5*STOP)/2
C
C              ***** PROCESS ONE FILE                  *****
C      INITIALIZATION
    1 LEVEL=1
       IOUT=1
       LEVELG=1
       NGLOT=0
       NPLS=0
       INTER=0
       INTERG=0
       CK2=0
       LASTP=-5000
       COMPR=(RECP*LASTP)/100
       IF (COMPR.GT.MINLR) COMPR=MINLR
       COMPG=(GLOTP*LASTP)/100
       IF (COMPG.GT.MINLG) COMPG=MINLG
```

```
      LASTP=0
C     PAGE HEADERS
      WRITE (P,76) EVENT,STOP,GLOTP,RECP
   76 FORMAT('1EVENT 'I3,', STOP= 'I4,', GLOTP='I3,', RECP='I3)
      WRITE (P,78)
   78 FORMAT(' PTIME COMPG P PER'I9X,'PULSE POSITIONS'
     1,33X,'PULSE INTERVALS')
C
C              ***** PROCESS ONE RECORD                ***** 
    2 CALL TPGETZ(INSET,XY)
      CALL TPCHKZ(INSET,NB,EOF)
      IFLAG=0
      DO 445 IT=1,2048
      XX=XY(IT)
C     KEEP TRACK OF MAX VALUE
      IF(XX.LT.LASTP) LASTP=XX
      CK2=CK2+1
C     RESET COMPR AND COMPG IF NECESSARY
      IF (CK2.LT.CK1) GO TO 8
      COMPR=(RECP*LASTP)/100
      IF (COMPR.GT.MINLR) COMPR=MINLR
      COMPG=(GLOTP*LASTP)/100
      IF (COMPG.GT.MINLG) COMPG=MINLG
    8 CONTINUE
C
C              ***** TEST FOR A RECOGNITION INTERVAL *****
      INTER=INTER+1
      IF (XX.GT.COMPR) GO TO 443
      IF (LEVEL.EQ.1) GO TO 444
C
      LEVEL=1
C     EXAMINE INTERVAL
      DIFF(IOUT)=INTER
      NPLS=NPLS+INTER
      NGLOT=NGLOT+INTER
      DUR(IOUT)=NGLOT
      IOUT=IOUT+1
C
C     END OF EXAMINATION
      INTER=0
      GO TO 444
  443 LEVEL=0
  444 CONTINUE
C
C                ***** TEST FOR A GLOTTAL INTERVAL *****
      INTERG=INTERG+1
      IF (XX.GT.COMPG) GO TO 423
      IF (LEVELG.EQ.1) GO TO 424
      IF (INTERG.LT.STOP) GO TO 425
C
C     WE NEED TO PRINT
      IF (IOUT.LE.1) GO TO 425
      IOUT=IOUT-1
      IF (IOUT.GT.10) GO TO 63
      WRITE (P,66) NPLS,COMPG,NGLOT,(DUR(M),DIFF(M),M=1,IOUT)
      GO TO 64
```

```
   63 WRITE (P,65) NPLS,COMPG,NGLOT,(DUR(M),M=1,10),
     1 (DIFF(M),M=1,10)
      WRITE (P,67) (DUR(M),DIFF(M),M=11,IOUT)
   64 NGLOT=0
      IOUT=1
      COMPR=(RECP*LASTP)/100
      IF (COMPR.GT.MINLR) COMPR=MINLR
      COMPG=(GLOTP*LASTP)/100
      IF (COMPG.GT.MINLG) COMPG=MINLG
      LASTP=0
      CK2=0
  425 INTERG=0
      GO TO 424
  423 LEVELG=0
  424 CONTINUE
C
C     CHECK IF WE WENT BY 20 PULSES
      IF (IOUT.LE.20) GO TO 60
C     YES, SO PRINT THEM
      WRITE (P,69) NPLS,COMPG,(DUR(M),M=1,10),
     1 (DIFF(M),M=1,10)
      WRITE (P,68) (DUR(M),M=11,20),(DIFF(M),M=11,20)
C
      IOUT=1
      COMPR=(RECP*LASTP)/100
      IF (COMPR.GT.MINLR) COMPR=MINLR
      COMPG=(GLOTP*LASTP)/100
      IF (COMPG.GT.MINLG) COMPG=MINLG
      LASTP=0
      CK2=0
   60 CONTINUE
C
C     END OF ONE-RECORD LOOP
  445 CONTINUE
      GO TO 2
C
C     END OF FILE EXIT
  101 IF (IFLAG.EQ.1) GO TO 3
      IFLAG=1
C     PRINT THE LAST FEW PULSES
      IF (IOUT.EQ.1) GO TO 103
      IOUT=IOUT-1
      IF (IOUT.GT.10) GO TO 104
      WRITE (P,66) NPLS,COMPG,COMPR,(DUR(M),DIFF(M),M=1,IOUT)
      GO TO 103
  104 WRITE (P,65) NPLS,COMPG,COMPR,(DUR(M),M=1,10),
     1 (DIFF(M),M=1,10)
      WRITE (P,67) (DUR(M),DIFF(M),M=11,IOUT)
  103 EVENT=EVENT+1
      IF (NFILES.EQ.NFILE) GO TO 4
      NFILE=NFILE+1
      GO TO 1
C
C     TWO END OF FILES IN A ROW
    3 CALL TPCLSZ(INSET)
      GO TO 10
```

```
C
C     END OF FILE ON SYSIN
   90 STOP
C
C
   66 FORMAT(1X,I5,I6,I6,2X,I5,T74,I5,T26,I5,T79,I5,T31,I5,T84,
     1I5,T36,I5,T89,I5,T41,I5,T94,I5,T46,I5,T99,I5,T51,I5,T104,
     2I5,T56,I5,T109,I5,T61,I5,T114,I5,T66,I5,T119,I5)
   67 FORMAT(19X,'*'I5,T74,I5,T26,I5,T79,I5,T31,I5,T84,
     1I5,T36,I5,T89,I5,T41,I5,T94,I5,T46,I5,T99,I5,T51,I5,T104,
     2I5,T56,I5,T109,I5,T61,I5,T114,I5,T66,I5,T119,I5)
   68 FORMAT(19X,'#'10I5,3X,10I5)
   65 FORMAT(1X,I5,I6,I6,2X,10I5,3X,10I5)
   69 FORMAT(1X,I5,I6,6X,2X,10I5,3X,10I5)
   75 FORMAT(1X,8I7)
C
      END
/*
//GO.FT25F001 DD UNIT=TAPE,LABEL=(,NL),VOL=(,RETAIN,SER=025858)
//GO.SYSIN DD *
25
      20    42     70     20 -3000 -3000
      20    89     70     20 -3000 -3000
      20   110     70     20 -3000 -3000
      20    51     70     20 -3000 -3000
      20    66     70     20 -3000 -3000

/*
```

```
/*ID PS=2335,DEPT=EE,CODE=DIFEQU,NAME=ATWOOD,360=5.,LINES=10000,
/* IOREQ=5000,MSGLEVEL=(1,1),LINECT=66
/*SETUP UNIT=TAPE,ID=(025858,NORING,NL)
// EXEC FORTLDGO,TIME.GO=5
//FORT.SYSIN DD *
C     OPEN THE TAPE INPUT DATASET
      CALL TPOPIZ(25)
C     CALL THE PULSE POSITION CHART SUBROUTINE
      CALL PPC(5,6,18)
C     CLOSE THE TAPE INPUT DATASET
      CALL TPCLSZ(25)
      STOP
      END
      SUBROUTINE PPC(R,P,S)
C     ALL LOGIC IS REVERSED DUE TO SIGN REVERSAL BY MULTIPLEXOR
C     ON 1800
C     THE BEGINNING OF A GLOTTAL FRAME IS DEFINED BY THE FIRST
C     TRANSITION THROUGH COMPG, WITH NEGATIVE SLOPE, WHICH
C     CORRESPONDS TO THE LEADING EDGE OF THE FRAME.
      INTEGER*2 XY(2048)
      INTEGER EVENT,STOP,EOF,XX,DUR(30),COMPR
      NP=30
C     NP AND THE DIMENSION OF DUR MUST BE THE SAME
C     FORMAT STATMENT 63 IS ALSO AFFECTED BY THE SIZE OF NP
      INTEGER COMPG,GLOTP,RECP,R,P,S,VOICP
      INTEGER CK1,CK2,CKNP
      ASSIGN 101 TO EOF
C     R IS THE READ DATASET
C     P IS THE PRINT DATASET
C     S IS THE STORAGE DATASET
C     IT IS ASSUMED THAT THE FILES POINTED TO BY THE VARIABLE
C     'INSET' ARE OPEN AND AT LOAD POINT
C     THEY WILL BE LEFT IN THAT CONDITION
C
C             ***** GET DATASET NUMBER AND OPEN TAPE *****
   10 READ (R,89,END=90) INSET
C     INSET IS THE INPUT DATASET NUMBER
   89 FORMAT(I2)
      IF (INSET.EQ.0) GO TO 10
      EVENT=1
      IFLAG=0
C
C             ***** GET GROUP PARAMETERS             *****
    4 READ (R,91) NFILES,STOP,GLOTP,RECP,MINLG,MINLR,VOICP
C     NFILES IS THE NUMBER OF FILES TO BE PROCESSED
C     NFILES=0 STOPS PROCESSING ON THIS INSET
C     STOP IS THE STOP NUMBER
C     GLOTP IS THE PERCENTAGE OF CURRENT PEAK VALUE USED
C     FOR PITCH EXTRACTION
C     RECP IS THE PERCENTAGE FOR RECOGNITION
C     MINLG IS THE MINIMUM THRESHOLD LEVEL FOR PITCH EXTRACTION
C     MINLR IS THE MINIMUM THRESHOLD LEVEL FOR RECOGNITION
C     VOICP IS THE VOICING CRITERION IN PERCENT
   91 FORMAT(2X,7I6)
      IF (NFILES) 6,3,5
C     NEGATIVE NFILES WILL SKIP NFILES EVENTS
```

```
    6 NFILES=-NFILES
      DO 7 I=1,NFILES
      EVENT=EVENT+1
    7 CALL TPFSFZ(INSET)
      IFLAG=1
      GO TO 4
C
    5 NFILE=1
C     AUXILLIARY STOP NUMBER
      CK1=(5*STOP)/2
C
C             ***** PROCESS ONE FILE                          *****
C     INITIALIZATION
    1 LEVEL=1
      IOUT=1
      LEVELG=1
      LPER=1
      NGLOT=0
      INTER=0
      INTERG=0
      CK2=0
      CKNP=0
      LASTP=-5000
C     RECOGNITION THRESHOLD
      COMPR=(RECP*LASTP)/100
      IF (COMPR.GT.MINLR) COMPR=MINLR
C     PITCH EXTRACTION THRESHOLD
      COMPG=(GLOTP*LASTP)/100
      IF (COMPG.GT.MINLG) COMPG=MINLG
      LASTP=0
C     PAGE HEADER
      WRITE (P,76) EVENT,STOP,GLOTP,RECP,VOICP
   76 FORMAT('1EVENT 'I3,', STOP= 'I4,', GLOTP='I3,', RECP='I3,
     *', VOICP='I3)
C
C             ***** PROCESS ONE RECORD                        *****
    2 CALL TPGETZ(INSET,XY)
      CALL TPCHKZ(INSET,NB,EOF)
      IFLAG=0
      DO 445 IT=1,2048
      XX=XY(IT)
C     KEEP TRACK OF MAX VALUE
      IF(XX.LT.LASTP) LASTP=XX
      CK2=CK2+1
C     RESET COMPR AND COMPG IF NECESSARY
      IF (CK2.LT.CK1) GO TO 8
      COMPR=(RECP*LASTP)/100
      IF (COMPR.GT.MINLR) COMPR=MINLR
      COMPG=(GLOTP*LASTP)/100
      IF (COMPG.GT.MINLG) COMPG=MINLG
    8 CONTINUE
C
C             ***** TEST FOR A RECOGNITION INTERVAL *****
      INTER=INTER+1
      IF (XX.GT.COMPR) GO TO 443
      IF (LEVEL.EQ.1) GO TO 444
```

```
C
      LEVEL=1
C     EXAMINE INTERVAL
      NGLOT=NGLOT+INTER
      DUR(IOUT)=NGLOT
      IOUT=IOUT+1
C
C     END OF EXAMINATION
      INTER=0
      GO TO 444
  443 LEVEL=0
  444 CONTINUE
C
C               ***** TEST FOR A GLOTTAL INTERVAL *****
      INTERG=INTERG+1
      IF (XX.GT.COMPG) GO TO 423
      IF (LEVELG.EQ.1) GO TO 424
      IF (INTERG.LT.STOP) GO TO 425
C
C     WE NEED TO PRINT
      IF (IOUT.LE.1) GO TO 425
      IOUT=IOUT-1
C
C               ***** PITCH PERIOD CORRECTION *****
C     STATEMENT 21 IS THE FAIL EXIT
C     WAS THE LAST PRINT LINE NP PULSES?
      IF (CKNP.EQ.1) GO TO 30
C     NEW PERIOD
   27 NPER=DUR(IOUT)
C     CHECK FOR VOICING
C     IS NPER WITHIN VOICP PERCENT OF LPER?
      IF (IABS(NPER-LPER).GT.((LPER*VOICP)/100)) GO TO 22
C     YES, SO WE PROBABLY HAVE VOICING
C     CHECK THAT THERE ARE AT LEAST TWO PULSES
      IF (IOUT.LE.1) GO TO 21
C     NEXT LOWER PULSE
      MPER=DUR(IOUT-1)
C     CHECK FOR SKIPPING
C     IS NEXT LOWER PULSE CLOSER?
      IF (IABS(MPER-LPER).GT.IABS(NPER-LPER)) GO TO 21
C     CHECK THAT THERE ARE AT LEAST THREE PULSES
      IF (IOUT.LE.2) GO TO 29
C     SECOND LOWER PULSE
      JPER=DUR(IOUT-2)
C     CHECK IF JPER IS CLOSER TO LPER THAN MPER IS
      IF (IABS(JPER-LPER).GT.IABS(MPER-LPER)) GO TO 29
C
C     DOUBLE SKIPPING HAS OCCURRED
C     MOVE TWO EXTRA PULSES TO THE NEXT PITCH PERIOD
      IOUT=IOUT-2
C     AND STORE THE REST
      WRITE (S,63) IOUT,(DUR(I),I=1,IOUT)
C     RESET THINGS
      LPER=JPER
      NGLOT=NPER-JPER
      DUR(2)=NGLOT
```

```
      DUR(1)=MPER-JPER
      IOUT=3
      GO TO 64
C
C     PREVIOUS LINE WAS NP PULSES,SO WE CANNOT ATTEMPT ANY
C     CORRECTION BASED ON VOICING
C     TURN OFF THE FLAG
   30 CKNP=0
      NPER=1
      GO TO 21
C
C     SINGLE SKIPPING HAS OCCURRED
C     SO MOVE THE EXTRA PULSE TO THE NEXT PITCH PERIOD
   29 IOUT=IOUT-1
C     AND STORE THE REST
      WRITE (S,63) IOUT,(DUR(I),I=1,IOUT)
C     RESET THINGS
      LPER=MPER
      NGLOT=NPER-MPER
      DUR(1)=NGLOT
      IOUT=2
      GO TO 64
C
C     CHECK FOR DOUBLE PITCH PERIOD
C     HAVE TO HAVE AT LEAST 2 PULSES
   22 IF (IOUT.LT.2) GO TO 21
C     IS NEW PITCH PERIOD WITHIN VOICP PERCENT OF TWICE LPER?
      IF (IABS(NPER-2*LPER).GT.((2*LPER*VOICP)/100)) GO TO 21
C     YES, PROBABLY HAVE A DOUBLE PITCH PERIOD
C     FIND THE FIRST PULSE GREATER THAN LPER
      DO 23 I=1,IOUT
      ICLOSE=I
      IF (DUR(I).GT.LPER) GO TO 24
   23 CONTINUE
      GO TO 21
C
C     FIND WHICH PULSE IS CLOSER TO LPER
   24 IF (ICLOSE.LE.1) GO TO 25
      IF (IABS(DUR(ICLOSE)-LPER).GT.IABS(DUR(ICLOSE-1)-LPER))
     &ICLOSE=ICLOSE-1
C     CHECK THAT THE CLOSEST PULSE IS NOT THE LAST ONE
   25 IF (ICLOSE.EQ.IOUT) GO TO 21
C     IS THE LENGTH OF THE FIRST PITCH PERIOD WITHIN VOICP
C     PERCENT OF LPER?
      IDUR=DUR(ICLOSE)
      IF (IABS(IDUR-LPER).GT.((LPER*VOICP)/100)) GO TO 21
C     DEFINITELY HAVE A DOUBLE PITCH PERIOD
C     STORE THE FIRST PITCH PERIOD
      WRITE (S,63) ICLOSE,(DUR(I),I=1,ICLOSE)
      IOUT=IOUT-ICLOSE
C     MOVE THE SECOND PITCH PERIOD INTO PLACE
      DO 26 I=1,IOUT
      I1=ICLOSE+I
   26 DUR(I)=DUR(I1)-IDUR
      LPER=IDUR
C     BACK TO THE TOP
```

```
      GO TO 27
C
C     WRITE OUT PITCH PERIOD
   21 WRITE (S,63) IOUT,(DUR(I),I=1,IOUT)
   63 FORMAT(3X,I2,15I5/5X,15I5)
      LPER=NPER
      NGLOT=0
      IOUT=1
   64 CONTINUE
C     RECOMPUTE THRESHOLDS
      COMPR=(RECP*LASTP)/100
      IF (COMPR.GT.MINLR) COMPR=MINLR
      COMPG=(GLOTP*LASTP)/100
      IF (COMPG.GT.MINLG) COMPG=MINLG
      LASTP=0
      CK2=0
  425 INTERG=0
      GO TO 424
  423 LEVELG=0
  424 CONTINUE
C
C     CHECK IF WE WENT BY NP PULSES
      IF (IOUT.LE.NP) GO TO 60
C     YES, SO PRINT THEM
      IOUT=NP
      WRITE (S,63) IOUT,DUR
C
      NGLOT=0
      IOUT=1
C     RECOMPUTE THRESHOLDS
      COMPR=(RECP*LASTP)/100
      IF (COMPR.GT.MINLR) COMPR=MINLR
      COMPG=(GLOTP*LASTP)/100
      IF (COMPG.GT.MINLG) COMPG=MINLG
      LASTP=0
      CK2=0
      CKNP=1
      LPER=1
   60 CONTINUE
C
C     END OF ONE-RECORD LOOP
  445 CONTINUE
      GO TO 2
C
C     END OF FILE EXIT
  101 IF (IFLAG.EQ.1) GO TO 3
      IFLAG=1
      ENDFILE S
      REWIND S
C     PRINT THE PULSE POSITION CHART FOR THIS EVENT
      CALL PRPPC(P,S)
      REWIND S
  103 EVENT=EVENT+1
      IF (NFILES.EQ.NFILE) GO TO 4
      NFILE=NFILE+1
      GO TO 1
```

```
C
C      TWO END OF FILES IN A ROW
     3 CALL TPREWZ(INSET)
       GO TO 10
C
C      END OF FILE ON SYSIN
    90 RETURN
C
       END
       SUBROUTINE PRPPC(Q,S)
       DATA ASTER,SLASH/'*','/'/
       INTEGER DUR(30),Q,S
       REAL LINE(131),CLEAR(131)
       DATA CLEAR/9*' ','|',9*' ','|',9*' ','|',9*' ','|',
      2 9*' ','|',9*' ','|',9*' ','|',9*' ','|',9*' ','|',
      3 9*' ','|',9*' ','|',9*' ','|',9*' ','|',' '/
C      PAGE HEADER
       WRITE (Q,78) (I,I=20,260,20)
    78 FORMAT(' 0'13I10)
     8 READ (S,9,END=7) IOUT,(DUR(I),I=1,IOUT)
     9 FORMAT(3X,I2,15I5/5X,15I5)
C      CLEAR LINE
       DO 1 I=1,131
     1 LINE(I)=CLEAR(I)
C      PUT ASTERISKS ON LINE
       DO 3 I=1,IOUT
       K=DUR(I)/2
       IF (K.GT.130) GO TO 5
       IF (K.LE.0) GO TO 11
     3 LINE(K)=ASTER
     6 WRITE (Q,4) LINE
     4 FORMAT(' |'131A1)
       GO TO 8
C      TOO FAR OVER
     5 LINE(131)=SLASH
       GO TO 6
C
C      INCORRECT K
    11 WRITE (Q,12) K
    12 FORMAT('  K=' I10)
C
     7 RETURN
       END
/*
//GO.FT18F001 DD UNIT=2314,DSN=&&SYSUT1,SPACE=(800,(25,10)),
// DCB=(LRECL=80,BLKSIZE=800,RECFM=FB)
//GO.FT25F001 DD UNIT=TAPE,LABEL=(,NL),VOL=SER=025858
//GO.SYSIN DD *
25
     20    42    70    20 -3000 -3000    20
     20    89    70    20 -3000 -3000    20
     20   110    70    20 -3000 -3000    20
     20    51    70    20 -3000 -3000    20
     20    66    70    20 -3000 -3000    20

/*
```

```
      SUBROUTINE PITG(D,OUTSET,FACTOR)
C     PITG GENERATES PULSE INTERVAL TABLES ACCORDING TO
C     INFORMATION IN DATASET D
C     D IS THE READ DATASET FOR CONTROL CARDS
      INTEGER D,OUTSET,DUR(31),PIT4(100)
      DATA DUR(1),IPIT/0,1/
      INTEGER*2 PIT(50)
C
C              ****** GET CONTROL CARD ******
    4 READ (D,91,END=7) N,IOUT,(DUR(J+1),J=1,IOUT)
   91 FORMAT(I3,I2,15I5/5X,15I5)
C              ****** GENERATE PULSE INTERVAL TABLE ******
      DO 81 N1=1,N
C     IF N IS BLANK (I.E., ZERO) THIS LOOP IS DONE ONCE
      DO 81 J=1,IOUT
      INTER=(DUR(J+1)-DUR(J))*FACTOR
   42 IF (INTER.GT.254) GO TO 41
      IF (INTER.LT.1) INTER=1
      PIT4(IPIT)=INTER
      IF (IPIT.LT.100) GO TO 81
      DO 82 M=1,50
      K=2*M
   82 PIT(M)=256*PIT4(K-1)+PIT4(K)
      CALL TPPUTZ(OUTSET,PIT,100)
      CALL TPWTEZ(OUTSET)
      IPIT=0
   81 IPIT=IPIT+1
      GO TO 4
C
C     INTERVAL LONGER THAN 254 COUNTS
   41 PIT4(IPIT)=255
      IF (IPIT.LT.100) GO TO 44
      DO 45 M=1,50
      I=2*M
   45 PIT(M)=256*PIT4(I-1)+PIT4(I)
      CALL TPPUTZ(OUTSET,PIT,100)
      CALL TPWTEZ(OUTSET)
      IPIT=0
   44 IPIT=IPIT+1
      INTER=INTER-254
      GO TO 42
C
C     END OF FILE ON DATASET D
    7 DO 61 J=1,40
      PIT4(IPIT)=255
      IF (IPIT.LT.100) GO TO 61
      DO 62 M=1,50
      I=2*M
   62 PIT(M)=256*PIT4(I-1)+PIT4(I)
      CALL TPPUTZ(OUTSET,PIT,100)
      CALL TPWTEZ(OUTSET)
      IPIT=0
   61 IPIT=IPIT+1
C     NOW RETURN TO CALLING PROGRAM
      RETURN
C     THIS ENTRY POINT IS USED TO WRITE THE LAST TAPE RECORD,
```

```
C     AND REWIND THE OUTPUT TAPE
      ENTRY PITE
      PIT4(IPIT)=0
      DO 83 M=1,50
      I=2*M
   83 PIT(M)=256*PIT4(I-1)+PIT4(I)
      CALL TPPUTZ(OUTSET,PIT,100)
      CALL TPWTEZ(OUTSET)
      CALL TPEOFZ(OUTSET)
      CALL TPEOFZ(OUTSET)
      CALL TPREWZ(OUTSET)
      IPIT=1
      RETURN
      END
```

REFERENCES

1. Lindgren, N., "Machine Recognition of Human Language," IEEE Spectrum, 2 (3) 114-136,(4) 44-59 (1965).

2. Babcock, M.L., "Search and Evaluation of Significant Event Sequences in Automated Speech Analysis" in Accomplishment Summary 1969/70, BCL Report 70.2, Biological Computer Laboratory, Department of Electrical Engineering, University of Illinois, Urbana, pp. 73-80 (1970).

3. Babcock, M.L., et al., "An Approach to the Elimination of the Free Response Distortion" in Search and Evaluation of Significant Event Sequences in Automated Speech Analysis, NASA Report 1, Department of Electrical Engineering, University of Illinois, Urbana, pp. 50-58 (1969).

4. Babcock, M.L., et al., "Adaptive Sampling of Speech" in Search and Evaluation of Significant Event Sequences in Automated Speech Analysis, NASA Report 1, Department of Electrical Engineering, University of Illinois, Urbana, pp. 59-66 (1969).

5. Babcock, M.L., "Asynchronous Sampling of Speech and its Evaluation" in Accomplishment Summary 1969/70, BCL Report 70.2, Biological Computer Laboratory, Department of Electrical Engineering, University of Illinois, Urbana, pp. 263-270 (1970).

6. Mergler, H.W., Digital Feature Analysis of a Limited Speech Vocabulary, Report 1-66-37B, Digital Systems Laboratory, Case Institute of Technology, Cleveland (1966).

7. Cohen, J.R., An Automatic Process for the Extraction of Pitch Periods of Speech, NASA Report 2, Department of Electrical Engineering, University of Illinois, Urbana, 67 pp. (1970).

8. Flanagan, J.L., Speech Analysis,Synthesis and Perception, Academic Press, New York, p. 17 (1965).

9. Babcock, M.L., personal communication (1969).

10. Wells, R.A., C.E. Carter and H.G. Friedman, Illinet Analog Facilities, Report 280, Department of Computer Science, University of Illinois, Urbana (1968).

11. Wells, R.A., The University of Illinois Attached Machine Operating System: Design Criteria and Program Logic, Report 334, Department of Computer Science, University of Illinois, Urbana (1969).

12. Wells, R.A., "Attached Machine Operating System, Users' Guide (Version 3.1)" in ARAMIS Manual (Section 3-00-2), Department of Computer Science, University of Illinois, Urbana (1970).

13. ARAMIS Manual, Department of Computer Science, University of Illinois, Urbana (1970).

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* University of Illinois Biological Computer Laboratory Urbana, Illinois 61801	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

PULSE INTERVAL REPRESENTATIONS OF SPEECH EVENTS

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Scientific; ; interim

5. AUTHOR(S) *(First name, middle initial, last name)*

John W. Atwood

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
October 15, 1970	67	13

8a. CONTRACT OR GRANT NO. AFOSR 70-1865 b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) Technical Report 22 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)* UILU-ENG-70-294, NASA 4

10. DISTRIBUTION STATEMENT

Distribution of this document is unlimited

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
co-sponsor: NGR 14-005-111	Air Force Office of Scientific Res. Dir. of Elect. & Solid St. Sciences Arlington, Virginia 22209

13. ABSTRACT

A new technique for the analysis, synthesis, and transmission of human speech is described which is based on the computation of time intervals between threshold transitions of the derivative in the speech waveform. The position taken here is that the sequence in which significant speech events occur, and their temporal relationships to each other, are for the description of speech as important as its frequency-domain properties, i.e., the dynamic properties and the sustained properties of the speech waveform each are considered to contribute to our understanding of human speech.

The described method of coding speech, in which only the intervals between transitions are recorded, is shown to have advantages in certain areas of speech analysis, synthesis, and transmission, over other methods which rely essentially on phenomena in the frequency domain.

In particular, it is shown that this coding method provides a basis for transmitting speech with high intellidibility at low transmission rates permits the measurement of many properties of the speech waveform which are difficult or impossible to evaluate using conventional techniques, and allows synthesis, by rule, of many speech sounds.

DD FORM 1 NOV 65 1473

UNCLASSIFIED
Security Classification

UNCLASSIFIED
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Speech						
Speech Analysis						
Speech Synthesis						
Speech Transmission						
Frequency Domain of Speech						
Time Domain of Speech						
Intelligibility of Coded Speech						
Vocoder						
Pulse Coding of Speech						
Wave Forms of Speech						

UNCLASSIFIED
Security Classification