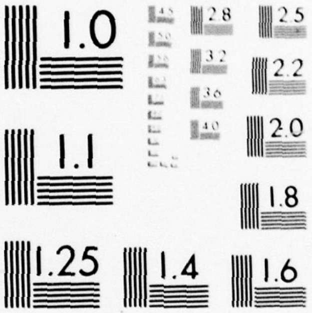
AD-A055 364 PHILCO-FORD CORP WILLOW GROVE PA F/G 17/2
PHONEMIC CATEGORY EXPERIMENTS.(U)
JAN 70 L R FOCHT F30602-69-C-0169
UNCLASSIFIED RADC-TR-70-84 NL

1 OF 1
AD
A055 364

END
DATE
FILMED
8-78
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

FOR FURTHER TRAN

380

②

⑱ ⑲

RADC-TR-70-84

⑨ Final Technical Report. Jan 69 - Jan 70,
July 1970

⑥ PHONEMIC CATEGORY EXPERIMENTS.

Philco-Ford Corporation ← use
Willow Grove, Pa.

⑩ Louis R./Focht

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DDC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

⑪ Jan 70

⑮ F30602-69-C-0169 ⑫ 82p.

~~Each transmittal of this document outside the Department of Defense must have prior approval of RADC (EMITP), GAFB, NY 13440.~~

This document has been approved for public release and sale; its distribution is unlimited.

STOCK COPY

Rome Air Development Center
Air Force Systems Command
Griffiss Air Force Base, New York

389 130

PHONEMIC CATEGORY EXPERIMENTS

Louis R. Focht
Philco-Ford Corporation

~~Each transmittal~~ of this document
outside ~~the Department of Defense~~
~~must have prior approval of RADC~~
(EMITP), GAFB, NY 13440.

S-191, 810
STOCK COPY

FOREWORD

This final technical report was prepared by Philco-Ford Corporation, Willow Grove, Pennsylvania under Contract F30602-69-C-0169, Project 7055, Task 705504. The RADC project engineer was Mr. Richard Vonusa (EMITP).

— non RDT/E
PE

This work covers a period of time from January 1969 to January 1970 and represents an effort of the Advanced Engineering and Research Laboratory of Philco-Ford's Data Recognition and Processing Engineering Department, managed by Mr. Frank Teklits. Mr. Louis R. Focht directed and carried out the technical program under the supervision of Mr. Thomas J. Harley, Jr., Supervisor of the Advanced Engineering and Research Laboratory.

[redacted].

This technical report has been reviewed and is approved.

Richard S. Vonusa

Approved: RICHARD S. VONUSA
Project Engineer
EM Processing Section

H Davis

Approved: HOWARD DAVIS
Acting Chief
Intel & Recon Division

FOR THE COMMANDER: Irving J Gabelman
IRVING J. GABELMAN
Chief, Plans Office

Abstract

This report describes the efforts undertaken to improve the experimental model "Voice Sound Recognizer" originally built under Contract AF 30602-67-C-0300. This equipment utilized the techniques of Single Equivalent Formant parameter extraction, phonemic category recognition, and category-sequence word recognition.

Extensive hardware and software modifications to the basic recognizer system were made during the program which include the use of semiautomatic speaker adaptation by means of distance functions defined by sets of phonemic category strings and nearest neighbor word recognition decisions.

The final recognizer configuration displayed a reduced speaker sensitivity and an average recognition rate for four speakers of 85% when using a 25-word vocabulary.

EVALUATION

The objective of this program was to make necessary modifications and improvements to an automatic isolated word recognizer fabricated under a previous contract. Words are identified by recognizing basic sound classes and their order or sequence of recognition. This program was aimed at making the necessary hardware and software modifications so that higher recognition rates could be achieved in recognizing some of the more difficult sound categories. Previously, the nasal category representing the sounds "m", "n", and "ng" was recognized with a very low probability of correct recognition. During this effort the reliability of the nasal category detector was greatly improved. In addition, the vowel and stop detectors were significantly modified to improve their recognition rates.

Software programs were written so that the sound recognizer hardware could be evaluated more efficiently using different speakers and varying the quantity of library words.

Word recognition is performed by using an algorithm which uses distance functions generated by groups of sound category sequences and nearest neighbor word recognition decisions.

The existing system is now capable of recognizing four different speakers uttering 25 test words with word recognition ratesvarying from 72% to 92%.

In its present form, it is a usable system as an interface between man and machine using a preselected vocabulary of approximately 25-30 words.

Word recognition rates could still be improved if a feedback system was incorporated. The feedback system would utilize various linguistic rules, apriori knowledge of the library words, and certain command language restraints. Such a system could very well be implemented and activate various systems by means of voice control.

Richard S. Vonusa
RICHARD S. VONUSA
Project Engineer
EC Processing Section

TABLE OF CONTENTS

ILLUSTRATIONS

TABLES

SECTION I

INTRODUCTION

The building of a real-time isolated word speech recognizer, utilizing the Single Equivalent Formant (SEF), phonemic category recognition and category-sequence word recognition techniques, was accomplished during a previous contract F30602-67-C-0300. This recognizer was entirely hardware implemented and capable of recognizing a total of any 12 words chosen from a larger 112-word vocabulary of Fortran programming words. The recognizer design concepts were such that the vocabulary could be expanded to the full 112 words; however, the equipment was limited to 12 words for reasons of implementation cost. The changing of vocabulary words was accomplished by altering hardwired category-sequence word-recognition logic to conform to design data obtained from sample utterances of the desired vocabulary words. The design of this recognizer and the underlying concepts are fully described in the final reports for contracts F30602-67-C-0300 and AF 30(602)4170 and should be considered a part of this report.

Test results from the phonemic category word recognizer demonstrated a tendency towards speaker sensitivity. As shown in Table 1, the equipment achieved a recognition rate of 93% (for a 12-word vocabulary) during the testing of the speaker who supplied word logic design data. However, for new speakers, the recognition rates dropped to the 70% region. To overcome this problem, RADC desired that three efforts be undertaken during the program described in this report:

1. Improvement of the existing phonemic category recognition logic.
2. Implementation of the category-sequence word recognition logic in software.
3. Provide a means to adapt the recognizer to individual speakers.

The first of these efforts was directed towards finding and eliminating general problems in the phonemic category detectors. The latter two efforts

Table 1

Recognition Rates for the Phonemic Category Word Recognizer

		Correct Response	Non or Multiple Response	Error Response
(This speaker supplied the training data)	Speaker #1	93%	4.6%	2.4%
	#2	68%	16.0%	16.0%
	#3	72%	18.0%	10.0%
	#4	71%	18.0%	12.0%

were undertaken to reduce the speaker sensitivity of the word logic. The technique of adaptation to individual speakers was felt to be a reasonable system compromise in applications where the speaker's speech characteristics can be determined prior to use. This is, of course, practical only if the process of adaptation is accomplished rapidly and automatically.

Early in the program, a technique for word logic adaptation was conceived and tested by RADC project personnel. It proved feasible for use in the recognizer and was subsequently used during the program. The technique involved the generation of a "training library" by each speaker, which consisted of a number of samples for each word in the vocabulary. These training samples are stored in the computer memory in the form of phonemic category sequences. A separate training set is stored for exclusive use by each speaker whenever he wishes to use the recognizer. These training sets are used during recognition as examples against which the unknown test word must be matched by the category-sequence word logic. The variations within an individual speaker's training library had already shown by the previous program to be within the match capability of the category-sequence word recognition logic. Thus, speaker-to-speaker variation was effectively eliminated. Furthermore, training can be accomplished automatically and as rapidly as the speaker can utter a set of library words.

Table 2 shows the typical improvement obtained by this technique at an early point in its development. The test results are for three speakers uttering 10, 20, and 30 word vocabularies, and show average recognition rates of 91, 85, and 80 percent, respectively, and relatively small differences from speaker to speaker.

The training library technique provides an additional important advantage in that a vocabulary may also be changed as rapidly as the speaker can utter a new set of training library samples.

Figure 1 shows the block diagram of the hardware/software speech recognizer that evolved during the course of the program. It consists of the phonemic category recognition logic, a computer interface, the RADC computer, tape deck, and teletype output. The operational sequence consists of a speaker uttering a set of training words into the phonemic category recognizer. The detected category sequence is fed to the computer through the interface

Table 2

Recognition Rates for Individual Speakers Using a Training Library for Logic Adaptation

	10-Word Vocabulary	20-Word Vocabulary	30-Word Vocabulary
Speaker #1	95%	90%	93%
Speaker #2	95%	90%	80%
Speaker #3	85%	80%	77%
Average	91%	85%	80%

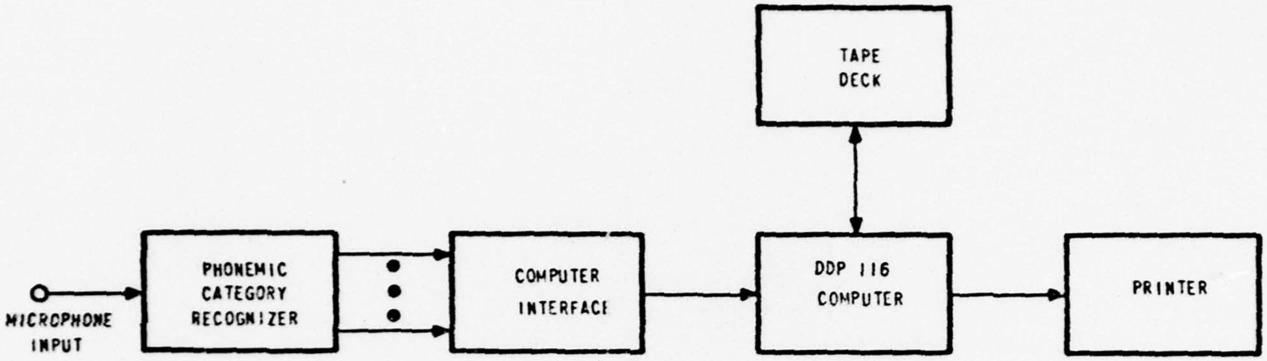


Figure 1

System Block Diagram

and stored in memory. If the speaker wishes to use the recognizer at a later date, his training library is transferred to tape so that he will not have to generate a training library a second time. Once the training library has been generated, the recognizer operates in the recognition mode, at which time the software category sequence logic carries out a matching process on each word as it is uttered. The teletype is used to print out the final recognition decision. The entire operation is essentially a real-time process.

The phonemic category logic recognizes a total of 13 phonemic categories. The categories are shown in Table 3 and are generally divided into groups denoting manner of articulation, i.e., voiced stops, unvoiced stops, nasals, etc. The vocabulary utilized during the program consisted of words taken from the list of Fortran programming words shown in Table 4.

Section II of this report describes hardware modification while Section III describes the word logic concepts and software developed during the program. The remaining Sections IV and V describe the results, conclusions, and recommendations.

Table 3

Phonemic Categories

Phonemic Category Number	Phonemes in the Category
1	s
2	ʃ
3	f, θ, h
4	p, t, k
5	b, d, g
6	m, n, r
7	i, j
8	I, ɛ, æ
9	ʌ, a, ɝ
10	l, ɔ, U
11	u, w
13	V - vowel
14	F - fricative

Table 4

Fortran Vocabulary

1. OH
2. ONE
3. TWO
4. THREE
5. FOUR
6. FIVE
7. SIX
8. SEVEN
9. EIGHT
10. NINE
11. ASTERISK
12. SLASH
13. PLUS
14. MINUS
15. PERIOD
16. COMMA
17. SEMICOLON
18. SPACE
19. EQUAL
20. OPEN
21. CLOSED
22. BRACKET
23. A
24. B
25. C
26. D
27. E
28. F
29. G
30. H
31. I
32. J
33. K
34. L
35. M
36. N
37. O
38. P
39. Q
40. R
41. S
42. T
43. U
44. V
45. W
46. X
47. Y
48. Z
49. TRUE
50. FALSE
51. IDENTITY
52. NOT
53. LESS
54. THAN
55. GREATER
56. OR
57. AND
58. GO
59. ASSIGN
60. IF
61. RETURN
62. END
63. DO
64. CONTINUE
65. PAUSE
66. STOP
67. COMPLETE
68. DIMENSION
69. SUBROUTINE
70. EXTERNAL
71. BLOCK
72. READ
73. PRINT
74. PUNCH
75. WRITE
76. ENDFILE
77. BACKSPACE
78. REWIND
79. FORMAT
80. CALL
81. FUNCTION
82. EQUIVALENCE
83. COMMON
84. DATA
85. INTEGER
86. REAL
87. DOUBLE
88. PRECISION
89. COMPLEX
90. LOGICAL
91. RELEASE
92. REPORT
93. SYSOPS
94. ABNORMAL
95. MAGTAPE
96. NOCARDS
97. NOPROGRAM
98. MAP
99. EDIT
100. NODEBUG
101. BATCH
102. FORTRAN
103. LIB
104. TAPE
105. JOB
106. PROGRAM
107. ENDCOMP
108. CARDS
109. NOIDENTITY
110. CMPERRS
111. F4DUMP
112. HALT
113. COBOL
114. CONIN
115. REM
116. TAC
117. LOW

SECTION II

PHONEMIC CATEGORY DETECTOR MODIFICATIONS

2.0 Introduction

Figure 2 shows the overall logic for the phonemic category recognizer (developed during the previous program). There are four basic levels of logic. The first of these is the parameter extractor logic. A total of three parameters are extracted: the SEF, the SEF amplitude, and voicing.

The second level of logic is the detection of acoustic features. This level concerns itself with the quantization of parameter levels and slopes. These features are the simplest individual unit of information processed by the recognizer.

The third level of logic is termed acoustic event detection. Strings of features are combined to form units that are perceptually significant but do not necessarily possess the same perceptual value.

The fourth logic level is the detection of phonemic categories. In this logic the various acoustic events are subdivided and then recombined to form categories of identical perceptual events. Figure 3 shows in greater detail the individual functional blocks and their interconnections.

2.1 Error Analysis

At the onset of the program, it was observed that the unvoiced stop detector (category 4) did not respond for certain speakers. An investigation of the problem uncovered a special case logical error in the release detector (card 10) which is described in Section 2.3. This problem showed the need for a general performance analysis of the phonemic category detectors.

This analysis consisted of examining the detected phonemic category sequences, as printed out by the computer, for four speakers uttering one sample each of 20 different words. The detected sequences were then compared with the sequence which would be expected by considering the word pronunciation. Confusion matrices were constructed from which the summary shown in Table 5 was derived. As shown, both the probability of detecting an uttered phoneme

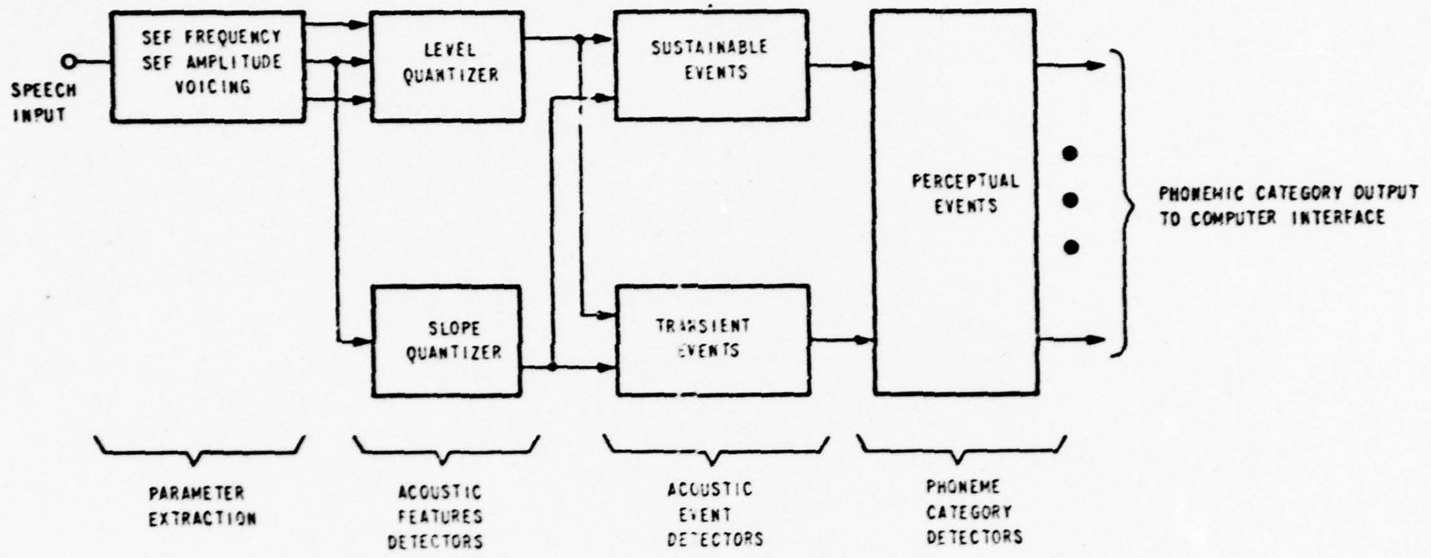


Figure 2

Overall Phonemic Category Recognizer

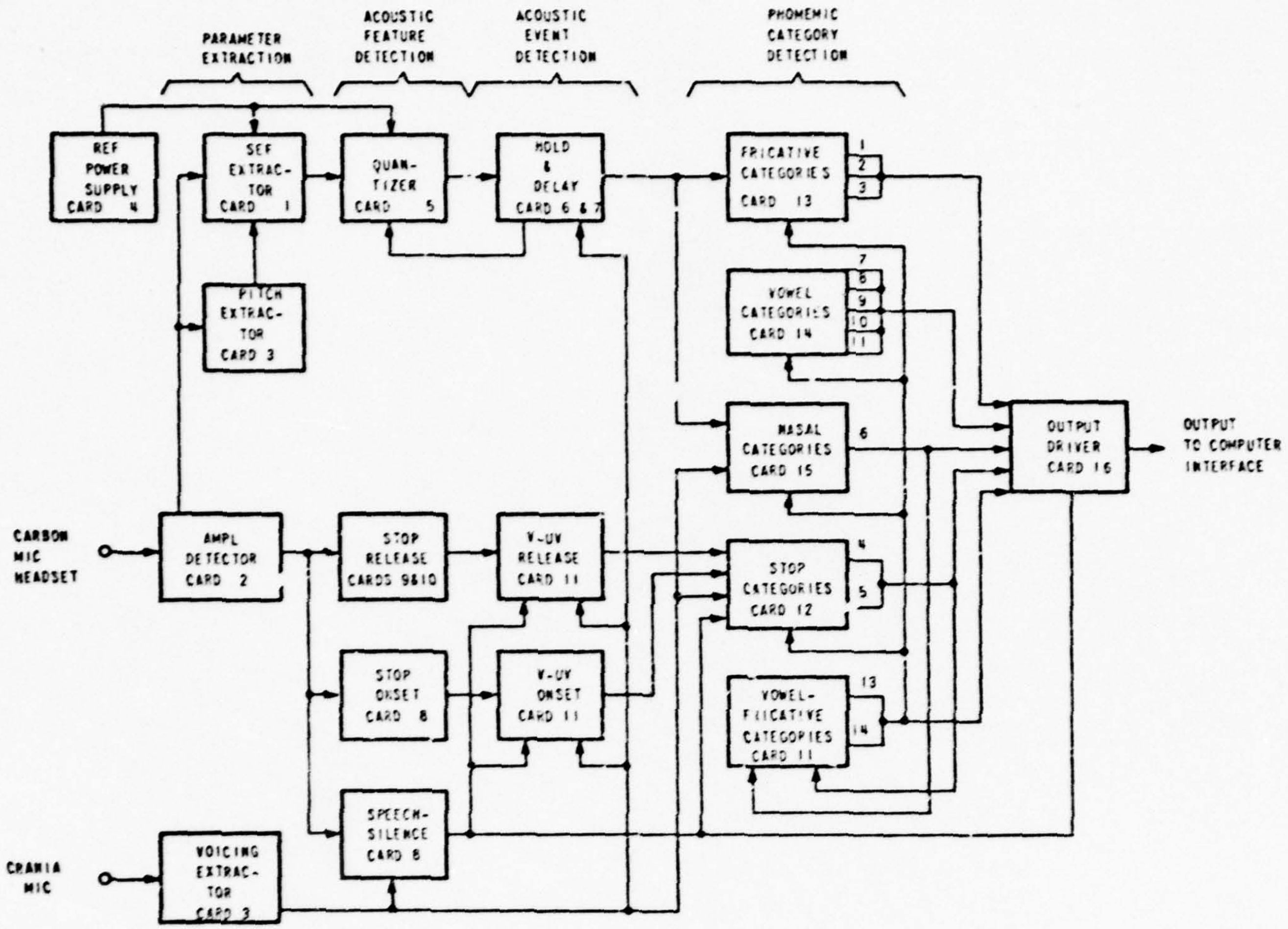


Figure 3

Detailed Phonemic Category Recognizer

Table 5

Performance Analysis Results

Phoneme Category	Probability of Correct Detector Reponse to Uttered Phoneme	Probability that the Detector Response Resulted from the Utterance of that Phoneme
1	.95	1
2	.75	1
3	.33	1
4	.89	.85
5	1	.57
6	.39	.86
7	1	.53
8	1	.67
9	.89	.52
10	1	.76
11	1	.59
13	.97	.99
14	.95	.95

and the probability that a detector response was the result of an uttered phoneme are shown.

Three major problem areas are evident from this data. First and most important, the nasal category (#6) has a very low probability of detection; however, if it is detected, there is a high probability that the phoneme was uttered. Secondly, vowels have just the reverse characteristic, producing many "extra" vowel detections. Finally, the voiced stop detector (category #5) also tends to produce many extra voiced stop detections. The previously mentioned unvoiced stop detector is not evident because a fix was made for this problem before the error analysis was undertaken.

These tests prompted further circuit efforts to eliminate these problems. The following sections describe these efforts.

2.2 Nasal Category Detector Studies

The original nasal category detector in the equipment delivered on contract F30602-67-C-0300 worked by detecting excursions of the SEF signal into the frequency region associated with the articulation of nasals. This SEF region is detected by quantizer circuit 9 in cards 5, 6, and 7. The final output from card 7 is designated Q_9 and requires the presence of the SEF signal in the nasal region for more than 120 ms. This rather long duration requirement was imposed to inhibit sperious nasal detections for certain short duration noiselike excursions of the SEF signal into the nasal region.

Nasal detection failures observed during the performance analysis were found to occur when the articulated nasal was less than 120 ms in duration. This situation frequently occurs for mid-position nasals such as the second nasal in the word "minus".

A new card was made and installed in the phonemic category recognizer to help reduce this problem. (The card is located in position 15A in the recognizer.) This card attempts to differentiate short mid-position nasal characteristics from the previously mentioned spurious SEF excursions. Figure 4 shows the functional block diagram for this card.

The logic utilizes the mid-nasal's characteristic small drop in the amplitude and the excursion of the SEF into the nasal region during this drop. The amplitude rise at the end of nasal articulation is detected

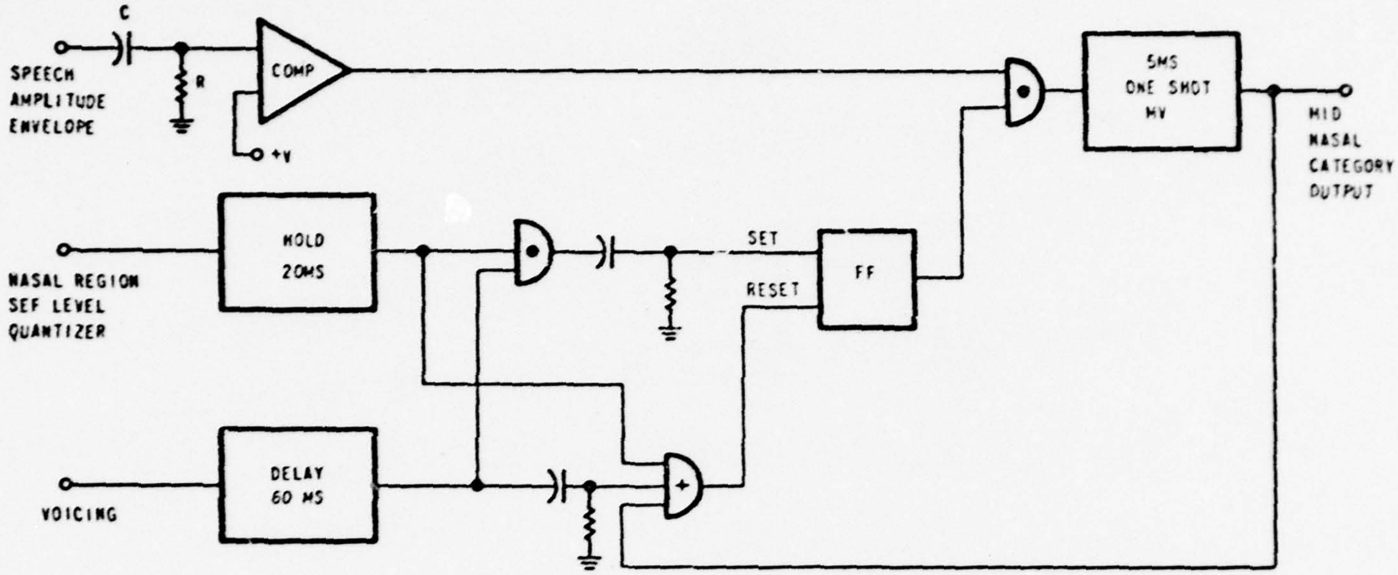


Figure 4

Mid Nasal Detector

by the RC differentiator and threshold detector at the bottom of the figure. Quantized SEF information from the Q_9 hold card (#6) rather than the Q_9 delay card (#7) eliminates the 120 ms duration requirement imposed by the original equipment. The SEF information is passed through a 20 ms hold circuit to insure its presence during the amplitude rise at the end of the nasal. As a further restriction to inhibit spurious SEF responses, it is required that the nasal be preceded by a voiced sound of at least 60 ms duration. This is accomplished by the 60 ms voicing delay circuit. If both preceding voicing and Q_9 occur, the Flip-Flop is set by means of the left-hand "and" gate. This, in turn, enables the right-hand "and" gate to pass the detected amplitude rise which in turn triggers the output one shot. Flip-Flop reset is accomplished through the three input "or" gate. The output of this card is then "or" gated with the output of the original nasal card (#15).

This card provides an improvement when the SEF nasal signal is detected or the drop in amplitude is pronounced enough to be detected. These two conditions were found to be frequently missing. The worst case occurs when the word is pronounced so rapidly that the nasal SEF level is totally missing. Nothing short of the recommendation in Section V is felt to be a satisfactory solution for this latter case.

It was found, however, that for the case of undetectable amplitude changes, a modification could be made that would provide some indication of a nasal's presence. This change involved changes in both the category and the word logic. (The category-sequence word logic changes are described in Section III.) The hierarchy of the category detectors assumes that a word is composed of strings of vowels and fricatives interspersed with nasals an stops. The beginning of a fricative or vowel is always signaled by the phonemic recognizer with a general vowel or fricative category (13 and 14, respectively) and then followed by the specific vowel or fricative category (7, 8, 9, 10, 11 or 1, 2, 3). Thus, a typical vowel fricative sequence would be 13-8-9-14-1. If a nasal occurs between two vowels, the general vowel category #13 detection is generated before both vowels, i.e., 13-7-6-13-8. This is accomplished in the vowel fricative detector (card #11) by resetting the vowel and fricative duration detectors whenever a nasal is detected by card 15 and 15A. However, if the SEF nasal quantizer is not present for more than 120 ms prohibiting an output on card 15 and the amplitude drop is too small to satisfy the detector on card 15A, no resetting of the vowel fricative detector will occur.

The above category vowel-nasal-vowel sequence is then detected as 13-7-8. This situation is particularly hard for the word logic to handle and leads to very poor match scores. To reduce the penalty imposed upon the word logic in such cases, a modification has been made to provide some indication of the occurrence of a nasal when the nasal SEF detector is satisfied but the amplitude criteria is not. The vowel-fricative detector (Figure 5) was modified so that reset occurs by the occurrence of an output from the unfiltered SEF nasal region quantizer (Q_9) rather than the actual output from the nasal detector. Thus, if a short nasal is not detected because the amplitude change is too small even though there is an output in the Q_9 quantizer, then the vowel category (13) will be inserted in the category sequence, i.e , 13-7-13-8. By proper word logic design, the 13 in the middle of the vowel string may be interpreted to mean the presence of a nasal in the word sequence.

2.3 Stop Detector Modification

During the initial testing of the phonemic category word recognizer after delivery to RADC, it was noticed that the unvoiced stop category detector did not respond for certain people's articulation of "p" and "k". Investigation showed that a transient in the voicing detector microphone occurred for these people. This transient produced a 10 to 30 ms duration voiced indication at the onset of the plosive release of the unvoiced energy during the articulation of "p" and "k". This voiced indication was sufficiently long to reset the Flip-Flop for the class 5 release detector through the three input "or" gates shown in Figure 6. The resetting of this Flip-Flop inhibited the detection of the release and thus the unvoiced stop. This situation was cured by changing the voicing input to the delayed voicing signal which requires that voicing be present a minimum of 40 ms before an indication of voicing is produced. Thus, the short bursts of voiced indication were inhibited from effecting the detection of unvoiced stop categories.

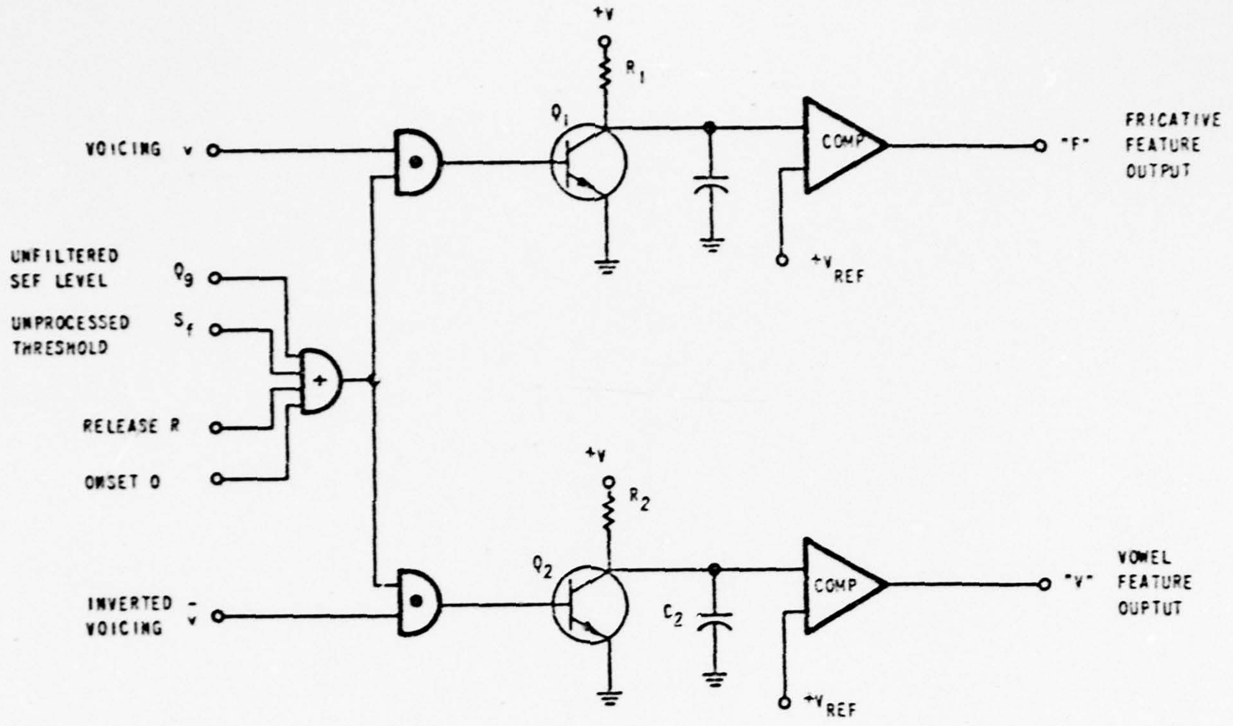


Figure 5

Block Diagram for Modified Vowel-Fricative Feature Detector

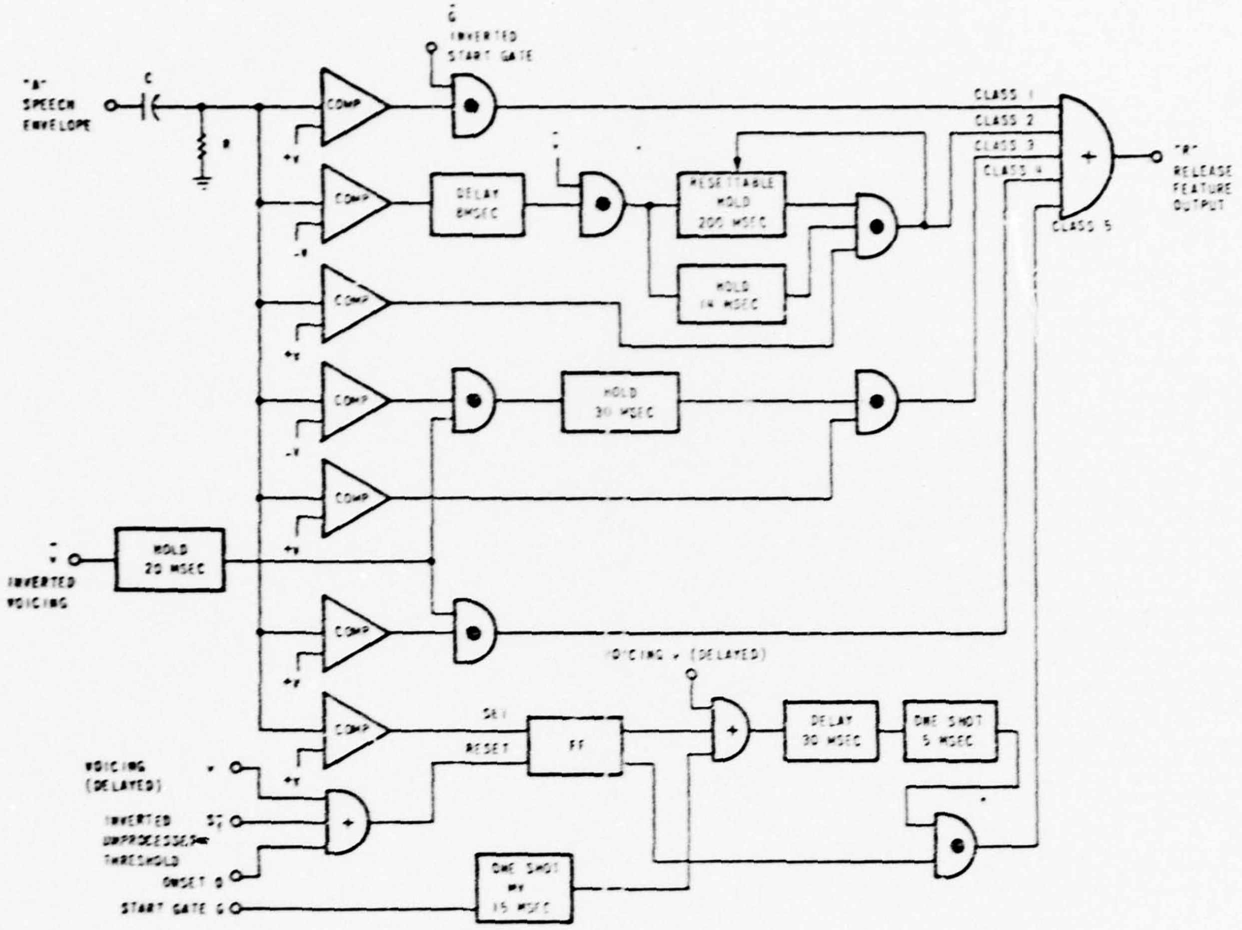


Figure 6

Block Diagram for Modified Release Feature Detector

2.4 SEF Filter

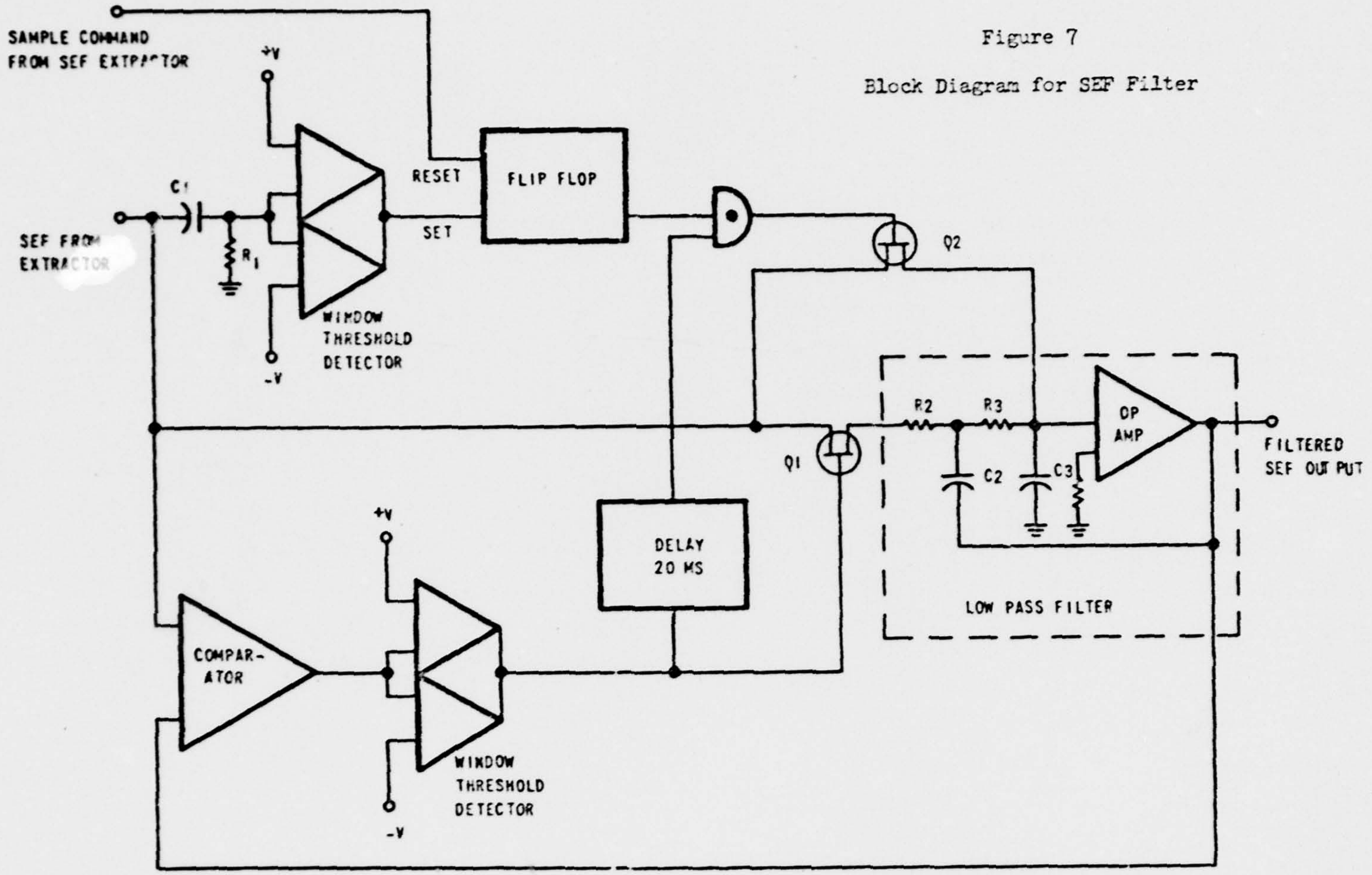
The large number of extra vowel recognitions indicated by the phonemic category error analysis were felt to be caused by the circuit techniques utilized to quantize and filter the SEF signal. The filtering is required to eliminate a small amplitude noise component (100 to 300 Hz) which is found on the basic 20 Hz SEF signal. This noise cannot be removed by simple low pass filtering because there are also significant large amplitude step-like changes in the SEF signal (100 to 300 Hz) which must be preserved. The existing phonemic category recognizer attempted to cope with this problem by quantizing the SEF signal into a number of frequency regions and then filtering each quantizer output with the hold and delay circuits on cards 6 and 7. These circuits first filled in 20 ms or less gaps in the quantizer outputs and secondly required at least 100 ms of signal to be present before a vowel recognition was indicated. This type of processing has the disadvantage that two vowels can be detected simultaneously if the noise and SEF signal are in such a position that two quantizers are activated at the same time. This action tends to produce a switching back and forth between two vowels which results in extra vowel indications from the vowel category detectors.

An attempt was made during the current program to solve this problem by filtering the SEF signal before quantization. This prequantizer filter was a 20 Hz active low pass filter which could be switched in and out of the circuit depending on the presence or absence of the large amplitude step changes in the SEF signal. Thus, small amplitude noise components could be filtered out without losing important step information.

Figure 7 shows the functional block diagram of the circuit developed to perform this switchable filtering action. The SEF signal is fed to a 20 Hz active low pass filter through a FET switch Q_1. When a SEF step change occurs, switch Q_2 is turned on and Q_1 off so that the SEF signal bypasses the filter and is applied directly to the output terminal. Switch Q_1 is controlled by a comparator and window threshold detector. The SEF input signal and the low pass filtered SEF output signal are subtracted by the comparator. The output of the comparator consists of the noise and step function components of the SEF. These are fed to an upper and lower limit threshold detector whose threshold limits are set by +V and -V. These limits are set high enough that they

Figure 7

Block Diagram for SEF Filter



are only exceeded by the large amplitude step changes in the SEF signal. When such a change occurs (in either direction), switch Q_1 is turned off by the window threshold detector output. This is done to protect the output signal from influences of the step change until it is determined to be a step change and not an isolated large amplitude noise spike. Two conditions must be satisfied to verify the step change and turn on Q_2. First, one full sampling interval (a pitch period) of no change must occur between the detection of a step SEF change and its verification. This rejects the influence of one sampling interval SEF impulses that go up and down during adjacent sampling intervals. Secondly, the turning on of Q_2 can occur no sooner than 20 ms after the detection of the SEF change. These requirements are accomplished by means of a window threshold detector, Flip-Flop, "and" gate, and 20 ms delay as shown in Figure 7.

The operation of this circuit proved quite satisfactory in filtering the SEF signal without introducing objectional distortions. The circuit was not, however, installed in the phonemic category recognizer because of the numerous other changes required for installation. It is, however, recommended in Section V that such a change be ultimately added to the phonemic category recognizer.

2.5 Go Light

The purpose of the go light is to provide (1) the speaker with a visual indication that the computer is ready to accept the next word and (2) to disable the hardware while the computer is processing data, i.e., the teletype printing out results. Figure 8 shows the block diagram for this circuit. When the computer is operating, Q_1 is turned inhibiting the operation of the category recognizer by clamping the amplitude detector in a no speech condition. The go light operation is initiated by the computer ready gate. A comparator is used for level conversion of the computer signal. The comparator triggers a .75 second delay to allow sufficient time for all noise from the teletype to die out. The .75 second delayed ready gate turns on the "go" light by means of the lamp driver and turns off transistor switch Q_1. Switch Q_1 enables the amplitude gate detector to operate in its normal manner. As soon as a word is spoken, the go light is turned out by the computer ready

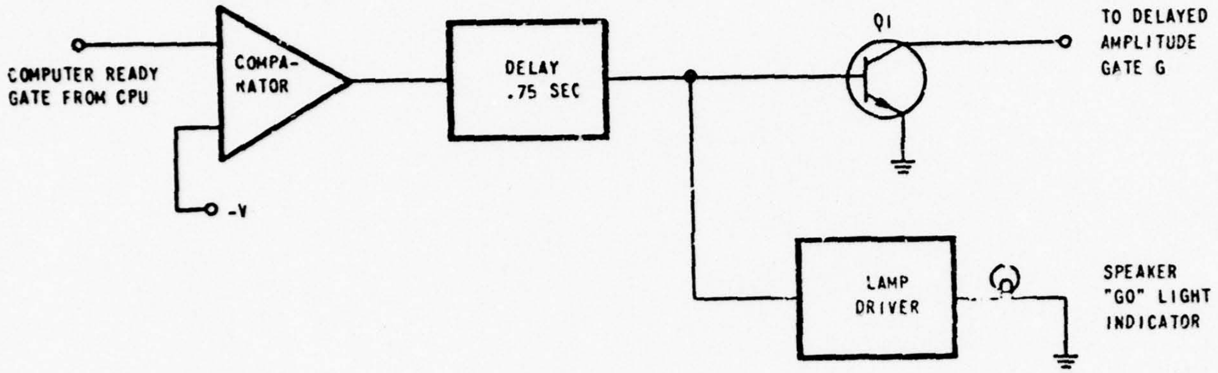


Figure 8

"Go" Light Block Diagram

gate signal. This action again turns on Q_1 disabling the phoneme category recognizer by forcing a no-speech condition on the amplitude gate signal line.

2.6 Voicing Detector Modifications

The voicing detector utilized in the phonemic category recognizer operated by means of a microphone in contact with the top of the head to pick up mechanical vibrations associated with vocal cord activity. (See final report for Contract AF 30(602)4170.) This technique, while extremely accurate, precludes the use of the recognizer for prerecorded speech where this special microphone was not available. An attempt was made during the program to develop a voicing circuit that would operate without the use of the cranium microphone.

The functional block diagram for the new circuit is shown in Figure 9. Three parameters were utilized to form the output decision:

1. the derivative of the dual time constant peak-detected amplitude parameter.
2. the zero crossing rate of the clipped speech.
3. the low frequency component of the amplitude parameter.

The peak-detected amplitude waveform has a spectral component, the frequency of which is a function of the state of voicing. For unvoiced sounds, the peak detector tends to charge in small increments at an average rate of 1000 to 2000 Hz. During voiced sounds the peak detector is influenced by the pitch period and thus tends to charge in fewer but much larger steps with an average rate of 100 to 300 Hz. This difference in peak detector charging frequency is detected by triggering a 500 μs one shot MV with the differentiated (R_1C_1) and amplified amplitude parameter. Low pass filtering of the one shot output converts the pulse rep rate to a baseband parameter whose absolute value correlates with the state of voicing.

This signal is resistively added to a second signal derived from the clipped speech signal. The average zero crossing rate is extracted by triggering the 50 μs one shot MV each time the

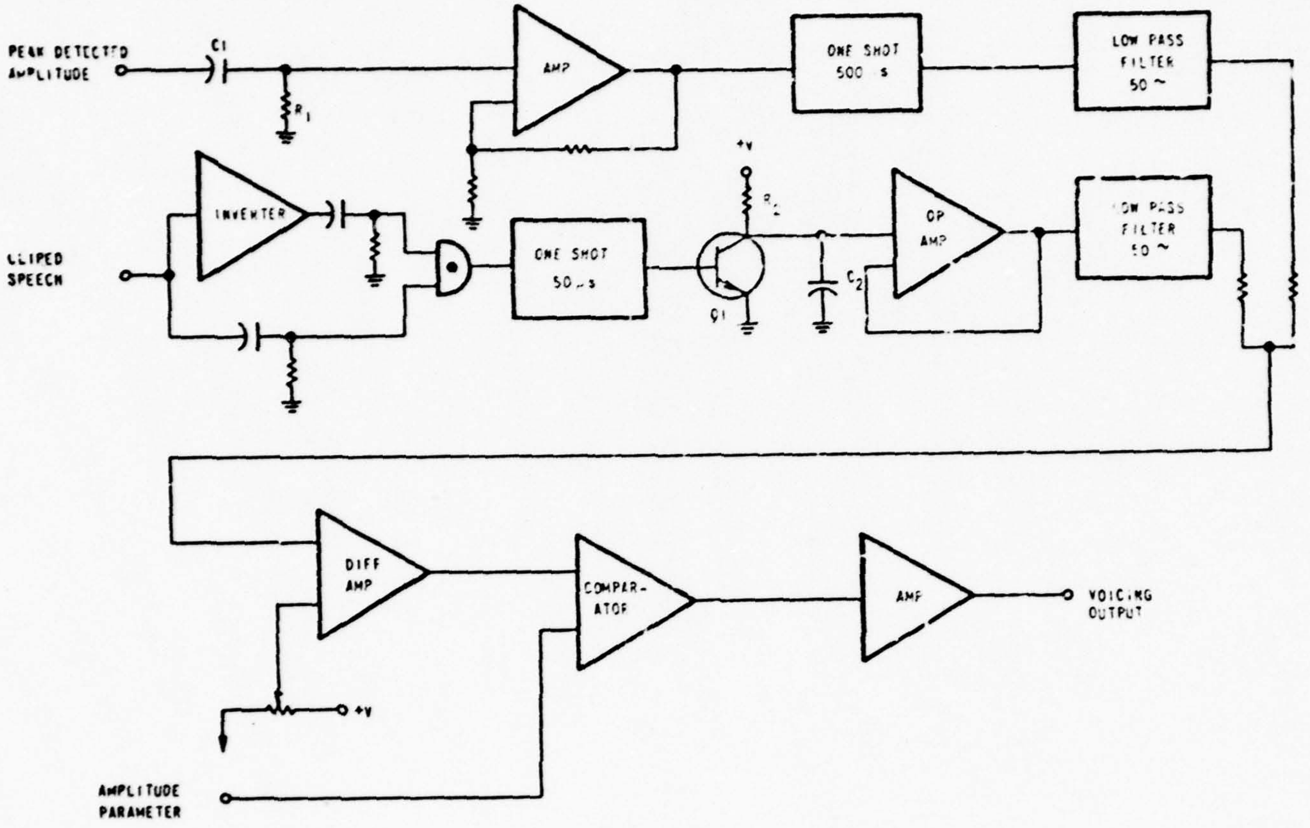


Figure 9

Voicing Detector Block Diagram

clipped speech signal changes state. The one shot is used to reset a ramp generator Q_1, C_2 and R_2 whose average energy is proportional to the period between zero crossings. Low pass filtering is again used to reduce the signal to a baseband parameter.

The summed output of these two signals is subtracted from an adjustable DC signal in a difference amplifier. This DC voltage serves as a threshold adjustment for the circuit. The resultant signal is fed to a comparator along with the amplitude parameter. The amplitude parameter also correlates with voicing in that unvoiced sounds are of lower amplitude than voiced sounds (when considering the system frequency response characteristics). The comparator output is a bilevel signal indicating the voiced and unvoiced states.

This circuit was tested and found to work satisfactorily in good signal-to-noise environemnts when compared to the cranium microphone. However, because of performance losses under poor signal-to-noise conditions, this circuit was not installed in the equipment.

SECTION III

CATEGORY SEQUENCE WORD LOGIC SOFTWARE STUDIES

3.1 Category Sequence Matching Concepts

The phonemic category recognition rates indicated by the error analysis of Section II are obviously something less than perfect. Such category recognition rates most certainly would present serious difficulties in a word recognition logic which requires all phonemes in a word to be correctly detected. The effects of such a requirement are shown by the lower dotted line in Figure 10. Here, the size of the word, in number of phonemes, versus the probability of having correctly detected all of the phonemes within the word is plotted. The figure assumes that the average probability of detecting each phoneme is 0.9. As shown, a five-phoneme word would be correctly recognized less than 60% of the time, while an eight-phoneme word would be recognized a little better than 40% of the time. Such results are, of course, very discouraging and seem contrary to the fact that longer words should be more easily recognized by virtue of their additional redundancy. In order to improve word recognition rates, it is obvious that a word logic must be capable of exploiting vocabulary redundancies.

The effect of using such redundancy is shown by considering the recognition rate of the example just described when any one phoneme in the word is allowed to be missing from the detected phonemic string. The middle dotted line in Figure 10 shows these results. Now, a five-phoneme word would be recognized better than 90% of the time, while an eight-phoneme word is recognized better than 80% of the time. A further refinement of this word recognition technique would be to carry along with the recognition indication the number of phonemes found to be missing. This could be used as part of a measure of recognition confidence or match quality. It might be argued that if a word in the vocabulary (word A) had a correct phonemic spelling identical to the word with a missing phoneme (word B), then a multiple recognition would occur whenever the word "A" were spoken. This, naturally, could be resolved by noting that the word "B" has a missing phoneme and thus a lower match quality. If the reverse occurred, i.e., word "B" were spoken and the phonemic

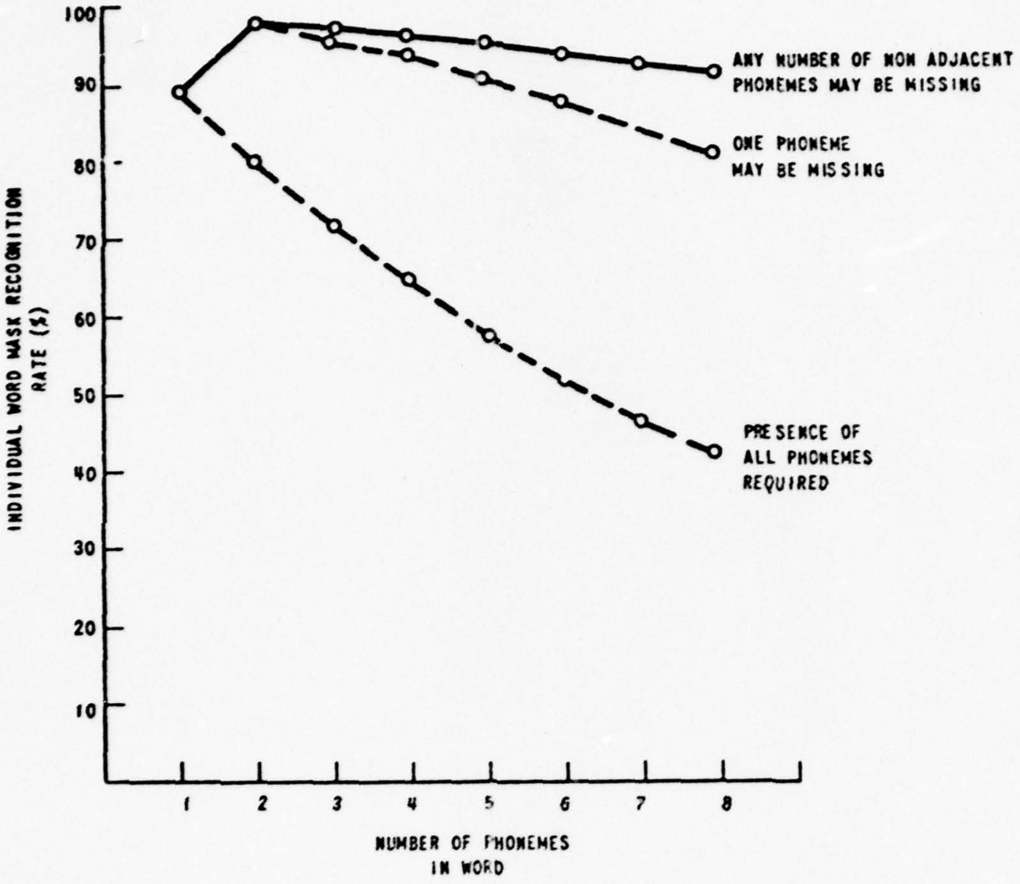


Figure 10

Predicted Word Mask Recognition Rate

category detectors failed to recognize a phoneme such that it produced a sequence identical to the spelling of word "A", then no harm is done because it would have been misrecognized by either system.

The next logical extension of this concept is to allow any non-adjacent phoneme in the word to be missing. The effects of this procedure on recognition rates are shown in the top line of Figure 10. Now, the eight-phoneme word is recognized 93% of the time. The word recognition rate has now exceeded the phonemic category rate for all words less than about 10 phonemes in length.

The category string word recognition logic chosen for this program is based upon these concepts. They are summarized as follows:

1. Word matches will be made in spite of missing or extra phonemes in the detected phonemic category string.
2. The number and type of missing and extra phonemes will be preserved and carried as a match quality for final recognition decision.
3. The final word recognition will be based upon the relative distribution of matched, missing, and extra phonemes.

3.2 Library Concept

The library training concept proposed and implemented by RADC personnel provides a set of examples (several for each library word) against which a test word is matched. Each speaker using the recognizer must provide a set of library samples before the word recognition mode is initiated. This technique reduces the speaker sensitivity of the recognizer.

The optimum number of library samples required for each vocabulary word was determined to be three by experimentation at RADC. More than three library samples per vocabulary word do not appear to increase the recognition rate by any appreciable amount. This number is, of course, a function of phonemic category logic design and thus any major changes in its design would be expected to change these results.

3.3 Matching Algorithm for Library and Test Words

The matching of test word and library word phonemic category sequences is complicated by the fact that there are usually differences between the two sequences. This is obviously the case when two different words are being matched, but also frequently occurs when the words being matched are the same. Variability in both the human articulation and equipment operation produces small but significant differences in the detected phonemic category sequence for two utterances of the same word even when spoken by the same person. Thus, in matching the phonemic category sequences, it has been found desirable to first match gross features in the phonemic category sequence (vowel and fricative categories) and then match the fine detail between these gross phonemic features. In this manner, general similarities between the test and library words may be found without interference from the minor differences between the words.

The category sequence matching algorithm achieves this objective by utilizing an observed general phonemic structure of words. This general phonemic structure can be characterized by a sequence of vowels and fricatives interspersed with stops and nasals, i.e., vowel-stops and/or nasals-fricatives-stops and/or nasals, etc. In terms of phonemic categories used in the phonemic category recognizer, a word may be desribed by the general sequence shown below by assuming that the sequence may be started and ended at any point and that elements may be deleted.

Vowel	Specific Vowel	Nasal and/or Stop	Fricative	Specific Fricative
(13),→	(7 through 11),→	(4 through 6),→	(14),→	(1 through 3),→

Nasal and/or Stop	Vowel	Specific Vowel	Nasal and/or Stop
(4 through 6),→	(13),→	(7 through 11),→	(4 through 6) →
A			**B**

Thus, for example, the word 'time', which has a phonemic category sequence (4) (13) (9,7) (6), matches the general sequence beginning at point A and ending at point B.

The category sequence matching algorithm uses rules derived from the structural characteristics of this general sequence. These rules have been devised to force a match based primarily upon gross features of the phonemic category sequence. These rules have been derived from the following considerations.

During the matching of each phoneme category in the library and test word, the phonemic categories will either be the same or different. If they are the same, a match is considered made and the next pair of phonemic categories in the test and library sequences is compared. If the phonemic categories are different, then it must be assumed that an extra phoneme has occurred in either the test word or library word. The total number of extra phonemes that must be assumed to complete the word match is dependent upon the choice of where the extra phoneme occurred—the test word or the libra word.

To demonstrate this situation, consider the matching of the following sequences:

library word	13-7-4-14-1-12	
test word	13-7---14-1-12	(note the missing 4)

The 13 and 7 of both sequences would be matched. To continue the match, either the 4 in the library word or the 14 in the test word must now be assumed to be extra. If the 4 is called extra, the 14 and 1 would be matched leaving a total of one extra phoneme for the word match. If the 14 in the test word is called extra, then only the 1's could be matched leaving a total of 3 extras for the final word match (14, 14, 4). By consulting the previously described general sequence, the choice of assumed extra library or test phonemes which will produce a minimum number of extra phonemes can be predicted. Examination of each possible combination of matches in the general sequence provides this information. For example, in the general sequence shown below, a test word (14) matched in the position shown library (13)(7-11)(4-6)(14)(1-3)(4-6)(13)(7-11)(4-6)(14)

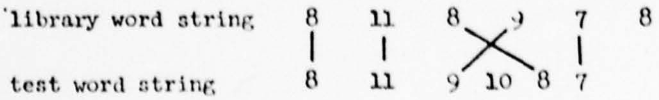
test (13)(7-11)(14)

produces a minimum of extra categories by assuming that an extra (4 through 6) has occurred in the library word.

The rules resulting from this comparison have been tabulated in a 6 x 6 matrix for use in the category sequence word matching algorithm and is shown in Table 6. For each combination of test word and library word phonemic category, an algorithm action is indicated. As shown, the fine detail within the general sequences, i.e., specific fricative (1,2,3) stop and/or nasal (4,5,6) and specific vowel (7,8,9,10,11), is matched in a separate routine called string matching.

3.4 String Matching

Strings are any combination of 7, 8, 9, 10, and 11 or 4, 5, and 6 or 1, 2, and 3 in an unbroken sequence. The intention of the string routine is to systematically find the maximum number of phonemic category matches that can be made solely within the string. This process is accomplished by starting at the end of the string and moving toward the beginning of the string (front of the word), making all possible matches until an extra phoneme is encountered. All matches to this point are recorded. The matching next proceeds to the front of the string and moves toward the end of the string. Again, each phoneme in the test word string is compared with each phoneme in the library word string. When matches are made, the phonemes in question are made unavailable for matches with other unmatched phonemes. After all matches are made, cross linkages between matched pairs are noted. If any occur, a penalty is applied to the match in the form of one negative match. Thus, in the following example, there are a total of four matched categories and two extra categories.



Library Word
Phonemic Category

	1,2,3	4,5,6	7 - 11	12	13	14
1,2,3	-1	0	0	0	0	1
4,5,6	1	-2	1	0	0	0
7 - 11	1	0	-3	0	1	0
12	1	1	1	M	1	1
13	1	1	0	0	M	1
14	0	1	1	0	0	M

Matrix Logic	Action
-1	1,2,3 String
-2	4,5,6 String
-3	7 - 11 String
0	Extra Phoneme in Test Word
1	Extra Phoneme in Library Word
M	Match

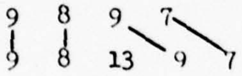
Table 6

6 x 6 Matrix
for Category Sequence
Word Recognition Algorithm

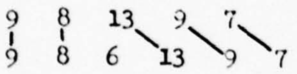
As shown, there are 4 matches but the crossover of linkages reduces the match count by one. A maximum of one negative match is given even when more than one cross linkage occurs.

3.5 Modification to String Routine When a Nasal is Present

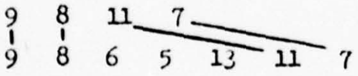
Software work was also directed toward the elimination of word errors that occur when a spoken nasal is detected in either the library word or test word but not both. Under these conditions, the word logic frequently produces an incorrect word decision because the matching of gross features is disrupted and produces very poor matching scores. The effect of this problem was reduced by assuming that a nasal could be missing if a nasal occurred in either the test or library word and not the other. Thus, as a special case, an exception is made to the definition of vowel strings (combinations of 7, 8, 9, 10 or 11) when a 13, 6-13 or 6-5-13 occurs within a sequence of vowels. For example, the category sequences 9-10-6-13-7-8, 9-10-13-7-8 and 9-10-6-5-13-7-8 are considered to be continuous vowel strings. String matching under these conditions is made as in a string match, but the scoring is different. Extra 13's are not counted but extra 6's and 5's are counted. For example,



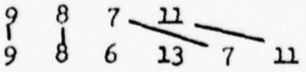
4 matches



5 matches
1 extra nasal (6)



4 matches
2 extra nasals (5,6)



4 matches
1 extra nasal (6)

3.6 Match Quality Calculation

A match using the techniques described in the preceding sections is made between a test word and every word in the training library. The match quality for each word is calculated and the final recognition choice is made by selecting the library word with the best match score. The calculation of the match quality or "Q" is performed by dividing the number of extra categories by the number of match categories for each library word. Two variations to this basic routine for calculating Q have been tried. The first method is weighted by the relatively greater importance of the consonant categories (1, 2, 3, 4, 5 and 6) and the general fricative and vowel categories (13 and 14). Additional weight is applied to these categories by multiplying the appropriate extra category count by two. The value of Q is given by the following equation.

$$Q = \frac{2(E_1 + E_2 + E_3 + E_4 + E_5 + E_6 + E_{13} + E_{14}) + E_7 + E_8 + E_9 + E_{10} + E_{11}}{M_1 + M_2 \ldots M_{13} + M_{14}}$$

where E_1 = extra category 1

E_2 = extra category 2

etc.

M_1 = matched category 1

M_2 = matched category 2

etc.

The second match scoring method makes use of the data obtained from the phonemic category error analysis. Weights are given to extra and matched categories according to their probability of detection and non-detection. The equation for this method of scoring is given by:

$$Q = \frac{2(E_1 + E_2 + E_3 + E_4 + E_6) + 4(E_{13} + E_{14}) + E_5 + E_7 + E_8 + E_9 + E_{10} + E_{11}}{4(M_6) + M_1 + M_2 + M_3 + M_4 + M_5 + M_7 + M_8 + M_9 + M_{10} + M_{11} + M_{13} + M_{14}}$$

The category sequence word recognition software written for the program can provide as a option either of these two forms of Q calculation.

3.7 Software Options

Five basic program options are provided for in the software. These options are:

1. Two different methods of calculating "Q".
2. Adjustable limits of Q values for acceptable recognition.
3. A library evaluate routine in which each library word is tested against every other word in the library to determine the quality of the library samples.
4. A tabulate routine that lists the total number of matches and deletions for each phonemic category which occurred during the testing of a series of words.
5. A phonemic category combiner routine to reduce the total number of categories.

Table 7 shows the methods of calling for each option via computer hardware sense switches and data words. Sense switch 1 is used to continue or stop the program as desired. Sense switch 2 is used to control the number of words printed out after each recognition attempt. Position 1 prints out the three library words which have the highest Q values, whereas position 2 prints out only the first choice word. Sense switch 3 is used to change the values of "DELTA" and "THRESH". These two data words provide a method to reject recognitions which have excessively poor (large) Q values or recognition whose Q values are too close to each other. Position 1 of the sense switch 3 is used for continued use of THRESH and DELTA values, while position 2 is used to change these values. When all library matches exceed either of these two limits during the matching of a test word, the word "Eh?" is printed out.

Data word "CONTROL" is used to change several program options. The first of these is called "old" and "new". The old option uses the first

Table 7

Option Listings

Computer Hardware Sense Switch	Sense Switch Position	Option
1	1	Return and Stop
1	2	New Test Word
2	1	Print 3 Largest Q Value Words
2	2	Print Best Q Value Word
3	1	Use Previous Delta and Thresh Values
3	2	Change Delta and Thresh Values

Data Word	Option
"THRESH" 0.0 → 10	Maximum Acceptable Q Value
"DELTA" 0.0 → 10	Minimum Acceptable Q Value Difference Between Words
"CONTROL"	
[illegible]	"Old" Q Calculation and Nasal String Fix
11	"New" Q Calculation and Nasal String Fix
12	"Old" and "New" Q Calculation and Nasal String Fix
21	"New" Q Calculation and "Tabulate"
22	"Old" and "New" Q Calculation and "Tabulate"
- 1	"Evaluate" Library With "Old" and "New"
-12	"Evaluate" Library With "New" and Proceed as 12
-11	"Evaluate" Library With "New" and Proceed as 11
-21	"Evaluate" Library With "New" and Proceed as 21
-22	"Evaluate" Library With "New" and Proceed as 22
"COARSE"	
FALSE	Normal Categories
TRUE	Combined Categories

method described for the calculating of "Q", while new uses the error analysis weighting method of Q calculation and the nasal string routine described in Section 3.6. CONTROL values 2 and 11 are used to call for old and new Q calculation methods, respectively. When CONTROL is equal to 21, both methods are used and the results printed out sequentially.

CONTROL values 21 and 22 are used to call for a data collection routine in which a summary is printed out showing the total number of times each phoneme category was matched during correct word recognitions and the total number of times each phoneme category was matched in the second choice or incorrect words. It is hoped that this option will provide an understanding of the relative importance of each phonemic category.

The placing of negative signs in front of each CONTROL value initiates the "EVALUATE" option. In this option, the quality of the library is first tested by extracting each phonemic category sequence of the library one at a time and using it as a test word. Obviously, if the vocabulary samples are of good quality, the first choice recognitions should be the other two samples of the particular word remaining in the library. RADC personnel adapted this program to their basic library generation program to automatically check for bad library samples. The addition of the EVALUATE technique to the library generation provided substantial improvements in the overall recognition rates.

The final option provided for in the software is a phonemic category combiner called "COARSE". In this option, any number of categories may be lumped together into a larger category: that is, stop categories 4 and 5 might be lumped into a single new category. This option is to be used to determine the optimum number of categories for a recognizer using a particular vocabulary.

SECTION IV

RESULTS

4.1 Introduction

The performance level of the phonemic category word recognizer was considerably improved during the program with respect to speaker-to-speaker variability. This gain was accomplished, however, with the additional expense of a speaker sample library. Such a library was shown to be easily generated and useful in adapting the recognizer to the speaker. Typical speaker performance for a 25-word library, utilizing the final form recognizer, is shown in Table 9. The results for a single speaker are analyzed in detail in the following subsections.

4.2 Training Library Data

The vocabulary utilized for this test consisted of 25 Fortran words. The vocabulary list was uttered three times, providing three samples of each vocabulary word. The "Evaluate" technique was used to test the samples and those which were confused with other vocable words were rejected and samples were then inserted in their place. Three iterations of the evaluate routine were required to optimize the library. The final 75-word sample library is shown in Table 10. The library ID number, word and category sequence are shown in this table.

4.3 Recognition Results

The recognition tests were carried out by uttering four samples for each of the 25 vocabulary words. Table 11 shows these results for both methods of calculating Q. The first group of these words shows the top three choices and their Q values for the first method of Q calculation. The recognized word is considered to be the word with the lowest Q value. The second group of words shows the results for the error analysis weighted Q value calculation method. Again, the lowest Q value word is the recognized word. The

Table 8

Final Recognition Results for Individual Speakers

<u>25-Word Vocabulary</u>

Speaker #1	85%
Speaker #2	92%
Speaker #3	91%
Speaker #4	72%
Average	85%

Table 9

Library Sequence
for 25-Word Vocabulary

WORD = 1 MINUS
6 13 9 7 13 9 14 1 12

WORD = 2 ONE
13 11 10 9 8 7 6 12

WORD = 3 COMPARE
4 5 13 9 10 6 5 4 13 8 9 12

WORD = 4 EQUAL
13 7 4 13 11 10 9 14 12

WORD = 5 ZERO
13 7 6 13 10 9 10 9 12

WORD = 6 TIMES
4 13 6 8 9 8 7 6 14 1 12

WORD = 7 DIVIDE
13 10 5 4 6 12

WORD = 8 SEVEN
14 1 13 8 7 5 13 8 10 11 6 12

WORD = 9 FALSE
14 2 13 10 4 14 1 12

WORD = 10 LOGICAL
13 11 10 9 8 9 10 5 13 7 4
13 10 12

WORD = 11 FOUR
13 11 10 9 12

WORD = 12 COMMA
4 13 9 11 6 13 9 12

WORD = 13 TAKE
4 13 8 7 4 14 2 12

WORD = 14 NINE
6 13 8 9 8 7 12

WORD = 15 ASSIGN
13 9 7 14 1 13 9 8 7 11 6 12

WORD = 16 OPTIONS
4 13 9 4 14 2 13 8 10 11 6 13 7 14 1 12

WORD = 17 SIX
14 1 13 8 4 14 1 12

WORD = 18 SLASH
14 1 13 10 9 8 7 10 8 4 14 2 12

WORD = 19 EIGHT
13 7 4 14 12

WORD = 20 RETURN
13 11 8 7 4 13 8 11 6 12

Table 9 (Continued)

WORD = 21 STOP
14 1 4 13 8 9 8 9 4 12

WORD = 22 INPUT
13 7 9 6 4 5 13 10 9 8 9 7 4 12

WORD = 23 FIVE
14 2 13 9 8 9 10 11 7 4

WORD = 24 LOCATE
13 11 6 13 9 10 11 4 5 13 7 4 14 12

WORD = 25 FORTRAN
14 2 13 10 9 10 8 4 13 8 7 9 6 12

WORD = 26 MINUS
13 8 7 10 7 9 14 1 12

WORD = 27 ONE
13 11 10 9 8 9 7 6 12

WORD = 28 COMPARE
4 5 13 9 10 8 6 4 13 8 9 10 13 12

WORD = 29 EQUAL
13 7 4 13 11 10 12

WORD = 30 ZERO
13 7 9 8 9 10 12

WORD = 31 TIMES
4 13 8 7 11 14 1 12

WORD = 32 DIVIDE
5 13 11 8 7 6 13 11 9 5 6 12

WORD = 33 SEVEN
44 1 13 8 7 11 5 6 13 10 8 10 11 6 12

WORD = 34 FALSE
14 2 13 10 14 1 12

WORD = 35 LOGICAL
13 11 10 9 8 9 5 13 7 4 5 13 9 10 9 12

WORD = 36 FOUR
14 13 10 9 12

WORD = 37 COMMA
4 13 9 8 9 12

WORD = 38 TAKE
4 13 8 7 4 14 2 12

WORD = 39 NINE
6 13 8 9 7 11 12

WORD = 40 ASSIGN
13 11 7 14 1 13 9 8 7 10 11 6 12

Table 9 (Continued)

WORD = 41 OPTIONS
5 13 9 4 14 2 13 8 9 10 11 6 14 1 12

WORD = 42 SIX
14 1 13 7 4 14 1 12

WORD = 43 SLASH
14 1 13 11 10 9 8 10 8 7 10 8 7 14 2 12

WORD = 44 EIGHT
5 13 7 4 14 2 12

WORD = 45 RETURN
13 8 7 4 13 8 10 11 6 12

WORD = 46 STOP
14 1 4 13 8 9 8 9 4 12

WORD = 47 INPUT
13 7 6 4 5 13 10 9 8 4 12

WORD = 48 FIVE
14 2 13 9 8 10 8 10 11 14 2 12

WORD = 49 LOCATE
6 13 9 10 4 5 13 8 7 4 14 12

WORD = 50 FORTRAN
14 13 10 9 10 4 13 8 7 10 11 12

WORD = 51 MINUS
6 13 9 8 7 6 13 8 10 7 14 1 12

WORD = 52 ONE
13 11 9 8 7 11 6 12

WORD = 53 COMPARE
4 13 8 6 4 13 8 9 12

WORD = 54 EQUAL
5 7 4 13 11 10 12

WORD = 55 ZERO
13 7 9 10 9 8 9 10 12

WORD = 56 TIMES
4 13 8 7 10 6 14 1 12

WORD = 57 DIVIDE
13 8 9 11 9 8 5 12

WORD = 58 SEVEN
14 1 13 8 6 5 13 8 10 11 6 12

WORD = 59 FALSE
14 2 13 10 7 14 1 12

WORD = 60 LOGICAL
13 9 8 5 6 13 7 8 5 13 9 12

Table 9 (Continued)

WORD = 61 FOUR
14 2 13 10 9 11 12

WORD = 62 COMMA
4 5 13 9 11 9 12

WORD = 63 TAKE
4 13 8 7 4 14 2 12

WORD = 64 NINE
6 13 8 9 8 7 6 12

WORD = 65 ASSIGN
4 14 1 13 4 8 7 6 12

WORD = 66 OPTIONS
4 13 9 8 9 4 14 2 13 8 10 11 6 14 1 12

WORD = 67 SIX
14 1 13 6 4 14 1 12

WORD = 68 SLASH
14 1 13 10 9 8 10 4 14 2 12

WORD = 69 EIGHT
13 7 4 14 2 4 12

WORD = 70 RETURN
13 11 7 4 13 8 11 6 12

WORD = 71 STOP
14 1 4 13 8 9 4 12

WORD = 72 INPUT
13 7 6 4 5 13 9 8 7 4 12

WORD = 73 FIVE
14 2 13 9 8 10 8 14 2 12

WORD = 74 LOCATE
13 9 10 4 5 13 8 7 4 12

WORD = 75 FORTRAN
14 2 13 10 9 4 13 8 7 10 11 6 12

Table 10

Match Scores for Four Test
Examples for Each Library Word

```
              MINUS                                       ONE

6  13  9  8  7  10  7  14  1  12         | 13  11  10  9  8  11  6  13  7  12
                                         |
WORD  =  26 MINUS          Q = 0.5714    | WORD  =   2 ONE            Q = 0.8333
WORD  =  56 TIMES          Q = 1.3333    | WORD  =  52 ONE            Q = 0.8333
WORD  =  31 TIMES          Q = 1.6000    | WORD  =  27 ONE            Q = 1.0000
                                         |
WORD  =  26 MINUS          Q = 0.4286    | WORD  =  12 COMMA          Q = 0.8750
WORD  =   1 MINUS          Q = 0.4444    | WORD  =   2 ONE            Q = 1.0000
WORD  =  56 TIMES          Q = 1.0000    | WORD  =  27 ONE            Q = 1.1667
                                         |
                                         |
13  8  7  10  8  7  10  11  8  14  1  12 | 13  11  10  9  7  8  12
                                         |
WORD  =  26 MINUS          Q = 0.7143    | WORD  =  11 FOUR           Q = 0.5000
WORD  =  31 TIMES          Q = 1.1667    | WORD  =   2 ONE            Q = 0.8000
WORD  =  43 SLASH          Q = 1.3333    | WORD  =  27 ONE            Q = 1.0000
                                         |
WORD  =  26 MINUS          Q = 0.7143    | WORD  =  11 FOUR           Q = 0.5000
WORD  =  31 TIMES          Q = 1.1667    | WORD  =   2 ONE            Q = 0.6000
WORD  =  56 TIMES          Q = 1.3333    | WORD  =  27 ONE            Q = 0.8000
                                         |
                                         |
6  13  9  8  7  10  5  14  1  12         | 13  11  10  9  8  6  13  7  4  12
                                         |
WORD  =  26 MINUS          Q = 1.1667    | WORD  =   2 ONE            Q = 1.0000
WORD  =  56 TIMES          Q = 1.5000    | WORD  =  10 LOGICAL        Q = 1.1250
WORD  =  31 TIMES          Q = 1.8000    | WORD  =  27 ONE            Q = 1.1667
                                         |
WORD  =   1 MINUS          Q = 0.4444    | WORD  =  10 LOGICAL        Q = 1.1250
WORD  =  26 MINUS          Q = 0.8333    | WORD  =   2 ONE            Q = 1.1667
WORD  =  56 TIMES          Q = 1.0000    | WORD  =  27 ONE            Q = 1.3333
                                         |
                                         |
6  13  9  8  7  10  5  14  1  12         | 13  11  10  9  8  9  6  12
                                         |
WORD  =  26 MINUS          Q = 1.1667    | WORD  =  27 ONE            Q = 0.1429
WORD  =  56 TIMES          Q = 1.5000    | WORD  =   2 ONE            Q = 0.3333
WORD  =  31 TIMES          Q = 1.8000    | WORD  =  52 ONE            Q = 0.8000
                                         |
WORD  =   1 MINUS          Q = 0.4444    | WORD  =  27 ONE            Q = 0.1000
WORD  =  26 MINUS          Q = 0.8333    | WORD  =   2 ONE            Q = 0.2222
WORD  =  56 TIMES          Q = 1.0000    | WORD  =  52 ONE            Q = 0.5000
```

Table 10 (Continued)

COMPARE	EQUAL
4 13 9 10 6 4 13 8 7 9 10 12	5 13 7 4 13 10 12
WORD = 28 COMPARE Q = 0.6000	WORD = 29 EQUAL Q = 0.6000
WORD = 3 COMPARE Q = 0.6667	WORD = 54 EQUAL Q = 0.6000
WORD = 53 COMPARE Q = 0.7143	WORD = 4 EQUAL Q = 1.2000
WORD = 28 COMPARE Q = 0.2308	WORD = 29 EQUAL Q = 0.4000
WORD = 3 COMPARE Q = 0.3333	WORD = 45 RETURN Q = 1.0000
WORD = 53 COMPARE Q = 0.5000	WORD = 54 EQUAL Q = 1.0000
4 13 9 10 6 4 13 7 9 10 8 12	13 7 4 13 10 9 12
WORD = 12 COMMA Q = 0.7143	WORD = 29 EQUAL Q = 0.4000
WORD = 53 COMPARE Q = 1.1667	WORD = 4 EQUAL Q = 0.5000
WORD = 28 COMPARE Q = 1.2500	WORD = 47 INPUT Q = 1.1667
WORD = 28 COMPARE Q = 0.6364	WORD = 29 EQUAL Q = 0.4000
WORD = 53 COMPARE Q = 0.7778	WORD = 4 EQUAL Q = 0.8333
WORD = 3 COMPARE Q = 0.8000	WORD = 47 INPUT Q = 0.8333
4 13 9 11 6 4 13 8 7 10 8 10 12	13 7 4 13 10 9 12
WORD = 12 COMMA Q = 1.3333	WORD = 29 EQUAL Q = 0.4000
WORD = 53 COMPARE Q = 1.3333	WORD = 4 EQUAL Q = 0.5000
WORD = 28 COMPARE Q = 1.3750	WORD = 47 INPUT Q = 1.1667
WORD = 28 COMPARE Q = 0.7273	WORD = 29 EQUAL Q = 0.4000
WORD = 53 COMPARE Q = 0.8889	WORD = 4 EQUAL Q = 0.8333
WORD = 3 COMPARE Q = 0.9000	WORD = 47 INPUT Q = 0.8333
4 13 9 11 8 6 4 13 8 7 10 8 14 12	13 7 4 13 10 12
WORD = 53 COMPARE Q = 1.1429	WORD = 29 EQUAL Q = 0.2000
WORD = 28 COMPARE Q = 1.2222	WORD = 4 EQUAL Q = 0.8000
WORD = 51 MINUS Q = 1.6250	WORD = 45 RETURN Q = 1.0000
WORD = 28 COMPARE Q = 0.8333	WORD = 29 EQUAL Q = 0.2000
WORD = 53 COMPARE Q = 1.0000	WORD = 45 RETURN Q = 0.8000
WORD = 3 COMPARE Q = 1.3000	WORD = 4 EQUAL Q = 1.2000

Table 10 (Continued)

ZERO		TIMES	
13 7 10 8 9 8 9 10 7 12		4 13 9 8 11 6 14 1 12	
WORD = 55 ZERO	Q = 0.8333	WORD = 31 TIMES	Q = 0.6667
WORD = 30 ZERO	Q = 1.0000	WORD = 56 TIMES	Q = 0.6667
WORD = 14 NINE	Q = 1.2000	WORD = 6 TIMES	Q = 0.8333
WORD = 55 ZERO	Q = 0.8333	WORD = 56 TIMES	Q = 0.4444
WORD = 14 NINE	Q = 1.0000	WORD = 31 TIMES	Q = 0.5000
WORD = 30 ZERO	Q = 1.0000	WORD = 6 TIMES	Q = 0.5556
13 7 9 8 9 10 12		4 13 8 11 14 1 12	
WORD = 30 ZERO	Q = 0.0000	WORD = 31 TIMES	Q = 0.1667
WORD = 55 ZERO	Q = 0.3333	WORD = 56 TIMES	Q = 1.0000
WORD = 27 ONE	Q = 1.0000	WORD = 6 TIMES	Q = 1.2000
WORD = 30 ZERO	Q = 0.0000	WORD = 31 TIMES	Q = 0.1667
WORD = 55 ZERO	Q = 0.3333	WORD = 56 TIMES	Q = 0.8000
WORD = 27 ONE	Q = 0.8000	WORD = 6 TIMES	Q = 1.0000
13 7 9 8 9 10 12		4 13 9 8 11 14 1 12	
WORD = 30 ZERO	Q = 0.0000	WORD = 31 TIMES	Q = 0.3333
WORD = 55 ZERO	Q = 0.3333	WORD = 56 TIMES	Q = 1.2000
WORD = 27 ONE	Q = 1.0000	WORD = 6 TIMES	Q = 1.4000
WORD = 30 ZERO	Q = 0.0000	WORD = 31 TIMES	Q = 0.3333
WORD = 55 ZERO	Q = 0.3333	WORD = 56 TIMES	Q = 1.0000
WORD = 27 ONE	Q = 0.8000	WORD = 6 TIMES	Q = 1.2000
13 7 9 8 10 9 10 12		4 13 9 8 11 6 14 1 12	
WORD = 30 ZERO	Q = 0.1667	WORD = 31 TIMES	Q = 0.6667
WORD = 55 ZERO	Q = 0.5000	WORD = 56 TIMES	Q = 0.6667
WORD = 27 ONE	Q = 1.2000	WORD = 6 TIMES	Q = 0.8333
WORD = 30 ZERO	Q = 0.1667	WORD = 56 TIMES	Q = 0.4444
WORD = 55 ZERO	Q = 0.5000	WORD = 31 TIMES	Q = 0.5000
WORD = 27 ONE	Q = 1.0000	WORD = 6 TIMES	Q = 0.5556

Table 10 (Continued)

```
               DIVIDE                                             SEVEN

5 13 8 10 11 7 6 13 10 9 8 5 13 7 4 12 | 14  1  13  8  5  13  10  11  8  7  4  12

WORD  =  24 LOCATE          Q = 1.4444 | WORD  =   8 SEVEN         Q = 1.0000
WORD  =  32 DIVIDE          Q = 1.6250 | WORD  =  58 SEVEN         Q = 1.1250
WORD  =  28 COMPARE         Q = 2.0000 | WORD  =  33 SEVEN         Q = 1.5000
                                       |
WORD  =  32 DIVIDE          Q = 1.5000 | WORD  =   8 SEVEN         Q = 0.8750
WORD  =   5 ZERO            Q = 1.5556 | WORD  =  58 SEVEN         Q = 0.8750
WORD  =  51 MINUS           Q = 2.0000 | WORD  =  21 STOP          Q = 1.8333
                                       |
                                       |
13 8 10 11 6 13 8 10 9 8 7 5 6 12      | 14 1 13 8 10 11 5 13 10 11 13 10 8 12
                                       |
WORD  =  32 DIVIDE          Q = 1.5714 | WORD  =   8 SEVEN         Q = 1.2500
WORD  =  58 SEVEN           Q = 2.0000 | WORD  =  33 SEVEN         Q = 1.3333
WORD  =  40 ASSIGN          Q = 2.1429 | WORD  =  58 SEVEN         Q = 1.3750
                                       |
WORD  =  32 DIVIDE          Q = 0.7692 | WORD  =   8 SEVEN         Q = 0.8750
WORD  =   2 ONE             Q = 0.8000 | WORD  =  58 SEVEN         Q = 0.8750
WORD  =  58 SEVEN           Q = 1.0769 | WORD  =  33 SEVEN         Q = 1.6250
                                       |
                                       |
13  8  7  9  8  5  4  12               | 14  1  13  8  9  5  13  10  11  12
                                       |
WORD  =  57 DIVIDE          Q = 1.0000 | WORD  =   8 SEVEN         Q = 0.6250
WORD  =  21 STOP            Q = 2.0000 | WORD  =  58 SEVEN         Q = 0.7500
WORD  =  46 STOP            Q = 2.0000 | WORD  =  33 SEVEN         Q = 1.1250
                                       |
WORD  =  57 DIVIDE          Q = 1.0000 | WORD  =   8 SEVEN         Q = 0.5000
WORD  =  14 NINE            Q = 1.5000 | WORD  =  58 SEVEN         Q = 0.5000
WORD  =  64 NINE            Q = 1.7500 | WORD  =  71 STOP          Q = 2.0000
                                       |
                                       |
13 8 10 9 7 5 13 11 10 9 8 9 7 4 12    | 14  1  13  8  10  5  13  10  11  12
                                       |
WORD  =  22 INPUT           Q = 0.9000 | WORD  =   8 SEVEN         Q = 0.6250
WORD  =  47 INPUT           Q = 1.2500 | WORD  =  58 SEVEN         Q = 0.7500
WORD  =  72 INPUT           Q = 1.2500 | WORD  =  33 SEVEN         Q = 1.1250
                                       |
WORD  =  27 ONE             Q = 1.5714 | WORD  =   8 SEVEN         Q = 0.5000
WORD  =  55 ZERO            Q = 1.5714 | WORD  =  58 SEVEN         Q = 0.5000
WORD  =   5 ZERO            Q = 1.8333 | WORD  =  48 FIVE          Q = 2.6667
```

Table 10 (Continued)

FALSE	LOGICAL
14 2 13 10 9 14 1 12	13 9 8 5 13 7 4 5 13 10 12
WORD = 34 FALSE Q = 0.1667	WORD = 35 LOGICAL Q = 0.5000
WORD = 59 FALSE Q = 0.3333	WORD = 10 LOGICAL Q = 0.6667
WORD = 9 FALSE Q = 0.5000	WORD = 60 LOGICAL Q = 0.8750
WORD = 34 FALSE Q = 0.1667	WORD = 35 LOGICAL Q = 0.5000
WORD = 59 FALSE Q = 0.3333	WORD = 10 LOGICAL Q = 0.5556
WORD = 9 FALSE Q = 0.5000	WORD = 45 RETURN Q = 1.5000
14 2 13 10 14 1 12	13 10 9 8 5 4 13 8 7 4 5 13 10 7 12
WORD = 34 FALSE Q = 0.0000	WORD = 35 LOGICAL Q = 0.7273
WORD = 59 FALSE Q = 0.1667	WORD = 10 LOGICAL Q = 0.9000
WORD = 9 FALSE Q = 0.3333	WORD = 60 LOGICAL Q = 1.5000
WORD = 34 FALSE Q = 0.0000	WORD = 74 LOCATE Q = 2.1429
WORD = 59 FALSE Q = 0.1667	WORD = 25 FORTRAN Q = 2.3750
WORD = 9 FALSE Q = 0.3333	WORD = 22 INPUT Q = 2.5714
14 2 13 10 14 1 12	13 10 9 8 5 13 7 4 5 13 1
WORD = 34 FALSE Q = 0.0000	WORD = 35 LOGICAL Q = 0.3636
WORD = 59 FALSE Q = 0.1667	WORD = 10 LOGICAL Q = 0.5000
WORD = 9 FALSE Q = 0.3333	WORD = 60 LOGICAL Q = 1.0000
WORD = 34 FALSE Q = 0.0000	WORD = 35 LOGICAL Q = 0.3636
WORD = 59 FALSE Q = 0.1667	WORD = 10 LOGICAL Q = 0.4000
WORD = 9 FALSE Q = 0.3333	WORD = 45 RETURN Q = 1.6667
14 2 13 10 7 14 1 12	13 10 9 8 5 13 8 4 13 9 10 7 12
WORD = 59 FALSE Q = 0.0000	WORD = 10 LOGICAL Q = 0.7778
WORD = 34 FALSE Q = 0.1667	WORD = 35 LOGICAL Q = 0.8000
WORD = 9 FALSE Q = 0.5000	WORD = 60 LOGICAL Q = 1.2500
WORD = 59 FALSE Q = 0.0000	WORD = 35 LOGICAL Q = 0.7000
WORD = 34 FALSE Q = 0.1667	WORD = 10 LOGICAL Q = 0.7778
WORD = 9 FALSE Q = 0.5000	WORD = 28 COMPARE Q = 1.8571

Table 10 (Continued)

FOUR	COMMA
14 13 11 10 9 8 12	4 13 9 8 11 5 12
WORD = 36 FOUR Q = 0.5000	WORD = 37 COMMA Q = 1.0000
WORD = 11 FOUR Q = 0.7500	WORD = 57 DIVIDE Q = 1.5000
WORD = 2 ONE Q = 1.0000	WORD = 62 COMMA Q = 1.5000
WORD = 36 FOUR Q = 0.5000	WORD = 37 COMMA Q = 0.7500
WORD = 2 ONE Q = 1.2000	WORD = 62 COMMA Q = 1.0000
WORD = 11 FOUR Q = 1.2500	WORE = 52 ONE Q = 1.5000
14 2 13 10 9 8 12	4 13 9 11 8 6 13 9 12
WORD = 61 FOUR Q = 0.4000	WORD = 12 COMMA Q = 0.1429
WORD = 36 FOUR Q = 0.7500	WORD = 53 COMPARE Q = 0.8333
WORD = 73 FIVE Q = 1.4000	WORD = 28 COMPARE Q = 1.4286
WORD = 61 FOUR Q = 0.4000	WORD = 12 COMMA Q = 0.1000
WORD = 36 FOUR Q = 0.7500	WORD = 32 DIVIDE Q = 0.8889
WORD = 73 FIVE Q = 1.8000	WORD = 37 COMMA Q = 1.0000
14 2 13 11 10 11 9 12	4 13 8 9 11 6 13 9 12
WORD = 61 FOUR Q = 0.6000	WORD = 12 COMMA Q = 0.1429
WORD = 36 FOUR Q = 1.0000	WORD = 53 COMPARE Q = 0.8333
WORD = 11 FOUR Q = 1.2500	WORD = 3 COMPARE Q = 1.6667
WORD = 61 FOUR Q = 0.6000	WORD = 12 COMMA Q = 0.1000
WORD = 36 FOUR Q = 1.0000	WORD = 62 COMMA Q = 1.2000
WORD = 11 FOUR Q = 1.7500	WORD = 32 DIVIDE Q = 1.2500
14 2 13 9 10 11 8 12	4 13 9 6 13 9 10 12
WORD = 61 FOUR Q = 0.6000	WORD = 12 COMMA Q = 0.3333
WORD = 73 FIVE Q = 1.0000	WORD = 53 COMPARE Q = 1.2000
WORD = 48 FIVE Q = 1.3333	WORD = 28 COMPARE Q = 1.2857
WORD = 61 FOUR Q = 0.6000	WORD = 12 COMMA Q = 0.2222
WORD = 73 FIVE Q = 1.3333	WORD = 5 ZERO Q = 1.1429
WORD = 48 FIVE Q = 1.6667	WORD = 37 COMMA Q = 1.5000

Table 10 (Continued)

TAKE	NINE
4 13 8 7 4 14 2 4 12	6 13 8 7 12
WORD = 13 TAKE Q = 0.2857	WORD = 14 NINE Q = 0.5000
WORD = 38 TAKE Q = 0.2857	WORD = 39 NINE Q = 0.5000
WORD = 63 TAKE Q = 0.2857	WORD = 64 NINE Q = 1.0000
WORD = 13 TAKE Q = 0.2857	WORD = 14 NINE Q = 0.2857
WORD = 38 TAKE Q = 0.2857	WORD = 39 NINE Q = 0.2857
WORD = 63 TAKE Q = 0.2857	WORD = 64 NINE Q = 0.4286
4 13 8 7 4 14 2 4 12	6 13 8 9 8 7 6 12
WORD = 13 TAKE Q = 0.2857	WORD = 64 NINE Q = 0.0000
WORD = 38 TAKE Q = 0.2857	WORD = 14 NINE Q = 0.3333
WORD = 63 TAKE Q = 0.2857	WORD = 39 NINE Q = 0.8000
WORD = 13 TAKE Q = 0.2857	WORD = 64 NINE Q = 0.0000
WORD = 38 TAKE Q = 0.2857	WORD = 14 NINE Q = 0.1111
WORD = 63 TAKE Q = 0.2857	WORD = 39 NINE Q - 0.3750
4 13 8 7 4 14 12	6 13 8 9 8 7 12
WORD = 13 TAKE Q = 0.3333	WORD = 14 NINE Q = 0.0000
WORD = 38 TAKE Q = 0.3333	WORD = 64 NINE Q = 0.3333
WORD = 63 TAKE Q = 0.3333	WORD = 39 NINE Q = 0.4000
WORD = 13 TAKE Q = 0.3333	WORD = 14 NINE Q = 0.0000
WORD = 38 TAKE Q = 0.3333	WORD = 64 NINE Q = 0.1111
WORD = 63 TAKE Q = 0.3333	WORD = 39 NINE Q = 0.2500
4 13 8 7 4 14 2 12	13 7 9 8 7 10 12
WORD = 13 TAKE Q = 0.0000	WORD = 30 ZERO Q = 0.4000
WORD = 38 TAKE Q = 0.0000	WORD = 55 ZERO Q = 0.8000
WORD = 63 TAKE Q = 0.0000	WORD = 26 MINUS Q = 1.2000
WORD = 13 TAKE Q = 0.0000	WORD = 30 ZERO Q = 0.4000
WORD = 38 TAKE Q = 0.0000	WORD = 55 ZERO Q = 0.8000
WORD = 63 TAKE Q = 0.0000	WORD = 2 ONE Q = 1.2500

Table 10 (Continued)

```
              ASSIGN                                      OPTIONS

13  11  7  14  1  13  8  9  8  7  12      |  4  13  8  9  4  14  2  13  8  11  14  1  12

WORD  =  40 ASSIGN        Q = 0.8750      |  WORD  =  66 OPTIONS       Q = 0.3333
WORD  =  15 ASSIGN        Q = 0.1429      |  WORD  =  41 OPTIONS       Q = 0.9000
WORD  =  72 INPUT         Q = 2.3333      |  WORD  =  13 TAKE          Q = 1.6667

WORD  =  40 ASSIGN        Q = 0.7500      |  WORD  =  66 OPTIONS       Q = 0.2500
WORD  =  14 ASSIGN        Q = 1.0000      |  WORD  =  41 OPTIONS       Q = 0.7000
WORD  =  72 INPUT         Q = 2.3333      |  WORD  =  13 TAKE          Q = 2.3333


13  7  14  1  13  9  8  7  12             |  4 13 8 10 4 14 2 13 8 11 6 14 1 12

WORD  =  15 ASSIGN        Q = 0.5000      |  WORD  =  66 OPTIONS       Q = 0.3333
WORD  =  40 ASSIGN        Q = 0.6250      |  WORD  =  41 OPTIONS       Q = 0.9000
WORD  =  72 INPUT         Q = 2.0000      |  WORD  =  16 OPTIONS       Q = 1.6667

WORD  =  15 ASSIGN        Q = 0.3750      |  WORD  =  66 OPTIONS       Q = 0.2667
WORD  =  40 ASSIGN        Q = 0.5000      |  WORD  =  41 OPTIONS       Q = 0.6154
WORD  =  72 INPUT         Q = 2.0000      |  WORD  =  20 RETURN        Q = 1.7000


13  7  14  1  13  9  8  7  10  6  12      |  4 13 9 4 14 2 13 8 10 11 6 14 1 12

WORD  =  40 ASSIGN        Q = 0.2000      |  WORD  =  66 OPTIONS       Q = 0.1538
WORD  =  15 ASSIGN        Q = 0.3333      |  WORD  =  41 OPTIONS       Q = 0.4167
WORD  =  45 RETURN        Q = 1.6667      |  WORD  =  16 OPTIONS       Q = 1.0000

WORD  =  40 ASSIGN        Q = 0.1538      |  WORD  =  66 OPTIONS       Q = 0.1250
WORD  =  15 ASSIGN        Q = 0.2500      |  WORD  =  41 OPTIONS       Q = 0.2667
WORD  =  45 RETURN        Q = 1.3333      |  WORD  =  45 RETURN        Q = 1.7000


13  10  14  1  13  9  8 10 11 8 7 10 12   |  4 13 8 13 4 14 2 13 8 10 11 6 14 1 12

WORD  =  40 ASSIGN        Q = 1.1250      |  WORD  =  10 LOGICAL       Q = 3.6667
WORD  =  15 ASSIGN        Q = 1.4286      |  WORD  =  16 OPTIONS       Q = 4.1667
WORD  =  50 FORTRAN       Q = 2.5000      |  WORD  =  35 LOGICAL       Q = 4.1667

WORD  =  40 ASSIGN        Q = 1.0000      |  WORD  =  66 OPTIONS       Q = 0.1250
WORD  =  15 ASSIGN        Q = 1.2857      |  WORD  =  41 OPTIONS       Q = 0.4286
WORD  =  50 FORTRAN       Q = 3.1667      |  WORD  =  45 RETURN        Q = 1.3636
```

Table 10 (Continued)

```
                SIX                                          SLASH

14  1  13  9  8  4  14  1  12               | 14 1 13 11 10 9 8 7 10 8 7 4 14 2 12
                                            |
WORD  =  17 SIX              Q = 0.1429     | WORD  =  18 SLASH          Q = 1.6667
WORD  =  67 SIX              Q = 0.1429     | WORD  =  43 SLASH          Q = 0.3077
WORD  =  42 SIX              Q = 0.5000     | WORD  =  68 SLASH          Q = 0.4000
                                            |
WORD  =  17 SIX              Q = 0.1429     | WORD  =  18 SLASH          Q = 0.1667
WORD  =  67 SIX              Q = 0.1429     | WORD  =  43 SLASH          Q = 0.3077
WORD  =  42 SIX              Q = 0.5000     | WORD  =  68 SLASH          Q = 0.4000
                                            |
                                            |
14  1  13  8  7  4  14  1  12               | 14  1  13  11  9  88 10  11 14  2  12
                                            |
WORD  =  17 SIX              Q = 0.1429     | WORD  =  48 FIVE           Q = 0.8750
WORD  =  42 SIX              Q = 0.1429     | WORD  =  68 SLASH          Q = 1.0000
WORD  =  67 SIX              Q = 0.1429     | WORD  =  73 FIVE           Q = 1.0000
                                            |
WORD  =  17 SIX              Q = 0.1429     | WORD  =  48 FIVE           Q = 0.8750
WORD  =  42 SIX              Q = 0.1429     | WORD  =  68 SLASH          Q = 1.0000
WORD  =  67 SIX              Q = 0.1429     | WORD  =  73 FIVE           Q = 1.0000
                                            |
                                            |
14  1  13  9  8  4  14  1  12               | 14 1 13 10 9 8 10 11 7 4 14 2 12
                                            |
WORD  =  17 SIX              Q = 0.1429     | WORD  =  68 SLASH          Q = 0.2000
WORD  =  67 SIX              Q = 0.1429     | WORD  =  18 SLASH          Q = 0.4000
WORD  =  42 SIX              Q = 0.5000     | WORD  =  23 FIVE           Q = 0.8000
                                            |
WORD  =  17 SIX              Q = 0.1429     | WORD  =  68 SLASH          Q = 0.2000
WORD  =  67 SIX              Q = 0.1429     | WORD  =  18 SLASH          Q = 0.4000
WORD  =  42 SIX              Q = 0.5000     | WORD  =  23 FIVE           Q = 0.8000
                                            |
                                            |
4  14  1  13  8  4  14  1  12               | 14 1 13 11 10 9 8 7 10 7 4 14 2 12
                                            |
WORD  =  17 SIX              Q = 0.2857     | WORD  =  18 SLASH          Q = 0.2727
WORD  =  67 SIX              Q = 0.2857     | WORD  =  68 SLASH          Q = 0.3000
WORD  =  42 SIX              Q = 0.6667     | WORD  =  43 SLASH          Q = 0.6364
                                            |
WORD  =  17 SIX              Q = 0.2857     | WORD  =  18 SLASH          Q = 0.2727
WORD  =  67 SIX              Q = 0.2857     | WORD  =  68 SLASH          Q = 0.3000
WORD  =  42 SIX              Q = 0.6667     | WORD  =  43 SLASH          Q = 0.6364
```

Table 10 (Continued)

EIGHT		RETURN	
13 7 4 14 2 12		13 8 7 4 13 8 10 11 12	
WORD = 44 EIGHT	Q = 0.4000	WORD = 45 RETURN	Q = 0.2500
WORD = 69 EIGHT	Q = 0.4000	WORD = 20 RETURN	Q = 0.5714
WORD = 19 EIGHT	Q = 0.5000	WORD = 29 EQUAL	Q = 0.8000
WORD = 44 EIGHT	Q = 0.2000	WORD = 45 RETURN	Q = 0.1250
WORD = 69 EIGHT	Q = 0.4000	WORD = 20 RETURN	Q = 0.4286
WORD = 19 EIGHT	Q = 0.5000	WORD = 70 RETURN	Q = 0.6667
13 8 7 4 14 1 12		13 11 8 7 4 13 8 11 12	
WORD = 19 EIGHT	Q = 0.7500	WORD = 20 RETURN	Q = 0.2500
WORD = 17 SIX	Q = 1.0000	WORD = 70 RETURN	Q = 0.4286
WORD = 26 MINUS	Q = 1.0000	WORD = 45 RETURN	Q = 0.5714
WORD = 19 EIGHT	Q = 0.7500	WORD = 20 RETURN	Q = 0.1250
WORD = 26 MINUS	Q = 1.0000	WORD = 70 RETURN	Q = 0.2857
WORD = 31 TIMES	Q = 1.0000	WORD = 45 RETURN	Q = 0.4286
13 7 4 14 1 12		13 8 7 4 13 8 11 12	
WORD = 19 EIGHT	Q = 0.5000	WORD = 20 RETURN	Q = 0.4286
WORD = 42 SIX	Q = 0.8000	WORD = 45 RETURN	Q = 0.4286
WORD = 9 FALSE	Q = 1.5000	WORD = 29 EQUAL	Q = 0.6000
WORD = 19 EIGHT	Q = 0.5000	WORD = 20 RETURN	Q = 0.2857
WORD = 42 SIX	Q = 1.2000	WORD = 45 RETURN	Q = 0.2857
WORD = 1 MINUS	Q = 1.2500	WORD = 70 EQUAL	Q = 0.5000
13 7 4 14 4 12		13 8 7 4 13 8 10 13 7 12	
WORD = 69 EIGHT	Q = 0.4000	WORD = 45 RETURN	Q = 0.8571
WORD = 19 EIGHT	Q = 0.5000	WORD = 29 EQUAL	Q = 1.2000
WORD = 44 EIGHT	Q = 1.5000	WORD = 20 RETURN	Q = 1.3333
WORD = 69 EIGHT	Q = 0.4000	WORD = 45 RETURN	Q = 0.4286
WORD = 19 EIGHT	Q = 0.5000	WORD = 29 EQUAL	Q = 0.8000
WORD = 44 EIGHT	Q = 1.2500	WORD = 20 RETURN	Q = 0.8333

Table 10 (Continued)

STOP	FIVE
14 1 4 13 9 4 12	14 13 9 8 7 11 14 2 12
WORD = 71 STOP Q = 0.1667	WORD = 48 FIVE Q = 0.8571
WORD = 21 STOP Q = 0.5000	WORD = 73 FIVE Q = 1.0000
WORD = 46 STOP Q = 0.5000	WORD = 18 SLASH Q = 1.1429
WORD = 71 STOP Q = 0.1667	WORD = 48 FIVE Q = 0.8571
WORD = 21 STOP Q = 0.5000	WORD = 73 FIVE Q = 1.0000
WORD = 46 STOP Q = 0.5000	WORD = 18 SLASH Q = 1.1429
14 1 4 13 9 4 12	13 13 9 8 9 8 7 14 2 12
WORD = 71 STOP Q = 0.1667	WORD = 73 FIVE Q = 0.7143
WORD = 21 STOP Q = 0.5000	WORD = 48 FIVE Q = 1.0000
WORD = 46 STOP Q = 0.5000	WORD = 23 FIVE Q = 1.1250
WORD = 71 STOP Q = 0.1667	WORD = 73 FIVE Q = 0.7143
WORD = 21 STOP Q = 0.5000	WORD = 48 FIVE Q = 1.0000
WORD = 46 STOP Q = 0.5000	WORD = 23 FIVE Q = 1.1250
14 1 4 13 8 4 12	14 2 13 9 8 7 10 8 11 7 14 2 12
WORD = 71 STOP Q = 0.1667	WORD = 48 FIVE Q = 0.3000
WORD = 21 STOP Q = 0.5000	WORD = 73 FIVE Q = 0.3333
WORD = 46 STOP Q = 0.5000	WORD = 23 FIVE Q = 0.7000
WORD = 71 STOP Q = 0.1667	WORD = 48 FIVE Q = 0.3000
WORD = 21 STOP Q = 0.5000	WORD = 73 FIVE Q = 0.3333
WORD = 46 STOP Q = 0.5000	WORD = 23 FIVE Q = 0.7000
14 1 4 13 8 7 9 4 12	14 2 13 9 8 10 11 14 2 12
WORD = 71 STOP Q = 0.1429	WORD = 48 FIVE Q = 0.2222
WORD = 21 STOP Q = 0.4286	WORD = 73 FIVE Q = 0.2500
WORD = 46 STOP Q = 0.4286	WORD = 23 FIVE Q = 0.6667
WORD = 71 STOP Q = 0.1429	WORD = 48 FIVE Q = 0.2222
WORD = 21 STOP Q = 0.4286	WORD = 73 FIVE Q = 0.2500
WORD = 46 STOP Q = 0.4286	WORD = 23 FIVE Q = 0.6667

Table 10 (Continued)

```
            LOCATE                                        FORTRAN

5  6  13  9  11  4  13  8  7  4  12           | 14  2  13  10  9  8  4  13  8  7  10 12

WORD  =  74 LOCATE            Q = 0.4444      | WORD  =  75 FORTRAN        Q = 0.4000
WORD  =  49 LOCATE            Q = 1.0000      | WORD  =  25 FORTRAN        Q = 0.5000
WORD  =  22 INPUT             Q = 1.8571      | WORD  =  50 FORTRAN        Q = 0.5556

WORD  =  74 LOCATE            Q = 0.1667      | WORD  =  75 FORTRAN        Q = 0.3000
WORD  =  49 LOCATE            Q = 0.7273      | WORD  =  25 FORTRAN        Q = 0.4000
WORD  =  22 INPUT             Q = 1.2857      | WORD  =  50 FORTRAN        Q = 0.5556


6  13  9  10  4  5  13  8  7  4  12           | 13  10  9  10  4  13  8  7  10 11 6 12

WORD  =  49 LOCATE            Q = 0.2000      | WORD  =  50 FORTRAN        Q = 0.4000
WORD  =  74 LOCATE            Q = 0.2222      | WORD  =  74 FORTRAN        Q = 0.5000
WORD  =  22 INPUT             Q = 1.1250      | WORD  =  45 RETURN         Q = 0.8571

WORD  =  74 LOCATE            Q = 0.1667      | WORD  =  50 FORTRAN        Q = 0.5000
WORD  =  49 LOCATE            Q = 0.3077      | WORD  =  75 FORTRAN        Q = 0.5385
WORD  =  72 INPUT             Q = 0.8571      | WORD  =  45 RETURN         Q = 0.6000


6  13  9  10  4  5  13  8  7  4  12           | 13  10  9  10  8  4  13  8  7  10  6 12

WORD  =  49 LOCATE            Q = 0.2000      | WORD  =  25 FORTRAN        Q = 0.6000
WORD  =  74 LOCATE            Q = 0.2222      | WORD  =  50 FORTRAN        Q = 0.6667
WORD  =  22 INPUT             Q = 1.1250      | WORD  =  75 FORTRAN        Q = 0.7778

WORD  =  74 LOCATE            Q = 0.1667      | WORD  =  45 RETURN         Q = 0.6000
WORD  =  49 LOCATE            Q = 0.3077      | WORD  =  25 FORTRAN        Q = 0.6154
WORD  =  72 INPUT             Q = 0.8571      | WORD  =  75 FORTRAN        Q = 0.7500


13  7  10  9  10  4  5  13  8  7  4  12       | 14 2 13 10 9 8 4 13 8 7 10 6 12

WORD  =  49 LOCATE            Q = 0.6667      | WORD  =  75 FORTRAN        Q = 0.1818
WORD  =  72 INPUT             Q = 0.7500      | WORD  =  25 FORTRAN        Q = 0.2727
WORD  =  74 LOCATE            Q = 0.7500      | WORD  =  50 FORTRAN        Q = 0.7778

WORD  =  72 INPUT             Q = 0.6250      | WORD  =  75 FORTRAN        Q = 0.1429
WORD  =  74 LOCATE            Q = 0.6250      | WORD  =  25 FORTRAN        Q = 0.2143
WORD  =  22 INPUT             Q = 0.6667      | WORD  =  50 FORTRAN        Q = 0.6667
```

Table 10 (Continued)

INPUT

```
13  7  6  4  13  9  8  10  7  4

WORD  =  72 INPUT                Q = 0.3333
WORD  =  47 INPUT                Q = 0.6250
WORD  =  22 INPUT                Q = 0.6667

WORD  =  72 INPUT                Q = 0.1667
WORD  =  47 INPUT                Q = 0.3636
WORD  =  22 INPUT                Q = 0.4167


13  7  6  4  5  13  10  9  8  4

WORD  =  47 INPUT                Q = 0.0000
WORD  =  72 INPUT                Q = 0.2222
WORD  =  22 INPUT                Q = 0.3000

WORD  =  47 INPUT                Q = 0.0000
WORD  =  72 INPUT                Q = 0.1667
WORD  =  22 INPUT                Q = 0.2308


13  7  9  8  6  4  5  13  11  9  10  7  4  12

WORD  =  72 INPUT                Q = 0.5556
WORD  =  22 INPUT                Q = 0.6000
WORD  =  47 INPUT                Q = 0.8750

WORD  =  72 INPUT                Q = 0.4167
WORD  =  22 INPUT                Q = 0.4615
WORD  =  47 INPUT                Q = 0.6364


13  7  6  4  5  13  10  9  8  7  4  12

WORD  =  47 INPUT                Q = 0.1000
WORD  =  72 INPUT                Q = 0.1000
WORD  =  22 INPUT                Q = 0.1818

WORD  =  47 INPUT                Q = 0.0769
WORD  =  72 INPUT                Q = 0.0769
WORD  =  22 INPUT                Q = 0.1429
```

detected category sequence is also shown for each test word just before the Q values.

The raw recognition rates for the 100 test utterances were 94% recognition and 6% errors for Q option #1, and 92% recognition and 8% errors for Q option #2. This data does not make use of the threshold value provided in the program. When this provision is used, an advantage appears in the use of the second method of Q calculation. Figures 11 and 12 shows the effect of varying the threshold value. The figures show the percentage of recognition error and non-recognition rate for various values of threshold applied to the match Q value. By comparing the two curves, it can be seen that if it is desired to hold the error rate to 2%, for example, then option #1 has a recognition rate of 59%, a non-recognition rate of 39%, and an error rate of 2%. Option #2, however, has a recognition rate of 75%, a non-recognition rate of 23%, and an error rate of 2%. Thus, obvious improvements are available by using the nasal string and error analysis weighted Q calculation provided for by option #2.

The relative importance of the nasal category errors can be seen in Figures 13 and 14. Here, the 25 word vocabulary data has been reworked to eliminate vocabulary words containing nasals. The results have changed considerably. The unthresholded recognition rates are now 98% for both option 1 and option 2 Q calculation. There still appears to be an advantage with option 2 when the Q threshold is applied as shown in Figures 13 and 14; however, this is due only to the advantages of error analysis weighting of Q calculation.

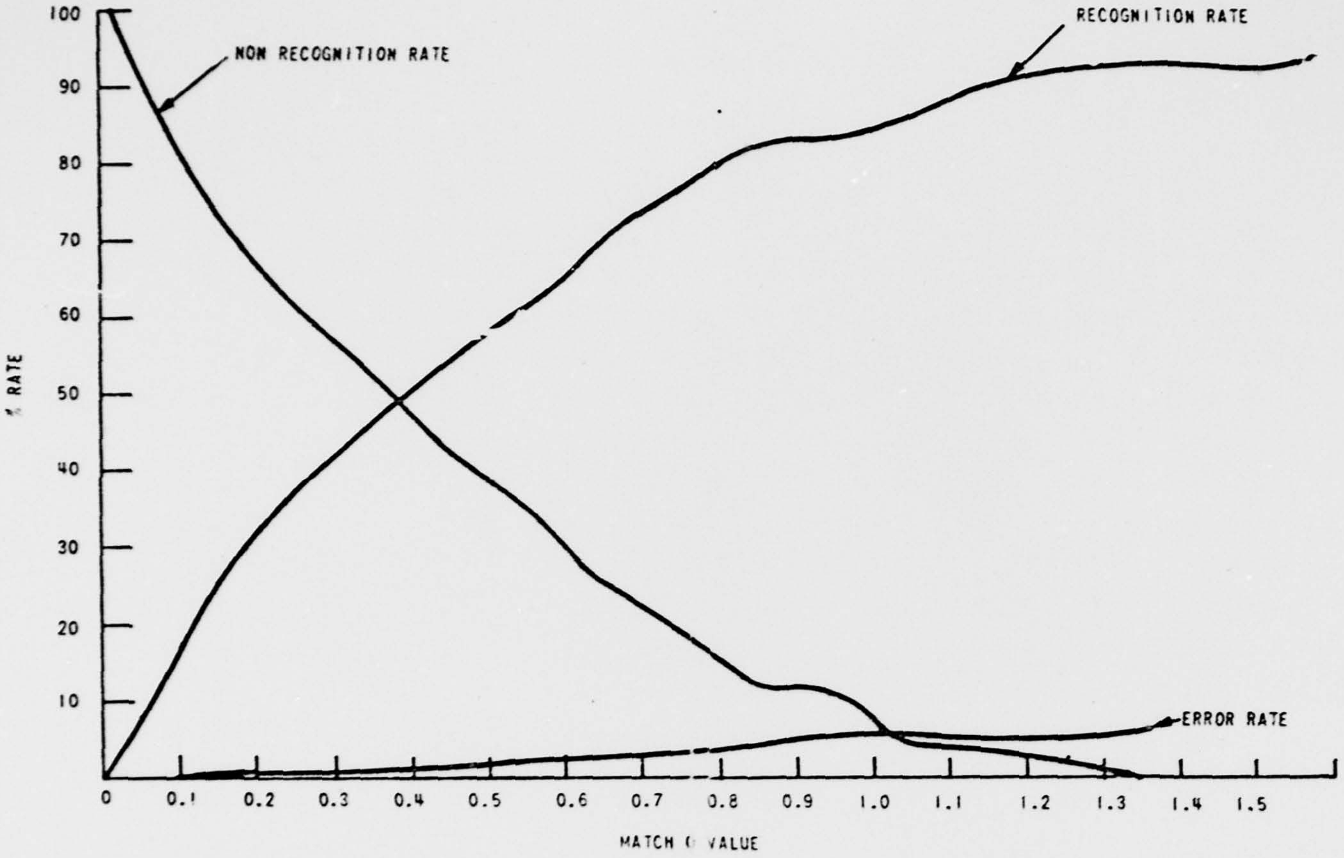


Figure 11

Recognition Rate No Nasal Interpolation - 25-Word Vocabulary

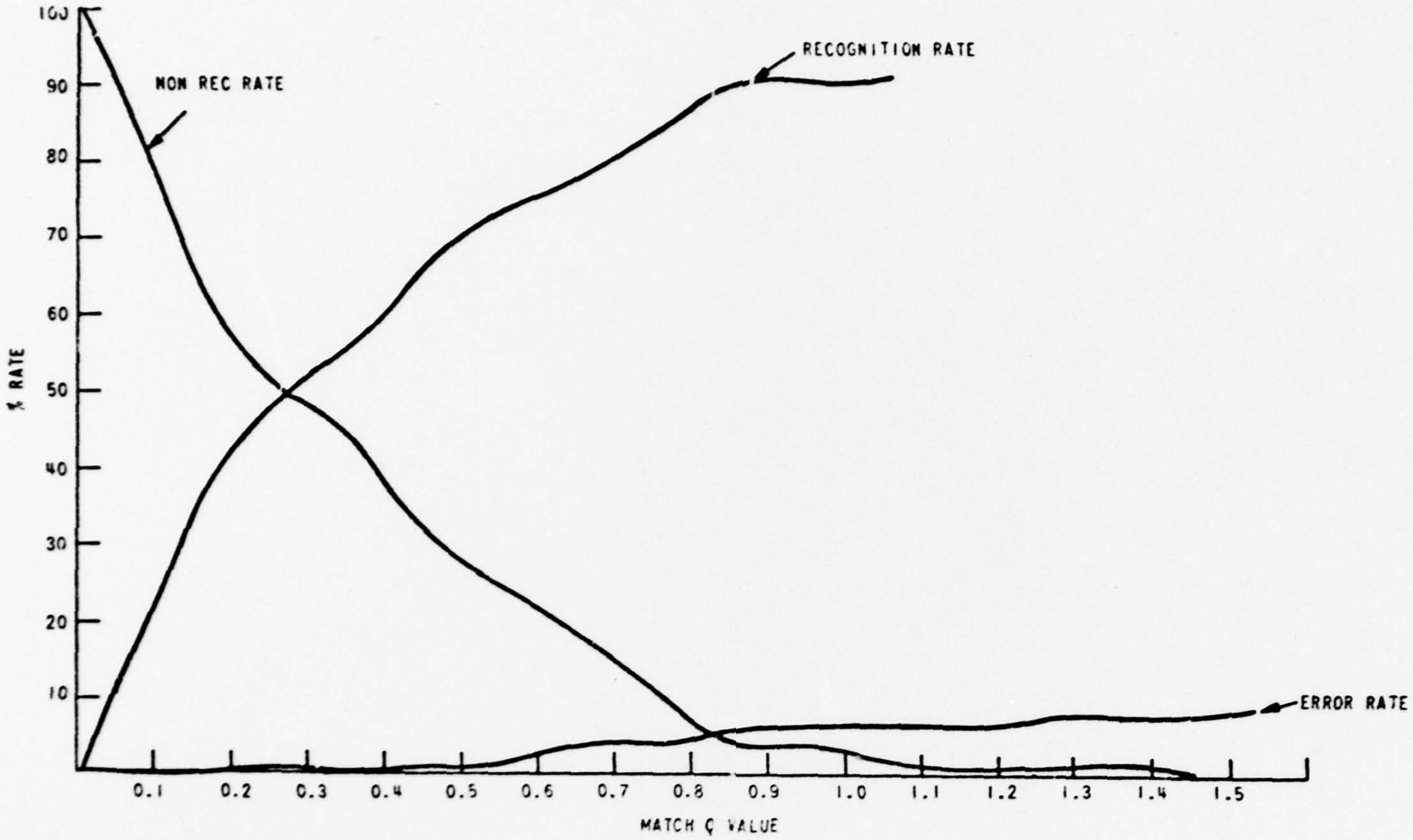


Figure 12

Recognition Rate Nasal Interpolation - 25-Word Vocabulary

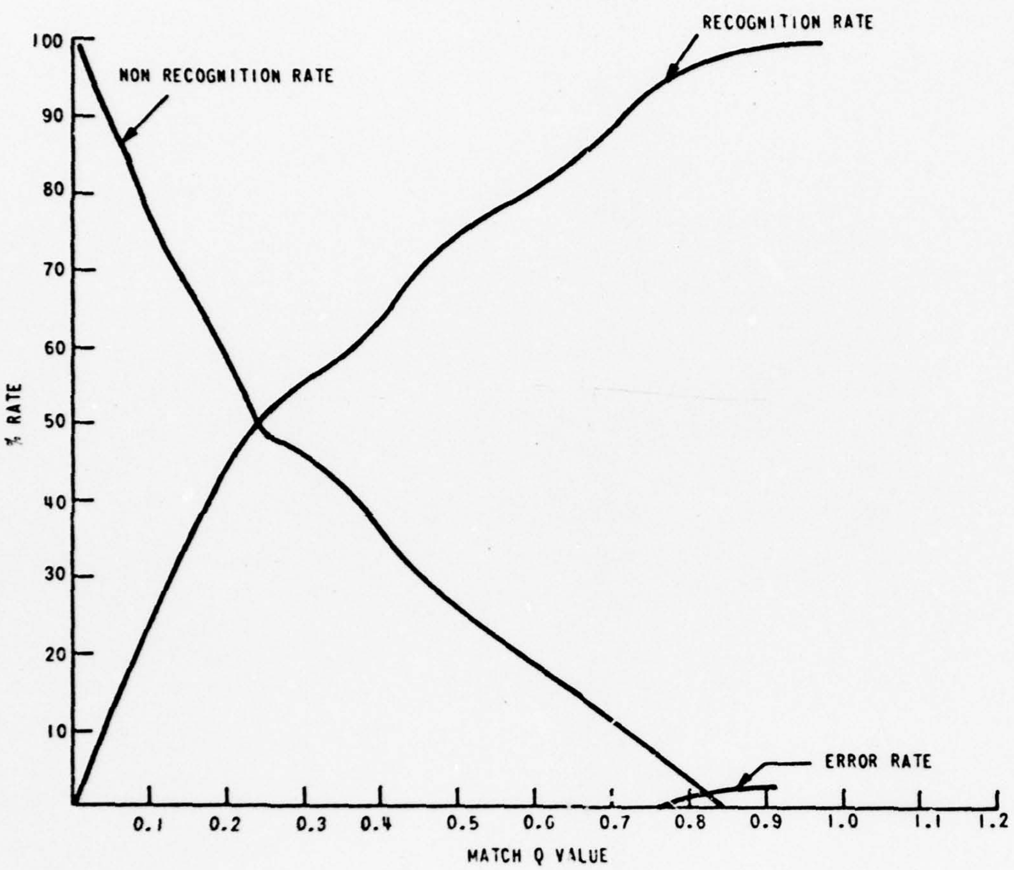


Figure 13

Recognition Rate No Nasal Interpolation - No Nasals in Vocabulary

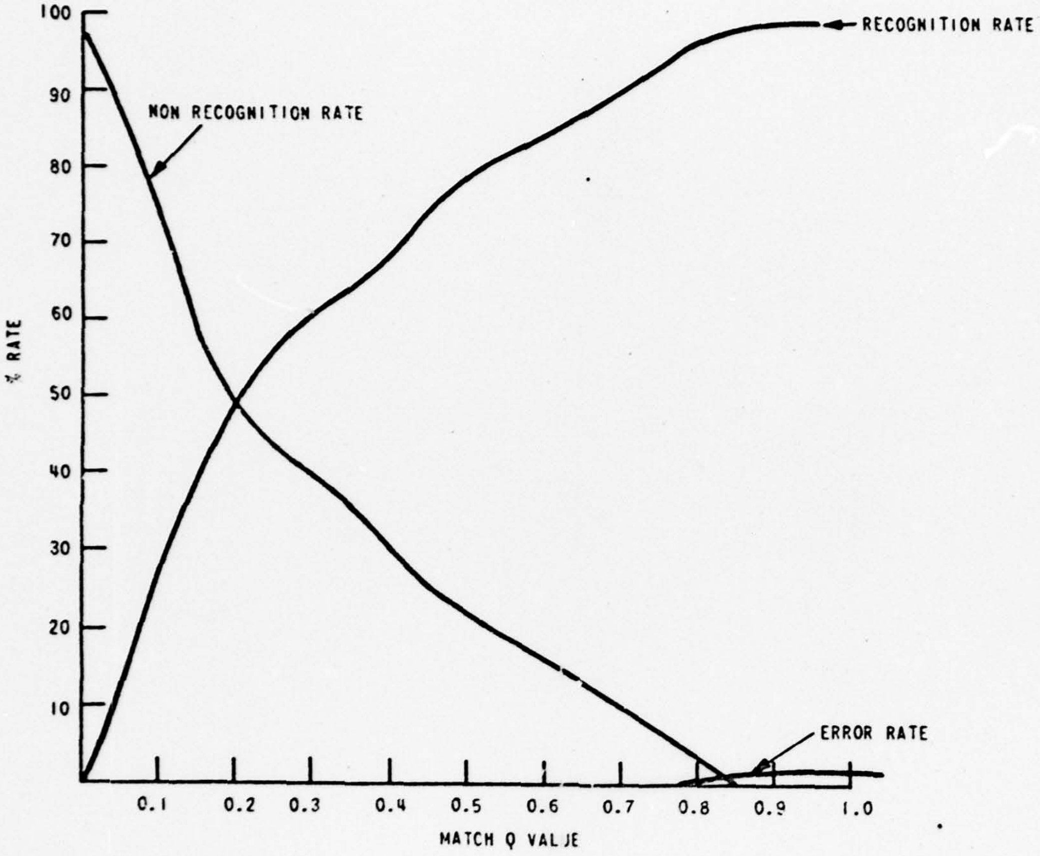


Figure 14

Recognition Rate Nasal Interpolation - No Nasals in Vocabulary

SECTION V

CONCLUSIONS AND RECOMMENDATIONS

The category-sequence word recognition software was pursued to a relatively high dgreee of refinement during the program. Analysis of test data taken at the end of the program has lead to the conclusion that the next major step in the recognizer performance will come with improvements in the phonemic category recognition hardware. However, work directed at improving the phonemic category recognizers during the course of the program has led to the conclusion that extensive changes must be made in the category detectors before a significant reduction in their contribution to word errors can be made. It is felt that the implementation of these changes will require a complete redesign and rebuilding of the phonemic category recognizer. These changes are aimed at four specific problem areas:

1. The elimination of certain design limitations.
2. The reduction of random errors caused by the parameter extractors.
3. The reduction of random errors caused by the category extractors.
4. The reduction of the recognizer's sensitivity to external acoustic noise.

While each of these areas is important to the overall performance, it is felt that the first area, design limitations, is the most important to the nasal category recognition problem inasmuch as it will permit the inclusion of additional information upon which to base the nasal recognition decision.

Four design limitation changes are felt to be necessary. These are:

1. New Category Extractors

The most severe deficiency in the present recognizer is the inability to recognize nasals with satisfactory reliability. The inclusion of two new phonemic categories should greatly help to reduce this problem. These categories are really the semi-stop features encountered in the articulation of nasals. These semi-stops are divided into semi-onset and semi-release classes. While these features do not uniquely identify the occurrence of a nasal, they always occur when a nasal occurs. (Unfortunately, semi-stops are also found during the articulation of l's and r's in certain phonemic environments.) This overlap is not severe because the occurrence of the feature is sufficiently consistent to appear in both test and library words. Thus, their use as a category should provide back-up for the existing nasal category. It is also suggested that the intervocalic pause also be carried as a separate new category. In the current logic it is used as a feature. This tends to rob the word logic of a reliable bit of information relating to the phonetic structure of the word.

2. Inclusion of Phoneme Durational Information

The present category extractors do not include phoneme duration as part of the information transmitted to the word logic. It has become evident from working with the detected strings of phonemic category that the inclusion of vowels and fricatives duration would provide significant held at the word logic level in determining the relative importance of vowel and fricative detections. At present, for example, many extra vowels are detected. The relative importance of these vowels cannot be determined because of the lack of duration information. (In general, the longer the vowel the more significant.) It is suggested that such information can be incorporated by providing the phonemic category logic with the capability of repeating a category detection in proportion to its duration. The word logic may then weight the relative importance of any detection by the number of repetitions. Thus, the category sequence 13-8-9-10-14-1-12 might become 13-8-9-9-9-10-14-1-12 or 13-8-8-8-9-10-14-1-12, depending upon the relative duration of the 8 and 9 category.

3. Fricative Detector Modifications

Fricative detections should be made on an instantaneous parameter basis rather than on an average parameter value during the fricative interval as is done in the current logic. The present fricative detectors produce one fricative decision per fricative utterance, and thus only represent the average value of the fricative. It has become clear that the fricatives should be detected much the same way as the vowels. Thus, if a change in the fricavalue occurs during the articulation, an indication of that change should be sent to the word logic.

4. Additional Parameters

An additional parameter has been found that improves the discrimination of the fθh category. This parameter is a measure of the randomness of the period of the clipped speech signal. The more random the period, the more likely that an fθh category is present. This information should be incorporated as a supplement to the present SEF spectral information.

The elimination of random errors produced by the parameter extractors involve three changes:

1. SEF Filter

The SEF extractor output contains a certain component of noise that is difficult to filter out with normal techniques. This is because the signal is characterized by large, very fast changes that contain significant information as well as small fast changes that are noise. Simple low-pass filtering to eliminate the noise also eliminates the large fast changes. A programmable filter has been breadboarded that heavily filters the signal in the absence of large changes and greatly increases its bandwidth during the fast changes. This technique shows a major reduction in the signal noise without losing any significant information.

2. Amplitude Detector

The present amplitude detector uses a peak detector envelope filtering technique. It has been found that low-pass filtering produces a more consistent signal response. It is suggested that this filtering system be adopted.

3. SEF Vowel and Fricative Quantizers

The proposed change in the SEF filter makes possible a better SEF quantizer that should considerably reduce the number of extra vowel detections. In the past, because of SEF noise, it was impossible to employ hysteresis in the threshold detectors. By using the SEF filter, a good deal of noise present in the output of the quantizers can be eliminated by the use of hysteresis.

There are two changes that would help to eliminate errors in the phonemic category detectors. These are:

1. Background Noise Level Shift for the Stop Category Detectors

The stop category detectors utilize the derivative amplitude parameter information to make decisions. The amplitude of the derivative is a function of the background level; thus, stop category decisions based upon derivative amplitude are modified by background level. It is suggested that background level can be compensated for by moving the stop detector threshold in accordance with a measurement of the background level.

2. Vowel-Fricative Detector Phonetic Environment Errors

The vowel-fricative detector is based upon the detection of fixed minimum durations of voiced and unvoiced energy. Unfortunately, these minimum durations vary depending upon the phonetic environment. It is suggested that these various environment conditions be used to alter the minimum time duration required to make vowel-fricative category decisions.

One change is recommended to reduce the noise sensitivity of the recognizer. This involves a modification of the speech silence detector so that the presence of a minimum duration period of voicing information is required to establish the presence of a spoken word. If this requirement is met before the amplitude parameter returns to silence, the word logic is reset.

SECTION VI

CIRCUIT DIAGRAMS

The circuit diagrams for the two cards added to the recognizer are shown on the following pages along with a table of the backpan wiring changes. Also included are the circuits developed for the SEF filter and the voicing circuits which were not installed in the recognizer.

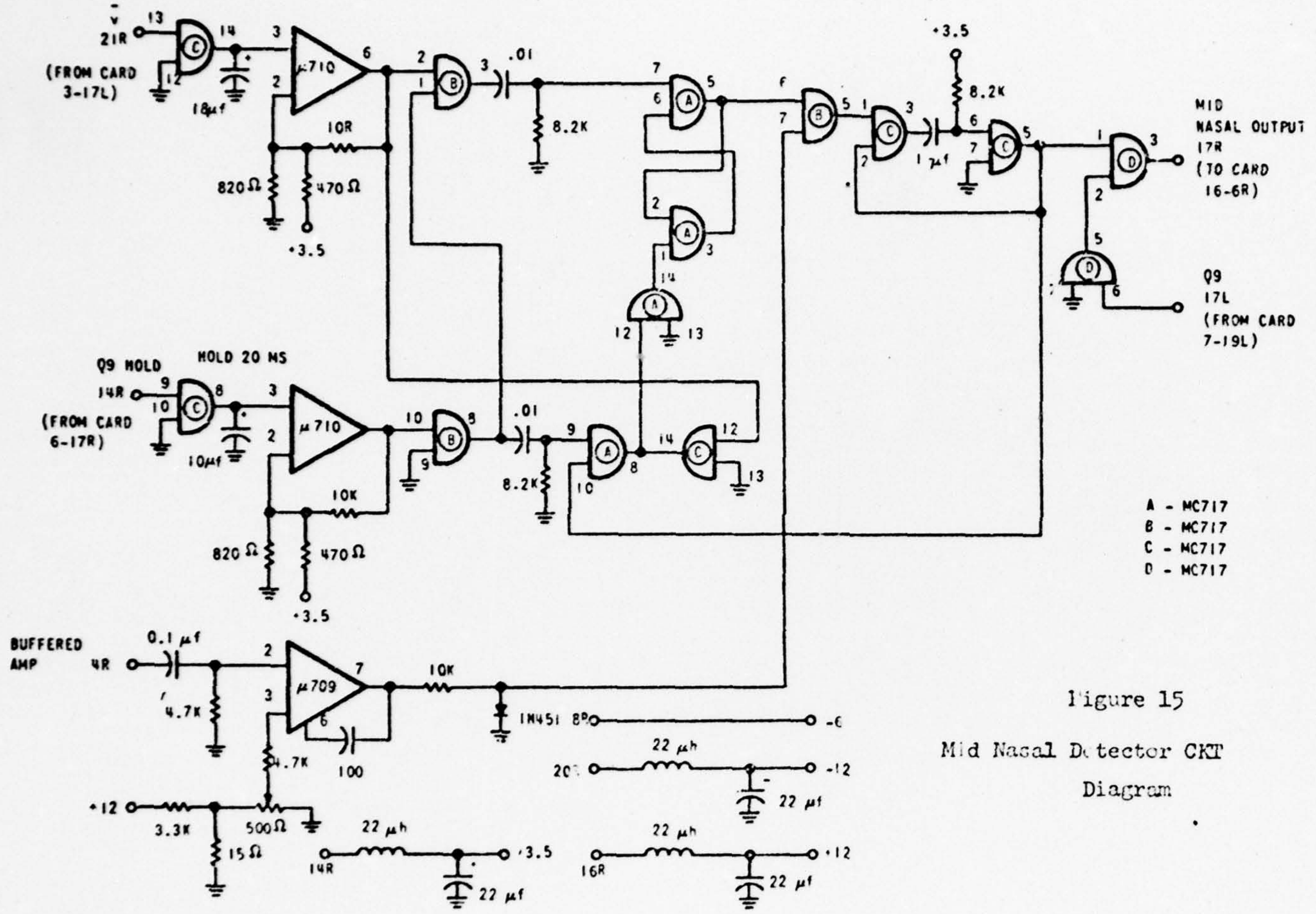


Figure 15

Mid Nasal Detector CKT Diagram

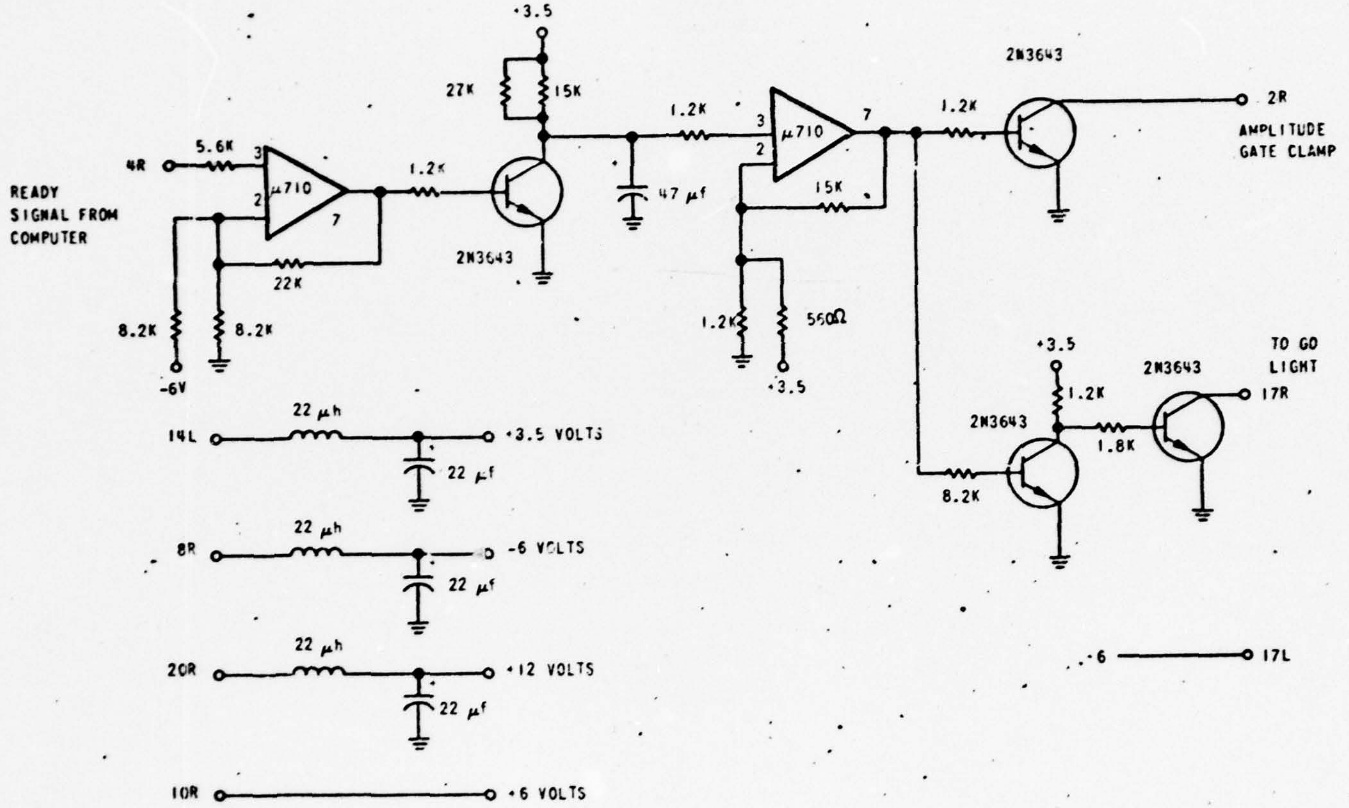


Figure 16

Go Light CKT Diagram

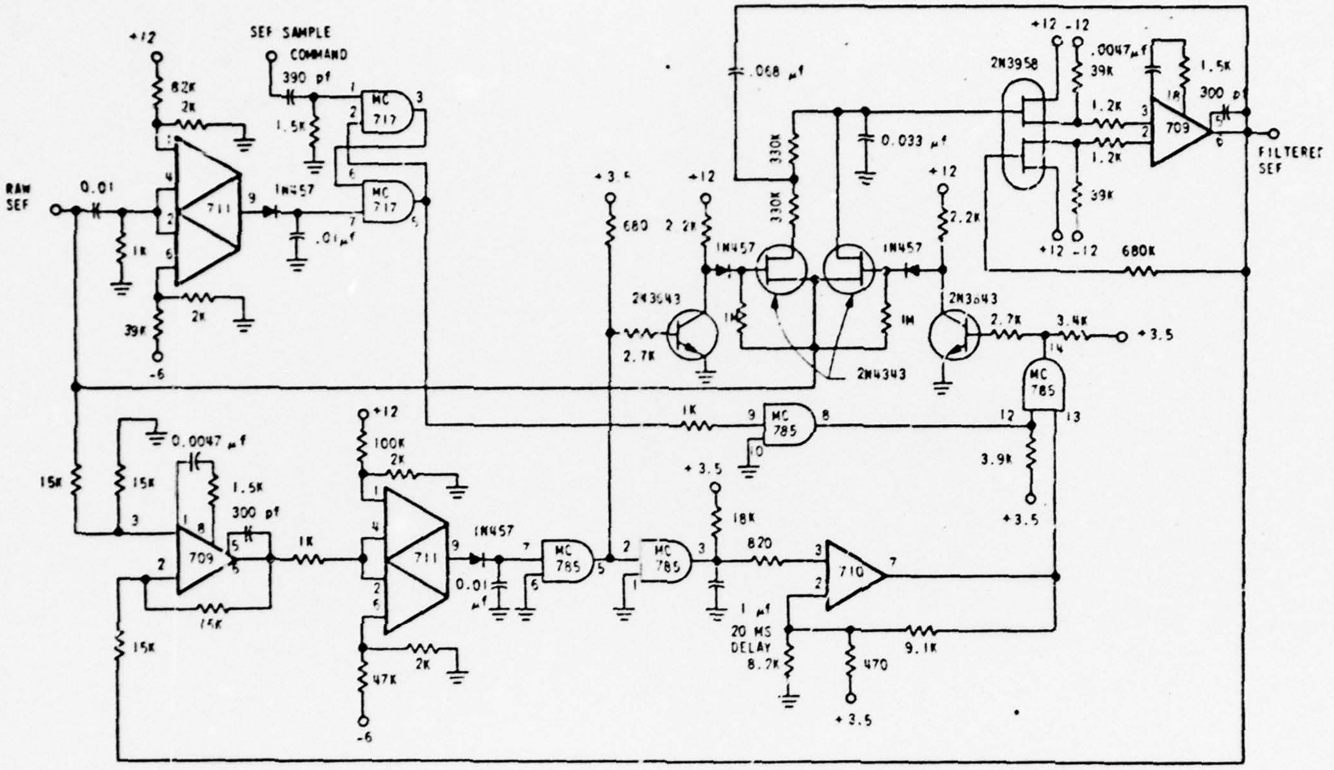


Figure 17

SEF Filter CKT Diagram

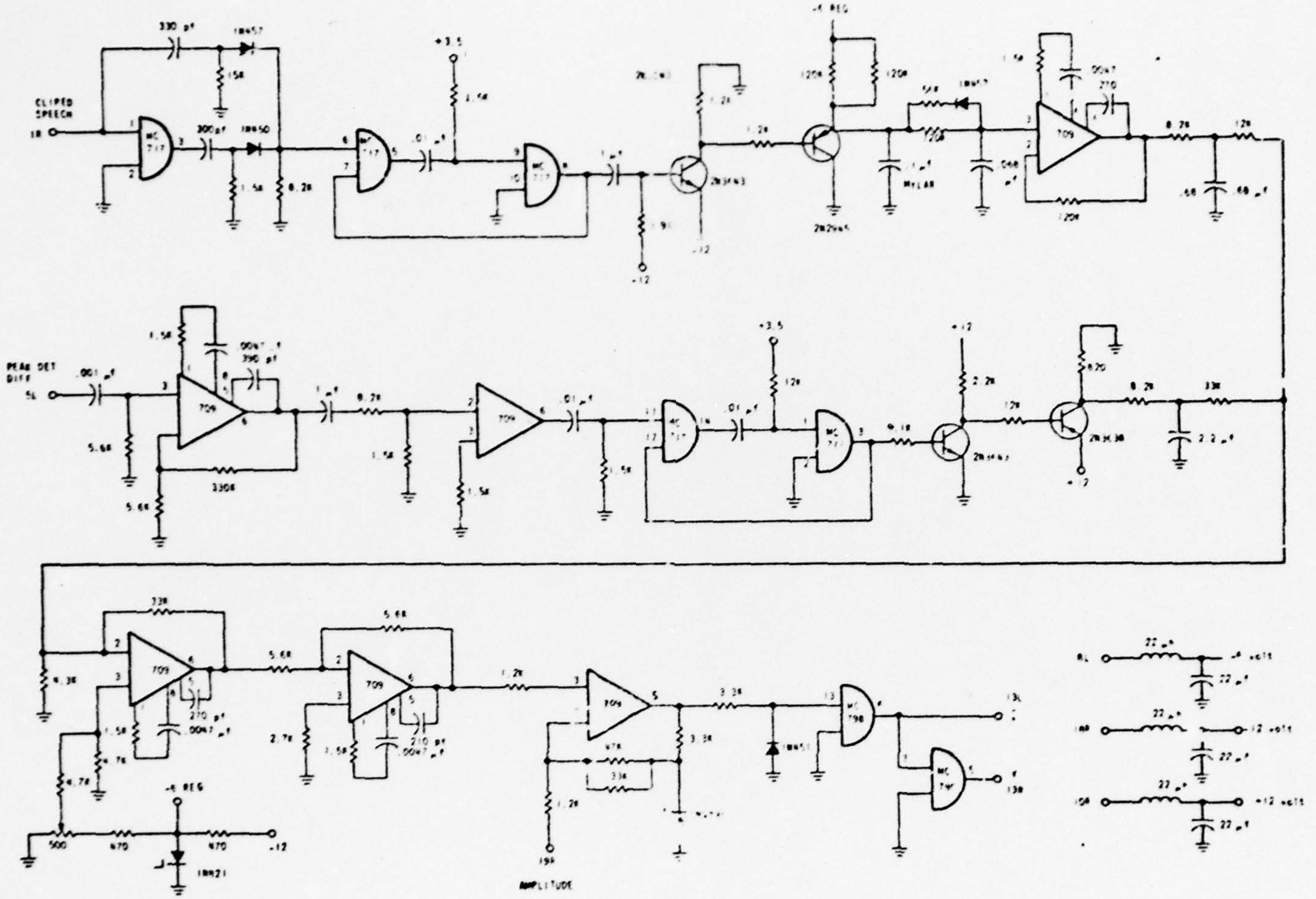


Figure 18

Voicing Detector CKT Diagram

Table 12

Wiring Changes on the Experimental Voice Sound Recognizer

TO		FROM		PURPOSE
Card	Pin	Card	Pin	
15A	4R	9	4R	A
15A	17R	15	6R	$\bar{Q}_9$
15A	17L	7	19L	$\bar{Q}_9$ (DELAY)
15A	19R	5	17R	Q_9
15A	21R	3	17L	S_f
15A	21L	8	21R	S_D
13	4R	11	11R	$\bar{F}$
XX	17L	GO LIGHT		
XX	17R	GO LIGHT		
XX	4R	COMPUTER "READY" INPUT		
XX	2R	8	19R	

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) Philco-Ford Corporation, Comm. & Tech. Svcs Div., 3900 Welsh Road Willow Grove, PA 19090	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP n/a

3. REPORT TITLE

Phonemic Category Experiments

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

Final Report, Jan 1969 - Jan 1970

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

Louis R. Focht

6. REPORT DATE July 1970	7a. TOTAL NO. OF PAGES 71	7b. NO. OF REFS -
8a. CONTRACT OR GRANT NO. F30602-69-C-0169 b. PROJECT NO. 7055 c. Task 705504 d.	9a. ORIGINATOR'S REPORT NUMBER(S) - 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) RADC-TR-70-84	

10. DISTRIBUTION STATEMENT

~~Each transmittal of this [illegible] the Department of Defense must have prior approval of RADC (EMITB), GAFB, NY 13440.~~

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES Report on speech recognizer experiments	12. SPONSORING MILITARY ACTIVITY Rome Air Development Center (EMITP) Griffiss AFB, NY 13440

13. ABSTRACT

This report describes the efforts undertaken to improve the experimental model "Voice Sound Recognizer" originally built under Contract AF30602-67-0300. This equipment utilized the techniques of Single Equivalent Formant parameter extraction, phonemic category recognition, and category-sequence word recognition.

Extensive hardware and software modifications to the basic recognizer system were made during the program which include the use of semiautomatic speaker adaptation by means of distance functions defined by sets of phonemic category strings and nearest neighbor word recognition decisions.

The final recognizer configuration displayed a reduced speaker sensitivity and an average recognition rate for four speakers of 85% when using a 25-word vocabulary.

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
Phonemic Category Recognition						
Single Equivalent Formant						
DESCRIPTORS:						
Acoustics						
Sound Signals						
Sound Pitch						
Computers and Data Systems						
Mathematics						
Statistical Analysis						
Statistical Data						
Statistical Distributions						

Unclassified
Security Classification