A MICROCOMPUTER-BASED DIGIT RECOGNITION SYSTEM

A Thesis Presented to
The Faculty of the College of Engineering and Technology
Ohio University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

by
Abdullahi M. Muhtar
June, 1984
# CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ABSTRACT</strong></td>
<td>i</td>
</tr>
<tr>
<td><strong>1.</strong> INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td><strong>2.</strong> SPEECH RECOGNITION SYSTEMS: A GENERAL PERSPECTIVE</td>
<td>5</td>
</tr>
<tr>
<td>2.1 Isolated speech recognition</td>
<td>5</td>
</tr>
<tr>
<td>2.2 Recognition algorithms</td>
<td>10</td>
</tr>
<tr>
<td>2.3 Limitations</td>
<td>11</td>
</tr>
<tr>
<td>2.4 Future developments</td>
<td>13</td>
</tr>
<tr>
<td><strong>3.</strong> SPEECH RECOGNITION HARDWARE</td>
<td>14</td>
</tr>
<tr>
<td>3.1 Mic preamplifier</td>
<td>16</td>
</tr>
<tr>
<td>3.2 Preamplication and signal detector stage</td>
<td>19</td>
</tr>
<tr>
<td>3.3 Bandpass filter stage</td>
<td>25</td>
</tr>
<tr>
<td>3.4 Sample-and-Hold System</td>
<td>36</td>
</tr>
<tr>
<td>3.5 Timing and control circuit</td>
<td>54</td>
</tr>
<tr>
<td><strong>4.</strong> ALGORITHM DEVELOPMENT (1)</td>
<td>59</td>
</tr>
<tr>
<td>4.1 Template matching</td>
<td>59</td>
</tr>
<tr>
<td>4.2 Amplitude normalization</td>
<td>73</td>
</tr>
<tr>
<td>4.3 Word identification</td>
<td>81</td>
</tr>
<tr>
<td>CHAPTER</td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>---</td>
</tr>
<tr>
<td>5. ALGORITHM DEVELOPMENT (11)</td>
<td>101</td>
</tr>
<tr>
<td>5.1 Systems hardware modification and connection</td>
<td>101</td>
</tr>
<tr>
<td>5.2 Data collection</td>
<td>106</td>
</tr>
<tr>
<td>5.3 Data reduction</td>
<td>114</td>
</tr>
<tr>
<td>5.4 Recognition software</td>
<td>134</td>
</tr>
<tr>
<td>6. SUMMARY AND CONCLUSION</td>
<td>149</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>153</td>
</tr>
<tr>
<td>APPENDICES</td>
<td>158</td>
</tr>
</tbody>
</table>
Chapter 1

INTRODUCTION

Speech recognition is a process in which human speech is translated from an analog waveform to a digital form that the computer can recognize. It has been a subject of research for about 30 years. Initial experimental systems utilized large computers because of the necessary extensive computations and the large amount of memory required. With the advent of the microcomputer, which contain processors with a lot of computing power, low cost, microcomputer-based speech recognition systems have become possible. The advantages of such systems are plenty. They have been found to be the simplest, most accurate and economical way to data entry into a computer in applications that require the operator's full attention or keeping both eyes on some process while inputting data on that process to a computer such as in manufacturing operations. They can be used in other varied applications such as airline reservations and security.

The audio waveform created from speech is a complex signal of considerable variation, depending on the speaker's age, sex, educational background, ethnic origin, regional dialect, rate of speaking, and physical characteristics. Speech is created by the vibration of vocal cords, which gives rise to voiced sounds, such as vowels. In addition, frictive sounds, like the "f" and "s" sounds, are produced by air passing through constrictions formed by the tongue, nose, and lips. These basic sounds are modified by the size and shape of the
of the cavities in the mouth, nose, and other air passages. Differences in all these physical factors cause voice patterns to vary from speaker to speaker and even from day to day in the same person. No present day recognition system could handle such differences efficiently and economically. To be practical, a speech recognition system requires inputs that are constrained and speech variations that are limited.

Isolated word recognition systems are currently the simplest, most accurate, and least expensive. They are usually programmed to recognize a small vocabulary of individual words, spoken carefully, and with controlled loudness by the user. These systems are usually speaker-dependent, responding only to the voice of the individual who trained the system. The purpose of this thesis is to design and build a microcomputer-based, isolated word, speaker-dependent, speech recognition system, capable of recognizing a vocabulary of the digits zero through nine.

A general view of speech recognition systems is given in Chapter 2. The basic steps involved in speech recognition are discussed. These include feature extraction and preprocessing of the speech signal, time normalization, and decision strategy. The preprocessing stage involves selecting certain features of the speech signal for sampling. A common approach is to use a bank of bandpass filters to provide a measurement of the frequency spectrum. The approach is used in this thesis. Six filter channels are used for spectrum analysis. The filter center frequencies chosen are:
64 Hz (channel 1), 125 Hz (channel 2), 250 Hz (channel 3), 500 Hz (channel 4), 1 kHz (channel 5), and 2 kHz (channel 6). Methods of time normalization and decision strategy are discussed.

Chapter 3 discusses the hardware circuitry for the speech recognition system. This consists of the microphone preamplifier for speech input into the system, a second stage of preamplification, which is used for speech input from tape, the six bandpass filters, six channels of rectifier and sample-and-hold stage, one for each filter, and control and timing circuitry. Rectification is done to reduce the variations in the speech signal, thus allowing sampling at a lower rate. Sampling is done by the sample-and-hold amplifier to ensure that all the filters are digitized at the same times.

Initial experiments were performed to evaluate some of the methods in speech recognition. These are discussed in Chapter 4. The experiments use known experimental data to represent both the vocabulary words and unknown words. These would give a better understanding of these methods before applying the varying signals encountered in speech. Methods of amplitude normalization, time normalization, and word matching are evaluated. Emphasis is placed on simple word matching techniques which are computationally simple and can be implemented within a reasonable time on a microcomputer.

Chapter 5 discusses some hardware modifications, interfacing the system to the Commodore PET microcomputer, and additional experiments to evaluate the stages of speech recognition. Real speech data, sampled at 6.7 kHz, is used in these experiments. Because of the enormous data generated by sampling six filter channels at 6,700
samples per second, it is desirable to find some technique of reducing the amount of data stored without losing the information needed for subsequent recognition. Two methods of data reduction are proposed and investigated. One of the goals of this system is to make a recognition decision by comparing as few filter channels as possible. This will reduce the computation needed and thus speed up the recognition process. The filter channels which provide the most unique representation of each vocabulary word are found by viewing the signals on an oscilloscope and by evaluating data on distance measures between spoken words and the template words obtained in experiments. Decisions are then made based on comparison of one or more of these channels. Based on the results of the experiments performed in Chapter 4 and this chapter, a final recognition algorithm is selected. The algorithm is implemented in BASIC and assembly language.

The performance and limitations of the system are discussed in Chapter 6. Future developments of the system are suggested. Data sheets of some ICs used in the hardware circuitry and the recognition programs are given in the Appendices.
Chapter 2

SPEECH RECOGNITION SYSTEMS: A GENERAL PERSPECTIVE

Speech recognition systems fall into two main groups: isolated (discrete) and continuous (connected) speech recognition. Isolated word recognizers recognize individual utterances (or words) separated by short pauses. The pauses help the recognizer to find the end of a word and the beginning of the next. Continuous speech recognition systems recognize words or phrases when they are spoken rapidly as in natural speech. Since the system in this thesis is for isolated word recognition, isolated word recognizers are discussed in this chapter. Their limitations, various recognition algorithms for matching words, and expected future developments of speech recognition systems are also discussed.

2.1 Isolated Speech Recognition

Isolated speech systems are currently the simplest, most accurate, and least expensive. These systems recognize a small vocabulary of individual utterances.

As mentioned above, utterances in isolated speech recognition systems must be isolated by pauses. The pause varies depending on the system, usually being at least 100 ms to ensure that the system doesn't confuse the pause between words with pauses sometimes occurring within words. A standard value for pauses is 200 ms [5].

Isolated speech recognition systems must be initially trained with the vocabulary they are expected to recognize. Most commercial
recognition systems are speaker dependent, thus each speaker using the system must train it to understand how he says each word. The training consists of the system listening to the way you say each word in the vocabulary and storing a reference pattern (template) in memory that is associated with each word. Old words may be "retrained" if recognition performance is unsatisfactory, and additional words can be added to vocabulary set at any time (within the limits of available memory).

A typical word recognition system uses template matching in the actual recognition of speech. A template characterizing each training word is formed and stored in memory. The system then attempts to match the word spoken to one of the existing templates when the word is spoken. The general recognition process used in most commercial systems is shown in Fig. 2.1. First, a form of preprocessing is done on the speech signal which includes digitizing it. After digitizing a word, if the system is in training mode, the digital pattern of that word is stored in memory as a template. If the system is in recognition mode the pattern of the incoming word is compared to previously digitized words stored in memory. The word in memory that most closely matches the incoming word is said to have been recognized.

The preprocessing may be accomplished in different ways. In general, certain features of the speech signal, e.g. loudness, duration, timing and pitch, are selected (extracted) for sampling. One common approach is to use a bank of bandpass filters (often 8 or 16) which divide up the speech energy into separate frequency ranges (spectrum analysis). The signals from the filters are then digitized.
Fig. 2.1 A general block diagram for speech recognition.
Another preprocessing method is linear predictive coding (LPC). This is the most widely used parametric means of representing speech. It is also the most compact, requiring about one-tenth the bit rate of waveform encoding. LPC comprises a technique of analyzing human speech by viewing the vocal tract as a time varying filter. Since the vocal tract, and thus, the formant frequencies, change slowly with time, analysis frames of finite duration can be sampled [7].

The speech analyzer in an LPC-based system looks for linear-predictive-coding features for each word spoken to it. It is these features that actually comprise the templates. Until recently, the process of extracting LPC features took a large computer and special algorithms. Programmable single-chip digital signal processors are now available that can carry out LPC encoding, and other voice processing functions such as speech synthesis and recognition, and voice verification. One of these is Texas Instruments' TMS 320 which is a single-chip microcomputer optimized for high-performance signal processing. It uses auto-correlation analysis and a modified Le Roux-Geuguen algorithm to perform feature extraction [6]. This and similar chips should revolutionalize computer speech technology.

Another method (used in the speech lab recognition board) extracts four features of the speech input (three amplitude values and one zero crossing value). Three filters, tuned to the midpoint of the first three characteristic speech frequencies, are used to find the amplitude values. A zero-crossing detector finds the signals zero-crossing rate, which provides an overall frequency measurement. These selected
features are then sampled and digitized, each spectral sample represented by a four-dimensional vector.

The actual word boundaries in the sample must also be determined. A threshold value, based on a combination of features (amplitude or zero-crossing rate), is used to determine the starting and end point of each word. A common difficulty with word boundaries appears with words which have stop consonants, such as the unvoiced "t" in the word "eight". This stop may be of considerable duration, and the method must be modified to find an end point only when the signal drops and stays below the threshold for a certain duration, typically 100 ms.

Another preprocessing step which is done after sampling is amplitude normalization. This is found to improve recognition performance [1], [2]. It compensates for signal level differences normally encountered in the speech data. A common normalization method for isolated word recognition is the mean normalization [2]. In this method, each spectral time sample is amplitude normalized independent of the other spectral time samples by subtracting the overall amplitude of the sample under consideration from each component in that sample. The normalized components then represent the spectrum shape of each sample independent of amplitude level. The disadvantage of this method is that it destroys the intrautterance amplitude relationships.

An alternate normalization method is called the proportional normalization and is explained in [4]. This method compensates for interutterance signal level differences and also preserves the intrautterance amplitude relationship.
Comparison of an unknown utterance and a prototype is accomplished by measuring the degree of similarity between sounds in the same time unit. The sound similarity is proportional to the distance separating two sounds in a vector space defined by the selected features (e.g. bandpass filter parameters).

There are various comparison methods which are normally done under program control. The choice of a particular method is based on the application of the recognition system. Simple differencing algorithms can be used for systems with limited vocabulary and where exact matching is not required. Other systems use math-modelling techniques with math intensive programs that require a mainframe computer to execute. These systems are expensive and slow.

The following algorithms are commonly used for speech sound similarity measure:

1. **Chebyshev norm**

   This means summing the absolute value of the differences of the coordinate values between the unknown and template for each time slice. If we have a vector space of six dimensions, and $X_i$ represents the template sample at time slice $i$ and $Y_i$ represents the unknown utterance, i.e. $X_i = (x_1, \ldots, x_6)$ and $Y_i = (y_1, \ldots, y_6)$, then the Chebyshev distance between $X$ and $Y$ is

   $$|XY| = (|x_1 - y_1| + \ldots + |x_6 - y_6|)$$

   This method has computational simplicity.
2. Euclidean norm

For six-dimensional Euclidean space, say, defined by six filter banks, if \( X = (x_1, \ldots, x_6) \) and \( Y = (y_1, \ldots, y_6) \), then the distance between \( X \) and \( Y \) is given by

\[
|XY| = [(x_1 - y_1)^2 + \ldots + (x_6 - y_6)^2]^{1/2}
\]

The multiplications required for this method can be costly for a large vocabulary.

3. Dynamic Programming

The dynamic programming algorithm provides a method of non-linear matching of speech patterns. The equation used is given by [1]

\[
D_{ij} = d_{ij} + \min \{D_{i-1,j}, D_{ij-1}, D_{i-1,j-1}\}
\]

where \( d_{ij} \) is the distance between the first utterance at time slice \( i \) and the second utterance at time slice \( j \). \( D_{ij} \) is the total distance between the first and second utterances from their beginnings up to and including time \( i \) and \( j \). The dynamic programming method requires more computation but significantly improves recognition results for multi-syllabic utterances [1]. This method is discussed fully by Itakura [3].

2.3 Limitations

One of the limitations of speech recognition systems is vocabulary size. The vocabulary size, as well as the size of the word template, must be traded off against the amount of storage. Even if storage is sufficient for a large vocabulary, the time for comparison between an utterance template and the vocabulary templates can be prohibitive. A possible solution is partitioning the vocabulary set by hierarchical
structuring. An example of this is to count the number of syllables in an input utterance and compare this count with similar counts for all the words in the vocabulary. The words whose syllable count is wrong are eliminated from the candidate list before any other comparison [5]. This speeds the recognition process.

Another problem is that of misalignment between utterances. This may be due to poor detection of the beginning and endings of utterances, or due to slight variations in timing both in the overall length of words and in the length of phonemes within an utterance. This effect makes a straightforward distance measure, for example Chebyshev distance, very poor in recognition performance. Dynamic programming is commonly used to solve this problem. This algorithm is used to find the best alignment between the vocabulary templates and the unknown utterance template while matching them.

Recognition errors also have to be considered when using a speech recognizer. These are errors of rejection (no recognition), substitution (false recognition), or addition. A rejection error occurs when the system fails to recognize a word which is in its vocabulary. Substitution error is when the system makes a match with a word other than the one that was spoken. An addition error is the incorrect selection of a word in response to background noise or an utterance not in the vocabulary. These errors may be reduced by retraining some words and by studying the environment where the system is going to be used.

In general, performance evaluation of a speech recognition systems includes cost effectiveness, practicality and reliability, and
acceptability to the user. Most important is that the system must
demonstrate the capability to perform the intended function with the
necessary accuracy.

2.4 Future Developments

One principal application of speech recognition is in data entry.
In applications that require the operator's full attention, or keeping
both eyes on some process while inputting data on that process to a
computer, a voice data entry clearly has advantages over other data-
entry techniques. In all applications, voice data entry is faster and
less prone to error or misinterpretation, and since it takes much less
effort, reduces labor costs.

Areas which can benefit from speech input/output technology are
control of devices in manufacturing process, information retrieval,
security in form of speaker identification, and even word processing.
Chapter 3

SPEECH RECOGNITION HARDWARE

The hardware circuitry designed for the speech recognition system is discussed. Both theory, design procedures and, in some cases, testing are covered for each circuit stage. The stages are covered in the sequence they are built into the system.

A low-noise microphone (mic) preamplifier, which is necessary when using a low impedance mic, is discussed. Its output is amplified further by the preamplifier stage which is a single stage, non-inverting amplifier. This can be the only stage of preamplification if a high impedance mic is used. The preamplifier provides the input for the signal detector, which detects the beginning of a word and presents a WORD PRESENT signal at its output, and the six filter channels. Analog signals from the filter channels are sampled at the sample-and-hold stage, the output of which are connected to a multiplexer for channel selection and digitization.

The timing and control circuitry is also discussed. This provides all timing and control signals thus enabling the recognition system to be able to operate as a "stand alone" system.

A complete block diagram of the speech recognition system is shown in Fig. 3.1.
Fig. 3.1 Block diagram of the speech recognition system hardware.
3.1 Mic Preamplifier

3.1.1 Introduction

Microphones are divided into two groups: high impedance and low impedance. High impedance mics (~20 K) have high outputs (~20 mV). They place no special requirements on the pre-amp, thus amplification is done simply and effectively with standard amplifier configurations. Due to the large input levels, noise requirements of the amplifier are minimal. But being a high impedance source, these mics are susceptible to stray field pickup (e.g. 60 Hz). Thus their use must be restricted to short distances typically less than 10 feet of cable length.

Low impedance mics (~200 Ω) have low output (~2 mV). This imposes stringent noise requirements upon the preamp. For a signal-to-noise ratio of 65 dB with a 2 mV input signal, the total equivalent input noise (EIN) of the preamp must be 1.12 μV (10-19 kHz) [8]. Thus low noise preamps must be used. National LN387A low noise dual preamp is chosen for the mic preamp in this project. It has a guaranteed EIN of ≤ 0.9 μV and gives at least 67 dB S/N performance for 2 mV input level.

3.1.2 LM387 Low Noise Dual Preamplifier

The LM387/387A belongs to National's line of preamplifiers designed to meet the requirements of amplifying low level signals in low noise applications. Its total equivalent input noise is typically $0.65 \mu V_{RMS}$ ($R_s = 600 \Omega$, 100 Hz - 10 kHz) and supply rejection ratio is typically 110 dB ($f = 1$ kHz). Other outstanding features include high
gain (112 dB), large output voltage swing \((V_{CC} - 2V)_{p-p}\), and wide power bandwidth \((75 \text{ kHz}, 20 V_{p-p})\). The LM387 operates from a single supply across the wide range of 9 to 40 volts. It is internally compensated and short-circuit protected.

3.1.3 Mic Preamp Design

For low level signal applications requiring optimum noise performance, the non-inverting amplifier configuration is most suitable. The LM387A used as a non-inverting mic preamp for a low impedance mic with unbalance two wire output is configured as in Fig. 3.2

Referring to the configuration shown, resistors \(R_2\) and \(R_3\) provide negative input bias current and establish the DC gain, \(A_{VD}C\), where

\[
A_{VD}C = 1 + \frac{R_3}{R_2} \quad (3.1.1)
\]

\(R_2\) and \(R_3\) are chosen to be metal film for the low noise requirements. AC gain is set by the ratio of \(R_3\) to \(R_1\), while low frequency roll-off at \(f_0\) (low frequency - 3 dB corner) is determined by capacitor \(C_1\).

\[
A_{VA}C = 1 + \frac{R_3}{R_1} \quad (R_1 << R_2) \quad (3.1.2)
\]

\[
C_1 = \frac{1}{2\pi f_0 R_1} \quad (3.1.3)
\]

Capacitor \(C_0\) acts as an input AC coupling capacitor to block DC potentials in both directions and can equal 0.1 \(\mu F\) or larger. Capacitor \(C_S\) is used to provide power supply decoupling. This is necessary since the LM387A is a high gain amplifier. A 0.1 \(\mu F\) ceramic capacitor located close to the IC is sufficient.
3.1.4 Design Calculations

Required gain is $A_{\text{VAC}} = 400$ (52 dB), and power supply is $V_s = 12 \text{ V}$. Choosing $R_2 = 61 \text{ K}$, from equation (3.1.1)

$$R_3 = \left( \frac{V_s}{2.6} - 1 \right) \quad R_2 = \left( \frac{12}{2.6} - 1 \right) \times 61 \text{ K}$$

$$\therefore R_3 = 220 \text{ K}$$

Using equation (3.1.2),

$$A_{\text{VAC}} = 400 = 1 + \frac{R_3}{R_1} = 1 + \frac{220}{R_1}$$

$$\therefore R_1 = 560 \Omega$$

From equation (3.1.3),

$$R_1 = \frac{1}{2\pi f_0 C_1 R_1}$$

Choosing $f_0 = 20 \text{ Hz}$, $C_1$ is calculated as $\approx 14 \mu\text{F}$. $C_1$ is chosen to be $10 \mu\text{F}$. Also $C_0$ is chosen to be $1 \mu\text{F}$. Thus the values of the components for the mic preamp as configured in Fig. 3.2 are:

- $R_1 = 560 \Omega$
- $R_2 = 61 \Omega$
- $R_3 = 220 \Omega$
- $C_0 = 1 \mu\text{F}$
- $C_1 = 10 \mu\text{F}$
- $C_S = 0.1 \mu\text{F}$

The mic preamp is constructed using these values.
3.2 Preamplification and Signal Detector Stage

3.2.1 Introduction

The preamplification stage in this speech recognition system is a single-stage non-inverting feedback amplifier with adjustable gain. The signal feeding its input may be from the mic preamp discussed in Section 3.1 if a low impedance, low input microphone is used, direct from the output of a high impedance microphone. The output of the preamplifier feeds the filter inputs and a signal detector, which, as the name suggests, detects the beginning of a word spoken into the system and gives the signal to start the sampling process.

3.2.2 Preamplifier

The general non-inverting circuit is shown in Fig. 3.3. The signal is applied to the non-inverting input and a portion of the output signal is fed back to the inverting input. This feedback network then determines the overall closed-loop transfer function.

The following equations may be written:

\[ v_1 = i_1 Z_1 = v_0 \frac{Z_1}{Z_1 + Z_f} \]  \hspace{1cm} (3.2.1)

\[ v_0 = A(v_S - v_1) \] \hspace{1cm} (3.2.2)

These equations assume that amplifier input current is zero. Combining the above equations and allowing the amplifier gain \( A \) to be arbitrarily large yields

\[ \frac{v_0}{v_1} = \frac{Z_1 + Z_f}{Z_1} \quad \text{and} \quad v_S = v_1 \] \hspace{1cm} (3.2.3)
Fig. 3.2  Non-inverting transformerless mic preamp for unbalanced inputs.

Fig. 3.3  Non-inverting follower.
Therefore, voltage amplification of the circuit is given by

$$A_V = \frac{V_0}{V_S} = 1 + \frac{Z_F}{Z_I} \quad (3.2.4)$$

The advantage of non-inverting circuits is the impedance buffering property they provide, i.e. $Z_I = \infty$ and $Z_0 = 0$.

The amplifier designed for the system is shown in Fig. 3.4. It has an adjustable voltage gain, adjustability being obtained by adding a potentiometer in the feedback loop.

3.2.3 Signal Detector

The signal detector consists of an average level indicator and a comparator. The average level indicator produces a DC voltage output proportional to the average input level. This output is fed to the voltage comparator, which switches when the average input level is above a certain amplitude as discussed below. Its output is used to start the sampling process as will be discussed in Section 3.5.

The level indicator in the system consists of a voltage follower with unity gain feeding a half-wave rectifier as shown in Fig. 3.5.

The comparator is also shown in Fig. 3.5 and consists of two inputs (one a constant reference voltage $V_R$ and the other a time-varying signal $v_i$ which is the output signal from the level indicator) and one output $v_o$. An ideal comparator, whose characteristic is shown in Fig. 3.6, has an output which is constant with $v_o = V(0)$ if $v_i < V_R$, and it has a different constant value $v_o = V(1)$ if $v_i > V_R$, thus the input is compared with the reference. Voltages $V(0)$ and $V(1)$ compatible with TTL or MOS logic may be obtained.
Fig. 3.4 Preamplifier stage.

INPUT

4.7 µF

33 K

100 K

100 K

250 K

10 K

10 K

22 µF

LM1458

5

6

7

TO SIGNAL DETECTOR INPUT

TO FILTER INPUTS

INPUTS

TO SIGNAL DETECTOR INPUT

N

N
Fig. 3.5 Average level indicator and comparator.
Fig. 3.6 The transfer characteristics of (a) an ideal comparator and (b) a practical comparator.
To obtain limiting output voltages compatible with TTL, a resistor R is added and two back-to-back Zener diodes to clamp the output of the comparator, as shown in Fig. 3.5. The value of R was chosen so that the avalanche diodes operate at the recommended Zener current. The limiting voltages of $v_o$ are $V_{Z1} + V_D = V_0$ and $-(V_{Z2} + V_D) = -V_0$, where $V_D$ (≈ 0.7 V) is a p-n diode forward voltage. Z1 is chosen with $V = 6.9$ volts and Z2 with $V = 3.5$ volts. Thus $v'_0$ switches from -4.2 volts to 7.6 volts.

An LM1458 is used for the comparator stage. This is a comparator/amplifier IC.

### 3.3 Bandpass Filter Stage

#### 3.3.1 Introduction

The bandpass filters form the main part of the preprocessing stage in the speech recognition system. A bank of six bandpass filters divide the speech input into six channels. The six filters cover six octaves from 64 Hz to 2 kHz. The choice of these center frequencies is based on the range of frequencies that contain the most spectral information. It has been shown that there is little activity below 60 Hz and above 2 kHz in speech [9]. The bandpass filters used are multiple feedback path active RC filter types. These are second-order filters and thus have a rolloff slope of 12 db/octave.

The filters are designed with a $Q = 4$ with a provision for changing the $Q$ to 2. This is achieved by using variable resistors (trim-pots) and choosing their values to allow one to set $Q = 4$ or $Q = 2$ while using standard capacitors.
3.3.2 Theory of Active RC Bandpass Filters

An ideal bandpass filter has a constant response for \( f_L < f_0 < f_H \) and zero gain outside this range. This is practically impossible to obtain. An approximate narrow band characteristic can be obtained using a second-order LC resonant circuit as shown in Fig. 3.7. This circuit provides a filter with a bandpass response which peaks at a center frequency \( f_0 \) and drops off with frequency on both sides of \( f_0 \).

Assuming the amplifier has a gain \( A_0 = V_0 / V_i \) which is positive and constant for all frequencies, we find the transfer function of the circuit in Fig. 3.7 is

\[
A_V(j\omega) = \frac{V_0}{V_s} = \frac{V_0 V_i}{V_i V_s} = \frac{RA_0}{R + j(\omega L - 1/\omega C)} \tag{3.3.1}
\]

Also the center frequency \( f_0 = \omega_0 / 2\pi \) of this circuit is the frequency at which the inductance is equal to the capacitance in magnitude. Thus

\[
\omega_0^2 = \frac{1}{LC} \tag{3.3.2}
\]

The quality factor \( Q \) of this circuit is given by

\[
Q = \frac{\omega_0 L}{R} = \frac{1}{\omega_0 CR} = \frac{1}{R} \sqrt{\frac{L}{C}} \tag{3.3.3}
\]

Putting \( s = j\omega \) in Eq. (3-3-1) and substituting Eqs. (3-3-2) and (3-3-3) we get after rearranging,

\[
A_V(S) = \frac{(\omega_0 / Q) A_0 S}{S^2 + (\omega_0 / Q) S + \omega_0^2} \tag{3.3.4}
\]

The transfer equation of Eq. (3-3-4) can be implemented with the multiple-feedback RC circuit of Fig. 3.8.
Fig. 3.7  A resonant circuit.
In a maximally flat bandpass version of the circuit we let $C_1 = C_2 = C$. Referring to Fig. 3-8, the voltage across $R_3$ is $V_0$. \therefore $I_3 = V_0 / R_3$, and assuming the bias current is negligible,

$$V_a = -\frac{I_3}{SC} = -\frac{V_0}{SCR_3} \quad (3.3.5)$$

Applying KCL to node "a", the transfer function of the circuit is found to be

$$\frac{V_O(S)}{V_S} = \frac{-S/R_1C}{S^2 + (2/R_3C)S + 1/R_1R_3C^2} \quad (3.3.6)$$

where

$$R = R_1 || R_2 = \frac{R_1R_2}{R_1 + R_2} \quad (3.3.7)$$

Equating coefficients of $S$ in the numerators of Eqs. (3-3-4) and (3-3-6), we obtain

$$R_1C = \frac{Q}{\omega_0(-A_0)}$$

or

$$R_1 = \frac{Q}{(-A_0) 2\pi f_0 C} \quad (3.3.8)$$

Similarly, equating the coefficients of $S$ in the denominators of the equations, we get

$$\frac{R_3C}{2} = \frac{Q}{\omega_0}$$

or

$$R_3 = \frac{2Q}{2\pi f_0 C} \quad (3.3.9)$$
Fig. 3.8 Multiple feedback active RC bandpass filter.
Equating the constant terms in the denominators gives

$$R' R_3 C^2 = \frac{1}{\omega_0^2} \quad (3.3.10)$$

Eliminating $R_3$ by using Eq. (3.3.9) in (3.3.10), we get

$$2R'C = \frac{1}{\omega_0 Q} \quad (3.3.11)$$

Putting Eqs. (3.3.7) and (3.3.8) in (3.3.11), we find

$$R_2 = \frac{0}{(2Q^2 - A_0)} \frac{1}{2\pi f_0 C} \quad (3.3.12)$$

Choosing $C$ arbitrarily, we can determine the values of $R_1$, $R_2$ and $R_3$.

3.3.3 Design Considerations

The choice of components for filter design is very critical if high performance is required. Certain amplifier performance characteristics affect filter performance. These include dc voltage offset, bias current, voltage and current noise, and open loop gain. They have to be considered when choosing the amplifier for an active filter circuit.

A dc offset voltage is important in low pass filter application, thus will not be discussed here.

The output noise of active filter circuits is due to the internal voltage and current noise of the operational amplifier used. This noise is random and is generated by resistors and active elements in the amplifier, its characteristics depending on the internal circuit design and the bandwidth of interest.
Open-loop gain effects can be severe for bandpass realization with the multiple feedback circuit because of the large amount of loop gain required for ideal performance. The open-loop gain of operational amplifiers is not infinite. It can be proven that this finite open loop gain gives the bandpass filter a magnitude and phase error [13].

The finite input impedance of operational amplifiers also affect the performance of the filter since it is a feedback circuit. At higher frequencies the input capacitance become very significant in determining closed-loop frequency response.

All these effects point to the fact that a high-grade amplifier should be chosen for our filter stages. The idea is to obtain as high an input impedance as possible, wide frequency response, low voltage and current noise, and low bias currents. The choice here is the LF356 which is a member of National's LF156 series of Low Offset Monolithic JFET Input Operational Amplifiers. Because they are FET they have low bias current (50 pA max), low input noise current (0.01 pA/√Hz), low input noise voltage (12 nV/√Hz), and a high input impedance (10^{12} Ω). They also have a wide gain bandwidth (4 MHz), and a large DC voltage gain (106 dB) [Appendix B].

The choice of resistors and capacitors also affect the performance of filter circuits. For best filter performance, precision resistors and low-drift capacitors are used. Three types of resistors often used are carbon composition, metal film, and wire-wound resistors. Capacitors often used are silver mica, polycarbonate, and polyethylene. The final choice is based on a compromise between superior characteristics, cost and size.
Since the values of these components is critical, it was decided to use standard value capacitors and non-standard value resistors obtained from trim-pots. The type of capacitor used is silver mica.

3.3.4 Design Calculations

Fig. 3.9 shows a diagram of the filter circuit used in the speech recognition system. The equations for calculating the resistor and capacitor values were found in Section 3.3.2 and are reproduced here for convenience.

\[ R_1 = \frac{Q}{A_V} \cdot \frac{2\pi f_0}{C} \]  
\[ R_2 = \frac{Q}{(2Q^2 - A_V)} \cdot \frac{2\pi f_0}{C} \]  
\[ R_3 = 2Q/\omega_0 C = \frac{Q}{\pi f_0} C \]

Also, the bandpass gain is given by

\[ A_V = \frac{1}{(R_1/R_3)(1 + C_2/C_1)} \]

Letting \( C_1 = C_2 \), Eq. (3.3.16) becomes

\[ A_V = \frac{1}{(2R_1/R_2)} = \frac{R_3}{2R_1} \]

The component selection sequence is:

1. Determine the required \( Q \), \( f_0 \) and \( A_V \).
2. Set \( C_1 \) and \( C_2 \) (\( C_1 = C_2 \)) to a convenient value, \( C \).
3. Compute \( R_1 \).
4. Compute \( R_2 \).
5. Compute \( R_3 \).

As mentioned in Section 3.3.3, standard values of capacitors were used in the design. These are:

a) 0.01 \( \mu F \) for the filters with \( f_0 = 64 \) Hz, 125 Hz, and 250 Hz, and
b) 0.0047 \( \mu F \) for the filters with \( f_0 = 500 \) Hz, 1 kHz and 2 kHz.
Fig. 3.9 Practical active bandpass filter.
It was also decided to design the filters with QS of both 2 and 4. Using the formulas in Eqs. (3-3-13), (3-3-14) and (3-3-15), the values of the resistors were calculated for the filters. The component values are given in Table 3.1a and 3.1b. Trim-pots were selected to cover the resistor values for both $Q = 2$ and $Q = 4$.

3.3.5 Tuning and Testing

After constructing the filter circuits, it is necessary to tune them to get the required $f_0$, $Q$ and $A_V$. Bandpass stage tuning may be a bit difficult because of interactions among the elements, thus one has to achieve the correct values of $f_0$, $Q$ and $A_V$ by an iterative process.

A sine wave signal generator is connected to $V_{IN}$ in Fig. 3.9 and an oscilloscope and frequency counter at $V_{OUT}$. The signal at the output will be largest at about the desired center frequency for each filter. If the center frequency is not exactly on the designed value, $R_2$ is adjusted to obtain the desired center frequency. The gain value is set to the required value by adjusting $R_1$ and $R_3$ since from Eq. (3.3-17) $A_V = R_3/2R_1$. It is preferable to adjust $R_1$ because it has the least effect on the center frequency.

The $Q$ is adjusted at those frequencies which are -3 dB down from the peak response at $f_0$. Those frequencies are

$$f_1 = \frac{-f_0}{2Q} + \frac{f_0}{2Q} \sqrt{1 + 4Q^2} \quad (3.3.18)$$
<table>
<thead>
<tr>
<th>$f_0$ (Hz)</th>
<th>$C_1, C_2$ (µF)</th>
<th>$R_1$ (KΩ)</th>
<th>$R_2$ (KΩ)</th>
<th>$R_3$ (KΩ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>0.01</td>
<td>124</td>
<td>124</td>
<td>1000</td>
</tr>
<tr>
<td>125</td>
<td>0.01</td>
<td>64</td>
<td>64</td>
<td>509</td>
</tr>
<tr>
<td>250</td>
<td>0.01</td>
<td>31.8</td>
<td>31.8</td>
<td>254</td>
</tr>
<tr>
<td>500</td>
<td>0.0047</td>
<td>34</td>
<td>34</td>
<td>270</td>
</tr>
<tr>
<td>1000</td>
<td>0.0047</td>
<td>17</td>
<td>17</td>
<td>135</td>
</tr>
<tr>
<td>2000</td>
<td>0.0047</td>
<td>8.5</td>
<td>8.5</td>
<td>67</td>
</tr>
</tbody>
</table>

Table 3.1a  Component values for $Q = 2$.

<table>
<thead>
<tr>
<th>$f_0$ (Hz)</th>
<th>$C_1, C_2$ (µF)</th>
<th>$R_1$ (KΩ)</th>
<th>$R_2$ (KΩ)</th>
<th>$R_3$ (KΩ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>0.01</td>
<td>248</td>
<td>35</td>
<td>2000</td>
</tr>
<tr>
<td>125</td>
<td>0.01</td>
<td>127</td>
<td>18</td>
<td>1018</td>
</tr>
<tr>
<td>250</td>
<td>0.01</td>
<td>63.7</td>
<td>9</td>
<td>509</td>
</tr>
<tr>
<td>500</td>
<td>0.0047</td>
<td>67.7</td>
<td>9.7</td>
<td>542</td>
</tr>
<tr>
<td>1000</td>
<td>0.0047</td>
<td>34</td>
<td>4.8</td>
<td>270</td>
</tr>
<tr>
<td>2000</td>
<td>0.0047</td>
<td>17</td>
<td>2.4</td>
<td>135</td>
</tr>
</tbody>
</table>

Table 3.1b  Component values for $Q = 4$. 
The test circuit of Fig. 3.10 illustrates the method used to determine the filter Q. A train of impulse functions is applied to the input of the active filter circuit. The impulse makes the filter ring as shown in the figure. The time it takes for a decaying sinusoid to go from 100 percent of its maximum value to 37 percent \((1/e)\) of its maximum value is measured. This is the ring time, \(t_{\text{RING}}\). The Q is determined by the equation

\[
Q = \frac{\pi f_0}{t_{\text{RING}}}
\]

where \(Q = \frac{f_0}{\text{bandwidth}} = \frac{f_0}{(f_2 - f_1)}\) and \(f_1 f_2 = f_0^2\).

The filter circuits were tuned to give a gain \(A_V = 4\), a \(Q = 4\) and the various center frequencies using the procedures outlined above. As mentioned earlier, the correct values were obtained by an interactive process. After tuning the circuits, frequency response data was collected and frequency response curves for all the filter circuits plotted. The plots are shown in Fig. 3.11 and the data in Tables 3.2a-f.

3.4 Sample-and-Hold System

3.4.1 Introduction

The sample-and-hold system converts the analog signals from the filter channels into a constant voltage over the gating-time
Fig. 3.10 Test circuit for measuring $t_{RING}$.
Fig. 3.10 Frequency response of bandpass filters ($Q = 4$)

$\theta$, $l = Q$

$< e.$

$\rightarrow$ FREQ (Hz)

$+10, +5, 0, -5, -10$

GAIN (dB)
Table 3.2 a-f  Data for frequency response of filters with $Q = 4$.

<table>
<thead>
<tr>
<th>FREQUENCY (Hz)</th>
<th>GAIN (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>-6.0</td>
</tr>
<tr>
<td>35</td>
<td>-3.0</td>
</tr>
<tr>
<td>40</td>
<td>0.0</td>
</tr>
<tr>
<td>45</td>
<td>2.3</td>
</tr>
<tr>
<td>50</td>
<td>5.3</td>
</tr>
<tr>
<td>55</td>
<td>8.3</td>
</tr>
<tr>
<td>58</td>
<td>10.1</td>
</tr>
<tr>
<td>60</td>
<td>11.4</td>
</tr>
<tr>
<td>64</td>
<td>12.0</td>
</tr>
<tr>
<td>68</td>
<td>10.8</td>
</tr>
<tr>
<td>70</td>
<td>9.8</td>
</tr>
<tr>
<td>75</td>
<td>7.6</td>
</tr>
<tr>
<td>80</td>
<td>5.0</td>
</tr>
<tr>
<td>90</td>
<td>2.3</td>
</tr>
<tr>
<td>100</td>
<td>0.0</td>
</tr>
<tr>
<td>125</td>
<td>-3.7</td>
</tr>
<tr>
<td>150</td>
<td>-6.0</td>
</tr>
<tr>
<td>175</td>
<td>-8.0</td>
</tr>
</tbody>
</table>

(a) $f_0 = 64$ Hz
<table>
<thead>
<tr>
<th>FREQUENCY (Hz)</th>
<th>GAIN (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>-12.0</td>
</tr>
<tr>
<td>50</td>
<td>-6.0</td>
</tr>
<tr>
<td>60</td>
<td>-4.4</td>
</tr>
<tr>
<td>75</td>
<td>-0.9</td>
</tr>
<tr>
<td>85</td>
<td>1.6</td>
</tr>
<tr>
<td>100</td>
<td>5.6</td>
</tr>
<tr>
<td>110</td>
<td>8.6</td>
</tr>
<tr>
<td>120</td>
<td>11.6</td>
</tr>
<tr>
<td>125</td>
<td>12.0</td>
</tr>
<tr>
<td>130</td>
<td>11.6</td>
</tr>
<tr>
<td>140</td>
<td>9.2</td>
</tr>
<tr>
<td>150</td>
<td>7.2</td>
</tr>
<tr>
<td>175</td>
<td>2.9</td>
</tr>
<tr>
<td>200</td>
<td>0.0</td>
</tr>
<tr>
<td>250</td>
<td>-3.1</td>
</tr>
<tr>
<td>300</td>
<td>-6.0</td>
</tr>
</tbody>
</table>

(b) $f_0 = 125$ Hz
<table>
<thead>
<tr>
<th>FREQUENCY (Hz)</th>
<th>GAIN (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>75</td>
<td>-9.0</td>
</tr>
<tr>
<td>100</td>
<td>-7.0</td>
</tr>
<tr>
<td>125</td>
<td>-4.4</td>
</tr>
<tr>
<td>150</td>
<td>-1.4</td>
</tr>
<tr>
<td>175</td>
<td>1.6</td>
</tr>
<tr>
<td>200</td>
<td>5.6</td>
</tr>
<tr>
<td>220</td>
<td>8.8</td>
</tr>
<tr>
<td>230</td>
<td>10.0</td>
</tr>
<tr>
<td>240</td>
<td>11.7</td>
</tr>
<tr>
<td>250</td>
<td>12.0</td>
</tr>
<tr>
<td>260</td>
<td>11.7</td>
</tr>
<tr>
<td>270</td>
<td>10.6</td>
</tr>
<tr>
<td>280</td>
<td>9.2</td>
</tr>
<tr>
<td>300</td>
<td>6.4</td>
</tr>
<tr>
<td>350</td>
<td>2.3</td>
</tr>
<tr>
<td>400</td>
<td>-0.9</td>
</tr>
<tr>
<td>450</td>
<td>-2.5</td>
</tr>
<tr>
<td>500</td>
<td>-4.4</td>
</tr>
<tr>
<td>600</td>
<td>-7.0</td>
</tr>
</tbody>
</table>

(c) $f_0 = 250$ Hz
<table>
<thead>
<tr>
<th>FREQUENCY (Hz)</th>
<th>GAIN (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>150</td>
<td>-9.0</td>
</tr>
<tr>
<td>200</td>
<td>-6.0</td>
</tr>
<tr>
<td>250</td>
<td>-3.7</td>
</tr>
<tr>
<td>310</td>
<td>0.0</td>
</tr>
<tr>
<td>350</td>
<td>2.6</td>
</tr>
<tr>
<td>375</td>
<td>4.0</td>
</tr>
<tr>
<td>400</td>
<td>6.0</td>
</tr>
<tr>
<td>430</td>
<td>8.3</td>
</tr>
<tr>
<td>450</td>
<td>9.8</td>
</tr>
<tr>
<td>480</td>
<td>11.4</td>
</tr>
<tr>
<td>500</td>
<td>12.0</td>
</tr>
<tr>
<td>520</td>
<td>11.6</td>
</tr>
<tr>
<td>550</td>
<td>10.0</td>
</tr>
<tr>
<td>570</td>
<td>8.8</td>
</tr>
<tr>
<td>600</td>
<td>7.0</td>
</tr>
<tr>
<td>650</td>
<td>4.6</td>
</tr>
<tr>
<td>700</td>
<td>2.6</td>
</tr>
<tr>
<td>800</td>
<td>0.0</td>
</tr>
<tr>
<td>900</td>
<td>-2.0</td>
</tr>
<tr>
<td>1000</td>
<td>-3.7</td>
</tr>
</tbody>
</table>

(d) $f_0 = 500$ Hz
<table>
<thead>
<tr>
<th>FREQUENCY (Hz)</th>
<th>GAIN (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>250</td>
<td>-10.5</td>
</tr>
<tr>
<td>300</td>
<td>-8.0</td>
</tr>
<tr>
<td>500</td>
<td>-3.0</td>
</tr>
<tr>
<td>600</td>
<td>-0.9</td>
</tr>
<tr>
<td>650</td>
<td>1.6</td>
</tr>
<tr>
<td>750</td>
<td>4.6</td>
</tr>
<tr>
<td>800</td>
<td>6.4</td>
</tr>
<tr>
<td>850</td>
<td>8.3</td>
</tr>
<tr>
<td>1000</td>
<td>12.0</td>
</tr>
<tr>
<td>1100</td>
<td>10.0</td>
</tr>
<tr>
<td>1200</td>
<td>6.8</td>
</tr>
<tr>
<td>1250</td>
<td>5.6</td>
</tr>
<tr>
<td>1400</td>
<td>2.9</td>
</tr>
<tr>
<td>1500</td>
<td>2.3</td>
</tr>
<tr>
<td>1750</td>
<td>-0.9</td>
</tr>
<tr>
<td>2000</td>
<td>-3.0</td>
</tr>
<tr>
<td>2500</td>
<td>-6.0</td>
</tr>
</tbody>
</table>

(e) $f_0 = 1$ kHz
<table>
<thead>
<tr>
<th>FREQUENCY (Hz)</th>
<th>GAIN (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.50</td>
<td>-10.5</td>
</tr>
<tr>
<td>0.75</td>
<td>-5.2</td>
</tr>
<tr>
<td>1.00</td>
<td>-2.0</td>
</tr>
<tr>
<td>1.25</td>
<td>2.3</td>
</tr>
<tr>
<td>1.50</td>
<td>6.0</td>
</tr>
<tr>
<td>1.75</td>
<td>10.6</td>
</tr>
<tr>
<td>2.00</td>
<td>12.0</td>
</tr>
<tr>
<td>2.25</td>
<td>9.0</td>
</tr>
<tr>
<td>2.50</td>
<td>5.8</td>
</tr>
<tr>
<td>2.75</td>
<td>3.5</td>
</tr>
<tr>
<td>3.00</td>
<td>2.3</td>
</tr>
<tr>
<td>3.50</td>
<td>-0.9</td>
</tr>
<tr>
<td>4.00</td>
<td>-2.5</td>
</tr>
</tbody>
</table>

\[(f) \quad f_0 = 2 \text{ kHz}\]
interval. This makes it easier to digitize the signals and assures that information is obtained from the six filter channels at the same instantaneous time always.

It was decided that the outputs from the filter channels would be rectified first before being inputted to the sample-and-hold circuits. This should smooth the analog fluctuations (rapid variations) of sound signals and allow one to sample at a lower frequency. The rectifier circuit also eliminates the negative signal, thus allowing one to use a unipolar ADC giving a higher resolution. White and Neely used this method in their experiments [1].

3.4.2 Rectification Stage

The rectifier circuit used in the system is shown in Fig. 3.12a. This consist of a half-wave rectifier, a capacitor to store the energy during the conducting period and to deliver this energy to the output during the inverse, and a resistor which provides a discharge path for the capacitor. The choise of the time constant Z = RC, provided by the resistor-capacitor configuration for each channel, was done by experimental trial. The signal should decay with a time constant long enough to smooth the analog signal to a reasonable degree yet giving it a distinct look. An example is shown in Fig. 3.12b.

3.4.3 Sample-and Hold Fundamentals

A sample-and-hold system tracks an input signal and then holds the instantaneous input voltage upon command by a logic control
Fig. 3.12a  Rectifier circuit

Fig. 3.12b  A rapidly changing wave showing the effect of rectification.
signal. This is illustrated in Fig. 3.13. As can be seen from this figure, \( v_0 = v_i \) during the sampling period and in the hold mode \( v_0 \) remains constant at the value which \( v_i \) attained at the end of the sampling time.

The simplest sample-and-hold circuit is a switch in series with a capacitor as in Fig. 3.14. The voltage across the capacitor tracks the input voltage when a logic control signal closes \( S \) and holds the instantaneous value when \( S \) opens. Two of the factors influencing this circuit are aperture time and acquisition time. (A third factor is the type of hold-capacitor used. This is discussed later.)

![Fig. 3.14 Basic sample-and hold circuit.](image)

Aperture time is the delay between the time the control logic signals switch \( S \) to open and the time this actually happens. When very long (milliseconds) aperture time can be tolerated, \( S \) can be a relay. For short aperture time delays (typically less than 100 ns), a FET controlled by a gating-signal voltage, a diode-bridge gate or a bipolar transistor switch may be used. The aperture time causes a holding error.
Fig. 3.13  Input-output waveforms of a sample-and-hold circuit.
Acquisition time is the time required for the capacitor to change from one level of holding voltage to the new value of input voltage after the switch has closed. The acquisition time is determined by the driving source and charging current \( \frac{C v_0}{dt} < I_{MAX} \), the maximum current available from the source. If \( v_i \) has a source impedance \( R_S \), then \( C \) will charge to the instantaneous value of the input voltage with a time constant \( R_S C \).

A simple practical sample-and-hold system is shown in Fig. 3.15. This circuit has very high input impedance and has an acquisition time determined by the time constant \( R_{ON} C \), provided \( \frac{C v_0}{dt} < I_{MAX} \), where \( R_{ON} \) is the ON resistance of the FET switch and where \( I_{MAX} \) is the maximum current which the input OP AMP follower can deliver. This circuit has the disadvantage that the output cannot be loaded if small decay in HOLD is required.

An improvement of the circuit in Fig. 3.15 is shown in Fig. 3.16. This circuit provides output buffering and eliminates the \( R_{ON} C \) time-constant limitation. Putting switch \( S_1 \) inside the feedback loop enables \( A_1 \) to deliver its maximum current through \( S \) until \( C \) is fully charged. In the HOLD mode, \( S_1 \) is opened and \( S_2 \) is closed. \( S_2 \) provides feedback for \( A_1 \). This sample-and-hold has the advantage that the input impedance is very high.

3.4.4 Sample-and-Hold Circuit

In this project the LF398 monolithic sample-and-hold chips are used. The functional diagram is shown in Fig. 3.17. It can be
Fig. 3.15  Single OP AMP sample-and-hold.

Fig. 3.16  Improved sample-and-hold.
seen that this circuit is similar to the one discussed in Subsection 3.4.3, where switch S₂ of Fig. 3.16 is made of a sampling diode-bridge gate, and switch S₁ is controlled by the logic inputs on pins 7 and 8.

The LF398 uses BI-FET technology to obtain ultra-high dc accuracy (typically 0.002%) with fast acquisition time (as low as 6 μS to 0.01%) and low droop rate (as low as 5 mV/min with a 1 μF hold capacitor). The JFET's also have a low output noise in hold mode. Other features include low input offset voltage and wide bandwidth which are achieved by the use of a bipolar input stage, and a very high input impedance (10¹⁰ Ω) which allows the use of high source impedance to be used without degrading accuracy. The LF398 will operate with ±5 V to ±18 V supplies.

Referring to the functional diagram of the LF398, the sample-and-hold circuits in the project are connected as shown in Fig. 3.18. Since, for proper logic operation, one of the logic pins must always be at least 2 V below the positive supply and 3 V above the negative supply [Appendix C], the logic reference input (pin 7) is grounded. Fast acquisition time is considered here to be a priority with respect to output droop rate, so hold capacitors with value 0.001 μF are used. From the graphs in Fig. C1 this value should give an acquisition time of 5 μS to 0.01% of the input, and an output droop rate of about 2 x 10⁻² V/sec. The actual droop rate was measured to be 1.7 x 10⁻² V/sec which agrees with the graph. The choice of the type of hold capacitor is discussed in the next section.
Fig. 3.17 Functional diagram of the LF398.
Fig. 3.18  Rectifier and sample-hold circuits.
3.4.5 Hold Capacitor

As mentioned above, hold step, acquisition time, and droop rate are the major considerations in the selection of a hold capacitor. The curves in Appendix C should help in reaching a compromise for choosing a suitable value.

It is recommended that a capacitor with polycarbonate, polyethylene, Mylar, Mica or Teflon dielectric be used. In other capacitors a polarization phenomenon causes the stored voltage to decay with a time constant of several seconds.

Another significant error in an accurate sample-and-hold is dielectric absorption in the hold capacitor. This causes a capacitor to "remember" a fraction of its previous charge after a quick change in voltage. Dielectric with very low hysteresis, such as polystyrene, polypropylene, and Teflon are preferred due to this phenomenon. Mylar has higher hysteresis than the above mentioned dielectrics. The hysteresis error can be reduced (by up to a factor of 10) if the output of the LF398 is digitized quickly (within 1 ms) after the hold signal is applied.

After reviewing the above considerations, and due to availability, hold capacitors with Mylar dielectric are used.

3.5 Time and Control Circuit

3.5.1 Introduction

As mentioned, it was decided to make the speech recognition system be able to act as a stand alone system. This will allow it to be transferable to any microcomputer which has a A-D converter.
(The ADC could alternatively be added to our system easily.) The microcomputer will furnish the minimum or no timing or synchronization functions. Thus all timing and control signals must be furnished by hardware.

The timing and control circuit must identify when a word is present at the input and then provide the following functions: Supply the sample-and-hold control logic for the sample-and-hold circuits; control the conversion process; and select the sample filter channels and connect them to the ADC for conversion in an orderly manner.

3.5.2 Circuit Design

Referring to the requirements enumerated in Subsection 3.5.1, the timing circuit shown in Fig. 3.19 was designed. The explanation of how it works follows.

The WORD PRESENT signal, produced by the signal detector explained in Section 3.2, initiates a chain of events. WORD PRESENT will stay HIGH as long as the analog input level is above the reference level to which the signal detector is set. ICa is a 74LS123 dual one-shot set up as an astable multivibrator. When WORD PRESENT is HIGH it produces a pulse train with pulse width of 5 μS. This is the sample-hold control signal. The sampling frequency is also determined here. It is adjustable with the trim-pot, R_F. (The description of the 74LS123 and its various configurations and typical applications are given in Appendix D.)

WORD PRESENT going HIGH also triggers ICb which is another 74LS123. The first one-shot in the IC provides a single 5 μS pulse,
which coincides with the first $5 \mu S$ pulse of the sample-hold control logic, thus conversion will not start until the first value is sampled. The falling-edge of this pulse triggers the second one-shot which then outputs a $5 \mu S$ pulse termed START CONVERSION. The START CONVERSION signal causes the ADC to begin converting the analog signal present at its input to a binary value.

After the ADC has finished converting the signal, it issues an END OF CONVERSION (EOC) signal. Depending on the design of the ADC interface to the computer, the EOC signal may be used to signal the computer to read the converted value and store it. After the computer has read that value, it provides a pulse to pin 10 of ICb which retriggers the second one-shot to give another START CONVERSION signal. Since the EOC signal is negative going, its rising edge may be used to toggle the second one-shot instead of using a signal from the computer (Fig. 3.19). The falling edge of the EOC is then used as clock signal for ICc as discussed below.

The filter channel converted by the ADC is selected by ICc-e. ICc is a counter which provides the multiplexer select signals. The sample pulses of the sample-hold control signal reset the counter. Thus every time the signal is sampled the counter is reset to 0, so the ADC always starts conversion of a sampled value with Channel 0. The counter clock input is provided by the computer, after it has read and stored a converted value, or may be provided by the falling edge of the EOC. The low three bits of the output of ICc are used as MUX SELECT to select one of six channels at the input of the multiplexer ICe.
Fig. 3.19  Timing and control circuit.
The MUX SELECT lines also provide the input for ICd which is a 3-to-8 decoder. It detects when the count at the output of ICc has reached 6 (thus Channels 0-5 have been "scanned") and disables the second one-shot in ICb thus stopping the generation of START CONVERSION pulses until the counter has been reset.

As long as WORD PRESENT remains high, the events described are executed repeatedly. The channels from the output of the sample-and-hold circuits are sampled, converted and passed to the computer in an orderly manner. It should be noted that the threshold set to trigger the signal detector must be very low, or else using WORD PRESENT alone could cause problems when sampling words with stop consonants such as "eight", where the unvoiced "t" could fall below a threshold and thus cause WORD PRESENT to go low. Another method to solve this problem is to have the rising edge of WORD PRESENT trigger a monostable circuit which will remain high for the duration required for sampling a word. (This assumes that each utterance is sampled for a fixed time regardless of its duration.)
In searching for a speech recognition program, which would give a reasonable and acceptable performance, experiments were performed in order to evaluate the various methods in the stages of speech recognition. The experiments in this chapter use known experimental data to represent the vocabulary words and the unknown word in a recognition system, and evaluate methods of template matching, amplitude normalization, and word identification.

The experiments performed and the programs used are explained in the following sections.

4.1 Template Matching

As mentioned in Section 2.2, matching of each of the vocabulary templates and the unknown word, in a system with limited vocabulary, is based on some simple distance measures. Three of these distances measures are evaluated from the results generated by Program ABl shown in Fig. 4.1.

4.1.1 Program ABl

Program ABl is used as a basis for the experiments in template matching and amplitude normalization. Apart from generating experimental data to represent the vocabulary and unknown word, it contains the minimum elements necessary for speech recognition.

The data is obtained from arbitrarily chosen signals that represent speech signals. The signals for two vocabulary words are shown in Fig. 4.2. Each word has three components representing
signals from three filters. Twenty data points are taken from each of
the components to represent the data obtained from sampling speech
signals.

The data for the unknown word is formed from vocabulary word 1 by
multiplying the later by a factor which may be 100 for an exact match
or less than 100 to give a difference in amplitude so as to see the
effects of varying loudness when speaking the same word twice.

Lines 40-110 form data arrays of the unknown word and vocabulary
words. Template matching is done in lines 700-800, the distance
measure being calculated in line 770. Lines 900-970 perform the word
identification which, in this case, is simply the vocabulary word with
the smallest difference.

4.1.2 Chebyshev Distance Measure

Chebyshev distance is the sum of the absolute value of the
differences of the coordinate values between the unknown word and a
template word. If we have an N-dimensional vector space and \( X_i \) repre­1.sents the template sample at time slice i and \( Y_i \) represents the
unknown utterance, i.e. \( X_i = (x_{i1}, \ldots, x_{iN}) \) and \( Y_i = (y_{i1}, y_{i2}, \ldots, y_{iN}) \), then the Chebyshev distance between \( X_i \) and \( Y_i \) is given
by

\[
d_i = \sum_{j=1}^{N} |x_{ij} - y_{ij}|
\]

(4.1.1)

where \( x_{ij} \) and \( y_{ij} \) are the jth-component, at time slice i, of the
template sample and unknown utterance sample respectively. This
method is often used because of its computational simplicity.
10 REM PROGRAM ABl
20 REM BASIC RECOGNITION PROGRAM
30 REM TO EXPERIMENT WITH DISTANCE MEASURES
31 REM AND AMPLITUDE NORMALIZATION
35 DIM X(5,60),Y(3)
40 REM FORM TEMPLATE
45 M=1:REM VOCABULARY "WORD" 1
50 DATA 0,40,70,102,120,112,100,78,56
51 DATA 74,88,58,12,22,22,4,10,8,4,8
52 DATA 0,20,40,26,0,12,22,18,10,30
53 DATA 70,104,120,112,90,80,30,10,0,0
54 DATA 0,10,24,20,14,6,0,12,26,32
55 DATA 26,18,0,4,10,5,4,0,0
56 FOR I=0 TO 59
57 READ A
58 X(M,I)=A
59 NEXT I
60 M=2:REM VOCABULARY "WORD" 2
70 DATA 0,25,40,30,20,40,60,40,0,60,100
71 DATA 90,73,80,40,70,120,110,90,70
72 DATA 0,10,20,10,0,20,40,26,17,30,40
73 DATA 20,0,10,20,10,0,6,16,20
74 DATA 20,30,25,15,12,20,28,15,0,6,22
75 DATA 20,7,20,40,30,10,0,5,10
76 FOR I=0 TO 59
77 READ A
78 X(M,I)=A
79 NEXT I
80 M=0:REM FORM "UNKNOWN WORD"
81 INPUT "% OF VOC. WORD ":P
82 P=P/100
83 FOR I=0 TO 59
84 X(M,I)=P*X(I,I)
85 NEXT I
86 OPEN 4,4,I
700 REM RECOGNITION ROUTINE
705 REM USING CHEBYSHEV DISTANCE MEASURE
710 FOR J=1 TO 2
720 Y(J)=0
730 NEXT J
740 FOR J=1 TO 2
750 FOR I=0 TO 60
760 D=ABS(X(I,J)-X(J,I))
770 Y(J)=Y(J)+D
780 NEXT I
790 PRINT J,Y(J)
791 PRINT#4,J,Y(J)
800 NEXT J

Fig. 4.1 Program ABl.
170 REM AMPLITUDE NORMALIZATION
171 REM MEAN NORMALIZATION METHOD
180 FOR M=0 TO 2
190 FOR J=0 TO 19
200 SUMX=0
250 FOR I=J TO 60 STEP 20
300 SUMX=SUMX+X(M,I)
350 NEXT I
351 SUMX=SUMX/3
400 FOR I=J TO 60 STEP 20
410 IF SUMX=0 GOTO 550
450 X(M,I)=X(M,I)/SUMX
500 NEXT I
550 NEXT J
600 NEXT M

Fig. 4.1 (cont.)
Fig. 4.2 Arbitrary signals for experiments.
4.1.3 Euclidean Distance Measure

Euclidean distance is the square-root of the differences squared between the component values of the unknown and a template. It measures distances in spaces with more than two dimensions (six in this case) and tends to exaggerate differences between unlike parameters.

Suppose in an N-dimensional space \( X_i \) represents the template sample at time slice \( i \) and \( Y_i \) represents the unknown utterance sample, that is, \( X_i = (x_{i1}, \ldots, x_{iN}) \) and \( Y_i = (y_{i1}, \ldots, y_{iN}) \), then the Euclidean distance between \( X_i \) and \( Y_i \) is

\[
\begin{align*}
    d_i &= \left[ \sum_{j=1}^{N} (x_{ij} - y_{ij})^2 \right]^{1/2} \\
    &= \left[ \sum_{j=1}^{N} (x_{ij} - y_{ij})^2 \right]^{1/2} \\

\end{align*}
\]

(4.1.2)

Euclidean distance is obtained by changing line 770 to read:

770 \( Y(J) = Y(J) + D \ast D \)

To save time, the square root operation is not performed since it doesn't change the relative order of the measures.

4.1.3 Polynomial Distance Measure

Squaring the differences in the Euclidean measure, emphasizes those measurements which differ by the greatest amount. Thus a point which varies considerably from its expected value can contribute a large amount to the computed difference even though the bulk of the
points examined match quite well. The Polynomial measure attempts to improve on this situation by rewarding small differences and punishing large differences. The measure is of the form:

\[ P_i = \sum_{j=1}^{N} (x_{ij} - y_{ij})^2 - F \cdot (x_{ij} - y_{ij}) \]  \hspace{1cm} (4.1.3)

where \( P \) is the Polynomial distance, in an \( N \)-dimensional space, between a template word represented by the component \( x_j \) (\( j = 1 \) to \( N \)) at time slice \( i \) and an unknown word represented by the component \( y_j \) (\( j = 1 \) to \( N \)). \( F \) is a small factor.

This measure is obtained by changing line 770 in Program ABl to read

770 \hspace{1cm} Y(J) = Y(J) + D \cdot D - F \cdot D

and adding line 705

705 \hspace{1cm} INPUT F

Various values of \( F \), including 0, 1, 3 and 5 are tried. The optimal value for \( F \) is a matter for experimentation and depends on the vocabulary.

4.1.5 Experimental Results

As explained in Subsection 4.1.1, the vocabulary template consists of two words: word 1 and word 2. Experiments are carried out with the unknown word generated to be equal to 100% and 95% amplitude of word 1. Program ABl is run with the various distance measure methods discussed above. The results are shown in Table 4.1. The numbers on top are the actual distances between the unknown word and
each template word as measured by the methods. Since the distances measured are computed differently by each method, we cannot use these actual distances to compare one method to the other. In order to do this, the actual distances are converted to a common base in such a way that the relative order of the measures remain unchanged.

Conversion of the actual distances measured is obtained as follows: a perfect match is given a value of 0. (This is the case when the unknown word is equal to 100% of word 1 and the distance measured between them is 0.) The distance measured between the unknown word, when it is 100% of word 1 and word 2 using each method, is given a value of 100 and all the other distances measured for that method are calculated relative to it. The formula is

\[ D = \frac{D}{D_M} \times 100 \]  

(4.1.4)

where \( D_M \) is the distance between the unknown word and word 2 when the unknown is 100% of word 1, \( D \) is an actual distance measured, and \( D \) the converted value of the distance measured.

The converted distances are shown in parenthesis under the actual distances. They show how close the unknown word is to the template words in the three-dimensional space defined by the three components of each word, and since they are on a common base we can compare these numbers directly in order to compare the distance measure methods. When the unknown word is 95% of word 1 (or, in general, when the two are not equal in amplitude), the method that gives a better match is the one with the converted distance between the unknown and word 1 closer to 0.
Table 4.1 Results for distance measure method.

<table>
<thead>
<tr>
<th>DISTANCE MEASURE</th>
<th>WORD</th>
<th>100% OF AMPLITUDE</th>
<th>95% OF AMPLITUDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chebyshev</td>
<td>1</td>
<td>0 (0)</td>
<td>98 (5.20)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1892 (100)</td>
<td>1835 (97.00)</td>
</tr>
<tr>
<td>Euclidean</td>
<td>1</td>
<td>0 (0)</td>
<td>362 (0.30)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>121408 (100)</td>
<td>113577 (93.50)</td>
</tr>
<tr>
<td>Polynomial (N = 3)</td>
<td>1</td>
<td>0 (0)</td>
<td>66 (0.06)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>111948 (100)</td>
<td>108072 (96.53)</td>
</tr>
</tbody>
</table>
Looking at the actual distance measures for both the 100% and 95% of amplitude, it can be seen that the distances using the Euclidean norm are large. This shows the exaggeration of differences between unlike parameters when using the Euclidean distance measure, which accounts for its better performance relative to Chebyshev norm as can be seen by comparing the converted distances under the 95% of amplitude column.

It can also be seen that the Polynomial measure provides better matching relative to the Euclidean norm. This, as explained in Subsection 4.1.4, is because the Polynomial measure rewards small differences while punishing large differences. Optimal value for F is found to be 3. Note that this will be different for a different vocabulary.

4.1.6 Time Normalization

One problem which should be discussed here is time alignment. Two separate utterances of the same word will, in general, suffer from misalignment due to poor detection of the beginning and endings of utterances, and the timing of their constituent sounds (non-linear time translations). Thus, if we line up the beginnings of words before computing differences, time-dependent features may be extracted incorrectly or a large value may be computed as the distance measure because the unknown word and the template words are out of time registration.

Dynamic programming reduces non-linear translation, and thus achieves better interior matching [3]. But because of the extensive
computation required, it is not discussed or investigated in this thesis.

One method of overcoming misalignment due to poor detection of the beginning of utterance is translation. An attempt is made to get better fits, i.e. smaller difference measures, by sliding the unknown utterance against the template until we find the smallest difference.

Program AB2 shown in Fig. 4.3 is used to investigate this method. Lines 10-150 generate experimental data to represent words. The unknown word is translated forward and backward, between -S and +S, against each of the vocabulary word until the smallest difference is found. This is accomplished in Lines 740-786. A sample run of Program AB2 for S = 2 is shown in Fig. 4.4. The value J identifies the word: 0 unknown, 1 and 2 vocabulary.

A variation of the above method is to find the first peak of the unknown word and, knowing the time of the first peak of a template word, determine the difference, . The unknown is then shifted by so that the first peaks coincide. This is shown in Fig. 4.5. The method is used in the experiments of Section 4.3.
1 REM PROGRAM AB2
2 REM EXPERIMENT WITH TRANSLATION
5 DIM X(3,8),Y(3)
10 DATA 1,1,3,2,2,1,1
20 FOR I=0 TO 6
30 READ A
40 X(0,I)=A
45 PRINT X(0,I)
50 NEXT I
55 PRINT
60 DATA 1,3,2,2,1,0
70 FOR I=0 TO 6
80 READ B
90 X(I,I)=B
95 PRINT X(I,I)
100 NEXT I
105 PRINT
110 DATA 0,0,1,2,2,3,3
120 FOR I=0 TO 6
130 READ C
140 X(2,I)=C
145 PRINT X(2,I)
150 NEXT I
155 PRINT
700 REM SLIDING AND RECOGNITION
740 FOR J=1 TO 2
741 I=2
742 PRINT "Z=" ,I
744 F=300000
745 FOR K=-I TO I
746 Y(J)=0
750 FOR I=ABS(K) TO 6-ABS(K)
760 D=ABS(X(0,I+K)-X(J,I))
770 Y(J)=Y(J)+D
780 NEXT I
781 PRINT J,Y(J)
782 IF Y(J)<F THEN F=Y(J)
783 NEXT K
784 Y(J)=F
789 PRINT
790 PRINT "FOR J=" ,J," MIN. DIFF.=" ,Y(J)
800 NEXT J
999 END

Fig. 4.3 Program AB2: Time normalization using translation.
Fig. 4.4 Output of Program AB2.
Fig. 4.5 Example of misalignment due to poor detection of beginning of utterance.
4.2 Amplitude Normalization

Normalization compensates for signal level differences normally encountered in two or more utterances of the same word. In this section, a few normalization methods are compared in performance accuracy. Program AB1 (Fig. 4.2) is used in the experiments with Euclidean distance measure.

4.2.1 Mean Normalization

In this method, each time slice is normalized independently of the other time slice in the utterance by dividing the value of each component of the slice by the mean of all the other components in that slice. The equation to realize mean normalization in an N-dimensional space is given by

\[ v_{ij} = \frac{u_{ij}}{\bar{u}_i} , \quad j = 1, 2, \ldots, N \quad (4.2.1) \]

where \( u_{ij} \) is the unnormalized component \( j \) of the sample at time slice \( i \), \( v_{ij} \) is the normalized component, and \( \bar{u}_i \) is the mean of all the components in the sample and is given by

\[ \bar{u}_i = \left( \frac{\sum_{j=1}^{N} u_{ij}}{N} \right) / N \quad (4.2.2) \]

This normalization method is achieved by inserting lines 170-600 in Fig. 4.6 into Program AB1. Program AB1 is then run varying the amplitude of the unknown "utterance". The results of this and subsequent methods are discussed in Subsection 4.2.4.
170 REM AMPLITUDE NORMALIZATION
171 REM MEAN NORMALIZATION METHOD
180 FOR I=0 TO 2
190 FOR J=0 TO 19
200 SUMX=0
250 FOR I=J TO 60 STEP 20
300 SUMX=SUMX+X(N,I)
350 NEXT I
351 SUMX=SUMX/3
400 FOR I=J TO 60 STEP 20
410 IF SUMX=0 GOTO 550
450 X(N,I)=X(N,I)/SUMX
500 NEXT I
550 NEXT J
600 NEXT N

Fig. 4.6 Mean normalization routine for Program AB1.
4.2.2 Proportional normalization

In this method normalization is achieved by using the following equations:

\[ v_{ij} = u_{ij} + (u_{6j} - L)(M - \hat{u}_6) / (\hat{u}_6 - L), \quad \text{if } u_{6j} > L \]  

\[ v_{ij} = u_{ij}, \quad \text{if } u_{6j} \leq L \]

where \( u_{ij} \) represents the unnormalized data, \( j \) is the index of spectral time samples, \( i = 1, 2, \ldots, 6 \) is the parameter index. \( u_{6j} \) denotes the overall amplitude value of the sample \( j \) and \( \hat{u}_6 \) is the maximum amplitude value of each utterance, i.e.

\[ \hat{u}_6 = \max_j u_{6j} \]

\( v_{ij} \) represent the normalized values, and \( L \) and \( M \) are two values chosen as follows. \( L \) represents the silence level, and from equations (4.2.3) and (4.2.4) no normalization is performed when the signal falls below this level. \( M \) is chosen to satisfy \( M > \hat{u}_6 \) for all utterances.

From equation (4.2.3) it is seen that the maximum amplitude value \( \hat{u}_6 \) of each utterance is moved up by the normalization procedure to the constant value \( M \), and the other time samples are proportionately moved up.

Proportional normalization is accomplished by inserting the lines of program in Fig. 4.7 into Program ABl. Note that each utterance is scanned to determine its maximum amplitude value.
4.2.3 Peak Normalization

In this technique, normalization is achieved by finding the highest peak in the sound signal and storing all values relative to this. This method is similar to the proportional normalization method and is based on the theory that when measured in decibels (dB), the intensities of the formants maintain about the same relative intensities regardless of their absolute value.

When using the decibel scale the peak value in an utterance is found and all other values are subtracted from it. On a linear scale the same result is achieved by dividing each value by the peak value.

Linear scale normalization is achieved by the equation

$$v_{ij} = \frac{u_{ij}}{\hat{u}_j}, \quad j = 1, 2, \ldots, N$$  \hspace{1cm} (4.2.5)

where $u_{ij}$ is the unnormalized component $j$ of the sample at time slice $i$, $v_{ij}$ is the normalized component, and $\hat{u}_j$ is the maximum amplitude value of component $j$ of each utterance which is determined by scanning that component. Lines 170-600 in Fig. 4.8 are added to Program ABI to realize peak normalization.

4.2.4 Experimental Results

The experiments are carried out with the unknown word equal to 100%, 80% and 60% of the identical vocabulary word. Program ABI is first run with no normalization of data for comparison. It is then run with the normalization methods described to test their effectiveness. In all cases the Euclidean distance measure is used.
REM AMPLITUDE NORMALIZATION
REM PROPORTIONAL NORMALIZATION METHOD
REM TO FIND OVERALL AMPLITUDE VALUE OF EACH SAMPLE
FOR M=0 TO 2
FOR J=0 TO 19
SUMX(J)=0
NEXT J
FOR J=0 TO 19
FOR I=J TO 59 STEP 20
SUMX(J)=SUMX(J)+X(M,I)
NEXT I
NEXT J
FOR J=0 TO 19
IF SUMX(J)>MAX THEN MAX=SUMX(J)
NEXT J
REM TO FIND MAXIMUM AMPLITUDE VALUE
L=20
N=765
FOR J=0 TO 19
FOR I=J TO 59 STEP 20
IF SUMX(J)<L GOTO 470
X(M,I)=X(M,I)+(SUMX(J)-L)*(N-MAX)/(MAX-L)
NEXT I
NEXT J
NEXT M

Fig. 4.7 Porportional normalization routine for Program AB1.

REM AMPLITUDE NORMALIZATION
REM PEAK NORMALIZATION METHOD
FOR M=0 TO 2
FOR J=0 TO 19
Q=0
FOR I=0 TO 19
IF X(M,I+J*20)>Q THEN Q=X(M,I+J*20)
NEXT I
NEXT J
FOR I=0 TO 19
X(M,I+J*20)=X(M,I+J*20)/Q
NEXT I
NEXT J
NEXT M

Fig. 4.8 Peak normalization routine for Program AB2.
The results are shown in Table 4.2. The numbers in parenthesis represent the actual distances converted to a common base using the method described in Subsection 4.1.5. From this data we can see how close the unknown word is to the vocabulary words. For ideal amplitude normalization the distance measure between the unknown word and vocabulary word 1 should be zero, and constant (i.e. converted distance equal to 100) between unknown word and vocabulary word 2.

From the results it is seen that the peak normalization method is more effective, since the converted distances between the unknown word and word 1, when the unknown word is equal to 80\% and 60\% of word 1, are closer to zero (thus a better match) than the distances using other methods and are constant between the unknown word and word 2. The other methods also give acceptable results since the distances between the unknown word and word 1 are relatively small (not far from zero) for all the methods.

With no normalization the error rate is higher, and the program may recognize a wrong vocabulary word when the unknown utterance signal level is different from the template utterance, especially with a large vocabulary.

The peak normalization technique is used in the speech recognition program because it can be carried out in synchrony with the sampling of each channel, that is, independently of the other channels (components), and thus will be suitable to use with the methods of data collection that will be used. In the recognition system, data will be collected from only two channels per utterance of the same
<table>
<thead>
<tr>
<th>NORMALIZATION METHOD</th>
<th>WORD</th>
<th>DISTANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>100% OF AMPLITUDE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100% OF AMPLITUDE</td>
</tr>
<tr>
<td>NO NORMALIZATION</td>
<td>1</td>
<td>0 (0)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>121408 (100)</td>
</tr>
<tr>
<td>MEAN</td>
<td>1</td>
<td>0 (0)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>79.93 (100)</td>
</tr>
<tr>
<td>PROPORTIONAL</td>
<td>1</td>
<td>0 (0)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3672566 (100)</td>
</tr>
<tr>
<td>PEAK</td>
<td>1</td>
<td>0 (0)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>12.55 (100)</td>
</tr>
</tbody>
</table>

Table 4.2 Results for amplitude normalization experiments.
word. Using the proportional normalization method, the whole utterance has to be examined for the maximum amplitude value, thus all the channels have to be sampled before the normalization procedure can be applied. This is also the case in the mean normalization method where each component of a sample is normalized with respect to the other components in that sample.

The peak normalization method is also easier to implement in assembly language.

4.3 Word Identification

After template matching between an unknown word and the vocabulary words is done using one of the distance measure methods, a decision has to be made as to which vocabulary word is the closest match to the unknown word. Various decision (or identification) strategies have been proposed and used. The decision strategy and template matching usually go hand in hand.

The simplest template matching and word identification method was used in the experiments of Sections 4.2 and 4.3. In the template matching stage each time sample of the unknown word is compared to the same time sample of each of the template words. The differences of all the time samples are accumulated to form a single difference for each template word. In the word identification stage, the template word with the lowest accumulated difference is identified as the unknown word.

This method may not be sufficient. Two methods are proposed and investigated in this section.
4.3.1 Experimental Data

The data for the experiments in this section is obtained from arbitrarily chosen signals to represent speech signals. These are shown in Fig. 4.9. Three signals, with three components each, are chosen to represent three template words. A fourth signal represents the unknown word. This is similar to template word 1 except that it has a slightly different amplitude at sample points (to simulate variation in signal level difference), and it is shifted with respect to time (to simulate misalignment due to poor detection of the beginning of utterances). Data points are taken from these signals to represent the data obtained from sampling speech signals.

The data for the template words is normalized and saved as data files on the PET computer cassette. This, in the real speech recognition system, will allow us to store a large set of templates which would otherwise not fit into our system memory. The data for each template word could be loaded from cassette into memory, compared to the unknown word, and discarded to give room for the next template word. Only the result of the comparison is kept in memory. The data representing the unknown word is generated within the programs used.

4.3.2 Method 1

This method is based on the suggestion that for a particular set of vocabulary words there may be a filter channel output whose characteristics are unique for each of the words. If, by observing the signals for each channel for each word, say on an
Arbitrary signals for experiments.
Fig. 4.9 (cont.)
oscilloscope, then template matching could be based on only this channel, and word identification could be made solely by the results obtained. The results must be unambiguous to confidently make a decision. If a decision cannot be made then the other channels will have to be matched. If a decision could be made by considering only one channel, the recognition procedure would be speeded up considerably.

In the experiment performed, the first channel of the unknown word is compared with the first channel of each of the template words. The other channels are then compared if a decision cannot be made based on the first comparison. The flowchart in Fig. 4.10 shows this method. Program AB3 in Fig. 4.11 implements the flowchart with some slight omissions, specifically the decision stage after matching the first channel. Template matching is done in Lines 600-720 and includes sliding of the unknown word data so as to compensate for the shifting of the signal due to poor detection of its beginning. This has been explained in Subsection 4.1.6. Peak normalization is implemented in Lines 800-910.

The printout from a run of Program AB3 is shown in Fig. 4.12. The difference for one channel for each word and the difference for all three channels for each word are shown. From these results we see that a decision could have been made confidently after matching only the first channel, identifying word 1 as being the correct match. Using actual speech data an estimate of the closeness of a correctly identified word could be found for each set of vocabulary by experimentation. Based on this value an identification decision could be
START

Get data for unknown word

Amplitude Normalize

Get data for Channel 1 for template word M

Compare with Channel 1 of unknown word. Difference = Y(M)

Any more template words?

Y

N

Find minimum difference

Can identification decision be made?

N

Y

Make decision

END

Fig. 4.10 Flowchart for word identification method 1.
Get data for all channels for template word

Compare with all channels of unknown word

Any more template words?

Find minimum difference

Make a decision

END
10 REM PROGRAM AB3
15 REM RECOGNITION EXPT. WITH TEMPLATE ON CASSETTE
20 DIM X(60),Z(60),Y(4)
30 REM GET DATA FOR SIGNAL TO BE RECOGNIZED
40 DATA 0,14,20,10,2,40,70,102,120,112
41 DATA 100,60,56,70,66,80,12,20,24,4
42 DATA 40,30,20,4,0,22,45,30,0,10
43 DATA 22,20,10,20,106,124,115,100
44 DATA 62,36,30,20,10,0,12,50,24,18,6
45 DATA 2,12,28,34,25,12,2,8,10,11
50 FOR I=1 TO 60
60 READ A:X(I)=A
70 NEXT
90 M=0:C=1
100 GOSUB 800:REM PEAK NORMALIZE
110 REM GET WORD 1 FIRST CHANNEL &COMPARE
120 M=1:C=1:F=0
140 OPEN 4,1,0
150 GOSUB 500:REM INPUT DATA
160 CLOSE 4
180 INPUT"INPUT S: ",S
200 GOSUB 600:REM COMPARE WITH UNKNOWN
210 M=M+1
220 IF M<5 GOTO 140
230 IF F=1 GOTO 999
300 REM GET DATA FOR ALL CHANNELS & COMPARE
310 PRINT"REWIND TAPE & PRESS ANY KEY TO CONTINUE"
320 GET A#: IF A#=""THEN 320
330 M=1:C=3:F=1
340 GOTO 140
500 FOR I=1 TO 20*C
510 INPUT#4,A
520 Z(I)=A
530 NEXT
540 RETURN
600 F=1000000:SS=S
604 FOR K=-5 TO S
605 Y(H)=0
610 FOR I=1 TO 20*C
615 FOR J=ABS(SS) TO (I9-ABS(SS))
620 D(H)=ABS(X(I*20)+(J+K))-Z(I*20+J))
630 Y(H)=Y(H)+D(H)*D(H)
635 NEXT J
640 NEXT I
641 IF Y(H)<F THEN F=Y(H):MS=K
642 NEXT K
643 Y(H)=F
650 PRINT

Fig. 4.11 Program AB3: Recognition experiment.
660 PRINT"FOR",C,"CHAINLS,WORD",M,"DIFF.="",Y(M)
670 OPEN 4,4,1
680 PRINT#4
690 PRINT"FOR",C,"CHAINLS,WORD",M
700 PRINT#4,"DIFFERENCE="",Y(M)
710 CLOSE 4
720 RETURN
800 Q=0
810 FOR I=1 TO 20*C
820 IF X(I)>Q THEN Q=X(I)
830 NEXT I
840 FOR I=1 TO 20*C
850 X(I)=X(I)/Q
860 NEXT I
870 RETURN
999 END

Fig. 4.11 (cont.)
<table>
<thead>
<tr>
<th>S</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOR</td>
<td>1</td>
</tr>
<tr>
<td>DIFFERENCE</td>
<td>CHANNEL</td>
</tr>
<tr>
<td>MIN. DIFF. AT S</td>
<td>4</td>
</tr>
<tr>
<td>FOR</td>
<td>1</td>
</tr>
<tr>
<td>DIFFERENCE</td>
<td>.579697585</td>
</tr>
<tr>
<td>MIN. DIFF. AT S</td>
<td>-4</td>
</tr>
<tr>
<td>FOR</td>
<td>3</td>
</tr>
<tr>
<td>DIFFERENCE</td>
<td>CHANNEL</td>
</tr>
<tr>
<td>MIN. DIFF. AT S</td>
<td>4</td>
</tr>
<tr>
<td>FOR</td>
<td>3</td>
</tr>
<tr>
<td>DIFFERENCE</td>
<td>.70759895</td>
</tr>
<tr>
<td>MIN. DIFF. AT S</td>
<td>-4</td>
</tr>
</tbody>
</table>

Fig. 4.12 Output of Program AB3.
made after only the first match, or a decision to match all the channels made.

4.3.3 Method 2

This method is based on a method used by Shearme and Leach [2] to identify a word from matching scores in a speaker independent speech recognition system. In their experiments, a number of templates representative of each word to be recognized, as spoken by different talkers, are made. When an input word is spoken it is compared with all the templates of each vocabulary word. A majority vote of the N best fits is taken, with votes weighted inversely as the score. When each vocabulary word was represented by nine templates, the optimum value for N was found to be about 4.

In this thesis it is proposed to match the unknown word with the vocabulary words keeping the difference for each channel separately. A majority vote of the N best fitting channels (lowest scoring channel) is then taken as explained above. The word with the highest best fitting channels is then identified. The optimum value for N, for any number of channels, can be found experimentally. A slight elaboration on this, which can speed up the recognition process, is to match the first N channels and take a majority vote. If one of the words score N best fits then it is immediately identified. If none of the words has N best fits then the remaining channels are matched but only for the candidate words; that is, those words which could have N best fits after the second matching. Those words with none or low best fits are thrown away.
The flowchart in Fig. 4.13 shows the method where \( N \) is the number of channels to be compared in the first match and also the optimum value of best fits for correct recognition, \( V \) is the total number of channels per word, and \( Y(M,J) \) is the difference for channel \( J \) of word \( M \). This flowchart is implemented for our experiment by Program AB4 in Fig. 4.14.

Lines 300-400 of Program AB4 take the majority vote for best fits after matching 2 channels. Lines 405-430 find the word with maximum best fits and if this is 2 the word is identified as the correct word. If it is not then the words with \( N < 1 \) are thrown out. This is done in Lines 435-460. The remaining channels for the candidate words are then compared and the word with the overall maximum best fits is identified as the correct word. Two outputs of Program AB4 are shown in Fig. 4.15 (a) and (b). The first shows a decision taken after the first matching. In the second case Lines 405-430 and Line 445 are removed, thus no decision is taken after the first match, and all the words are compared in the second match.

A modification of Program AB4 could be made by inserting the lines in Fig. 4.16. This will change the time alignment method from sliding to get the minimum difference, to aligning the first peaks of the input word and template word as described in Subsection 4.1.6. A printout of the modified program is shown in Fig. 4.17.
START

Get input word

Peak normalize

Number of channels you wish to compare? C

Get template word and compare for each channel. Keep Y (M, J)

More template words?

Y

N

Find best fits for each word. NN(J)

Find max best fit, MBF

MBF > N? Y N

Fig. 4.13 Flowchart for word identification method 2.
A

Throw away words with NN(J) + (V-C) < N

Get (V-C) channels for candidate word

Compare each channel
Keep (M, J)

More candidate words?

Y

N

Find best fits for each word NN(J)

Find word with max NN(J)
Identify

B

END RECOGNITION

Fig. 4.13 (cont.)
10 REM PROGRAM AB4
20 REM IDENTIFICATION OF WORD FROM MATCHING SCORES
30 REM TEMPLATE ON CASSETTE
31 REM IMPROVEMENT ON PROG. ABB
35 DIM X(60),Z(60),Y(3,3),D(3,3),N(3),T(3)
40 REM GET DATA FOR SIGNAL TO BE RECOGNIZED
50 DATA 0,0,0,0,0,40,70,102,120,112
51 DATA 100,80,56,70,86,60,12,20,24,4
52 DATA 0,0,0,0,0,22,45,50,0,10
53 DATA 22,20,10,50,70,106,124,115,100
54 DATA 62,0,0,0,0,0,12,30,24,16,6
55 DATA 2,12,28,34,25,12,2,6,10,11
56 FOR I=0 TO 59
57 READ A
58 X(I)=A
59 NEXT I
60 INPUT "NO. OF CHANNELS PER WORD=";V
70 M=0:C=V:F=0:R=V
80 GOSUB 1000:REM NORMALIZE
85 INPUT "S=";S
90 INPUT "NO. OF CHANLS. YOU WISH TO COMPARE=";C
100 M=1:R=C:REM GET DF1 AND COMPARE
110 OPEN 4,1,0
120 GOSUB 780:REM COLLECT DATA
130 CLOSE 4
140 GOSUB 1000
150 REM COMPARE WITH X(I) & PRINT DIFF. FOR EACH CHAN.
200 GOSUB 800
220 M=M+1
230 IF M=4 GOTO 110
300 REM GET MAJORITY VOTE FOR BEST FITS
310 M=M-1
320 FOR I=1 TO C
325 MIND=100000
330 FOR J=1 TO M
335 IF Y(J,I)<MIND THEN MIND=Y(J,I);NN=J
340 NEXT J
345 N(1,NN)=N(1,NN)+1
350 NEXT I
355 OPEN 4,4,1
360 PRINT
365 PRINT "THE NO. OF BF FOR & WORD FOR",C,"CHANLS. ARE"
370 PRINT#4
375 PRINT#4,"THE NO. OF BF FOR & WORD FOR",C,"CHANLS. ARE"
380 FOR J=1 TO M
385 PRINT J,N(J)
390 PRINT#4,J,N(J)

Fig. 4.14 Program AB4.
395 NEXT J
400 CLOSE 4
405 REM FIND WORD WITH MAX. BF
410 MBF=0
415 FOR J=1 TO M
420 IF N(J)>MBF THEN MBF=N(J):WB=J
425 NEXT J
430 IF MBF>=2 GOTO 715
435 B=1
440 FOR J=1 TO M
445 IF N(J)<1 GOTO 460:REM THROW AWAY WORD
450 T(B)=J:REM STORE CANDIDATE WORD IN TABLE
455 B=B+1
460 NEXT J
500 REM GET REMAINING CHANLS. FOR CANDIDATE WORDS & COMPARE
501 PRINT "REWIND TAPE & PRESS ANY KEY TO CONTINUE"
502 GET A$:IF A$="" THEN 502
505 N=1:REM N IS THE ORDER OF DATA FILES
510 FOR JJ=1 TO B-1
520 N=T(JJ)
550 OPEN 4,i,0
560 GOSUB 1200:REM GET DATA
570 CLOSE 4
580 IF N(N)<N THEN N=N+1:GOTO 550
590 P=C:R=V
600 GOSUB 1000
610 GOSUB 800
615 N=N+1
620 NEXT JJ
625 REM GET MAJORITY VOTE FOR BF FOR CANDIDATE WORDS
630 FOR I=P+1 TO V
635 MIND=1000000
640 FOR J=1 TO B-1
645 M=T(JJ)
650 IF Y(M,J)>MIND GOTO 655
651 MIND=Y(M,J):NN=J
655 NEXT J
660 N(NN)=N(NN)+1
665 NEXT I
670 OPEN 4,4,1
671 PRINT
672 PRINT
673 PRINT "NO. OF BF FOR A CANDIDATE WORD FOR ALL CHANLS:";
673 PRINT #4
676 PRINT #4
677 PRINT #4,"NO. OF BF FOR CANDIDATE WORD FOR ALL CHANLS:";

Fig. 4.14 Program AB4 (cont.)
679 FOR J=1 TO B-1
680 PRINT T(J),N(J)
681 PRINT#4,T(J);N(J)
682 NEXT J
683 CLOSE 4
685 REM FIND WORD WITH MAX BF
690 MFB=0
695 FOR J=1 TO B-1
700 IF N(J)<MFB GOTO 705
701 MFB=N(J);WR=J
705 NEXT J
710 WFB=T(WR)
715 PRINT
720 PRINT
725 PRINT"RECOGNIZED:",WFB
729 OPEN 4,4,1
730 PRINT#4
740 PRINT#4
750 PRINT#4,"RECOGNIZED:",WFB
755 CLOSE 4
760 GOTO 2000
780 FOR I=0 TO 20*C-1
782 INPUT#4,A
783 I(I)=A
784 NEXT I
799 RETURN
800 FOR I=F TO R-1
810 F=1000000;SS=5
820 FOR K=-5 TO S
830 Y(M,I+1)=0
840 FOR J=SS TO I9-SS
850 D(M,I+1)=ABS(X(I*20+(J+K))-I(I*20+J))
860 Y(M,I+1)=Y(M,I+1)+D(M,I+1)
870 NEXT J
880 IF Y(M,I+1)<F THEN F=Y(M,I+1);MS=K
890 NEXT K
900 Y(M,I+1)=F
901 NEXT I
905 PRINT
910 PRINT"DIFF. FOR WORD",M,"FOR & CHANL."
915 OPEN 4,4,1
920 PRINT#4
925 PRINT#4,"DIFF. FOR WORD",M,"FOR & CHANL"
930 FOR I=F+1 TO R
940 PRINT I,Y(M,I)
945 PRINT#4,I,Y(M,I)
950 NEXT I

Fig. 4.14 (cont.)
960 PRINT "MIN. DIFF. AT S=", MS
965 PRINT#4
970 PRINT#4,"MIN. DIFF. AT S=", MS
975 PRINT#4
980 CLOSE 4
990 RETURN
1000 FOR I=F TO R-1
1010 Q=0
1020 FOR J=0 TO 19
1030 IF M=0 GOTO 1050
1040 IF Z(D+J)>Q THEN Q=Z(D+J)
1045 GOTO 1060
1050 IF X(D+J)>Q THEN Q=X(D+J)
1060 NEXT J
1070 FOR J=0 TO 19
1080 IF H=0 GOTO 1100
1090 Z(D+J)=Z(D+J)/Q
1095 GOTO 1110
1100 X(D+J)=X(D+J)/Q
1110 NEXT J
1120 NEXT I
1130 RETURN
1200 FOR I=0 TO 20*C-1
1210 INPUT#4,A
1220 NEXT I
1230 FOR I=20*C TO 20*V-1
1240 INPUT#4,B
1250 Z(I)=B
1260 NEXT I
1270 RETURN
2000 END

Fig. 4.14 (cont.)
DIFF. FOR WORD 1
1 .100000002
2 .15641612

DIFF. FOR WORD 2
1 3.19168867
2 3.03951613

DIFF. FOR WORD 3
1 .333333333
2 2.93709878

THE NO. OF BF FOR 3 WORD FOR 2 CHANL. ARE
1 2
2 0
3 0

RECOGNIZED: 1

Fig. 4.15 (a) Output of Program AB4.
DIFF. FOR WORD 1 FOR 3 CHANL
1 \( \cdot083333344 \)
2 \( \cdot0808451605 \)
MIN. DIFF. AT 5 = 4

DIFF. FOR WORD 2 FOR 3 CHANL
1 2.0418887
2 3.03951613
MIN. DIFF. AT 5 = 6

DIFF. FOR WORD 3 FOR 3 CHANL
1 \( \cdot33333333 \)
2 1.41774194
MIN. DIFF. AT 5 = -6

THE NO. OF BF FOR 3 WORD FOR 2 CHANLS. ARE
1 2
2 0
3 0

DIFF. FOR WORD 1 FOR 3 CHANL
3 \( \cdot483294118 \)
MIN. DIFF. AT 5 = 4

DIFF. FOR WORD 2 FOR 3 CHANL
3 1.98
MIN. DIFF. AT 5 = 1

DIFF. FOR WORD 3 FOR 3 CHANL
3 2.019680784
MIN. DIFF. AT 5 = 4

NO. OF BF FOR 3 CANDIDATE WORD FOR ALL CHANLS.:
1 3

RECOGNIZED: 1

Fig. 4.15 (b) Output of Program AB4 with no decision made after first stage of comparison.
82 GOSUB 1300: REM FIND 1ST PEAK OF UTTERANCE, FFE

READY.

150 GOSUB 1400: REM FIND 1ST PEAK OF TEMPLATE, FFT(M)

800 S=FFE-FFT(M)
810 PRINT FFE,FPT(M),S
820 FOR I=F TO R-1
830 Y(M,I+1)=0
840 FOR J=ABS(S) TO 19-ABS(S)
850 D(M,I+1)=ABS(X(I+20+(J+S))-Z(I+20+J))
860 Y(M,I+1)=Y(M,I+1)+D(M,I+1)
870 NEXT J
880 NEXT I

1300 I=0
1310 D=X(I+1)-X(I)
1320 IF D<0 THEN FPE=I: GOTO 1340
1330 I=I+1: GOTO 1310
1340 RETURN

1400 I=0
1410 D=Z(I+1)-Z(I)
1420 IF D<0 THEN FFT(M)=I: GOTO 1440
1430 I=I+1: GOTO 1410
1440 RETURN

Fig. 4.16 Program AB4 modified to use peak alignment method for time normalization.
In this chapter, real speech data is used for recognition experiments and investigations. The system hardware connection to the Analog-to-Digital converter (ADC) and to the PET microcomputer is discussed. Hardware modification, which is necessary because it is proposed to sample only two channels per utterance, is also discussed. Raw data collection programs and two problems encountered when sampling speech are discussed.

Because of the enormous raw data generated by sampling speech signal at the proposed 6,700 samples per second, it is desirable to find some technique of reducing the amount of data stored, without losing the information needed for subsequent recognition. Two methods of data reduction are proposed. The formation of the reduced data vocabulary templates is discussed and recognition experiments are performed to test the effectiveness of the data reduction methods. Finally a recognition algorithm is chosen based on the results of the experiments in this chapter and in Chapter 4. Recognition experiments are carried out to evaluate the performance of the system.

5.1 System Hardware Modification and Connection

Due to the storage limitations in the PET, only two channels can be digitized per utterance. The control and timing circuit in the system hardware is designed to scan six channels per utterance and provide control to the ADC for digitizing the six channels. Thus the hardware needs to be modified to scan only two channels. It is
decided that the control signals be provided by the PET under software control. Fig. 5.1 shows a block diagram of the system hardware connection to the PET, and Fig. 5.2 shows a block diagram of the modified system hardware.

As can be seen from Fig. 5.2, the multiplexer select signals are provided by the PET on pins PA0 – PA2 of the user port. Thus any of the six channels can be selected by software for digitizing. The WORD PRESENT signal is also provided by the computer (on pin PA4) instead of the signal detector in the original design. This signal, when high, enables the 74LS123 multivibrator in the control and timing circuit which generates the sample-and-hold logic signal. An interrupt is generated on each falling edge of the sample-and-hold logic signal by connecting this signal to CA1 of the user port. On detecting an interrupt (by reading a logic 1 in the CA1 interrupt flag bit), the computer sends a start conversion signal (SC) to the ADC on pin PA5. After conversion, the data is sent to the computer via pins DIO-0 to DIO-7 of the IEEE-488 bus.

The ADC used is the ADC^2HX12BSC which is a self contained, high performance, 12-bit A/D converter manufactured by Datel-Intersil. It is connected for 8 bits, unipolar operation with a range of 0 to +5 V. The connection diagram is shown in Fig. 5.3. The ADC specifications and characteristics are given in Appendix C.
Block diagram of system hardware connection to the PET.
Fig. 5.2  Block diagram of modified speech recognition system hardware.
SC (FROM USER PORT)

Fig. 5.3 Connection diagram for ADC.
5.2 Data Collection

In the data collection process, a word, spoken in the microphone or played back from a tape recorder into the preamplifier, is separated into six filter channels which are then sampled at 6.7 kHz. Each of two selected sampled channel is then digitized at the rate of 6,700 samples per second. Data is collected for 700 msec for each utterance even if the utterance is less than 700 msec long. Sampling at a fixed time for each utterance will eliminate the problem of end-of-word detection and that of stops in words with stop consonants.

5.2.1 Sampling Program

The sampling program is in assembly language and is given in Appendix D. It samples two channels per utterance and is called from BASIC. The channels to be digitized are passed to the sampling program by poking their numbers (1 to 6) into defined memory locations (FILTER and FILTER + 1) from BASIC. A basic routine which does this and calls the sampling program is shown in Fig. 5.4. This routine was used for testing the hardware and debugging the sampling program. The 60 Hz interrupt is first disabled in line 40 and enabled after returning from sampling (line 90). Line 50 puts a 1 in PA4 thus starts the sample-and-hold logic signal and thus the conversion process, since the falling edge of the pulses set an interrupt flag which, when detected by the computer, causes a conversion of the input to be performed.
A flowchart for the sampling program is shown in Fig. 5.5. When the sampling program is entered the first filter channel selected is continuously digitized until the sample data is above a set threshold value which is considered to be the value of noise. The two channels are then digitized alternately after each detection of an interrupt flag bit set by a falling edge of the sample-and-hold control signal until 4690 samples are collected for each channel. This is the number of samples per 700 msec, sampling at 6.7 kHz. The sampling program then jumps to a normalization routine.

5.2.2 Normalization Routine

The normalization routine amplitude normalizes the speech samples to compensate for signal level differences normally encountered in speech data, as explained in Sections 2.1 and 4.2. The method used is peak amplitude normalization. This method finds the highest peak in the signal and stores all values relative to this, as explained in Subsection 4.2.3.

The program to perform amplitude normalization is given in Appendix E. The flowchart for this program is shown in Fig. 5.6. The largest value (highest peak) in the sample is found and each value is divided by the largest. The results are then multiplied by 255, thus the highest peak has a value of 255 which is the maximum value that can be stored in a byte.
Fig. 5.5  Flowchart for sampling.
Fig. 5.5 (cont.)
Fig. 5.6  Flowchart for normalization routine.
5.2.3 Problems Encountered in Speech Sampling

Two problems to consider when sampling speech are noise and saturation (or dynamic range). Some of the variations in a speech signal is due to noise sources. These noises could be external, for example room noise, heavy breathing or brushing the microphone cord, or internal (electrical) noise within the hardware, or a combination of both. While speech recognition algorithms must be robust to be effective, experimentation is best done with minimum variation due to noise sources. These noise sources can make it hard to see what is going on. Therefore, every effort should be made to minimize external sources of noise and variation. The most important of this is found to be microphone position. The microphone should be kept a constant two inches from the mouth when speaking into the system or recording on tape to play back to the system [16]. Room noise should also be kept down.

The dynamic range of a system is the maximum and minimum amplitude that the system can attain. In this system the range is 255 (8 bits.) To get maximum information from the speech data, the signal should span approximately the same range. Speech that is too loud will saturate the system producing many large values (near 255) with very little variation, even though the original signal may have contained considerable variation. This effect is called saturation and is shown in Fig. 5.8a. Fig. 5.8b shows the same signal, but since it isn't saturating, more detail is visible. Also if the peak values of signals fall consistently below the maximum possible, information is lost by not taking advantage of the range possible in the hardware.
Fig. 5.7 Saturated signal.

Fig. 5.8 Unsaturated signal.
Care in holding the microphone at a constant position from the mouth will ensure the best result [16]. The dynamic range could be set by adjusting the gain control of the preamplifier stage.

5.3 Data Reduction

After a word had been sampled, it is represented in the computer by a matrix of integers (filter number vs. sampled time vs. amplitude). With multiple filters in the system and a high sampling rate, the amount of data representing each word can be enormous. Much of this data is irrelevant or redundant, especially when sampling slowly varying signals from the lower frequency filter channels at a high sampling rate. A property or feature extraction system is used to reduce this data to a level that makes it possible for the recognition algorithm to operate reasonably and for the recognition system to have a reasonable memory requirement, without losing the information needed for subsequent recognition.

Two sets of recognition properties are linguistics and non-linguistics types. A system based on the first extracts properties, which tend to describe the speech signal in more linguistic terms, changes of which describe features of the input message which can be understood in terms of the ordinary linguistic descriptions of such messages. An example of this is phonemes extraction and recognition. The second set of properties, which are more abstract, extract features of the spectral shape of speech but does not reflect the detailed phonetic content of the utterance.
In this section two methods of data reduction, based on non-linguistic feature extraction, are proposed. We have chosen the non-linguistic approach because of the computational simplicity of spectral properties.

5.3.1 Data Reduction Method 1

In this method we suggest lowering the sampling rate to 100 Hz by taking the average of the data over every 10 ms. This is based on the fact that the fundamental frequency of the vocal cords in men is approximately 100 Hz. Sampling more often than this would not significantly increase the information obtained, since the signal will not vary too much in this time [17]. White and Neely [1] used a sampling rate of 100 Hz in a recognition experiment with filter bank analysis and obtained a good result.

Plots of averaged data for the filter with 500 Hz center frequency, for words ONE to NINE, are shown in Figs. 5.9 (a)–(e). The utterances are initially sampled at 6.7 kHz over 700 ms and then averaged every 10 ms. From these plots we can see that there are significant differences in the spectral shapes of the words.

The program to reduce the raw data as explained above is given in Appendix F. It sums each 67 samples of raw data, obtained over 10 ms at 6.7 kHz sampling rate and finds an average. Thus each filter channel will be represented by 70 data points over a period of 700 ms, and an utterance will be represented by a maximum of 420 samples in the computer.
Fig. 5.9 (a) Reduced data plots for words ONE and TWO.

Fig. 5.9 (b) Reduced data plots for words THREE and TEN.
Fig. 5.9 (c) Reduced data plots for words FOUR and FIVE.

Fig. 5.9 (d) Reduced data plots for words EIGHT and NINE.
5.3.2 Method 2: Peak Characteristics

In this method spectral shape features with pattern recognition potential are selected. These are the peak characteristics of a signal. The method is based upon an approach used by F. Koperda in a voice-recognition system [18]. In his approach, the speech input signal is "tracked" in real time by a triangular wave of variable amplitude and frequency. This is achieved by comparing a received data from the ADC to the previous data to determine if a change in the sign of the slope has occurred. If it has, the amplitude value as well as the time interval since the last slope change are stored. In effect, the minimum and maximum peaks and the time between peaks are detected as shown in Fig. 5.10.

This approach also separates the amplitude and frequency characteristics of the speech signal to the extent that the signal frequency is related to the time intervals between slope reversals. This is called the frequency and amplitude separation (FAS) method of data collection. The effectiveness of this approach could not be determined because, at the time of publishing, the system had not been fully implemented. However, the FAS method does require that the time intervals be measured with reasonable precision so that the peaks could be closely tracked. It is found, experimentally, that for adequate time resolution, a sampling rate of at least 12.5 kHz is required [18]. The author used a sampling rate of 50 kHz.

The method proposed here is to scan the raw data collected for a filter output signal to find the maximum and minimum amplitude and the
Fig. 5.10 Frequency and amplitude separation.
time between them. This is done by comparing the sample at time slice \( i \) with the samples at \( i-1 \) and \( i+1 \) to see if the sample at \( i \) represents a maximum or minimum. If it does it is stored and the time since the last maximum or minimum is recorded. Fig. 5.11 is a flow-chart of an algorithm for finding the peak characteristics of the filter output signals. It can be seen that the first peak is determined initially and processing starts from that time slice. This reduces the problem of beginning-of-word detection. The algorithm is implemented in assembly language and is given in Appendix G.

5.3.3 Experiments

The experiments in this section are carried out to test the effectiveness of the data reduction techniques in recognizing the digits 0 to 9 while comparing only one filter channel at a time. A vocabulary word will be considered recognized if it has the lowest distance measure from the spoken word. By considering only one channel at a time for recognition, it is hoped to find the filter channels which provide the most distinction between the vocabulary words. The final recognition software could then make a decision based on one or more of these channels thus satisfying one of our system's goals as mentioned in Chapter 1. Four filter channels are initially considered in the experiments. These are channels 2 (\( f_0 = 125 \) Hz), 3 (\( f_0 = 250 \) Hz), 4 (\( f_0 = 500 \) Hz) and 5 (\( f_0 = 1 \) kHz). It was found, by observing all the channels on an oscilloscope, that they contain the most information.
Flowchart of data reduction (method 2) algorithm.

Fig. 5.11
For these experiments, and for the final recognition experiments discussed later in this chapter, all speech used was recorded on a Sony analog tape. The recordings were made in a room with noise from an air conditioner. The vocabulary of the digits zero through nine was spoken four times. A single repetition was used as an utterance prototype, and the remaining three repetitions were then used to supply unknown utterances to match against the prototype.

The first step in a recognition process is training which involves forming a vocabulary template. A template of the vocabulary word is formed and stored on a PET cassette tape for these experiments. The flowchart for this training process is shown in Fig. 5.12. The operator is first asked to enter the number of words in the vocabulary and the two filters to be sampled. These are then stored on a file on cassette and can subsequently be read by the recognition routine. The operator is then asked to speak (or play from tape) the first vocabulary word. The sampling, normalization and data reduction subroutines are called. The reduced data for the two channels are then saved on tape. This process is performed for all the vocabulary words for all four channels. The training program is written in BASIC and is given in Fig. 5.13. The subroutines that call the data reduction routines and save the reduced data on the PET cassette tape are given in Fig. 5.14 (a) for data reduction method 1 and in Fig. 5.14 (b) for method 2.

The recognition process consists of forming the template of a spoken unknown word and comparing it to the vocabulary template one
Fig. 5.12 Flowchart for training algorithm.
5 REM  PROGRAM TXFPT
10 REM RECOGNITION EXPERIMENTS: TRAINING
15 REM TO STUDY DATA REDUCTION METHODS
25 POKE 57457,255:REM MAKE VIA PAO-7 OUTPUT
30 INPUT"NO. OF WORDS:";N
35 A=1:GOTO 65
40 OPEN 4,1,1
45 PRINT#4,N
46 PRINT#4,F1
47 PRINT#4,F2
50 CLOSE 4
55 M=1
60 IF A=2 GOTO 90
65 PRINT"PLACE VOCABULARY CASSETTE IN RECORDER "
70 PRINT"PRESS ANY KEY TO CONTINUE"
75 GET A$:IF A$="" THEN 75
80 INPUT"FILTERS TO BE SAMPLED:";F1,F2
85 IF A$=1 THEN A=A+1:GOTO 40
90 POKE 538,(F1-1)+16:REM SELECT FILTER #F1
95 POKE 539,(F2-1)+16:REM SELECT FILTER #F2
100 PRINT"PRESS ANY KEY TO CONTINUE"
105 GET A$:IF A$="" THEN 105
110 POKE 57471,16:REM ENABLE 5-H LOGIC.
115 POKE 57411,80:REM DISABLE 60HZ INTERRUPT
120 PRINT"SAY WORD ",M
125 SYS(826)
130 POKE 57471,00:REM DISABLE 5-H
135 POKE 57411,61:REM ENABLE 60HZ INTERRUPT
140 GOSUB 600
150 GOSUB 1100
155 M=M+1
160 IF M=N GOTO 100
165 PRINT "YOU NOW HAVE YOUR VOC. STORED ON CASSETTE."
170 PRINT
175 PRINT"DO YOU WANT TO ENTER RECOGNIZE MODE (Y/N): "
180 GET A$:IF A$="" THEN 180
185 IF A$<>"Y" GOTO 500
190 PRINT"PLACE BASIC CASSETTE IN RECORDER"
195 PRINT"PRINT ANY KEY TO CONTINUE"
200 GET A$:IF A$="" THEN 200
205 LOAD "REXPT"
210 PRINT
220 END

Fig. 5.13 Training program for experiments.
600 REM TO FIND AVERAGE OVER 10MS.
610 SYS(20800)
620 RETURN
630 REM STORE TEMPLATE ON CASSETTE
640 FOR I=0 TO 139
650 A=PEEK(20972+I)
660 PRINT#4,A
670 NEXT I
680 CLOSE 4
690 RETURN

Fig. 5.14 (a) Data reduction subroutine for method 1.

600 REM TO FIND MAX/MIN VALUES
610 POKE 75,0:POKE 76,65+(CC*19)
620 SYS(20488)
630 CTI=PEEK(20485)
640 CI(CC,H)=CTI-1
650 RETURN
660 REM STORE TEMPLATE ON CASSETTE
670 OPEN 4,i,i
680 PRINT#4,CI(CC,H)
690 FOR I=1 TO CI(CC,H)
700 A=PEEK(30812+I):B=PEEK(31068+I)
710 PRINT#4,A,B
720 NEXT I
730 CLOSE 4
740 RETURN

Fig. 5.14 (b) Data reduction subroutine for method 2.
channel at a time. Two utterances of the unknown are required to compare all four channels since only two channels are sampled per utterance. The distance measures between the unknown word and all the vocabulary words for each channel are then recorded. It is from this data that a decision can be made as to which channels would distinctly recognize an unknown word. Fig. 5.15 is a flowchart of the algorithm for the recognition process, and Fig. 5.16 gives the program.

The subroutines in the recognition program are given in Fig. 5.17 (a) for data reduction method 1 and in Fig. 5.17 (b) for method 2. The subroutine starting at line 600 puts the unknown word template in an array. The subroutine starting at line 700 gets a vocabulary word template from tape and puts it in an array. It also stores the data in the computer memory so it doesn't have to be loaded again from cassette when recognizing another unknown word; the subroutine starting at line 1200 puts this stored data in an array. The subroutine that calculates the distance measures starts at line 800. The distance measure used is the Euclidean norm. For method 1, the averaged data over a 10 ms time slice for the unknown is compared with the data for the corresponding time slice for each word. This is done in line 840 of Fig. 5.17 (a). Lines 835 and 840 calculates the differences between both the peak and time between peaks for the same time slice for the unknown word and a vocabulary template.

These recognition experiments are performed for all the words in the vocabulary. The distance measures between three of the words (one, three and eight), when spoken, and the vocabulary words are shown
Flowchart for recognition algorithm.
Any more vocabulary words?

Any more channels?

End Recognition
5 REM PROGRAM REXPT
10 REM RECOGNITION ROUTINE FOR EXPERIMENTS
15 DIM Z(255), X(255), D(255), Y(10, 2), YY(10)
25 PRINT"PLACE VOC. CASSETTE IN RECORDER"
30 PRINT"PRESS ANY KEY TO CONTINUE"
35 GET A$: IF A$="" THEN 35
40 OPEN 4, 1, 0
45 INPUT#4, N
50 INPUT#4, F1
55 INPUT#4, F2
60 CLOSE 4
65 POKE 59459, 255: M=0
70 POKE 638, (F1-1)+16: POKE 839, (F2-1)+16
75 PRINT"PRESS ANY KEY TO CONTINUE"
80 GET A$: IF A$="" THEN 80
85 POKE 59471, 16: POKE 59411, 60
90 PRINT"SHY WORD"
95 SYS(826)
100 POKE 59471, 0: POKE 59411, 61
105 GOSUB 600: REM DATA REDUCTION
110 M=1
115 IF C=1 GOTO 128
120 OPEN 4, 1, 0
125 GOSUB 700: REM GET VOC. DATA
130 CLOSE 4
135 IF M=1 OR C=0 GOTO 130
136 IF M=1 OR C=0 GOTO 135
138 GOSUB 1200: REM PUT VOC. DATA IN ARRAY
139 GOSUB 800: REM COMPARE UNKNOWN & TEMPLATE
140 IF M<N+1 AND C=0 GOTO 115
141 IF M<N+1 GOTO 128
150 PRINT
160 INPUT"DO YOU WISH TO RECOGNIZE ANOTHER WORD?: "; A$
165 IF A$<>"Y" GOTO 500
170 C=1: GOTO 75
500 END

Fig. 5.16 Recognition program for experiments.
600 REM DATA REDUCTION OF UNKNOWN
610 SYS(20800)
620 FOR I=0 TO 139
630 X(I)=PEEK(Z0792+I)
640 NEXT I
650 RETURN

700 REM GET VOC. DATA
710 FOR I=0 TO 139
720 INPUT#4, A: Z(I)=A
725 POKE 16384+(M-I)*140+I, A
730 NEXT I
740 RETURN

800 REM COMPARE UNKNOWN & TEMPLATE
805 YY(M)=0
810 FOR II=0 TO I
820 Y(M,II)=0
830 FOR J=0 TO 69
840 D(II)=ABS(X(II*70+J)-Z(II*70+J))
850 Y(M,II)=Y(M,II)+D(II)*D(II)
860 NEXT J
870 PRINT
880 PRINT"DIFF. FOR WORD",M,"CHANL.",II,"="",Y(M,II)
890 YY(M)=YY(M)+Y(M,II)
900 NEXT II
905 PRINT
910 PRINT"TOTAL DIFF. FOR WORD",M,"IS",YY(M)
920 RETURN

1200 REM PUT VOC. DATA IN AN ARRAY
1210 FOR I=0 TO 139
1220 Z(I)=PEEK(16384+(M-I)*140+I)
1230 NEXT I
1240 RETURN

Fig. 5.17 (a) Subroutines for word identification method 1.
REM DATA REDUCTION OF UNKNOWN
FOR CC=0 TO 1
POKE 75,CC:POKE 76,83+(CC*19)
SYS(20486)
FOR K=0 TO 49
HP(0,K+(CC*50))=PEEK(30812+K):TP(0,K+(CC*50))=PEEK(31058+K)
NEXT K
NEXT CC
RETURN

REM GET VOC. DATA
INPUT#4,CI(CC,H)
R=17920+(CC*50)+(H-1)*100:S=R+1000
FOR K=0 TO CI(CC,H)-5
INPUT#4,A
INPUT#4,B
IF K>49 GOTO 755
HP(i,K+(CC*50))=A:TP(i,K+(CC*50))=B
POKE R+K,A:POKE S+K,B
NEXT K
NEXT
RETURN

REM COMPARISON OF UNKNOWN & TEMPLATE
FOR I=0 TO 1
F=1000000:5=R*I*50
FOR K=-5 TO 5
FOR J=ABS(S) TO (49-ABS(S))
YY(H,I)=YY(H,I)+(HP(0,R+J+K)-HP(0,R+J))\2
NEXT J
IF YY(H,I)<F THEN F=YY(H,I)
NEXT K
NEXT
PRINT "FOR WORD":H,"CHAPL.",I,"DIFF.="YY(H,I)
NEXT I
PRINT
RETURN

REM PUT VOC. DATA IN AN ARRAY
R=17920+(H-1)*100:S=R+1000
FOR I=0 TO 99
HP(0,1)=PEEK(R+1):TP(0,1)=PEEK(S+1)
NEXT I
RETURN

Fig. 5.17 (b) Subroutines for Program REXPT: Word identification method 2.
in Tables 5.1 (a)-(f) for data reduction method 1, and the distance measure for word one, when spoken, channels 2 and 4 is shown in Table 5.2 for method 2. These results are used for the discussion of experimental results. The channels which have the lowest distance measures are marked with asterisks.

5.3.4 Experimental Results

For data reduction method 1, it is seen from Tables 5.1 (a)-(f) that for the word one all three utterances have the lowest distance measures when compared with the same word in the vocabulary template, for all the channels, with the exception of the second utterance which has the second lowest distance measure (greater than the lowest with a factor of 1.002). This indicates that word one could be recognized by comparing only one channel or two.

For word three, the first utterance has the lowest distance measure with the same word in the vocabulary template for channel 4, the second lowest measure for channels 3 and 5, but has the lowest when the two channels are combined. It has the third lowest measure for channel 2. The second utterance has the lowest distance measure for all the channels. The third utterance has the second lowest distance for channel 4, fourth lowest distance for channel 5, and sixth lowest for channels 2 and 3. This means that the third utterance of word three cannot be recognized by comparing any one, or infact all of the channels. This may be due to the fact that the utterance was too loud.
For word eight the first and third utterance of the word have the lowest distance measures for channels 3, 4 and 5, and the second lowest for channel 2. The second utterance has the lowest distance for channels 3 and 4, the second lowest for channel 2, and the third lowest for channel 5.

Based on this result and the distance measures obtained with the other utterances, it was concluded that for data reduction method 1 any of the vocabulary words could be recognized by comparing only one of, or a combination of, channels 3, 4 and 5 of the spoken word with the same channels of the vocabulary words.

Table 5.2 shows the distance measures, calculated with method 2 as the data reduction method, between two utterances of the word one and the vocabulary template. From this result and other results obtained with other spoken words, it is concluded that utterances cannot be reliably recognized either by comparing only one channel or any combination of channels with data reduction method 2 used. This may be due partly to the variations in speech signals (peak cannot be detected accurately), or the distance measure method used.

Based on these results, it is decided to use data reduction method 1 in the final recognition algorithm. Recognition will be based on comparing one of, or a combination of, channels 3, 4 and 5.

5.4 Recognition Software

Based on the experiments performed in Chapter 4 and in Section 5.3, the final recognition software was selected. It consists of a menu program, a training program, and a recognition program. All
Table 5.1(b) Word "ONE"

<table>
<thead>
<tr>
<th>TEMPLATE WORDS</th>
<th>CHANL</th>
<th>DISTANCE MEASURE FOR EACH CHANNEL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1st UTTERANCE</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>21,304</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>28,566</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>192,149</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>234,483</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>118,600</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>264,135</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>146,600</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>359,160</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>110,800</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>144,122</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>166,658</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>399,013</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>373,464</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>376,611</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>181,634</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>323,575</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>302,320</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>60,373</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>122,985</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>282,351</td>
</tr>
</tbody>
</table>
Table 5.1(c) Word "THREE"

<table>
<thead>
<tr>
<th>TEMPLATE WORDS</th>
<th>CHANL</th>
<th>DISTANCE MEASURE FOR EACH CHANNEL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1st UTTERANCE</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>395,490</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>239,544</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>80,942</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>160,325</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>89,784</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>157,803</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>62,233</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>148,565</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>213,450</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>139,934</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>428,634</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>98,729</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>359,928</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>104,732</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>90,648</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>83,708</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>261,509</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>260,704</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>552,493</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>416,760</td>
</tr>
<tr>
<td>TEMPLATE WORDS</td>
<td>CHNL</td>
<td>DISTANCE MEASURE FOR EACH CHANNEL</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1st UTTERANCE</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>111,048</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>356,866</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>127,261</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>68,869</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>71,099</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>41,544</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>97,616</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>31,578</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>114,38</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>74,638</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>164,814</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>52,237</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>285,884</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>51,499</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>107,440</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>56,219</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>215,216</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>141,207</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>51,179</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>80,041</td>
</tr>
</tbody>
</table>
Table 5.1(e)  Word "EIGHT"

<table>
<thead>
<tr>
<th>Vocabular y Words</th>
<th>Chnl 1</th>
<th>Chnl 2</th>
<th>Chnl 3</th>
<th>Chnl 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>413.819</td>
<td>302.454</td>
<td>359.347</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>358.049</td>
<td>219.299</td>
<td>341.300</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>39.461</td>
<td>78.182</td>
<td>45.811</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>123.698</td>
<td>69.882</td>
<td>123.565</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>59.770</td>
<td>84.805</td>
<td>65.276</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>79.352</td>
<td>45.120</td>
<td>57.521</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>20.364</td>
<td>40.037</td>
<td>19.420</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>75.598</td>
<td>65.508</td>
<td>86.763</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>351.459</td>
<td>326.788</td>
<td>389.055</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>139.979</td>
<td>89.835</td>
<td>136.812</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>468.233</td>
<td>484.968</td>
<td>445.829</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>39.038</td>
<td>95.468</td>
<td>35.457</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>477.457</td>
<td>534.414</td>
<td>514.471</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>85.781</td>
<td>82.879</td>
<td>74.234</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>48.953</td>
<td>41.398</td>
<td>32.247</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>9.645</td>
<td>33.877</td>
<td>9.360</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>362.102</td>
<td>417.359</td>
<td>389.456</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>361.251</td>
<td>228.333</td>
<td>338.954</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>616.750</td>
<td>492.117</td>
<td>584.848</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>439.471</td>
<td>466.397</td>
<td>439.568</td>
</tr>
</tbody>
</table>
Table 5.1 (f)  Word "EIGHT"

<table>
<thead>
<tr>
<th>TEMPLATE WORD</th>
<th>CHANL</th>
<th>DISTANCE MEASURE FOR EACH CHANNEL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1st UTTERANCE OF UNKNOWN</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>160,818</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>454,258</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>43,351</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>29,839</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>91,856</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>47,189</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>146,988</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>8,230</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>157,796</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>66,290</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>363,424</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>25,609</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>400,614</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>11,063</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>13,506</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>4,349</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>285,854</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>155,388</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>143,005</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>39,463</td>
</tr>
</tbody>
</table>
Table 5.2  Distance measures for utterances of the word "ONE" with data reduction method 2.

<table>
<thead>
<tr>
<th>TEMPLATE WORDS</th>
<th>CHANNEL</th>
<th>DISTANCE MEASURE FOR EACH CHANNEL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1st UTTERANCE</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>555.764</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>206.908</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>483.237</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>334.651</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>530.715</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>473.789</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>385.637</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>358.824</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>506.804</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>241.147</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>415.143</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1,000,000+</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>290.735</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>838.101</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>432.293</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>516.038</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>417.382</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>194.563</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>440.295</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>686.190</td>
</tr>
</tbody>
</table>
these programs are interactive. The menu program is given in Appendix H. It simply asks the operator to select either training or recognition mode. It then loads the appropriate program from cassette which then starts to execute. Since it was decided to use data reduction method 2 for the system, the final training program is similar to the training routine used for the experiments in the last section. The only modification is the added ability to select recognition mode from training mode. The training program then loads the recognition program from the cassette which starts to run. The training program is given in Appendix I.

The final recognition program has two modifications from the recognition routine used for the experiments in the last section. The first is the addition of translation in the subroutine that calculates the distance measures. This aligns the unknown word and vocabulary words as explained in Subsection 4.1.6 and improves recognition even though it increases the time for the recognition process. The second modification is the addition of a word identification stage. This is discussed below.

Recognition experiments are performed to evaluate the performance of the system with the word identification technique chosen.

5.4.1 Word Identification

The word identification technique selected was based upon the results of the experiments in Section 5.3. By evaluating the distance measures between the spoken utterances of the vocabulary and
the vocabulary template for channels 3, 4 and 5, the following conclusions were made:

1. By comparing channel 3 of an unknown word with the vocabulary template, the spoken word can be recognized correctly as that vocabulary word which has the lowest difference (distance measure), provided the second lowest difference is at least three times the lowest differences. If the unknown word cannot be recognized as such, then one or more of the other channels must be compared.

2. By comparing channel 5 as above, the spoken word can be recognized correctly as that vocabulary word which has either the lowest difference, provided the second lowest difference is at least three times the lowest difference, or has the lowest total difference, when the differences of channels 3 and 5 are added, provided the second lowest total difference is at least one-and-half times the lowest difference. If the unknown word cannot be recognized as such, then channel 4 is compared.

3. By comparing channel 4 as above, the spoken word can be recognized correctly as that vocabulary word which has either the lowest difference, provided the second lowest difference is at least three times the lowest difference, or has the lowest total differences, when the differences for channels 3, 5 and 4 are added, provided that the vocabulary word has the lowest difference (best fit) in at least two of the three channels.
Based on the above points it was decided to compare channel 3 first. If a recognition decision cannot be made as explained in (1) above, the three vocabulary words with the lowest differences are selected as candidate words. Then the template for channel 5 for these words are loaded from the cassette and compared with the unknown word. If a decision cannot be made, the templates for channel 4 for the candidate words are loaded and compared with the unknown word as explained in (3). If a decision cannot still be made, the unknown word is rejected as not being in the vocabulary. The complete recognition program is given in Appendix J.

5.4.2 Recognition Experiments

For these experiments, two repetitions of an utterance of each vocabulary word as recorded on tape was used to train the recognition system. The three remaining utterances of each of the vocabulary words were then used to supply unknown utterances to test the accuracy of the recognizer. Thus 30 individual utterance tests were made.

5.4.3 Results

Fig. 5.18 shows the recognition results obtained. An X in a box means when the word in that column was spoken the word in the row of that box is recognized. The subscript gives the number of channels compared before a recognition decision could be made. As an example the $X_2 X_3 X_1$ in column 8, row 8 means when three utterances of
Fig. 5.18 Results of recognition experiments to test the performance of the system.
the word eight were spoken the first utterance was recognized correctly after comparing two channels (3 and 5), the second was recognized correctly after comparing three channels (3, 5 and 4), while the third was recognized correctly after comparing only channel 3. A "0" means that an utterance of a spoken word was recognized as a different word or rejected. As can be seen, all correct recognitions fall diagonally. False recognitions or rejections fall off the diagonal.

From these results it can be seen that most of the utterances were recognized after comparing only two channels, that is channels 3 and 5. It can be seen from Fig. 5.18 that seven out of the thirty utterances used in the test were either falsely recognized or rejected as not being in the vocabulary. In all cases it was found by listening to these words or watching their signals on an oscilloscope, that they were either saturated (too loud) or distorted. It can thus be concluded that if a word can be spoken with care it will be correctly recognized.

Fig. 5.19 shows the relationship between the three candidate words obtained after comparing the first channel (channel 3) in a recognition process. A 1 in the box indicates the template word along the row had the lowest difference, a 2 indicates the word with the second lowest difference, and a 3 the word with the third lowest difference. The numbers in parenthesis give the factor by which the second and third lowest differences are greater than the lowest difference. Thus one can get an idea of the words which are similar to each other and may be confused during recognition. An example
<table>
<thead>
<tr>
<th>TEMPLATE</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1(1)</td>
<td>3(25.1)</td>
<td></td>
<td></td>
<td></td>
<td>3(17.1)</td>
<td></td>
<td></td>
<td></td>
<td>2(2.4)</td>
</tr>
<tr>
<td>2</td>
<td>2(1.3)</td>
<td>1(1)</td>
<td>2(4.1)</td>
<td></td>
<td>3(2.1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>1(1)</td>
<td></td>
<td></td>
<td></td>
<td>2(6.1)</td>
<td>3(3.4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>3(1.6)</td>
<td>1(1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>1(1)</td>
<td></td>
<td>2(2.1)</td>
<td>3(6.7)</td>
<td>2(1.2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3(2.0)</td>
<td>2(5.1)</td>
<td>3(6.4)</td>
<td>1(1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>3(1.6)</td>
<td>1(1)</td>
<td></td>
<td>3(2.1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>2(1.2)</td>
<td></td>
<td></td>
<td></td>
<td>1(1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>2(1.3)</td>
<td>3(3.1)</td>
<td>1(1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1(1)</td>
</tr>
</tbody>
</table>

Fig. 5.19  Relationship between the three candidate words for each utterance for channel 3 ($f_0 = 250$ Hz).
of this is the similarity between the words two, six and one. From the data in the figure, it can be seen that for each of these three digits spoken, the other two digits are candidate words for recognition after comparing channel 3. This is also the case for the digits five, seven and nine.

The recognition process takes an average of 3 1/2 minutes per utterance. This does not include the time it takes to get data from the tape. The speed of recognition could be improved by writing the recognition routine in assembly language.

For our limited vocabulary, the implementation discussed in this thesis is computationally fast and gives satisfactory results.
Chapter 6

SUMMARY AND CONCLUSIONS

The purpose of this thesis was to build a microcomputer-based, speaker-dependent, isolated word speech recognition system capable of recognizing the digits 0 through 9. Spectral analysis using six bandpass filters was chosen for the preprocessing stage. The signals from these filters were also rectified and sampled at a rate of 6.7 kHz before digitizing. Amplitude normalization was implemented using the peak normalization method and time normalization was implemented using the translation technique. Due to memory limitations within the microcomputer, only two filter channels could be digitized at one time per utterance. Thus a modification of the hardware circuitry, whereby the microcomputer supplied some of the control signals, was necessary. The raw speech data was compressed by taking the average data over every 10 ms. Filter channels 3 ($f_0 = 250$ Hz), 4 ($f_0 = 500$ Hz) and 5 ($f_0 = 1$ kHz) were found to contain the most unique representations of the vocabulary words. Recognition decisions were made based on comparing one or more of these channels.

The experiments performed in this thesis were limited and used a small vocabulary. Thus the results obtained are preliminary. Further experimentation, which could involve modifications to the procedures described, may be necessary. It was not practical to use some approaches that have been investigated in the past because many require processing power in excess of that available with a
microprocessor or will require an unreasonable time duration to execute on a microprocessor.

Based on the experiments performed, the limited goals of the system have been achieved. The recognition process was performed on the microcomputer in a reasonable time—an average of 3 1/2 minutes per utterance, excluding the cassette read time. The recognition results are satisfactory. Most of the utterances were recognized based on the comparison of only filter channels 3 and 5. This shows that for the speaker who trained the system for the experiments these carry enough information to recognize the vocabulary words correctly. The channels may be different for a different vocabulary, or a different speaker. Channel 4 also carries important information of the vocabulary words. Some of the utterances, including two of the utterances for the digit 5, were recognized only after comparing that channel.

Some words were also rejected or falsely recognized. These rejections and false recognitions were found to be not so much as a failure of the system but due to the variation in human speech. Some of the words that were rejected were found to be too loud or distorted when played back from tape with a loudspeaker at the output of the tape. Based on the performance obtained the data reduction method was found to be satisfactory.

As was said earlier, the experiments in this thesis, and the results obtained, are preliminary. Numerous improvements could be made to the system and further experiments performed. Recognition speed could be improved by implementing the system with the vocabulary
template stored in ROM. The complete recognition program could also be written in assembly language. The memory constraint on the microcomputer could be avoided by implementing the data reduction in real-time. The samples digitized over every 10 ms period are averaged immediately and only the average value is stored in the computer memory. Thus the whole three channels could be digitized per utterance instead of the two utterances required in the thesis.

Since it was found that the vocabulary words could be recognized using one to three filter channels, a final system could be built with only these channels. However, these channels may give different performances for different speakers or for a different vocabulary. Further experimentation is necessary to investigate this. The system could be trained with a different vocabulary to see the difference in performance. White and Neely [1] found that there may be a difference in recognition performance due to vocabulary differences. A larger vocabulary could also be used for experiments.

Another area of improvement is to make the system insensitive to extraneous sounds. This is desirable so the system can be used in practical applications where background noise must be tolerated, for example, in a manufacturing environment. One way of achieving this is by examining the input patterns to distinguish between noise and voice. Most noise above 300 Hz is random and of short duration. On the basis of experimental results [17], if an isolated sound lasts less than 100 ms, a word is not in progress and the preceding data can be ignored. High-pass filters can also help reduce noise
generated by low-frequency mechanical equipment such as electric motors. These filters can be implemented by using analog techniques or digitally.

The ultimate recognition system is one which is low-cost, speaker-independent, with a large vocabulary, and capable of recognizing continuous speech. This type of system is yet to become commercially practical. Voice recognition developments are continuing and a lot of work remains to be done.
REFERENCES


Appendix A

SERVICING AND CONNECTION INFORMATION

For servicing purposes a board layout of the speech recognition system hardware is shown in Fig. A1. The IC types and edge connector terminal pin assignment are given below.

The connection diagrams of the headers used are shown in Fig. A2, and the connection diagrams of the ICs used are shown in Fig. A3.

**IC Types and Functions:**

<table>
<thead>
<tr>
<th>IC</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC1</td>
<td>LM3877A Low noise dual preamp</td>
</tr>
<tr>
<td>IC2, IC3</td>
<td>LM1458 Dual comparator op amp</td>
</tr>
<tr>
<td>IC4 (i)(vi)</td>
<td>LF156 JFET input op amp</td>
</tr>
<tr>
<td>IC5 (i)(vi)</td>
<td>LF398 Sample and hold circuit</td>
</tr>
<tr>
<td>IC6, IC7</td>
<td>74LS123 Retriggerable monostable multivibrator</td>
</tr>
<tr>
<td>IC8</td>
<td>CD 4049 Hex inverter</td>
</tr>
<tr>
<td>IC9</td>
<td>74LS138 3/8 decoder</td>
</tr>
<tr>
<td>IC10</td>
<td>CD4520 Dual binary up counter</td>
</tr>
<tr>
<td>IC11</td>
<td>CD4051 8-input multiplexer</td>
</tr>
</tbody>
</table>

**Edge Connector Terminals:**

<table>
<thead>
<tr>
<th>Power supply</th>
<th>+12V</th>
<th>Pin 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>+12V</td>
<td>Pin 36</td>
</tr>
<tr>
<td></td>
<td>+5V</td>
<td>Pin 7</td>
</tr>
<tr>
<td></td>
<td>GND</td>
<td>Pin 50</td>
</tr>
<tr>
<td>Description</td>
<td>Count</td>
<td>Pin</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>Speech input</td>
<td>(1)</td>
<td>2</td>
</tr>
<tr>
<td>Word present</td>
<td>(1)</td>
<td>3</td>
</tr>
<tr>
<td>Preamp input</td>
<td>(1)</td>
<td>4</td>
</tr>
<tr>
<td>Preamp output</td>
<td>(1)</td>
<td>5</td>
</tr>
<tr>
<td>Sample-hold outputs</td>
<td>(6)</td>
<td>8, 10, 12, 14, 16, 18</td>
</tr>
<tr>
<td>Filter outputs</td>
<td>(6)</td>
<td>9, 11, 13, 15, 17, 19</td>
</tr>
<tr>
<td>Counter clock</td>
<td>(1)</td>
<td>20</td>
</tr>
<tr>
<td>Pin 10 of IC7</td>
<td>(1)</td>
<td>21</td>
</tr>
<tr>
<td>Multiplexer output</td>
<td>(1)</td>
<td>23</td>
</tr>
<tr>
<td>Start conversion signal</td>
<td>(1)</td>
<td>24</td>
</tr>
<tr>
<td>Sample-hold control</td>
<td>(1)</td>
<td>25</td>
</tr>
<tr>
<td>Multiplexer select</td>
<td>(3)</td>
<td>26, 27, 28</td>
</tr>
</tbody>
</table>
Fig. A2 Connection diagrams of headers.
(c) Header H3

(d) Header H4

Fig. A2 (cont.)
Connection Diagram

Metal Can Package

Order Number LFT15SH, LFT156H, LFT355H or LFT356H
See NS Package M05C

Note. Pin 4 connected to case.

SN54123, SN54LS123...J OR W
SN54L123...J
SN74123, SN74LS123...J OR N
(TOP VIEW) (SEE NOTES 1 THRU 4)

Fig. A3
SN54LS139, SN54S139 ••• J OR W PACKAGE
SN74LS139, SN74S139 ••• J OR N PACKAGE
(TOP VIEW)

Select Data Outputs

Positive logic: see function table

Connection diagram

Dual-In-Line Package

Order Number LF198J, LF208J or LF398J
See NS Package J08A
Order Number LF398N
See NS Package NO8B

Fig. A4
Fig. A5
LF198/LF298/LF398 monolithic sample and hold circuits

**General Description**

The LF198/LF298/LF398 are monolithic sample and hold circuits which utilize BJT technology to obtain ultra-high dc accuracy with fast acquisition of signal and low droop rate. Operating as a unity gain follower, dc gain accuracy is 0.002% typical and acquisition time is as low as 5μs. A bipolar input stage is used to achieve low offset voltage and wide bandwidth. Input offset adjust is accomplished with a single pin and does not degrade input offset drift. The wide bandwidth allows the LF198 to be included inside the feedback loop of 1 MHz op amps without having stability problems. Input impedance of 10 MΩ allows high current impedances to be used without degrading accuracy.

P-channel JFET’s are combined with bipolar devices in the output amplifier to give droop rates as low as 5 mV/min with a 1μF hold capacitor. The JFET’s have much lower noise than MOS devices used in previous designs and do not exhibit high temperature instabilities. The overall design guarantees no feedthrough from input to output in the hold mode even for input signals equal to the supply voltages.

**Features**

- Operates from 15V to ±18V supplies
- Less than 10μs acquisition time
- TTL, PMOS, CMOS compatible logic input
- 0.5 mV typical hold step at $C_h = 0.01μF$
- Low input offset
- 0.002% gain accuracy
- Low output noise in hold mode
- Input characteristics do not change during hold mode
- High supply rejection ratio in sample or hold
- Wide bandwidth

Logic inputs on the LF198 are fully differential with low input current, allowing direct connection to TTL, PMOS, and CMOS. Differential threshold is 1.4V. The LF198 will operate from ±15V to ±18V supplies. It is available in an 8-lead TO-5 package.

**Functional Diagram**

![Functional Diagram](image)

**Typical Applications**

![Typical Connection](image) ![Acquisition Time](image)
absolute maximum ratings

Supply Voltage
±18V
Power Dissipation (Package Limitation) (Note 1)
500 mW
Operating Ambient Temperature Range
LF198
-55°C to +125°C
LF298
-25°C to +85°C
LF398
0°C to +70°C
Storage Temperature Range
-66°C to +150°C

Input Voltage
Equal to Supply Voltage

Logic To Logic Reference Differential Voltage
+7V, -30V
(Note 2)

Output Short Circuit Duration
Indefinite

Hold Capacitor Short Circuit Duration
10 sec

Load Temperature (Soldering, 10 seconds)
300°C

Input Offset Voltage, (Note 6)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>CONDITIONS</th>
<th>LF198/LF298</th>
<th>MIN</th>
<th>TYP</th>
<th>MAX</th>
<th>LF398</th>
<th>MIN</th>
<th>TYP</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Offset Voltage, (Note 6)</td>
<td>Tj = 25°C</td>
<td>Full Temperature Range</td>
<td>4</td>
<td>6</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tj = 25°C</td>
<td>Full Temperature Range</td>
<td>2</td>
<td>5</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input Bias Current, (Note 6)</td>
<td>Tj = 25°C</td>
<td>Full Temperature Range</td>
<td>510</td>
<td>25</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Full Temperature Range</td>
<td>75</td>
<td>25</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input Impedance</td>
<td>Tj = 25°C, Rl = 10k</td>
<td>Full Temperature Range</td>
<td>1010</td>
<td>250</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gain Error</td>
<td>Tj = 25°C, Rl = 10k</td>
<td>Full Temperature Range</td>
<td>0.002</td>
<td>0.004</td>
<td>0.001</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feedthrough Attenuation Ratio</td>
<td>Tj = 25°C, C0 = 0.1μF</td>
<td>Full Temperature Range</td>
<td>88</td>
<td>96</td>
<td>88</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>INF</td>
<td>90</td>
<td>90</td>
<td>INF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output Impedance</td>
<td>Tj = 25°C</td>
<td>Full Temperature Range</td>
<td>0.6</td>
<td>2</td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tj = 25°C</td>
<td>Full Temperature Range</td>
<td>4</td>
<td>2</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;HOLD&quot; Step, (Note 4)</td>
<td>Tj = 25°C, &quot;HOLD&quot; mode</td>
<td>Full Temperature Range</td>
<td>0.5</td>
<td>2.0</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tj = 25°C</td>
<td>Full Temperature Range</td>
<td>5.5</td>
<td>2.0</td>
<td>2.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supply Current, (Note 6)</td>
<td>Tj ≥ 25°C</td>
<td>Full Temperature Range</td>
<td>4.5</td>
<td>5.5</td>
<td>6.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logic and Logic Reference Input Current</td>
<td>Tj = 25°C</td>
<td>Full Temperature Range</td>
<td>2</td>
<td>10</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Leakage Current into Hold Capacitor (Note 6)</td>
<td>Tj = 25°C</td>
<td>Hold Mode</td>
<td>30</td>
<td>100</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tj = 25°C</td>
<td>Hold Mode</td>
<td>20</td>
<td>100</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Acquisition Time to 0.1%</td>
<td>∆VOUT = 10V, C0 = 1000 pF,</td>
<td>200 pA</td>
<td>100</td>
<td>20</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>C0 = 0.1μF</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hold Capacitor Charging Current</td>
<td>VIN - VOUT = 2V, VOUT = 0</td>
<td>VOUT = 0</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supply Voltage Rejection Range</td>
<td>VOUT = 0</td>
<td>INF</td>
<td>110</td>
<td>110</td>
<td>110</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Differential Logic Threshold</td>
<td>Tj = 25°C</td>
<td>INF</td>
<td>2.4</td>
<td>2.4</td>
<td>2.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note 1: The maximum junction temperature of the LF198 is 150°C, for the LF298, 115°C, and for the LF398, 100°C. When operating at elevated ambient temperature, the TO-5 package must be derated based on a thermal resistance (θJA) of 150°C/W.

Note 2: Although the differential voltage may not exceed the limits given, the common-mode voltage on the logic pins may be equal to the supply voltages without causing damage to the circuit. For proper logic operation, however, one of the logic pins must always be at least 2V below the positive supply and 3V above the negative supply.

Note 3: Unless otherwise specified, the following conditions apply. Unit is in "sample" mode, Vg = ±15V, Tj = 25°C, -11.5V ≤ VIN ≤ ±11.5V, C0 = 0.01μF, and Rl = 10 kΩ. Logic reference voltage = 0V and logic voltage = 2.5V.

Note 4: Hold step is sensitive to stray capacitive coupling between input logic signals and the hold capacitor. 1 pF, for instance, will create an additional 0.5 mV step with a 5V logic swing and a 0.01μF hold capacitor. Magnitude of the hold step is inversely proportional to hold capacitor value.

Note 5: Leakage current is measured at a junction temperature of 25°C. The effects of junction temperature rise due to power dissipation or elevated ambient can be calculated by doubling the 25°C value for each 11°C increase in chip temperature. Leakage is guaranteed over full input signal range.

Note 6: These parameters guaranteed over a supply voltage range of ±5 to ±18V.

typical performance characteristics
typical performance characteristics (con't)
logic input configurations

**TTL & CMOS**
\[ 3V \leq V_L (\text{Hi State}) \leq 7V \]

1. **Sample & Hold**
   - LF356
   - Threshold = 1.4V

2. **Sample & Hold**
   - LF356
   - Threshold = 1.4V

**CMOS**
\[ 7V \leq V_L (\text{Hi State}) \leq 15V \]

1. **Sample & Hold**
   - LF356
   - Threshold = 0.6(V+) + 1.4V

2. **Sample & Hold**
   - LF356
   - Threshold = 0.6(V+) - 1.4V

**Op Amp Drive**

1. **Sample & Hold**
   - LF356
   - Threshold = +4V

2. **Sample & Hold**
   - LF356
   - Threshold = -4V

**Typical Applications (Cont')**

**X1000 Sample & Hold**

**Sample and Difference Circuit**
(Output Follows Input in Hold Mode)

*For lower gain, the LM108 must be frequency compensated
100
Use \( \frac{100}{AV} \) pF from comp 2 to ground
FEATURES

- 12 Bits Resolution
- 8 or 20 μSec. Conversions
- 5 Input Ranges
- Internal Hi Z Buffer
- Short Cycle Operation

GENERAL DESCRIPTION

The ADC-HX12B and ADC-HZ12B are self-contained, high performance, 12 bit A/D converters manufactured with thin-film hybrid technology. They use the successive approximation conversion technique to achieve a 12 bit conversion in 20 and 8 microseconds respectively. Five input voltage ranges are programmable by external pin connection: 0 to +5V, 0 to +10V, ±2.5V, ±5V, and ±10V. An internal buffer amplifier is also provided for applications where 100 megohm input impedance is required.

These converters utilize a fast 12 bit DAC consisting of tightly matched monolithic quad current switches, a stable nichrome thin-film resistor network, and a precision zener reference source. The circuit also contains a fast monolithic comparator, a monolithic 12 bit successive approximation register, a clock, and a monolithic buffer amplifier. The thin-film resistor network is functionally trimmed by a laser to precisely set the 8-4-2-1 current weighting in the quad current switches. The close tracking of the thin-film resistor and quad current switches result in a differential nonlinearity tempco of only ±2ppm/°C. Gain tempco is ±20ppm/°C maximum.

Both models have identical operation except for conversion speed. They can be short-cycled to give faster conversion in lower resolution applications. Use of the internal buffer amplifier increases conversion time by 3μsec., the settling time of the amplifier. Output coding is complementary binary, complementary offset binary, or complementary 2's complement. Serial data is also brought out. The package is a 32 pin ceramic case. Eight different models are offered covering the operating temperature ranges of 0 to 70°C, -25 to +85°C, and -55 to +100°C.

MECHANICAL DIMENSIONS:

INPUT/OUTPUT CONNECTIONS
### INPUTS

<table>
<thead>
<tr>
<th>ADC-HX128</th>
<th>ADC-HZ128</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analog Input Ranges, unipolar</td>
<td>0 to +5V, 0 to +10V FS</td>
</tr>
<tr>
<td>Analog Input Ranges, bipolar</td>
<td>±2.5V, ±5V, ±10V FS</td>
</tr>
<tr>
<td>Input Impedance</td>
<td>2.5K (to ±5V, ±2.5V)</td>
</tr>
<tr>
<td>Input Impedance with Buffer</td>
<td>5K (to ±10V, ±5V)</td>
</tr>
<tr>
<td>Input Offset Voltage</td>
<td>±15V</td>
</tr>
<tr>
<td>Input Bias Current of Buffer</td>
<td>±100nA typ., ±250nA max.</td>
</tr>
<tr>
<td>Input Overvoltage</td>
<td>±15V</td>
</tr>
<tr>
<td>Start Conversion</td>
<td>2V min. to 5.5V max. positive pulse with duration of 100nsec. min. Rise and fall times &lt;30nsec.</td>
</tr>
<tr>
<td>Logic “1” to “0” transition resets converter and initiates next conversion. Loading: 1 TTL load</td>
<td></td>
</tr>
</tbody>
</table>

### OUTPUTS

| Parallel Output Data | 12 parallel lines of data held until next conversion command. |
| Coding, unipolar | Complementary Binary |
| Coding, bipolar | Complementary Offset Binary |
| Serial Output Data | Complementary Two’s Complement |
| End of Conversion (Status) | Conversion status signal. Output is logic “1” during reset and conversion and logic “0” when conversion complete. |
| Clock Output | Start of positive going +5V 100nsec. pulse. 600 kHz for ADC-HX128 and 1.5MHz for ADC-HZ128 (pin 7 grounded) |

### PERFORMANCE

| Resolution | 12 bits (1 part in 4096) |
| Nonlinearity | ±1/2 LSB max. |
| Differential Nonlinearity | ±1/2 LSB max. |
| Gain Error, before adjustment | ±0.1% |
| Offset Error, before adjustment | ±0.05% of FSR |
| Temp. Coef. of Gain | ±0.1% of FSR |
| Temp. Coef. of Offset, unipolar | ±200ppm/°C max. |
| Temp. Coef. of Offset, bipolar | ±200ppm/°C of FSR |
| DNL, Nonlinearity Temp. | ±2ppm/°C of FSR |

### POWER REQUIREMENT

| +15VDC ±0.3V @ 55mA |
| -18VDC ≥0.5V @ 45mA |
| +5VDC ±0.25 @ 100mA |

### PHYSICAL-ENVIRONMENTAL

| Operating Temperature Range | 0 to 70°C, -25 to +85°C |
| Storage Temperature Range | -55 to +100°C |
| Package Size | 1,700 x 1,100 x 0.160 inches |
| Package Type | 22 pin ceramic |
| Pins | 0.010 x 0.018 inch Kovar |
| Weight | 0.5 oz. (14g) |

### NOTES

1. All digital outputs can drive 2 TTL loads. |
2. Without buffer amplifier used, ADC-HZ128 may require external adjustment of clock rate. |
3. FSR is full scale range and is 10V for 0 to +10V or 25V input and 25V for 100V input. |
4. Shorted output operation.
INPUT CONNECTIONS

<table>
<thead>
<tr>
<th>VOLT. RANGE</th>
<th>INPUT WITHOUT BUFFER</th>
<th>WITH BUFFER</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 TO +5V</td>
<td>24</td>
<td>22 &amp; 25</td>
</tr>
<tr>
<td></td>
<td>23 &amp; 26</td>
<td>30</td>
</tr>
<tr>
<td>0 TO +10V</td>
<td>24</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>23 &amp; 26</td>
<td>30</td>
</tr>
<tr>
<td>±2.5V</td>
<td>24</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>23 &amp; 22</td>
<td>30</td>
</tr>
<tr>
<td>+5V</td>
<td>24</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>23 &amp; 22</td>
<td>30</td>
</tr>
<tr>
<td>±10V</td>
<td>25</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>23 &amp; 22</td>
<td>30</td>
</tr>
</tbody>
</table>

SHORT CYCLE OPERATION

<table>
<thead>
<tr>
<th>RESOLUTION</th>
<th>PIN 14 TO</th>
<th>RES. (BITS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 BITS</td>
<td>7</td>
<td>PIN 5</td>
</tr>
<tr>
<td>10 BITS</td>
<td>8</td>
<td>PIN 4</td>
</tr>
<tr>
<td>8 BITS</td>
<td>9</td>
<td>PIN 3</td>
</tr>
</tbody>
</table>

CALIBRATION PROCEDURE

1. Connect converter as shown in the Standard Connection diagrams. Use the Input Connection Table for the desired input voltage range and input impedance. Apply Start Convert pulses of 100 nsec. minimum duration to pin 21. The spacing of the pulses should be no less than the maximum conversion time.

2. Zero and Offset Adjustments

   Apply a precision voltage reference source between the selected analog input and ground. Adjust the output of the reference source to the value shown in the Calibration Table for the unipolar zero adjustment (zero % LSB) or the bipolar offset adjustment (±FS% LSB). Adjust the trimming potentiometer so that the output code flickers equally between 1111 1111 1111 and 1111 1111 1110.

3. Full Scale Adjustment

   Change the output of the precision voltage reference source to the value shown in the Calibration Table for the unipolar or bipolar full scale adjustment (+FS% LSB). Adjust the gain trimming potentiometer so that the output code flickers equally between 0000 0000 0001 and 0000 0000 0000.

CALIBRATION TABLE

<table>
<thead>
<tr>
<th>UNIPOLAR RANGE</th>
<th>ADJUST. INPUT VOLTAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 TO +5V</td>
<td>ZERO GAIN = -0.6 mV</td>
</tr>
<tr>
<td>0 TO +10V</td>
<td>ZERO GAIN = +1.2 mV</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BIPOLAR RANGE</th>
<th>OFFSET GAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>±2.5V</td>
<td>-2.4994 mV</td>
</tr>
<tr>
<td>±5V</td>
<td>-2.4988 mV</td>
</tr>
<tr>
<td>±10V</td>
<td>-3.9916 mV</td>
</tr>
</tbody>
</table>

CLOCK RATE ADJUSTMENT

- 600 kHz to 720 kHz (ADC-HX128)
- 1.5 MHz to 1.8 MHz (ADC-HZ128)

- 600 kHz to 880 kHz (ADC-HX128)
- 1.5 MHz to 2.2 MHz (ADC-HZ128)
APPENDIX D

* SAMPLE
* THIS PROGRAM SAMPLES TWO SELECTED FILTER CHANNELS,
* WHOSE VALUES ARE STORED IN FILTER AND FILTER+1,
* AT 6.7KHZ SAMPLING RATE STORING THE VALUES
* STARTING FROM LOCATIONS POINTED BY BUFFR1 AND
* BUFFR2. IT THEN JUMPS TO THE NORMALIZATION SUBROU-
* TIME

033A ORG $33A

033A 4C 4B 03 JMP START

033D DDRA2 EQU $E843
033D ORA EQU $E841
033D IFR EQU $E84D
033D IORB EQU $E820
033D BUFFR1 EQU $83
033D BUFFR2 EQU $3A
033D BUFFER EQU $49
033D BUFFE2 EQU $0F
033D 52 CNT HEX 52
033E 12 CNT+1 HEX 12
033F 90 NOISE HEX 90
0340 43 NSAMPL HEX 43
0341 46 CRT0 HEX 46
0342 00 OLY1 BSS 1
0343 00 DLY2 BSS 1
0344 00 TEMP1 BSS 1
0345 00 TEMP2 BSS 1
0346 00 00 FILTER BSS 2
0348 00 MAX BSS 1
0349 00 CTR BSS 1
034A 00 CTRFIL BSS 1

034B A9 FF START LDA =$FF
034D 8D 43 E8 STA DDRA2
0350 A9 00 LDA =$00
0352 85 88 STA BUFFR1
0354 A9 53 LDA =$53
0356 85 89 STA BUFFR1+1
0358 A9 00 LDA =$00
035A 85 8A STA BUFFR2
035C A9 66 LDA =$66
035E 85 83 STA BUFFR2+1
0360 A0 00 LDY =00
0362 A2 00 LDX =00
0364 AD 41 E8 LDA ORA
0367 AD 46 03 CHECK LDA FILTER
036A 8D 41 E8 STA ORA
036D AD 40 E8 C1 LDA IFR
0370 29 02 AND =2
0372 F0 F9 BEQ C1
0374 AD 41 E8 LDA ORA
0377 AD 46 03 LDA FILTER
037A 09 20 ORA =%00100000
037C 8D 41 E8 STA ORA
037F 29 17 AND =%00010111
0381 8D 41 E8 STA ORA
0384 20 F7 03 JSR DLY20
0387 AD 20 E8 LDA IOR3
038A 49 FF EOR =FF
038C CD 3F 03 CMP NOISE
038F 90 DC BCC C1
0391 A9 52 LDA =52
0393 8D 3D 03 STA CNT
0396 A9 12 LDA =12
0398 8D 3E 03 STA CNT+1
039B AD 46 03 SAMPLE LDA FILTER
039E 8D 41 E8 STA ORA
03A1 AD 40 E8 S11 LDA IFR
03A4 29 02 AND =2
03A6 F0 F9 BEQ S11
03A8 AD 41 E8 LDA ORA
03AB AD 46 03 LDA FILTER
03AE 09 20 ORA =%00100000
03B0 8D 41 E8 STA ORA
03B3 29 17 AND =%00010111
03B5 8D 41 E8 STA ORA
03B8 20 F7 03 JSR DLY20
03BB AD 20 E8 LDA IOR5
03BE 49 FF EOR =FF
03C0 91 83 STA (BUFFR1),Y
03C2 AD 47 03 LDA FILTER+1
03C5 8D 41 E8 STA ORA
03C8 09 20 ORA =%00100000
03CA 8D 41 E8 STA ORA
03CD 29 17 AND =%00010111
03CF 8D 41 E8 STA ORA
03D2 20 F7 03 JSR DLY20
03D5 AD 20 E8 LDA IOR3
03DA 91 8A STA (BUFFR2),Y
03DC CB INY
03DD D0 04 BNE NEXT
03DF E6 89 INC BUFFR1+1
03E1 E6 8B INC BUFFR2+1
03E3 CE 3D 03 NEXT DEC CNT
03E6 D0 83 BNE SAMPLE
03E9 AD 3E 03 LDA CNT+1
03EB F0 06 BEQ FIN
03ED CE 3E 03 DEC CNT+1
03F0 4C 93 03 JMP SAMPLE
03F3 4C 06 4F FIN JMP INITPN
03F6 60 RTS
*CALLING THIS SUBROUTINE GIVES A 12US TIME DELAY
*
*
03F7 EA    DLY20  NOP
03F8 60    RTS
*
APPENDIX E

* * *
PEAK NORMALIZATION
* THIS PROGRAM NORMALIZES THE RAW SPEECH DATA
* *

4F00 ORG $4F00
  *
  4F00 00 00 SUM BSS 2
  4F02 00 00 TEMP3 BSS 2
  4F04 00 00 NUM BSS 2
  *
  4F06 A9 00 INITPN LDA =00
  4F08 8D 4A 03 STA CTRFL
  4F0A 85 88 LDA BUFFER1
  4F0C 85 48 STA BUFFER
  4F0E 85 0F STA BUFFER2
  4F10 A5 89 LDA BUFFER1+1
  4F12 85 4C STA BUFFER+1
  4F14 85 10 STA BUFFER2+1
  4F16 A9 52 STRTPN LDA =$52
  4F18 8D 3D 03 STA CNT
  4F1A A9 12 LDA =$12
  4F1C 8D 3E 03 STA CNT+1
  4F1E A9 00 LDA =00
  4F20 8D 48 03 STA MAX
  4F22 A0 00 LDY =0
  4F24 A2 00 LDX =0
  4F26 B1 48 PEAK LDA (BUFFER),Y
  4F28 CD 48 03 CMP MAX
  4F2A 90 03 BCC P1
  4F2C 8D 48 03 STA MAX
  4F2E 03 C3 P1 INY
  4F30 D0 02 BNE P2
  4F32 EE 4C INC BUFFER+1
  4F34 CE 3D 03 P2 DEC CNT
  4F36 D0 EC BNE PEAK
  4F38 A0 3E 03 LDA CNT+1
  4F3A F0 06 BEQ P3
  4F3C CE 3E 03 DEC CNT+1
  4F3E 4C 2A 4F JMP PEAK
  4F42 A9 52 P3 LDA =$52
  4F44 8D 3D 03 STA CNT
  4F46 A9 12 LDA =$12
4F50 8D 3E 03 STA CNT+1
4F53 AD 48 03 LDA MAX
4F56 8D 04 4F STA NUM
4F59 A9 00 LDA =00
4F5B 8D 05 4F STA NUM+1
4F5E A0 00 LDY =0
4F60 81 0F P4 LDA (BUFFE2),Y
4F62 8D 02 4F STA TEMP3
4F65 A9 FF LDA =$FF
4F67 8D 03 4F STA TEMP3+1
4F6A 20 A2 4F JSR MULT
4F6D 20 BB 4F JSR DIVIDE
4F70 AD 00 4F LDA SUM
4F73 91 0F STA (BUFFE2),Y
4F75 C8 INY
4F76 D0 02 BNE P5
4F78 E6 10 INC BUFFE2+1
4F7A CE 3D 03 P5 DEC CNT
4F7D D0 E1 BNE P4
4F7F AD 3E 03 LDA CNT+1
4F82 F0 06 BEQ P6
4F84 CE 3E 03 DEC CNT+1
4F87 4C 60 4F JMP P4
4F8A AD 4A 03 P6 LDA CTRFIL
4F8D D0 12 BNE P8
4F8F A5 8A P7 LDA BUFFER2
4F91 85 4B STA BUFFER
4F93 85 0F STA BUFFER2
4F95 A5 83 LDA BUFFER2+1
4F97 25 4C STA BUFFER+1
4F99 85 10 STA BUFFER2+1
4F9B EE 4A 03 INC CTRFIL
4F9E 4C 17 4F JMP STRIPN
4FA1 60 P8 RTS
MULT
*THIS SUBROUTINE MULTIPLIES AN 8-BIT NUMBER
*IN TEMP3 BY ANOTHER 8-BIT NUMBER IN TEMP3+1
*THE 16-BIT RESULT IS STORED IN SUM & SUM+1
*
4FA2 A9 00    MULT    LDA  =00
4FA4 A2 08    LDX  =08
4FA6 4E 03 4F  NXT    LSR  TEMP3+1
4FA9 90 04    BCC  ALIGN
4FAB 18       CLC
4FAC 6D 02 4F  ADC  TEMP3
4FAF 6A       ALIGN  ROR  A
4FB0 6E 00 4F  ROR  SUM

4FB3 CA       DEX
4FB4 D0 F0    BNE  NXT
4FB6 8D 01 4F  STA  SUM+1
4FB9 60       RTS
DIVIDE

THIS SUBROUTINE DIVIDES A 16-BIT NUMBER STORED IN SUM & SUM+1 BY AN 8-BIT NUMBER STORED IN NUM. THE RESULT IS STORED IN SUM.

```
4FB8 8A   DIVIDE TXA
4FBC 48   PHA
4FBD 98   TYA
4FBE 48   PHA
4FBF AD 04 4F   LDA NUM
4FC2 F0 33   BNE DIVO
4FC4 A9 00   LDA =0
4FC6 8D 02 4F   STA TEMP3
4FC9 8D 03 4F   STA TEMP3+1
4FCC 42 10   LDX =16
4FCE 0E 00 4F   NXTBT ASL SUM
4FD1 2E 01 4F   ROL SUM+1
4FD4 2E 02 4F   ROL TEMP3
4FD7 2E 03 4F   ROL TEMP3+1
4FDA AD 02 4F   LDA TEMP3
4FDD 38   SEC
4FDE ED 04 4F   SBC NUM
4FE1 A8   TAY
4FE2 AD 03 4F   LDA TEMP3+1
4FE5 ED 05 4F   SBC NUM+1
4FE8 90 0A   BCC CNTDN
4FEA EE 00 4F   INC SUM
4FED 8D 03 4F   STA TEMP3+1
4FF0 98   TYA
4FF1 8D 02 4F   STA TEMP3
4FF4 CA   CNTDN DEX
4FF5 DD 07   BNE NXTBT
4FF7 68   DIVO PLA
4FF9 A8   TAY
4FF9 68   PLA
4FFA AA   TAX
4FF8 60   RTS
```
APPENDIX F

DATA REDUCTION(1)

* This program reduces the raw speech data by
* using data reduction method 1, that is, averaging the speech data every 10ms.

5140 ORG $5140
5140 LOC EQU $5200

5140 A9 00 MPADD LDA =00
5142 8D 4A C3 STA CTRFIL
5145 A9 43 LDA =67
5147 8D 40 03 STA NSAMPL
514A 8D 04 4F STA NUM
514D A9 00 LDA =00
514F 8D 05 4F STA NUM+1
5152 A9 00 LDA =0
5154 85 4B STA BUFFER
5156 A9 53 LOA =$53
5158 85 4C STA BUFFER+1
515A A2 00 LDX =0
515C A9 46 STRADD LDA =$46
515E 8D 41 03 STA CTR70
5161 A0 00 LDY =0
5163 18 AO CLC
5164 A9 00 LDA =00
5166 8D 00 4F STA SUM
5169 8D 01 4F STA SUM+1
516C AD 40 03 LDA NSAMPL
516F 8D 49 03 STA CTR
5172 B1 4B ADD LOA (BUFFER),Y
5174 60 00 4F ADC SUM
5177 90 03 BCC A1
5179 EE 01 4F INC SUM+1
517C 8D 00 4F A1 STA SUM
517F C8 INY
5180 D0 02 BNE A2
5182 E6 4C INC BUFFER+1
5184 CE 49 03 A2 DEC CTR
5187 D0 E9 BNE ADD
5189 20 B3 4F JSR DIVIDE
518C AD 00 4F LDA SUM
518F 90 00 52 STA LOC,X
5192 E8 INX
5193 CE 41 03 A3 DEC CTR70
5196 D0 CB BNE AO
5199 AD 4A 03 LDA CTRFIL
519B D0 0E BNE FINISH
519D A9 00 LDA =0
519F 85 4B STA BUFFER
51A1 A9 66 LDA =$66
51A3 85 4C STA BUFFER+1
51A5 EE 4A 03 INC CTRFIL
51A8 4C 5C 51 JMP STRADD
51AB 58 FINISH CLI
51AC 60 RTS
END
APPENDIX G

DATA REDUCTION (2)
* THIS PROGRAM REDUCES THE RAW SPEECH DATA
* USING DATA REDUCTION METHOD 2, THAT IS, IT
* FINDS THE MAX/MIN POINTS AND THE TIME
* BETWEEN PEAKS IN A SPEECH SIGNAL.

* *

5000 ORG $5000
5000 00 TEMP3 BSS 1
5001 00 TEMPO BSS 1
5002 00 00 FPMEM BSS 2
5004 00 FLAG BSS 1
5005 00 CI BSS 1
5006 DATA1 EQU $785C
5006 DATA2 EQU $795C

5006 A9 03 START2 LDA =3
5008 BD 49 03 STA CTR
5008 A0 00 LOY =0
5000 A2 00 LDX =0
500F A9 00 LDA =0
5011 BD 04 50 STA FLAG
5014 BD 02 50 STA FPMEM
5017 BD 03 50 STA FPMEM+1
501A BD 05 50 STA CI
5010 38 SET SEC
501E A9 4F LDA =$4F
5020 ED 02 50 SBC FPMEM
5023 BD 30 03 STA CNT
5026 A9 12 LDA =$12
5028 ED 03 50 SBC FPMEM+1
5029 BD 3E 03 STA CNT+1
502E EA NOP
502F EA NOP
5030 EA NOP
5031 B1 4B S0 LOA (BUFFER),Y
5033 BD 01 50 STA TEMPO
5036 C8 INY
5037 DO 02 BNE S1
5039 E6 4C INC BUFFER+1
5038 B1 4B S1 LOA (BUFFER),Y
503D BD 44 03 STA TEMP1
5040 C8 INY
5041 DO 02 BNE S2
5043 E6 4C INC BUFFER+1
5045 B1 4B S2 LOA (BUFFER),Y
5047 BD 45 03 STA TEMP2
504A C8 S3 INY
504B DO 02 BNE S4
504D E6 4C INC BUFFER+1
504F B1 4B S4 LOA (BUFFER),Y
5051 BD 02 4F STA TEMP3
5054 AD 04 50 LDA FLAG
5057 DO 06 BNE CHECK2
5059 20 E2 50 JSR FPEAK
505C 4C 1D 50 JMP SET
505F AD 44 03 CHECK2 LDA TEMP1
5062 CD 45 03 CMP TEMP2
5065 90 20 BCC MIN2
5067 38 SEC
5068 ED 01 50 SBC TEMPO
506B 90 0F BCC MIN1
506D C9 05 CMP =5
506F 90 26 BCC C22
5071 AD 44 03 MAX LDA TEMP1
5074 CD 02 4F CMP TEMP3
5077 90 1E BCC C22
5079 4C A6 50 JMP C32
507C AD 44 03 MIN1 LDA TEMP1
507F 38 SEC
5080 ED 01 50 SBC TEMPO
5083 C9 05 CMP =5
5085 90 10 BCC C22
5087 AD 44 03 MIN2 LDA TEMP1
508A CD 02 4F CMP TEMP3
508D B0 08 BCS C22
508F AD 01 50 LDA TEMPO
5092 CD 02 4F CMP TEMP3
5095 B0 0F BCS C32
5097 EE 49 03 C22 INC CTR
509A AD 49 03 LDA CTR
509D C9 FD CMP =253
509F D0 1C BNE CONT
50A1 A9 00 LDA =0
50A3 9D 45 03 STA TEMP2
50A6 AD 49 03 C32 LDA CTR
50A9 9D 5C 79 STA DATA2,X
50AC E8 INX
50AD F0 30 BEQ FIN2
50AF EE 05 50 INC C1
50B2 AD 45 03 LDA TEMP2
50B5 90 5C 78 STA DATA1,X
50B8 A9 00 LDA =0
50BA BD 49 03 STA CTR
50BD CE 30 03 CONT DEC CNT
50C0 D0 08 BNE CONT1
50C2 AD 3E 03 LDA CNT+1
50C5 F0 18 BEQ FIN2
50C7 CE 3E 03 DEC CNT+1
50CA AD 44 03 CONT1 LDA TEMP1
50CD BD 01 50 STA TEMPO
50D0 AD 45 03 LDA TEMP2
50D3 BD 44 03 STA TEMP1
50D6 AD 02 4F LDA TEMP3
50D9 BD 45 03 STA TEMP2
50DC 4C 4A 50 JMP S3
50DF 60 FIN2 RTS
50E0 EA NOP
50E1 EA NOP
* THIS SUBROUTINE FINDS THE FIRST PEAK IN
* THE SPEECH SIGNAL.
*
50E2 AD 44 03 FPEAK  LDA TEMP1
50E5 CD 45 03    CMP TEMP2
50E8 B0 12       BCS F2
50EA 38          SEC
50EB ED 01 50    SBC TEMPO
50EE 90 0C       BCC F2
50F0 C9 0A       CMP =10
50F2 90 08       BCC F2
50F4 AD 45 03 F1  LDA TEMP2
50F7 CD 02 4F    CMP TEMP3
50FA B0 25       BCS F4
50FC AD 44 03 F2  LDA TEMP1
50FF 8D 01 50    STA TEMPO
5102 AD 45 03    LDA TEMP2
5105 8D 44 03    STA TEMP1
5108 AD 02 4F    LDA TEMP3
510B 8D 45 03    STA TEMP2
510E EE 02 50    INC FPMEM
5111 C8          INY
5112 D0 05       BNE F3
5114 E6 4C       INC BUFFER+1
5116 EE 03 50    INC FPMEM+1
5119 B1 4B F3    LDA (BUFFER)*Y
511B 8D 02 4F    STA TEMP3
511E 4C E2 50    JMP FPEAK
5121 AD 45 03 F4  LDA TEMP2
5124 9D 5C 78    STA DATA1*X
5127 EE 05 50    INC CI
512A A9 01       LDA =1
512C 8D 04 50    STA FLAG
512F 60          RTS
APPENDIX H

20 REM SPEECH MENU
30 INPUT "PLEASE TYPE 1-FOR TRAINING MODE OR 2-FOR RECOGNIZE: "; S
40 IF S<2 GOTO 60
50 LOAD "TRAIN"
60 IF S>2 GOTO 30
70 LOAD "RECOGNIZE"
80 END
APPENDIX I

5 REM PROGRAM TRAIN
10 REM TRAINING PROGRAM FOR SYSTEM
25 POKE 57459,255:REM MAKE VIA PAO-7 OUTPUT
30 INPUT "NO. OF WORDS:";N
35 A=1:GOTO 65
40 OPEN 4,1,1
45 PRINT #4,N
46 PRINT #4,F1
47 PRINT #4,F2
50 CLOSE 4
55 H=1
60 IF A=2 GOTO 90
65 PRINT "PLACE VOCABULARY CASSETTE IN RECORDER"
70 PRINT "PRESS ANY KEY TO CONTINUE"
75 GET A:IF A="" THEN 75
80 INPUT "FILTER TO BE SAMPLED:";F1,F2
85 IF A=1 THEN A=A+1:GOTO 40
90 POKE 835,(F1-1)+16:REM SELECT FILTER #F1
95 POKE 835,(F2-1)+16:REM SELECT FILTER #F2
100 PRINT "PRESS ANY KEY TO CONTINUE"
105 GET A:IF A="" THEN 105
110 POKE 59471,16:REM ENABLE 5-H LOGIC
115 POKE 59411,65:REM DISABLE 50HZ INTERRUPT
120 PRINT "SAY WORD ",M
125 SYS(826)
130 POKE 59471,00:REM DISABLE 5-H
135 POKE 59411,61:REM ENABLE 50HZ INTERRUPT
140 GOSUB 800
150 GOSUB 1100
155 M=M+1
160 IF M<=N GOTO 100
165 PRINT "YOU NOW HAVE YOUR VOC. STORED ON CASSETTE."
170 PRINT
175 PRINT "DO YOU WANT TO ENTER RECOGNIZE MODE (Y/N):"
180 GET A:IF A="" THEN 180
185 IF A<"Y" GOTO 500
190 PRINT "PLACE BASIC CASSETTE IN RECORDER"
195 PRINT "PRESS ANY KEY TO CONTINUE"
200 GET A:IF A="" THEN 200
205 LOAD "RIEXIT"
600 REM TO FIND AVERAGE OVER IOHS.
610 SYS(20800)
620 RETURN
1100 OPEN 4,1,1
1110 FOR I=0 TO 139
1120 A=PEEK(20992+I)
1130 PRINT#4,A
1150 PRINT A
1140 NEXT I
1150 CLOSE 4
1160 RETURN
APPENDIX J

5 REM PROGRAM RECOG
10 REM RECOGNITION ROUTINE
15 DIM Z(255), X(255), D(255), Y(10,2), YY(10), TYY(10), T(11), TT(3)
20 P=0: C=0: CTR=0
25 PRINT "PLACE VOC. CASSETTE IN RECORDER"
30 PRINT "PRESS ANY KEY TO CONTINUE"
35 GET A+: IF A+=""THEN 35
40 OPEN 4, I, 0
45 INPUT #4, N
50 INPUT #4, F1
55 INPUT #4, F2
60 CLOSE 4
65 POKE 59459, 255: M=0
70 POKE 838, (F1-I)+16: POKE 839, (F2-I)+16
75 PRINT "PRESS ANY KEY TO CONTINUE"
80 GET A+: IF A+=""THEN 80
85 POKE 59471, I6: POKE 59411, 80
90 PRINT "SAY WORD"
95 SYS(826)
100 POKE 59471, 00: POKE 59411, 61
105 GOSUB 600: REM DATA REDUCTION
110 IF CTR=3 GOTO 165
115 IF C=I GOTO 128
120 IF M=N+1 OR C=0 GOTO 130
125 CLOSE 4
128 GOSUB 1200: REM PUT VOC. DATA IN ARRAY
129 II=0: YY(M)=0: NN(M)=0
134 GOSUB 800: REM COMPARE UNKNOWN & TEMPLATE
139 M=M+1: B=B+1: T(B)=M
140 IF M<N+1 AND C=0 GOTO 115
141 IF M<N+1 GOTO 128
145 GOSUB 1000: REM GOTO WORD IDENTIFICATION STAGE
150 IF CTR=3 AND C=0 THEN P=I: GOTO 25
155 IF CTR=3 AND C=I THEN P=I: F1=2: F2=4: GOTO 85
160 GOTO 235
165 IF C=I GOTO 200
170 REM GET DATA FOR REMAINING CHANS.
175 FOR M=1 TO 10
180 OPEN 4, I, 0
185 GOSUB 700
190 CLOSE 4
195 NEXT M
200 REM PUT DATA FOR CANDIDATE WORDS IN ARRAY
205 FOR JJ=1 TO 3
210 M=TT(JJ)
215 GOSUB 1200
220 GOSUB 800
225 NEXT JJ
230 GOSUB 1000
235 PRINT
240 INPUT "DO YOU WISH TO RECOGNIZE ANOTHER WORD: ":A#
245 IF A$="Y" GOTO 500
250 C=1:P=0;F1=3;F2=5;GOTO 65
500 END
505 REM DATA REDUCTION OF UNKNOWN
610 SYS(20800)
620 FOR I=0 TO 139
630 X(I)=PEEK(20992+I)
640 NEXT I
650 RETURN
700 REM GET VOC. DATA
710 FOR I=0 TO 139
720 INPUT#4,A:Z(I)=A
725 POKE 16384+P*1400+(M-1)*140+I,A
730 NEXT I
740 RETURN
800 REM COMPARE UNKNOWN & TEMPLATE
805 IF CTR=3 THEN II=1;YY(M)=0
810 F=6000000;S=3
815 FOR K=-5 TO 5
820 Y(M,II)=0
830 FOR J=3 TO 66
840 D(II)=ABS(X(II*70+J+K)-Z(II*70+J))
850 Y(M,II)=Y(M,II)+D(II)*D(II)
860 NEXT J
870 IF Y(M,II)<F THEN F=Y(M,II);MS=K
880 NEXT K
890 Y(M,II)=F
891 PRINT "MS=";MS
900 YY(M)=YY(M)+Y(M,II)
920 RETURN
930 REM WORD IDENTIFICATION
935 IF CTR=3 GOTO 1130
1010 REM DIFF. OF 1ST. CHANNEL
1015 GOSUB 1310;REM FIND THREE CANDIDATE WORDS
1020 T(1)=M1;T(2)=M2;T(3)=M3;B=4
1025 REM COMPARE 1ST. & 2ND. LOWEST. DESICION?
1030 IF Y(H2,II)>Y(H1,II)*3 THEN RW=H1
1035 PRINT"DESICION CANNOT BE MADE AT THIS STAGE"
1040 PRINT"CANDIDATE WORDS ARE AS FOLLOWS:"  
1045 PRINT
1050 PRINT H1,"DIFF.="",Y(H1,II)
1055 PRINT H2,"DIFF.="",Y(H2,II)
1060 PRINT H3,"DIFF.="",Y(H3,II)
1065 PRINT
1070 CTR=2:II=1
1071 FOR JJ=1 TO B-1
1072 M=T(JJ)
1073 GOSUB 1200
1075 GOSUB 800:REM COMPARE 2ND CHANL FOR CANDIDATES
1076 NEXT JJ
1080 GOSUB 1310
1085 TT(I)=H1:TT(2)=H2:TT(3)=H3
1090 IF T(I)<TT(I) GOTO 1100
1095 IF Y(H2,II)>Y(H1,II)*3 THEN RW=H1:GOTO 1190
1100 IF YY(H2)>YY(H1)*3/2 AND YY(H3)>YY(H1)*3/2 THEN RW=H1
1101 IF YY(H1)>YY(H2)*2 THEN RW=H2:GOTO 1190
1105 TYY(H1)=YY(H1):TTY(H2)=YY(H2):TTY(H3)=YY(H3)
1110 CTR=3:PRINT
1115 PRINT"DESICION CANNOT BE MADE. UNKNOWN WORD MUST BE"
1120 PRINT
1125 PRINT"FOR TWO CHANL. DIFFERENCES ARE:"  
1126 PRINT"WORD",H1,YY(H1)
1127 PRINT"WORD",H2,YY(H2)
1130 PRINT"WORD",H3,YY(H3)
1135 GOTO 1195
1140 B=4
1145 PRINT
1150 GOSUB 1310
1155 PRINT"FOR CHANL.4, DIFFERENCES ARE:"  
1160 PRINT"WORD",H1,Y(H1,II)
1165 PRINT"WORD",H2,Y(H2,II)
1170 PRINT"WORD",H3,Y(H3,II)
1175 PRINT TT(I)=H1:TT(2)=H2:TT(3)=H3
1180 FOR JJ=1 TO B-1
1185 M=TT(JJ)
1190 IF NN(H1)>=2 AND TYY(H1)<TTY(H2) THEN RW=H1:GOTO 1190
1195 IF NN(H2)>=2 AND TYY(H2)<TTY(H1) THEN RW=H2:GOTO 1190
1200 IF NN(H3)>=2 AND TTY(H3)<TTY(H1) THEN RW=H3:GOTO 1190
1205 PRINT
1210 PRINT"WORD REJECTED"
1215 GOTO 1193
1190 PRINT
1191 PRINT "RECOGNIZED: ", RW
1192 PRINT
1193 CTR=1
1194 RETURN
1200 REM PUT VOC. DATA IN AN ARRAY
1210 FOR I=0 TO 139
1220 Z(I)=PEEK(16384+F*1400+(H-1)*140+I)
1230 NEXT I
1240 RETURN
1310 REM FIND SMALLEST DIFF.
1320 H=600000: W0=0: W1=0: W2=0
1330 GOSUB 1500: REM FIND WORD WITH MIN. DIFF.
1340 W1=W0: H1=H0: NN(H1)=NN(H0)+1
1350 H=600000
1360 GOSUB 1500: REM FIND 2ND LOWEST
1370 W2=W0: H2=H0
1380 H=600000
1390 GOSUB 1500: REM FIND 3RD LOWEST
1400 W3=W0: H3=H0
1410 RETURN
1500 REM FIND SMALLEST DIFF.
1510 FOR J=1 TO B-1
1515 IF CTR=3 THEN H=TT(J): GOTO 1530
1520 H=T(J)
1530 IF H=W1 OR H=W2 GOTO 1560
1540 IF Y(H, II)>H GOTO 1560
1550 H=Y(H, II): W0=H
1560 NEXT J
1570 RETURN