AD-A084 743 ITT DEFENSE COMMUNICATIONS DIV NUTLEY N J F/G 9/2
SOLID STATE AUDIO/SPEECH PROCESSOR ANALYSIS.(U)
MAR 80 A R SMITH, B P LANDELL, G VENSKO F30602-78-C-0359
UNCLASSIFIED RADC-TR-80-75 NL

1 OF 2
AD
A084 743

RADC-TR-80-75
Final Technical Report
March 1980

SOLID STATE AUDIO/SPEECH PROCESSOR ANALYSIS

ITT Defense Communications Division

Dr. A. Richard Smith
Mr. B. Patrick Landell
Mr. George Vensko

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-80-75 has been reviewed and is approved for publication.

APPROVED: Melvin G. Manor, Jr.

MELVIN G. MANOR, Jr.
Project Engineer

APPROVED: Thadeus J. Domurat

THADEUS J. DOMURAT
Acting Chief, Intelligence and Reconnaissance Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (IRAA), Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: RADC-TR-80-75	2. GOVT ACCESSION NO.: AD-A084 743
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): SOLID STATE AUDIO/SPEECH PROCESSOR ANALYSIS.	5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report. 28 Sep 78 — 7 Dec 79
	6. PERFORMING ORG. REPORT NUMBER: N/A
7. AUTHOR(s): Dr. A. Richard Smith; Mr. B. Patrick Landell; Mr. George Vensko	8. CONTRACT OR GRANT NUMBER(s): F30602-78-C-0359
9. PERFORMING ORGANIZATION NAME AND ADDRESS: ITT Defense Communications Division, 492 River Road, Nutley NJ 07110	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62702F, 45941576
11. CONTROLLING OFFICE NAME AND ADDRESS: Rome Air Development Center (IRAA), Griffiss AFB NY 13441	12. REPORT DATE: Mar[illegible]1980
	13. NUMBER OF PAGES: 94
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Same	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Melvin Manor (IRAA)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Realtime Word Recognition	Low Cost Speech Recognition
Charge Couple Devices	Microprocessor Speech Systems
Dynamic Programming	Speech Analysis

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report details the progress made ~~by ITTDCD~~ in evaluating the feasibility of applying Charge Coupled Devices (CCD's) and microprocessors to reduce the cost and complexity of Automatic Speech Recognition (ASR) systems. The report answers two basic questions. First, can CCD devices be used to generate speech recognition parameters that are useful for accurate, low cost speech recognition? Second, what would be the cost and complexity for a realtime

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Automatic Word Recognition (AWR) system, using a CCD speech analyzer and current microprocessor technology?

To answer these questions, three speech analysis techniques were implemented with CCD analyzers. These techniques include a Discrete Fourier Transform analysis, a Cepstral analysis, and a Bandpass Filter analysis. The CCD analyzers were incorporated into a realtime laboratory AWR system based on a dynamic programming match algorithm.

Speaker dependent word recognition experiments were conducted for a performance comparison of the three CCD based speech analysis techniques. The data base used in the recognition experiments was based on two vocabularies of 26 and 20 words recorded by eight different speakers. Results indicated that the Bandpass Filter CCD analyzer provides the best parameters for isolated word recognition. A recognition accuracy of 99.4% was achieved on a 20 word vocabulary. The experiments showed that CCD speech analyzers can provide speech parameters which are useful for accurate realtime word recognition.

Experiments were also conducted to measure the speed versus accuracy tradeoffs of four speed-up techniques. The techniques were demonstrated to be worthwhile in an efficient realtime AWR system.

Finally, microprocessor architectures were designed to implement the realtime AWR system and then evaluated in terms of cost and complexity using three different microprocessors: the 8-bit Intel 8085A, the 16-bit Motorola MC68000, and a 16-bit configuration of the AMD 2901A. Of the three, the AMD 2901A proved preferable from both a cost and a performance standpoint. Hardware cost projections for an AWR system featuring an AMD 2901A architecture and a Bandpass Filter CCD analyzer indicate that the hardware components for such a system should range between $1,500 for a 52 word vocabulary, to about $12,700 for a 780 word vocabulary. These costs do not include custom chip development, detailed hardware design, construction or testing.

ITTDCD is very encouraged by the results obtained in this investigation. It does appear that an accurate, low cost AWR system could be developed using a CCD speech analyzer and a microprocessor recognition system.

ACKNOWLEDGEMENTS

We want to acknowledge the contributions of three people in various stages of this study. Dr. George White was responsible for the proposal effort leading up to the study and directed the inital stages of the work. Later after leaving ITTDCD to form his own company, he contributed to the work as a consultant. Dr. Robert Brodersen of the University of California, Berkley, was responsible for furnishing the simulation data of CCD speech analysis techniques and for the design and development of the hardware CCD analyzers used in this study. Mr. Allan Samuels performed the preliminary simulation studies and coded most of the realtime automatic word recognition system in the Quintrell processor.

EVALUATION

This contract was in support of TPO R1B, Signal Intelligence. Speech processors have applications to Air Force problems of data entry, secure personnel entry, communications, and intelligence. This effort determined the feasibility of using CCD and microprocessor technology for speech processors, in order to obtain significant reductions in cost, size, weight, and power consumption. Additional work is programmed to refine algorithms and hardware, and to construct a breadboard Automatic Speech Recognition System.

Melvin G. Manor, Jr.

MELVIN G. MANOR, Jr.
Project Engineer

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS

LIST OF TABLES

LIST OF TABLES

Chapter 1: EXECUTIVE SUMMARY

This document is the final report from ITTDCD to RADC on Contract Number F3062-78-C-0359, entitled Solid State Audio/Speech Processor Analysis. This report details the progress made by ITTDCD in evaluating the feasibility of applying Charge Coupled Devices (CCD'S) and microprocessors to reduce the cost and complexity of Automatic Speech Recognition (ASR) systems. ASR systems have the potential of solving many Air Force command, control, and communication problems. The report answers two basic questions. First, can CCD devices be used to generate speech recognition parameters that are useful for accurate low cost speech recognition? Second, what would be the cost and complexity for a realtime Automatic Word Recognition (AWR) system, using a CCD speech analyzer and current microprocessor technology?

To answer these questions, ITTDCD studied four speech analysis techniques. These techniques include a Discrete Fourier Transform (DFT) analysis, a Cepstral analysis, a Bandpass Filter (BPF) analysis, and Linear Predictive Coding (LPC). To provide more realistic data for performance assessment, the first three techniques were implemented using actual CCD hardware. For each of these CCD analyzers, software was developed to make the respective speech parameters more suitable for realtime word recognition.

ITTDCD then incorporated the CCD hardware and software into a realtime laboratory AWR system. The major hardware components of this system include the CCD speech analyzers, a Quintrell signal processor, a PDP-11/60 minicomputer, and two display terminals. Realtime word recognition software was developed for the Quintrell processor, a high speed signal processor originally designed by ITTDCD for narrowband speech transmission systems. The word recognition software is based on a dynamic programming match algorithm to provide effective time normalization. The laboratory AWR system was designed to meet the performance goals of the contract: realtime response on a 20 word vocabulary with better than 99% word recognition accuracy in a speaker dependent mode.

ITTDCD employed the laboratory AWR system to conduct a series of word recognition experiments. These experiments were designed with the intent of comparing alternative word recognition techniques and algorithms, rather than demonstrating the maximum achievable accuracy of a realtime AWR system. The data base used in the recognition experiments was based on two vocabularies of 26 and 20 words recorded by eight different speakers.

A first group of experiments was designed and conducted for a performance comparison of the three CCD based speech analysis techniques. Results obtained from these experiments clearly indicated that the Bandpass Filter CCD analyzer provides the best parameters for isolated word recognition. The laboratory AWR system was also used as a test vehicle for a second group of experiments measuring the speed versus accuracy tradeoffs of four speed-up techniques. All four of the speed-up algorithms which were studied were demonstrated to be worthwhile in an efficient realtime AWR system. A third group of experiments provided an additional performance evaluation of the Bandpass Filter AWR system. A recognition accuracy of 99.4% was achieved on a 20 word vocabulary, thus surpassing the contract performance goal. Overall, the experiments showed that CCD speech analyzers can provide speech parameters which are useful for accurate realtime word recognition.

Finally, ITTDCD designed and evaluated architectures for microprocessor based versions of the realtime AWR system. The microprocessor architectures were evaluated in terms of cost and complexity for solving various isolated word recognition problems using three different microprocessors: the 8-bit Intel 8085A, the 16-bit Motorola MC68000, and a 16-bit configuration of the AMD 2901A. Of the three, the AMD 2901A proved preferable from both a cost and a performance standpoint. Hardware cost projections were then made for an AWR system featuring an AMD 2901A architecture and a Bandpass Filter CCD analyzer. These projections indicate that the hardware components for such a system should range between $1,500 for a 52 word vocabulary, to about $12,700 for a 780 word vocabulary. These costs do not include custom chip development, detailed hardware design, construction or testing.

ITTDCD is very encouraged by the results obtained in this investigation. It does appear that an accurate, low cost AWR system could be developed using a CCD speech analyzer and a microprocessor recognition architecture. A logical next step would be to proceed with the detailed design, construction, and testing of a deliverable version of such an AWR system. The system could then be evaluated at an actual Air Force laboratory or operational site.

Chapter 2: INTRODUCTION AND TECHNICAL OVERVIEW

This report details the progress made by ITTDCD in evaluating the feasibility of applying Charge Coupled Devices (CCD'S) and microprocessors to improve the cost, size, weight, and power consumption of Automatic Speech Recognition (ASR) systems. ASR systems have the potential of solving many Air Force command, control, and communication problems. For many applications, however, such factors as cost, size, and power must be reduced. The report answers two basic questions. First, can CCD devices be used to generate speech recognition parameters that are useful for accurate low cost speech recognition? Second, what would be the cost and complexity for a realtime Automatic Word Recognition (AWR) system, using a CCD analyzer and current microprocessor technology?

Four tasks were carried out to answer these questions. In the first task, simulation studies were performed on an existing Automatic Word Recognition (AWR) system to develop speech recognition algorithms suitable for low cost microprocessor implementation. This task also provided experience with four speech processing techniques using data from simulated CCD analyzers. These techniques are Linear Predictive Coding (LPC), a Discrete Fourier Transform (DFT), a Cepstral analysis, and a Bandpass Filter analysis (BPF). In the second task, a realtime laboratory AWR system was designed and implemented on a Quintrell processor. This is a special purpose ITTDCD developed signal processor based on a AMD-2901 microprocessor. It can perform a full LPC analysis/synthesis in approximately one-half realtime. The AWR system used CCD analyzers to provide speech parameters, and was designed to meet the performance goals of the contract: realtime response on a 20 word vocabulary with better than 99% word recognition in a speaker dependent mode. The third task was to use this realtime AWR system to compare the recognition performance of actual CCD analyzers. In the final task, a component analysis of the realtime AWR system was performed to determine the speed/accuracy tradeoff of various parts of the recognition algorithm, the projected cost of the CCD hardware analyzers, and the cost and complexity of a microprocessor implementation of the system.

This work was enhanced in two ways by an ITTDCD IR&D program that focused on developing low cost speech recognition systems. First, the IR&D program developed the CCD analyzers and supporting realtime software. The availability of real CCD analyzers, though not necessary for completing this contract, permitted a more realistic comparison of CCD technology than could have been done with simulated data alone. CCD hardware was built for three of the speech processing techniques. (The LPC analysis was not implemented in hardware.) Second, a detailed microprocessor architecture study of the AWR system was conducted under IR&D for three microprocessors. These designs were evaluated in terms of cost and complexity for vocabulary sizes of up to 780 words.

The CCD hardware, the realtime laboratory AWR system, and the results of recognition experiments are discussed in subsequent chapters. The remainder of this chapter gives a technical overview of the ITTDCD AWR system, the speech processing techniques, and the test paradigm used throughout the study. Finally the simulation studies are discussed as background to the following chapters.

2.1 Overview of the ITTDCD Automatic Word Recognition System

Figure 2.1 shows a simple block diagram of the ITTDCD AWR system. Analog speech undergoes a parametric analysis to derive speech parameters that accurately describe the sounds present in each spoken word. These parameters are sampled periodically (usually every 10 to 25 ms) to produce "frames" that represent the time variation of the input. For this contract, CCD analyzers were evaluated for their ability to perform this speech analysis step. Separation of the parametric analysis from the rest of the recognition process is not only natural, but it also permits developing the AWR system independent of any particular speech analysis technique. In addition, the same AWR system could be used for each CCD analyzer technique to insure a valid comparison.

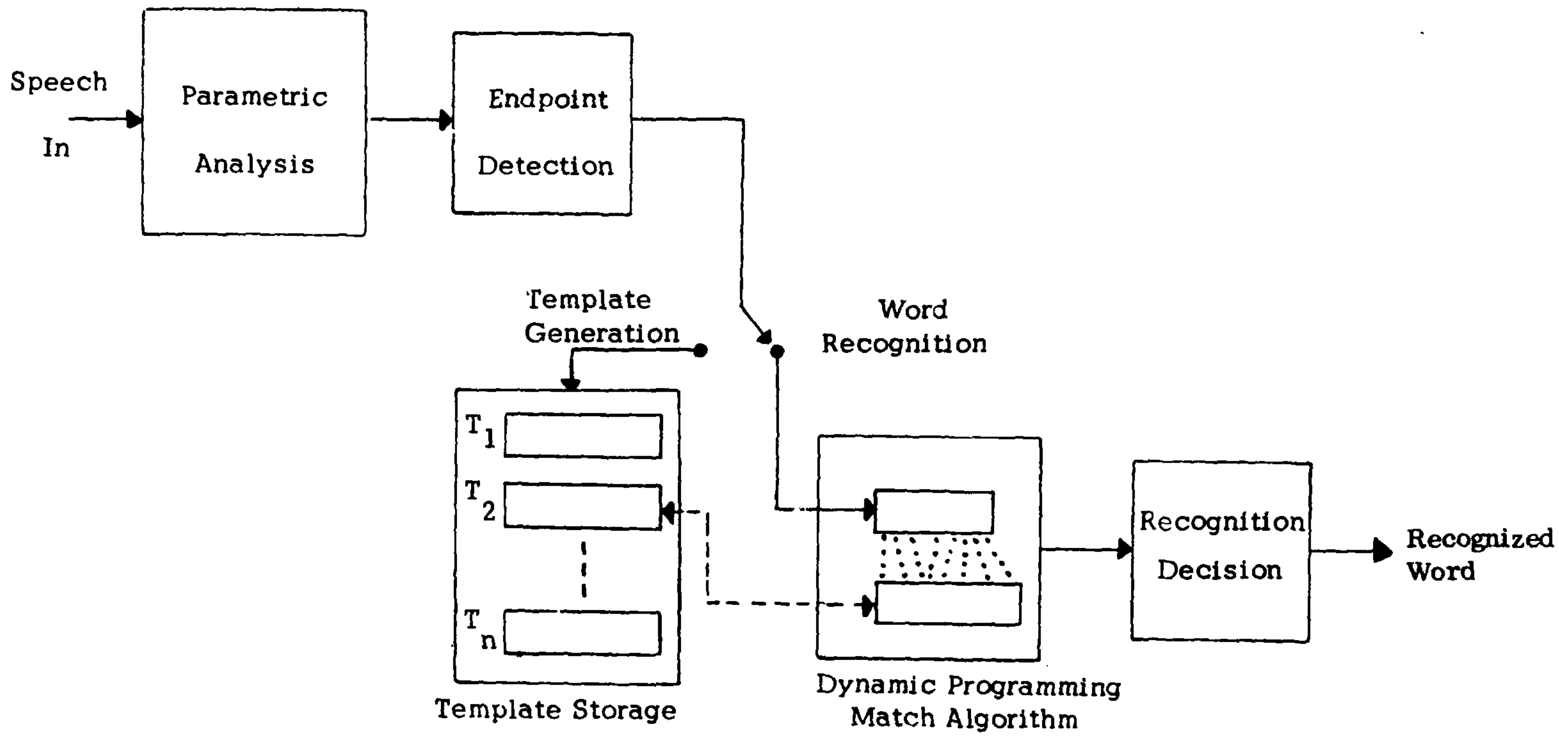


Figure 2.1 The ITTDCD Automatic Word Recognition System

The beginning and end of each word is found in the endpoint detection step by using an energy function derived in the parametric analysis. The word beginning is detected by summing a clipped energy function over about 200 ms. The energy function is clipped to prevent a high energy burst (e.g., lip smack) from triggering a false speech detection. Speech is detected when the sum passes a set threshold, and the beginning of the speech is marked as the first point in the 200 ms window where the energy is greater than a "silence" threshold. The word end is similarly detected when about 100 ms of the energy function remains below the "silence" threshold.

An unknown word is recognized by comparing it to the set of reference templates representing the vocabulary of the system. A template is generated by speaking the vocabulary word once and storing the parametric representation of the word as a template. The comparison between each template and unknown is performed with a non-linear time alignment process carried out by the dynamic programming match algorithm (discussed in the next section). The identity of the template best matching the unknown word is the system's response.

2.1.1 Dynamic Programming Match Algorithm

Non-linear time alignment between a word template and the unknown word is necessary to account for the natural time variations between different utterances of the same word. Figure 2.2 illustrates how a non-linear time alignment between a template and an unknown is achieved with dynamic programming. The time frames of the template on the y-axis and the time frames of the unknown on the x-axis form a matrix of frame-to-frame cells. (A dotted line is shown on the figure for every five frames of speech.) The piece-wise linear line cutting diagonally across the matrix shows one of many possible non-linear time alignment paths. The match score of the path is equal to the sum (or weighted sum) of the frame-to-frame "sound similarity" scores for each cell along the path. One common "sound similarity" score between the parameters of a template frame and the parameters of an unknown frame is simply the Euclidean. Dynamic programming finds the "best" non-linear time alignment path within set

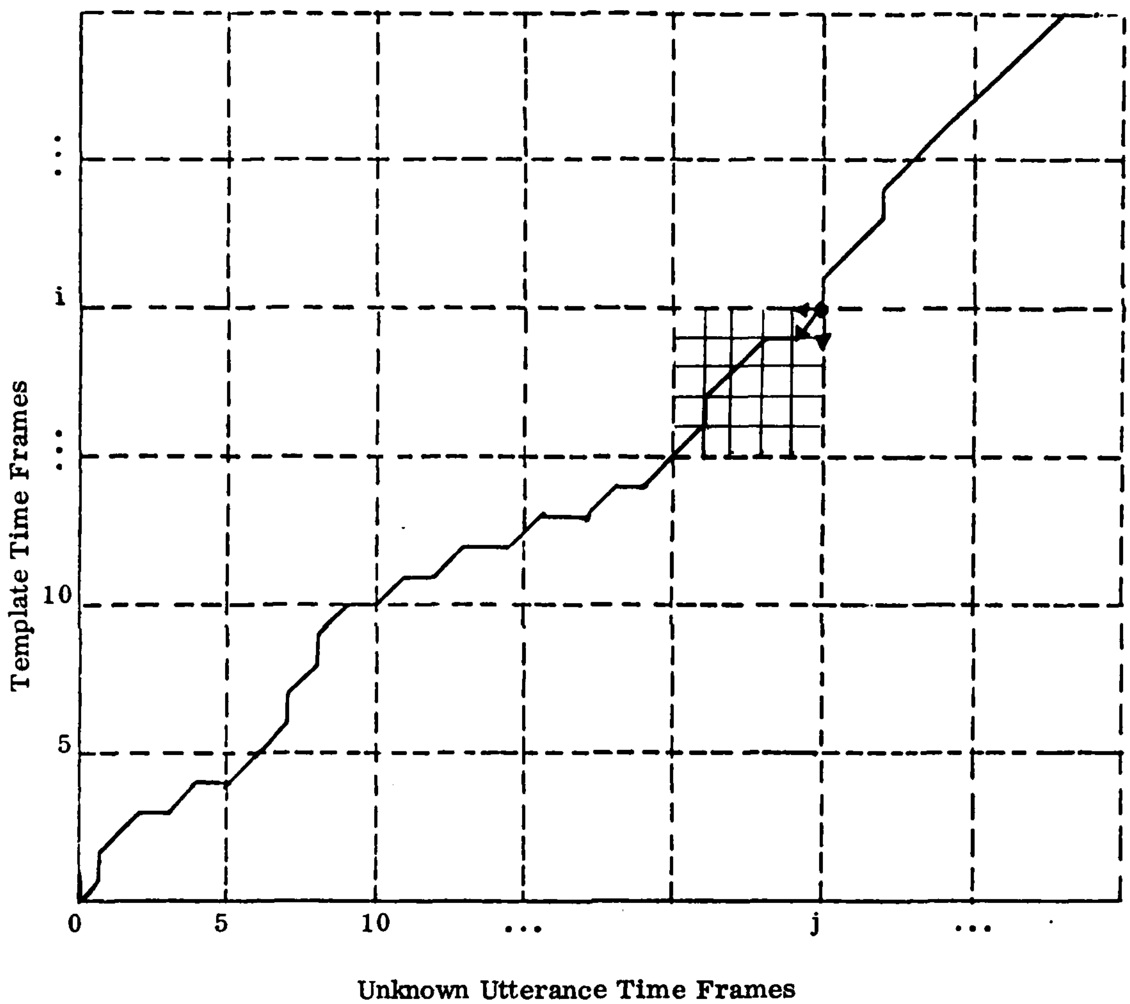


Figure 2.2 Dynamic Programming Matrix for Non-Linear Time Alignment

constraints by finding at each time cell (i,j) the best partial path in the three adjacent cells to extend to (i,j) as shown in the figure. This process begins in the lower left corner and continues up one column at a time until all of the unknown utterance is processed.

The particular dynamic programming algorithm used by the ITTDCD AWR system is one studied by Sakoe and Chiba [1]. Each path through the dynamic programming matrix is constrained so that a diagonal step has to be taken after a horizontal or vertical step. This constraint has the effect of restricting the slope of a path to be between one-half and two. Thus, the length of the spoken word must be between one-half and twice the length of its template. In addition, the constraint prevents more than two frames of the unknown word to be matched against one frame of the template and vice-versa.

The dynamic programming equation below shows how the scores of the partial paths ending at times (i-1,j), (i,j-1), and (i-1,j-1) are compared to see which path is the best one to extend to time (i,j).

$$S_{ij} = \min (S_{i-1,j} + d,\ S_{i,j-1} + d,\ S_{i-1,j-1} + 2d)$$

where

S_{ij} is a partial path score at cell (i,j)
d is the Euclidian distance for cell (i,j)

The constraints given above are used in conjunction with this equation to see which path is extended. The final path score is normalized by dividing by the sum of the lengths of the unknown and the template to yield a match score that can be compared across all templates.

2.1.2 Algorithm Speed-up and Storage Reduction Techniques

The dynamic programming recognition algorithm as presented above places a large computational burden on the AWR system. Also, the parametric frame representation of speech requires considerable data memory for template storage. Less expensive recognition algorithms exist and more compact data representations have been used, but with a corresponding degradation in performance. ITTDCD elected to use the more accurate algorithm and representation, and to depend on other techniques to reduce the computational and memory requirements. Four techniques were investigated: principal component dimensionality reduction, variable frame rate encoding, corner pruning, and template pruning. The first two techniques reduce the data rate resulting from the speech analysis and therefore reduce both the computational and storage burden on the system. The second two techniques of corner and template pruning reduce computation by limiting unneccessary matching in the dynamic programming algorithm. These techniques are described in the paragraphs below.

2.1.2.1 Principal Component Dimensionality Reduction

The n coefficients of each frame of speech from a front end analysis technique define a point in an n-dimensional feature space. The dimensionality of this feature space can be reduced while minimizing any reduction in the variation described by the speech data by the method of principal component analysis [2]. Pols first applied this analysis to speech recognition [3]. In the analysis, the variances along each dimension, as well as the covariances between the dimensions, are calculated over a large sample of speech data. The eigenvector of the covariance matrix with the largest eigenvalue defines a dimension in the original feature space which accounts for as much of the variance as possible. The eigenvector with the next largest eigenvalue defines a new orthogonal dimension which accounts for as much of the variance as possible that was not accounted for by the first eigenvector. Thus a set of m eigenvectors ($m<n$) can be found which account for a high percentage of the variance of the speech data. A point in the n-dimensional feature space is mapped to the reduced feature space by multipling the n-coefficient frame vector by the n-by-m eigenvector matrix. The resulting m-coefficient frame vector requires less data memory to store and less computation when it is

compared against other frame vectors in the dynamic programming algorithm.

2.1.2.2 Variable Frame Rate Encoding

Variable frame rate encoding achieves data reduction by reducing the number of frames in each template and unknown. This again reduces the amount of data memory required for storing templates and also reduces the size of the dynamic programming matrix. Variable frame rate encoding reduces the number of frames in areas where the speech features change slowly (e.g. in sustained sounds like vowels), but retains more frames in areas where the speech features change more rapidly (e.g. liquids and stops). This is accomplished by simply comparing (via a Euclidean distance metric) the next input frame to the last frame retained. If the new frame is quite similar to the last frame according to a set threshold, the new frame is rejected. However, if the new frame is different from the last frame, the new frame is passed on to the next step in the analysis process.

2.1.2.3 Corner Pruning

Figure 2.3 illustrates dynamic programming with corner pruning (the shaded portion). Corner pruning eliminates, with a minimum of added software, those frame-to-frame comparisons which are not part of a good time alignment path. The width of the remaining band in the dynamic programming matrix (measured by the number of horizontal frames across the band) can be adjusted to obtain the greatest decrease in computation costs while maintaining the same recognition performance. Obviously, if the bandwidth is too narrow, the best alignment path is cut off and performance suffers.

2.1.2.4 Template Pruning

In template pruning, as the unknown word is processed, the partial match scores of the templates are compared. If the partial match score of a particular template is sufficiently worse (according to a threshold) than the best partial match score over all the templates, that template is pruned. Recognition continues on the reduced set of templates. This is

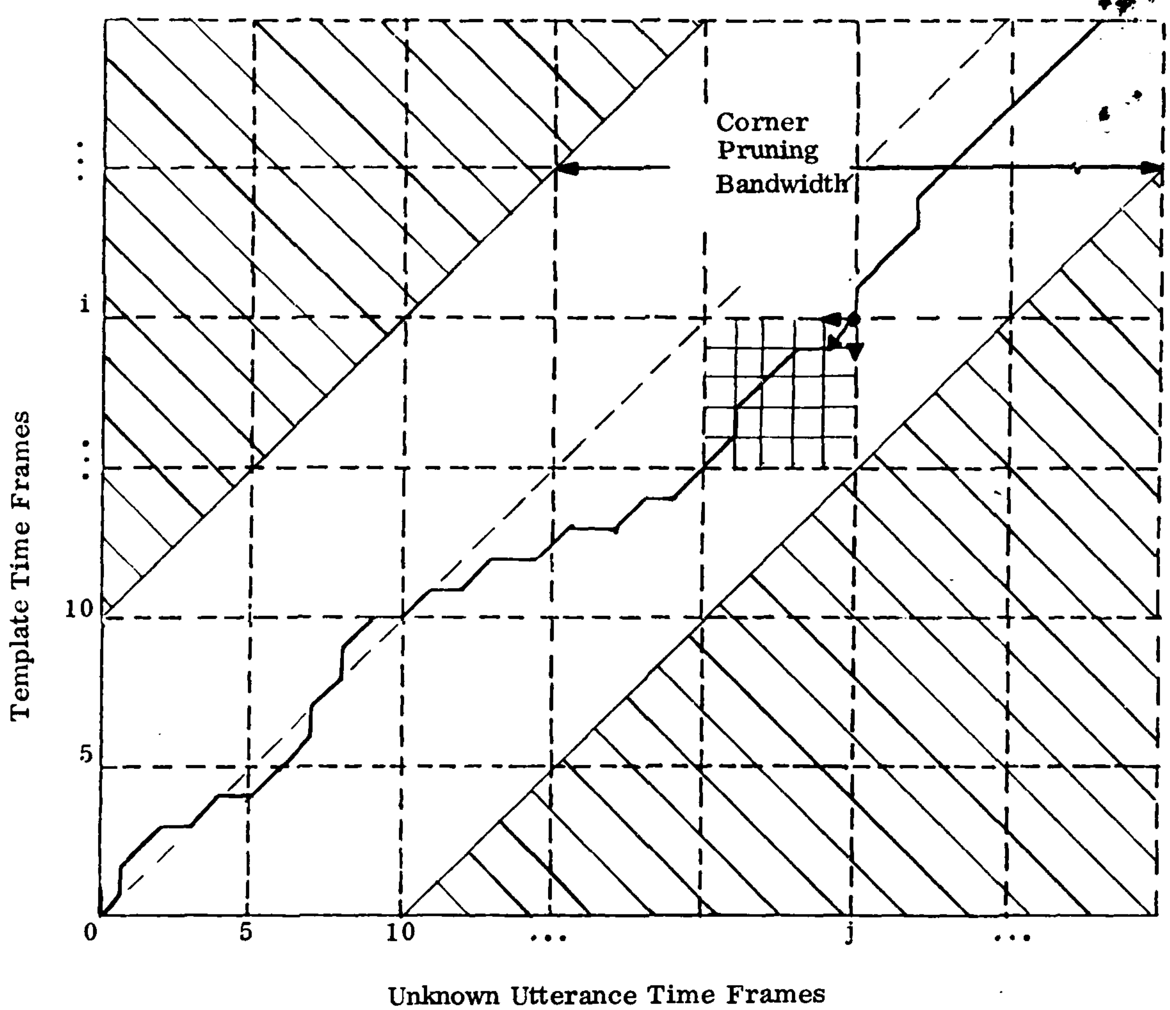


Figure 2.3 Dynamic Programming Matrix with Corner Pruning

illustrated in Figure 2.4. For each frame of the unknown utterance (time j), the score of the best partial path of a template t ending in time j is identified by finding the minimum score, r_{jt}, of the column. If for some template t, r_{jt} is greater than R_j (the minimum for all templates) by some constant threshold C, then template t is pruned. This method of template pruning is similar to one used by Itakura [4].

2.2 Speech Analysis Techniques

Four speech analysis techniques were investigated as alternative first stages in speech recognition systems. These techniques included Linear Predictive Coding (LPC), Bandpass Filtering (BPF), Discrete Fourier Transform (DFT), and Cepstral analysis. They all accept analog speech (time domain data) as input and produce signal parameters for each time frame that are used to represent a smoothed approximation of the speech spectrum. The lower plot of Figure 2.5 shows a speech spectrum for the vowel in "beet". The rapid oscillations of the plot are due to the pitch harmonics, whereas the overall shape of the plot is due to the shape of the vocal tract and is therefore characteristic of the sound being generated. To be useful for speech recognition, the spectrum must be smoothed so that the signal parameters most closely represent the sounds generated by a speaker and not the pitch of his voice.

2.2.1 Linear Predictive Coding

Linear Predictive Coding (LPC) analysis approximates the speech spectrum by an all pole model. The smoothed line shown in Figure 2.5 is an all pole approximation of the spectrum. The number of poles used in the model (generally between 8 and 14) determine how closely the model approximates the spectrum. Unique in LPC analysis is the fact that the model most closely matches the spectrum at the higher energys, i.e., at the vocal tract resonances, or formants of the spectrum. Thus LPC analysis can correctly model formants that are close together, while smoothing the spectrum between formants. This is generally not true for the other analysis techniques studied here. LPC analysis works best for the spectrum of non-nasalized voiced sounds. The spectrum of nasals contain zeros which

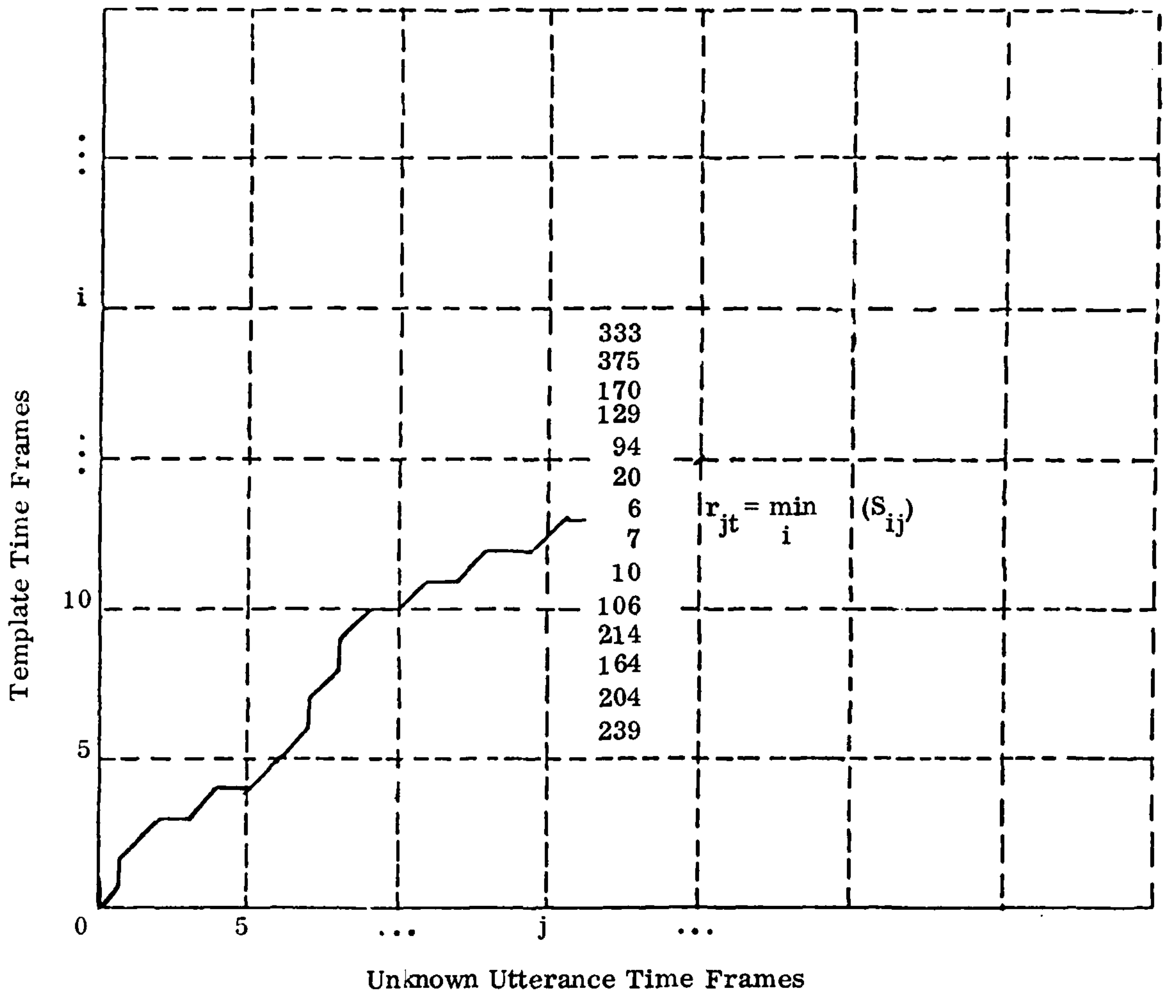


Figure 2.4 Dynamic Programming Matrix with Template Pruning

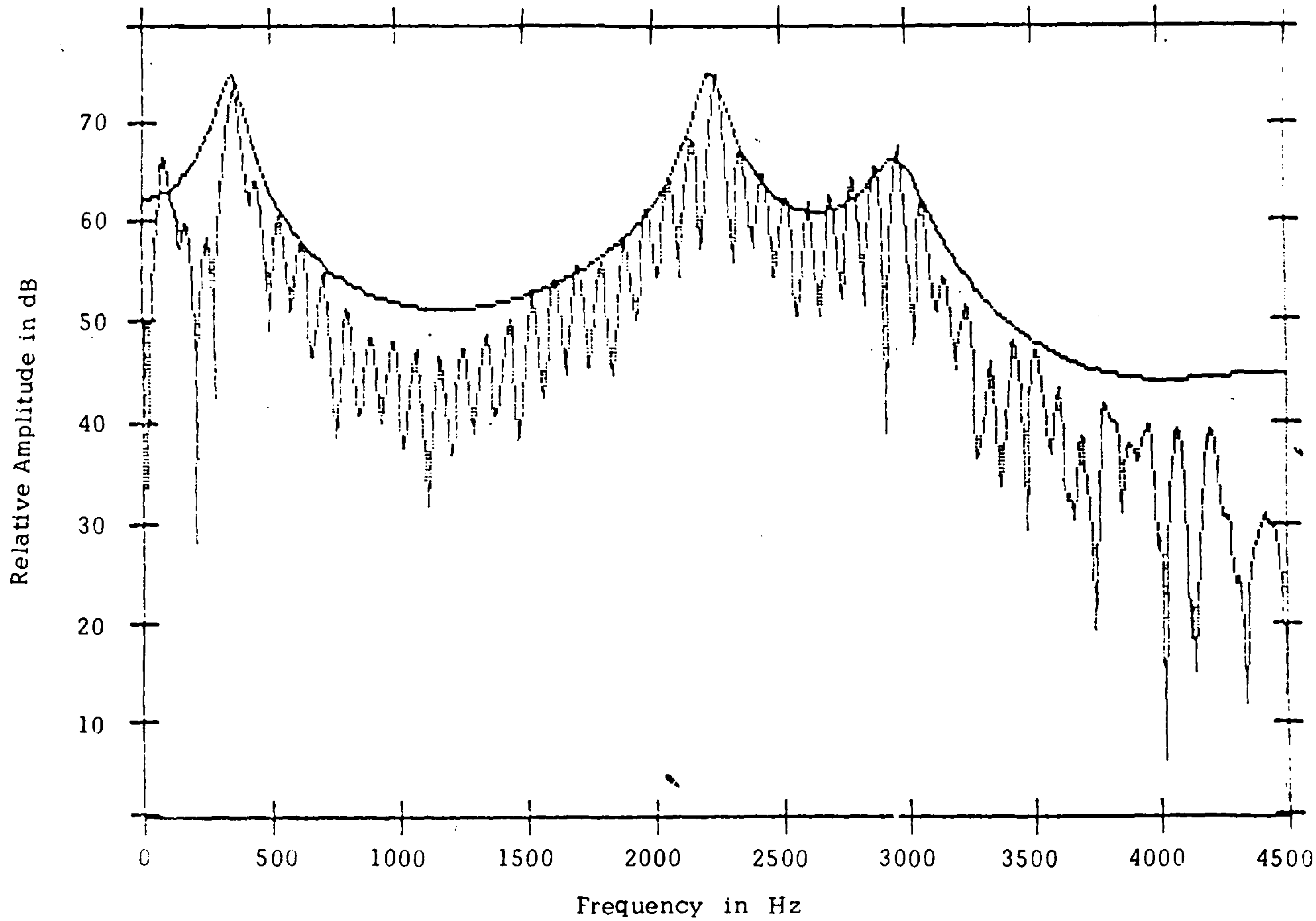


Figure 2.5 Speech Spectrum for the Vowel in "Beet"

cannot be correctly modeled by an all pole model. The LPC coefficients modeling two spectra are typically compared in speech recognition using the Itakura "log ratio of LPC residuals" [4].

2.2.2 Bandpass Filtering

Bandpass Filtering (BPF) analysis converts the speech spectrum into a power spectral density representation. The number and width of the bands determine the smoothing of the spectrum. By careful selection of the filter bands, variations due to pitch harmonics are avoided and the "critical bandwidths" of the ear are approximated. BPF coefficients from two different sounds can be compared using a Euclidean metric.

2.2.3 Discrete Fourier Transform

A Discrete Fourier Transform (DFT) analysis for speech generally has between 128 and 512 speech samples per transform. DFT coefficients, however, are seldom used directly. Rather, the coefficients are used to produce a power spectral density representation like that obtained by bandpass filtering. Each "bandpass filter" is obtained by summing a set of frequency adjacent DFT coefficients. Again a Euclidean metric is used to compare the representations of different sounds.

2.2.4 Cepstral Analysis

The cepstrum is defined as the Fourier transform of the logarithm power spectrum. In order to understand what the cepstral coefficients represent, consider again the spectrum of Figure 2.5. If this spectrum is treated as a time domain signal and processed by a Fourier transform, the resulting low "frequency" components (called quefrency components) will be related to the overall shape of the power spectrum, and the high "frequency" components will be related to the pitch harmonics of the power spectrum. These quefrency components of the transform of the power spectrum are called cepstral coefficients. In speech recognition, only the low quefrency cepstral coefficients are used in order to "smooth" out the pitch harmonics of the spectrum.

Cepstral coefficients have been used by several investigators [5,6] and have been reported to be equal to the best, if not superior to, the other techniques for encoding speech for recognition purposes. Cepstral coefficients are usually used with a Euclidean distance function to yield the "sound similarity" between two speech sounds. It was shown by Gray and Markel [7] that this is equivalent to measuring the Euclidean distance in the log RMS spectral power domain. Using only the low quefrency cepstral coefficients in the distance function is equivalent to measuring the Euclidean distance in a smoothed log RMS spectral power domain. In other words, the cepstral analysis in its preferred form differs only in minor ways from the Fourier transform and bandpass filter approaches.

In developing these speech analysis techniques for CCD implementation there are two questions of concern: How accurately do each of these signal parameterizations represent speech at a given data rate, and how effective is the CCD approach to computing these parameters?

The answer to the first question, how "good" are the four representations, depends on how they are implemented in a speech recognition system. There is no universally agreed upon "proper way" to use these representations. We have implemented them using generally accepted principles, which we think are the best for the given constraints. Our conclusion is that three of the four techniques (BPF, DFT, and Cepstral) produce speech parameters that are essentially equivalent, or can be made equivalent, in theory. In practice, differences will arise because of the differing effects of time windowing, dynamic range, and number of equivalent bits of accuracy in the internal operations of the devices.

The second question, how effective is the CCD approach in each case, is the topic of the next chapter.

2.3 Experimental Test Paradigm

The word recognition tests used for testing algorithms and for comparing CCD analyzer processing techniques are speaker dependent, with each speaker using simple one word templates. For each test vocabulary, a test speaker recorded the vocabulary five times. Each vocabulary repetition for that speaker was used in turn as a set of templates and compared against the other four repetitions by the same speaker as test utterances.

It has been well documented that template clustering and multiple template techniques improve recognition performance [8]. Therefore the results reported in Chapter 6 should not be taken to indicate the best possible performance for the techniques studied, but rather only their relative performance.

2.4 Developmental Simulation Studies

Two activities were undertaken to support the work that is reported in the following chapters. First, the four algorithm speed-up techniques described above were developed and studied on an existing AWR system implemented on a PDP-11/60. For speech analysis, this system used an LPC-10 analysis program to obtain 10 LPC reflection coefficients every 22.5 ms.

During the algorithm speed-up development, the following effects were observed. Because of the large frame size in this system, Variable Frame Rate Encoding was not very effective. Reducing an already slow frame rate degrades performance. However, the technique could be helpful for a CCD analyzer with a higher frame rate.

Similarly, principal component dimensionality reduction was not very effective for reflection coefficients. However, we expect that it might be more effective for something like bandpass filter analysis, where the coefficients are more highly correlated.

Corner pruning and template pruning showed great promise in the developmental studies. Approximately 60% of the dynamic programming matrix could be ignored by corner pruning without significantly degrading the performance. On the average, template pruning provided a 30% reduction in the number of templates processed without significantly degrading performance. The actual effect of these techniques in the realtime AWR system is discussed later.

The second activity of the developmental simulation studies was to generate and test simulated CCD analyzer outputs. CCD simulations for the four speech analysis techniques investigated in this contract were generated under a subcontract with Dr. Bob Brodersen at the University c California, Berkeley. An anlog tape of four speakers repeating a 26 phonetic word vocabulary five times was processed by the simulation software at Berkeley. The resulting digital tapes were delivered to ITTDCD for experimentation using the AWR system implemented on the PDP-11/60.

Only a small part of this simulated data (one speaker over all techniques) was studied for the following reasons. First and foremost was the fact that the actual hardware devices themselves were being constructed under our IR&D program. Results with actual hardware would be more meaningful then that from a simulation study. However, the CCD simulation data did enable us to become familar with the type of speech parameters that the hardware would generate before it became available. Also, the simulation data served as a backup in case the hardware development failed. A second reason for limited testing of the simulation data was that at the time of the testing the realtime AWR system had not yet been implemented. The PDP-11/60 AWR system required considerable time to process the data. Finally, problems were found with word endpoint detection in using the simulated data. Although these problems could have eventually been overcome, the required effort was not justified, once we decided to focus our investigation on the hardware devices themselves.

Chapter 3: DEVELOPMENT OF CCD SPEECH ANALYZERS

To provide more realistic data for performance assessment, three of the speech analysis techniques (BPF, DFT, and Cepstral) were implemented using actual CCD hardware. In addition, for each of the devices, special purpose software was developed to make the respective speech parameters suitable for a realtime recognition system. This chapter describes the detailed design of each CCD analyzer and its associated software.

It is not currently possible to implement the fourth speech processing technique, LPC analysis, in CCD hardware. The first step of the analysis, namely the extraction of autocorrelation coefficients, was attempted in CCD hardware by Dr. Brodersen at the University of California, Berkeley. The autocorrelation coefficients generated by this device proved unsatisfactory. However, recognition experiments were performed with software generated autocorrelation and LPC coefficients. Results of these experiments are presented in Section 6.2 for comparison with the other three speech analysis techniques implemented in CCD hardware.

3.1 CCD Analysis Hardware

All three of the speech analysis techniques implemented in CCD hardware make use of Reticon CCD devices. Two of the techniques (DFT and Cepstral) share a Reticon spectral analyzer board and are contained in one hardware unit, while the BPF CCD analyzer is a separate unit.

3.1.1 Bandpass Filter Hardware

The Bandpass Filter (BPF) design uses nineteen switched-capacitor bandpass filters to cover a frequency range of 100 Hz to 9500 Hz. Six R5604 integrated circuits are used for eighteen 1/3 octave filters, and one R5606 integrated circuit is used for a full octave filter covering the higher frequencies. Figure 3.1 shows a block diagram of the BPF CCD analyzer.

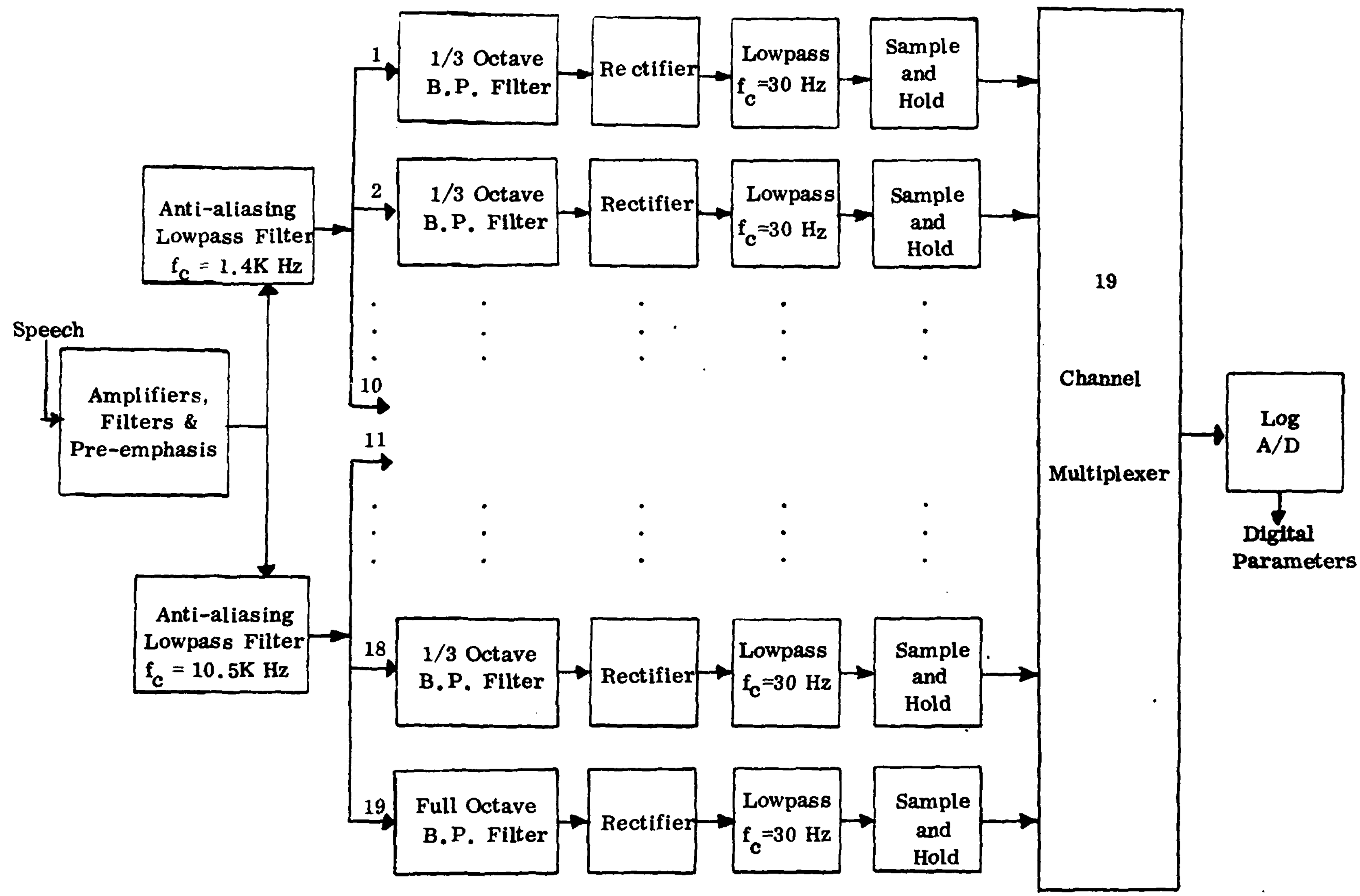


Figure 3.1 CCD Bandpass Filter Analyzer

The speech input is amplified, anti-aliased by a lowpass filter, and then pre-emphasized by a 6 dB/octave slope beginning at 500 Hz. The signal is then filtered by two switched-capacitor lowpass filters with cutoffs at 1.4 kHz for the lower frequency bandpass filters and 10.5 kHz for the higher frequency bandpass filters. These lowpass filters act as anti-aliasing filters to the bandpass filters. The output of each bandpass filter is processed to obtain an approximation to the smoothed RMS value by half wave rectification followed by a 30 Hz lowpass filter. The 19 analog signals are simultaneously sampled and held once each 10 ms. Then they are multiplexed into a single logarithmic A/D converter to produce 19 eight-bit values every 10 ms. The frequency characteristics of each filter are presented in Table 3.1.

3.1.2 Discrete Fourier Transform and Cepstral Hardware

The Discrete Fourier Transform (DFT) analyzer and the Cepstral analyzer share a Reticon RC5601 spectral analyzer board which is based on the R5601 chirp Z transformer (CZT) . The R5601 is an MOS intergrated circuit which performs the bulk of the computation required to calculate a 512-point DFT. The circuit contains two separate 512-point CCD's which are used to implement four transversal filters using the split-electrode weighting technique.

The block diagram of Figure 3.2 illustrates the DFT and Cepstral analyzers. The speech input is again preprocessed by amplification, anti-aliasing filtering, and pre-emphasis. The speech is then processed by the CCD spectrum analyzer, which is driven by a 20 kHz clock to obtain 512 Fourier magnitude coefficients every 25.6 ms. Switch 1 is then placed in either the log or linear position, depending on whether cepstral (log position) or Fourier (linear position) processing is desired. The logarithm is calculated in an Intersil 8048 logarithmic amplifier. This bipolar device uses the exponential characteristics of a diode to yield a logarithmic transfer characteristic. Unfortunately the device is sensitive to offsets as well as having a rather high noise level. To alleviate some of the offset problems, DC offset controls were added to the input and output of this device. In the bypass path (linear position of S), there

Table 3.1 Characteristics of the 19 Channel CCD Filter Bank

Filter	Low Frequency Cutoff (-3 dB)	High Frequency Cutoff (-3 dB)	Center Frequency	Bandwidth
1	100	126	111	26
2	126	156	141	30
3	156	200	178	44
4	200	252	223	52
5	252	308	280	56
6	308	400	354	92
7	400	504	447	104
8	504	618	561	114
9	618	800	709	182
10	800	1008	894	208
11	1008	1228	1118	220
12	1228	1600	1414	372
13	1600	2016	1788	416
14	2016	2456	2236	440
15	2456	3200	2806	744
16	3200	4032	3576	832
17	4032	4912	4472	880
18	4912	6400	5612	1488
19	5000	9500	7250	4500

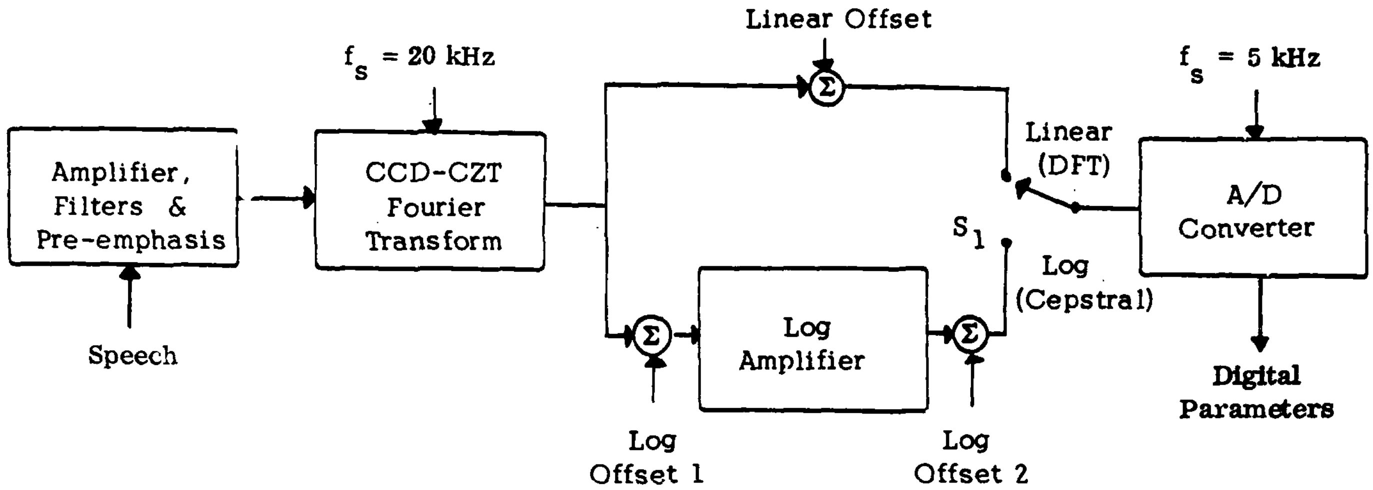


Figure 3.2 CCD Discrete Fourier Transform and Cepstral Analyzers

is also an offset control to obtain a DC output voltage compatible with the log amplifier path.

The original design of the CCD Cepstral analyzer required another CCD-CZT Fourier transform stage after log amplification to obtain cepstral coefficients. It was found, however, that the signal coming from the log amplifier could not survive another analog CCD-CZT stage and still be useful for speech recognition. The solution was to complete the cepstral analysis in software after the signal was digitized. An A/D converter accomplishes the digitization for both the linear and log spectrum signals. It is driven by a 5 kHz clock to obtain 128 spectral values per frame at the digital output.

3.2 CCD Analysis Software

The outputs of each of the CCD analyzers require additional processing to yield speech parameters that are more suitable for a low cost, realtime AWR system. This processing provides more effective recognition parameters and reduces their data rate so that memory and processing requirements can be minimized for an AWR system. In the case of the Cepstral technique, this processing is also needed to complete the analysis, since it is not possible to perform the final Fourier transform in CCD hardware.

3.2.1 Bandpass Filter Software

Figure 3.3 shows a block diagram of the software added to handle the output of the BPF CCD analyzer. The first step reduces the 19 filter channel outputs to 16 coefficients by summing the first three channels to obtain the first coefficient and summing the next two channels for the second coefficient. This procedure follows work done by Pols [3]. The combination of these low frequency filters reduces the sensitivity of the speech parameters to variations in the fundamental frequency of voiced speech, by ensuring that there are always at least two harmonics of the fundamental within each coefficient. An amplitude measure is also computed at this point by summing all filter channel outputs.

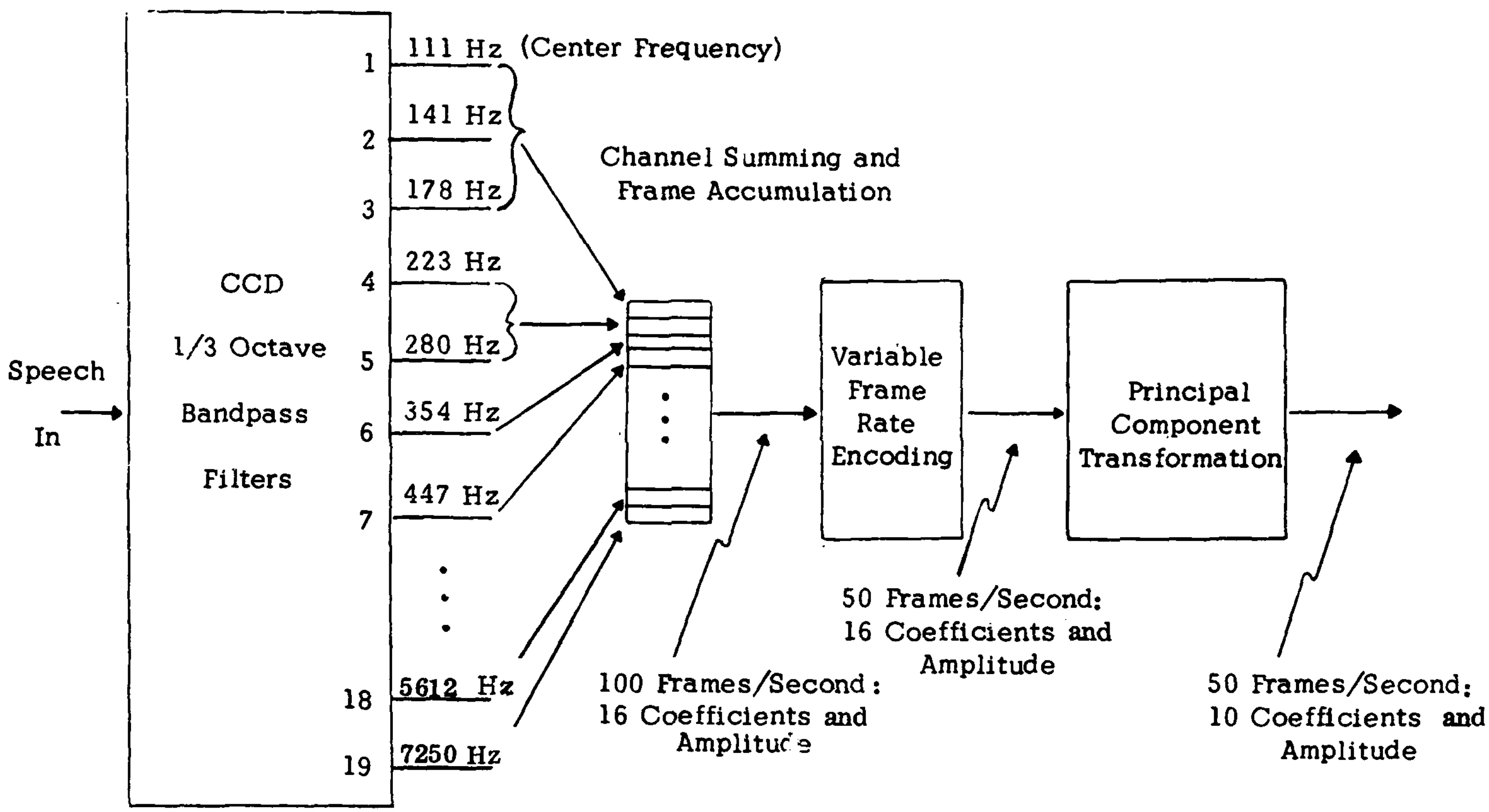


Figure 3.3 Software Processing for the CCD Bandpass Filter Analyzer

A new set of filter values representing a frame of data are produced by the BPF analyzer every 10 ms. This data rate is then reduced by the process of variable frame rate encoding discussed in the last chapter. The distance threshold in this process was set to retain only about half of the input frames.

Principal component dimensionality reduction is then applied to reduce the 16 coefficients to 10 principal components. Multiplying the 16 coefficient frame vector by a matrix of 10 eigenvectors produces a new vector of 10 coefficients which account for about 98% of the variance. The three steps of channel summing, variable frame rate encoding, and principal component dimensionality reduction, lower the original 15.2 kilobits/second data rate to 4 kilobits/second for the BPF technique.

3.2.2 Discrete Fourier Transform Software

The data rate of the DFT CCD analyzer is higher than that of the BPF analyzer. Every 26.5 ms, 128 eight-bit spectral values are produced by the analyzer (38.6 kilobits/second). Figure 3.4 shows how this data rate is reduced to 3 kilobits/second in two steps. The first step is again channel summing. However, in this case, overlapping groups of channels are summed in a manner that approximates a digital filter bank suggested by Klatt for spectral matching of speech sounds [9]. Table 3.2 shows characteristics of the resulting 32 channel filter bank. As with the BPF data, a principal component transformation then reduces these 32 coefficients to 10 coefficients which account for about 93% of the variance in the data.

3.2.3 Cepstral Software

The cepstrum is defined as the Fourier transform of the logarithm power spectrum. Typically the cepstrum is computed in three stages: a DFT of the signal produces an n-point power spectrum, then a log operation produces an n-point logarithm power spectrum, and finally another DFT produces an n-point cepstrum. As explained in Section 2.2.4, only the low quefrency cepstral components of the cepstrum are normally used for speech recognition, and we intended to use only the first 10 of these

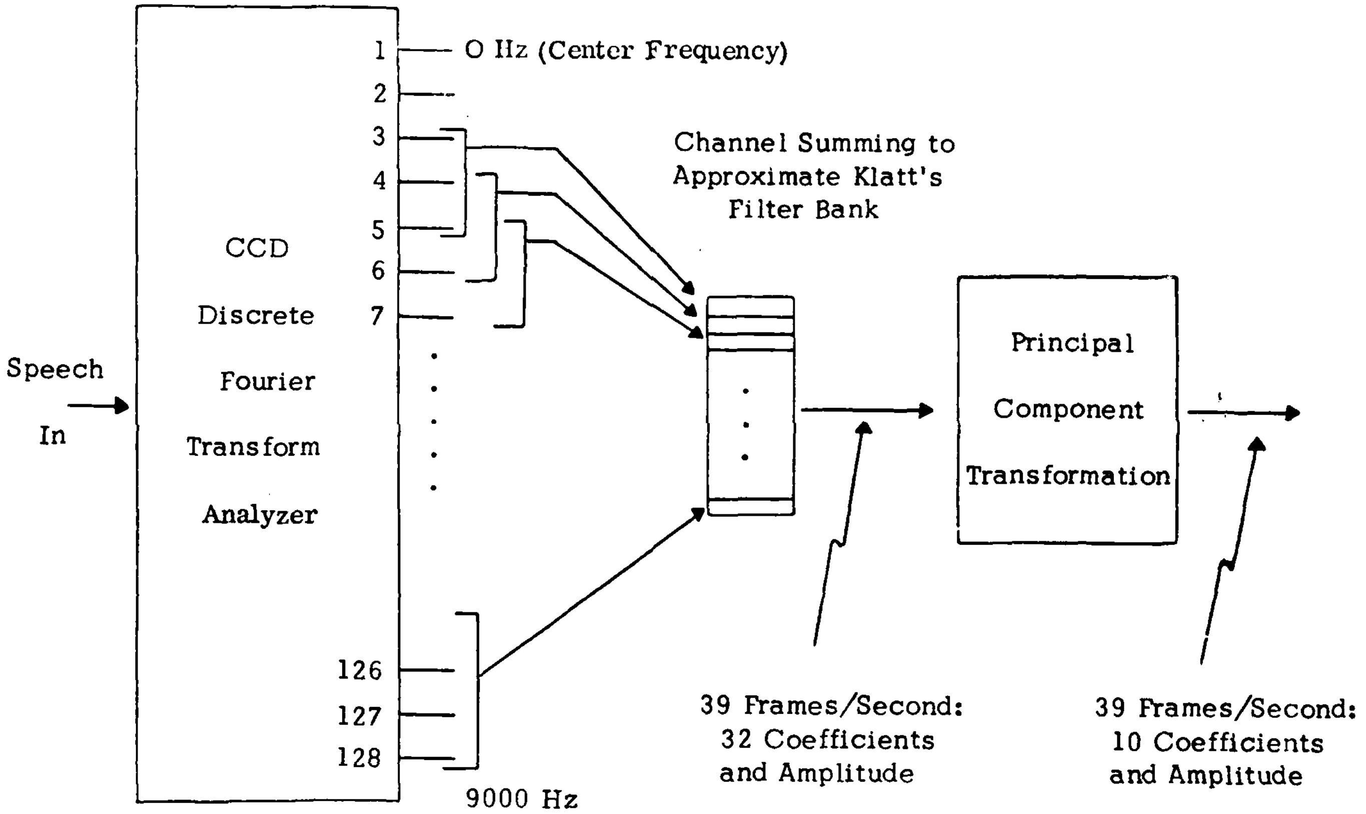


Figure 3.4 Software Processing for the CCD Discrete Fourier Transform Analyzer

Table 3.2 Characteristics of the 32 Channel Filter Bank Created from CCD DFT Samples

Filter	Center Frequency	Bandwidth	DFT Samples
1	281	241	3 - 5
2	352	241	4 - 6
3	422	241	5 - 7
4	492	241	6 - 8
5	563	241	7 - 9
6	633	241	8 - 10
7	703	241	9 - 11
8	773	241	10 - 12
9	844	241	11 - 13
10	914	241	12 - 14
11	984	241	13 - 15
12	1055	241	14 - 16
13	1195	241	16 - 18
14	1336	241	18 - 20
15	1477	241	20 - 22
16	1617	241	22 - 24
17	1758	241	24 - 26
18	2074	452	27 - 32
19	2215	452	30 - 35
20	2496	452	33 - 38
21	2707	452	36 - 41
22	2918	452	40 - 45
23	3270	452	44 - 49
24	3761	452	48 - 59
25	4113	452	53 - 64
26	4465	873	58 - 69
27	4887	873	64 - 75
28	5379	873	71 - 82
29	5941	873	79 - 90
30	6574	873	88 - 99
31	7277	873	98 - 109
32	8332	1366	109 - 128

coefficients. Therefore the most efficient software replacement for the final DFT stage of the cepstral analysis was not to perform a complete Fast Fourier Transform (FFT), but rather to compute a partial Fourier transform by convolving the log power spectrum with a set of 10 cosine transforms [5]. Figure 3.5 summarizes these steps, and illustrates that the cepstral analysis is realized by adding log amplification to the DFT signal within the DFT analyzer, and then computing in software the dot products between the 128 log DFT outputs and a set of 10 cosine tables.

While this type of cepstral representation has been sucessfully used for recognizing words, Davis and Mermelstein have shown [5] that better speech recognition is possible if the cepstral coefficients are computed from a power spectrum based on a mel frequency scale. The mel frequency scale represents the spectrum linearly between zero and 1000 Hz, but logarithmically from 1000 Hz to the highest frequency covered (9000 Hz in this case). The mel frequency scale is derived from perceptual data on the frequency response of the human ear, and thus mel frequency spectral coefficients should better represent perceptually relevant aspects of the short-time speech spectrum than should linear frequency coefficients. In order to obtain the best possible performance with the cepstral CCD analyzer, ITTDCD devised a method of obtaining "mel cepstral" coefficients without altering the frequency scale of the CCD analyzer technique. Figure 3.6 illustrates how this is done. The cosine curves which are convolved with the log power spectrum are stretched above 1125 Hz so that they cover that part of the spectrum in a log manner. In addition (but not shown in the figure), the individual cosine values along the stretched portion of the curve are normalized by dividing by a value proportional to the amount of stretching in that part of the curve. Thus equal parts of a cosine curve contribute equally to the final coefficient value, independent of the amount of stretching that each part undergoes.

Each CCD analyzer and its associated software were developed to provide speech parameters suitable for realtime word recognition and accordingly, each comprises a component in the realtime laboratory AWR system which is described in the next chapter.

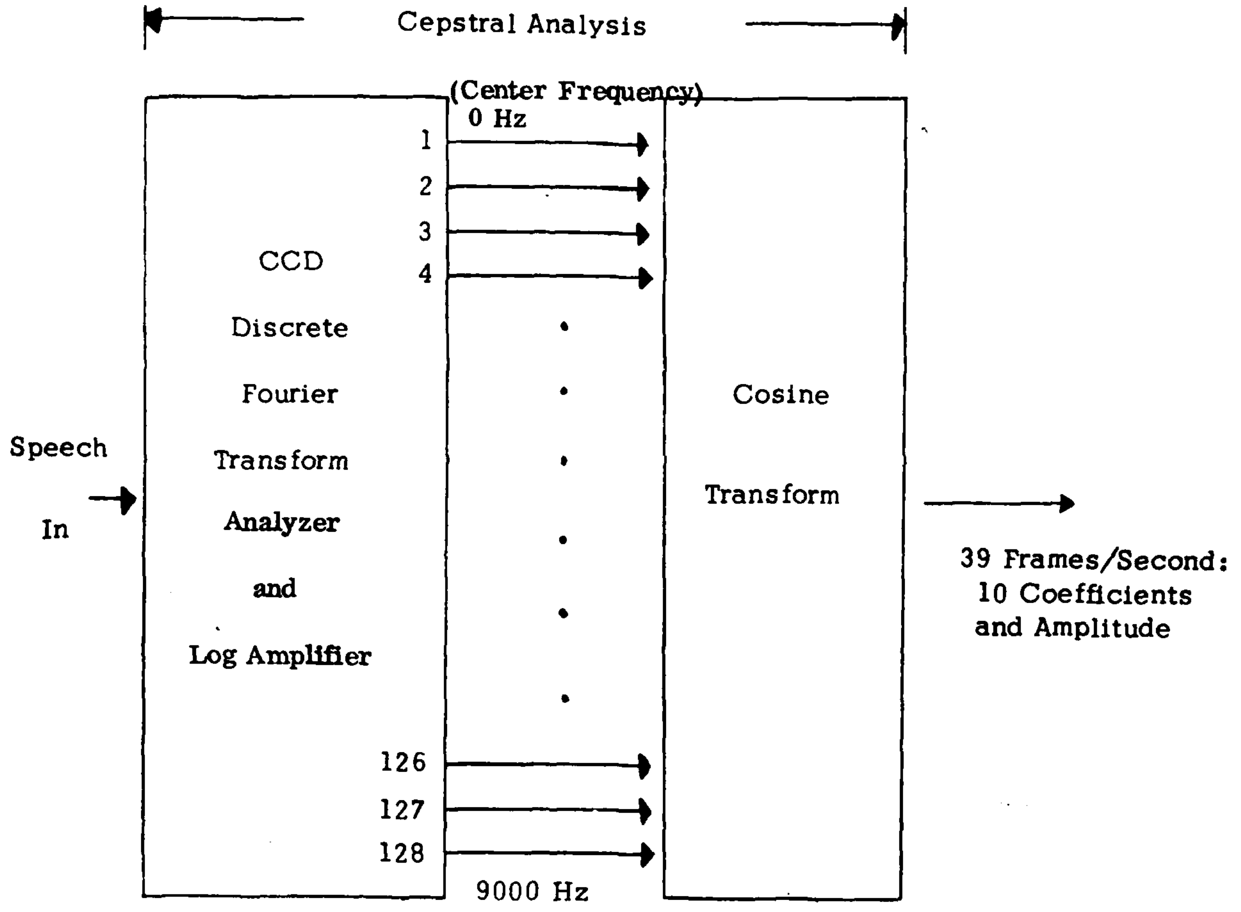


Figure 3.5 Software Processing for the CCD Cepstral Analyzer

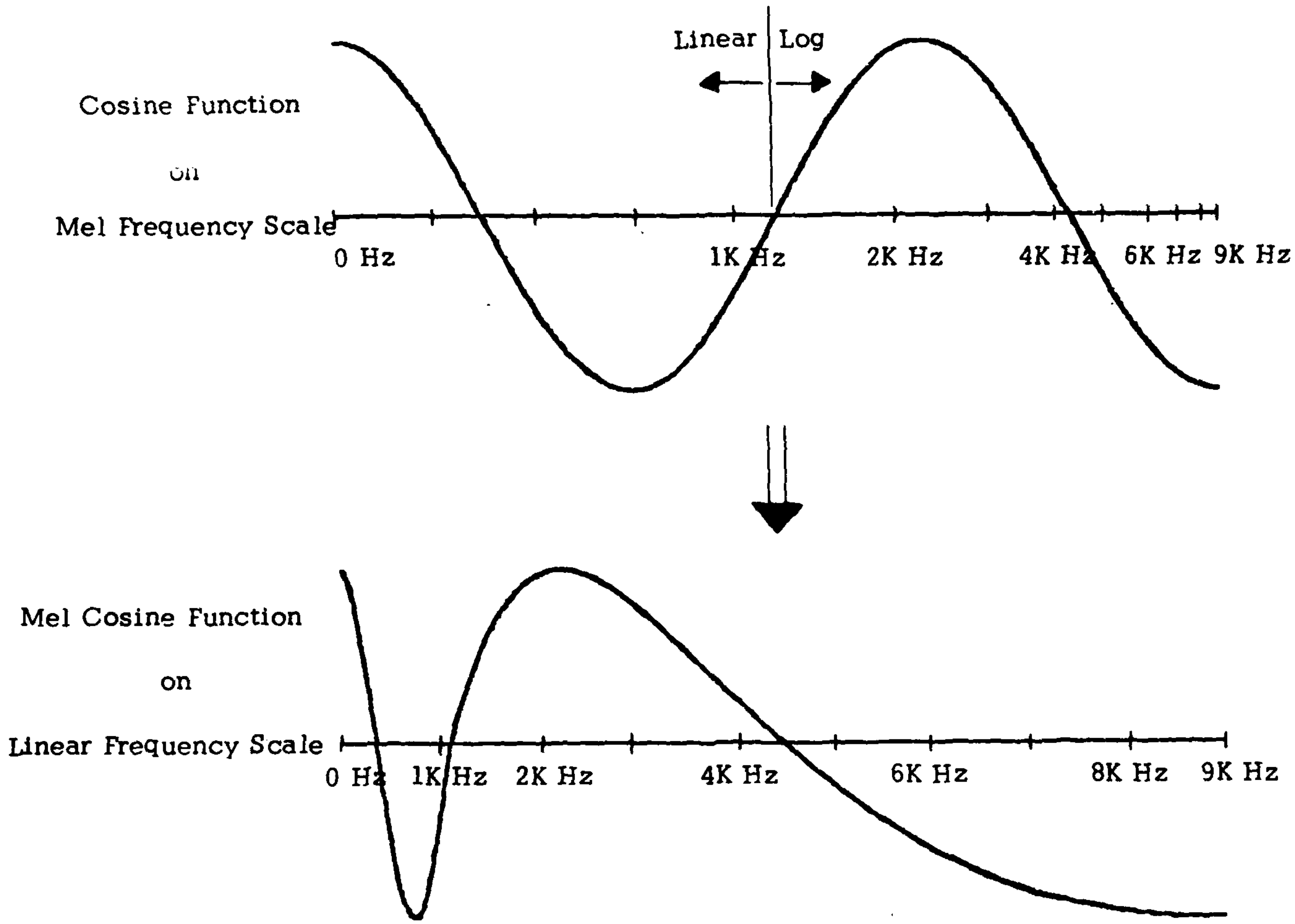


Figure 3.6 Transformation of a Cosine Function to a Mel Cosine Function

Chapter 4: A REALTIME LABORATORY WORD RECOGNITION SYSTEM

As the second task of this study, ITTDCD implemented a realtime laboratory AWR system. The major hardware components include the CCD speech analyzers, a Quintrell processor, a PDP-11/60 minicomputer, and two display terminals. Realtime word recognition software was developed for the Quintrell processor, a high speed signal processor originally designed by ITTDCD for narrowband speech transmission systems.

The system operates in three modes: a realtime recognition mode, a template generation mode, and an experimental mode. This chapter provides a description of the various components, configurations, displays, and operating modes of our versatile laboratory AWR system.

4.1 System Configuration

An overview of the various components of the AWR system is presented in Figure 4.1. The system can be configured with any one of the three CCD analyzers described in the preceding chapter. The software associated with each CCD analyzer resides in the Quintrell processor, as does the recognition software. Speech is input to the system via microphone or tape recorder. The speech is parameterized by the CCD analyzer and passed on to the Quintrell for further processing.

The system is controlled by the user from a PDP-11/60 display terminal. The PDP and Quintrell communicate via the DR11-B, a high speed direct memory access interface. PDP software controls this interface, which is primarily used for transmittal of speech parameters in the form of templates. PDP software also computes principal component (eigenvector) matrices and, in conjunction with the UNIX operating system, provides for the storage and retrieval of speech templates on disk.

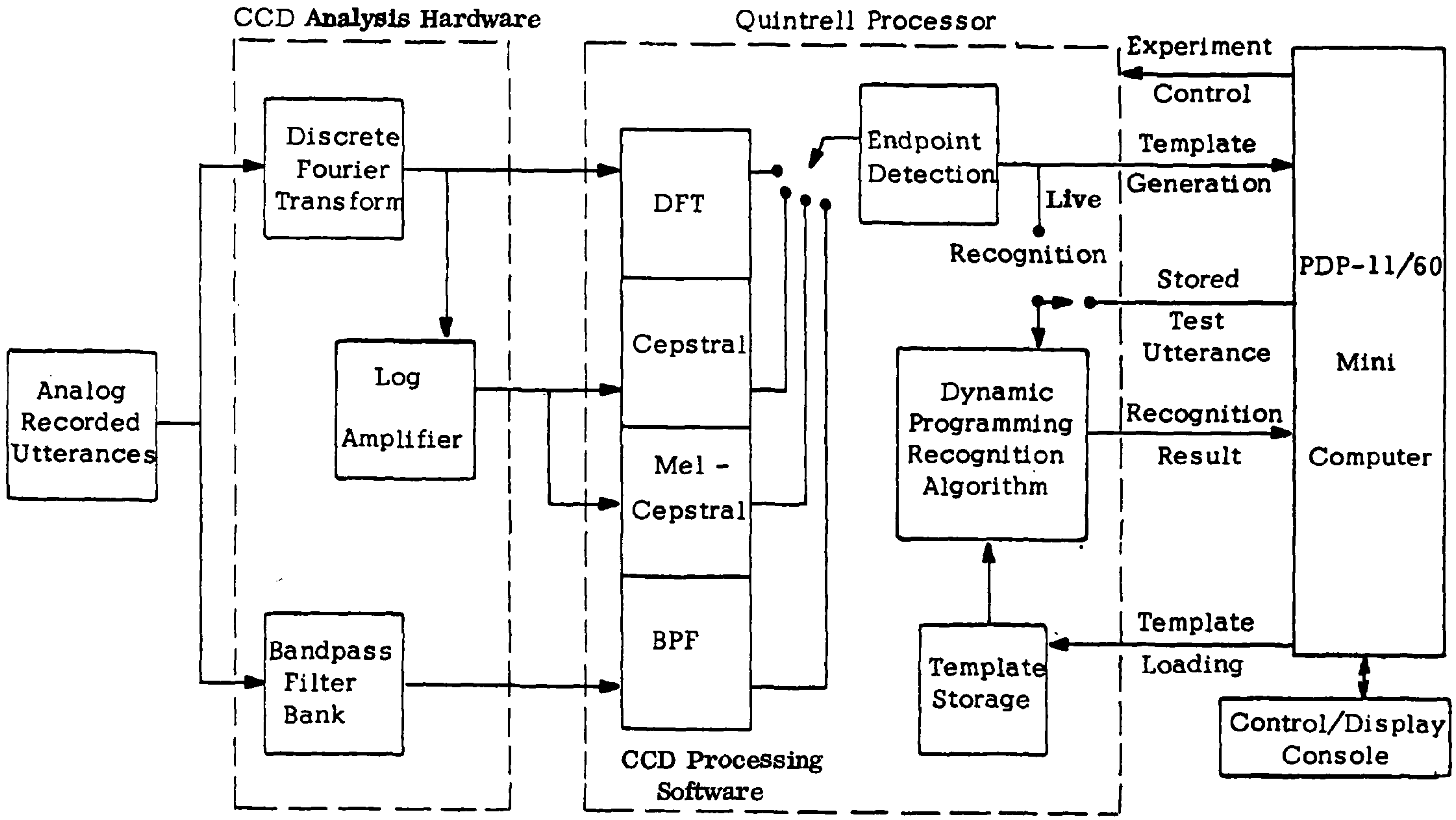


Figure 4.1 The ITTDCD Realtime Laboratory Automatic Word Recognition System

4.2 Recognition Software

The core of the recognition software operates in the Quintrell processor and is independent of a particular CCD analyzer. This software is a realtime implementation of the dynamic programming algorithm described in Section 2.1.1 and is depicted by functional flow in Figure 4.2. The recognition process involves a comparison of parameters of the unknown word to the parameters of a set of reference templates, each representing a specific word. These comparisons are performed on a frame by frame basis and account for the bulk of the computational load on the AWR system. The role of the recognition software in the overall system is elaborated below in a sequential description of the recognition mode.

At initialization of the recognition mode, a set of templates is sent (on command) from the PDP computer to the Quintrell processor where they are stored. If the system is configured with the BPF or DFT analyzer, the eigenvectors associated with the template set are also transferred to the Quintrell. The speaker then executes a "recognize" command from the PDP terminal and says a word into the microphone. The speech signal is parameterized by the selected CCD analyzer and sent to the Quintrell, where the analyzer dependent processing described in Section 3.2 takes place. The resulting speech parameters and energy function are then processed by the recognition software on a frame by frame basis. The recognition software monitors the energy function for detection of the beginning and end of the utterance, and executes the dynamic programming algorithm. When the end of the utterance is detected, the identity of the best matching template is communicated to the PDP and the recognized word is displayed on the PDP terminal.

4.3 Operator Displays

During the recognition mode described above, a graphic display is continuously updated by the Quintrell processor. An example of this display is shown in Figure 4.3. Upon execution of a "recognize" command, the energy profile of incoming speech moves from right to left across the screen. As the beginning and end of the utterance are detected, dots mark

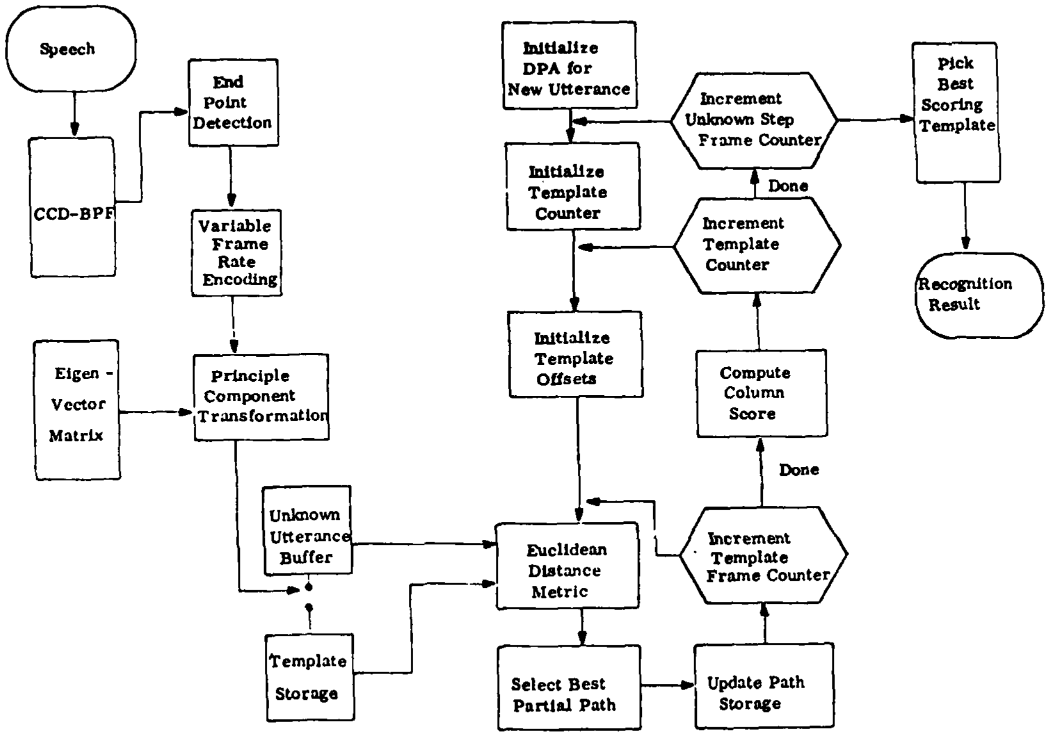


Figure 4.2 Major Steps in the Automatic Word Recognition System

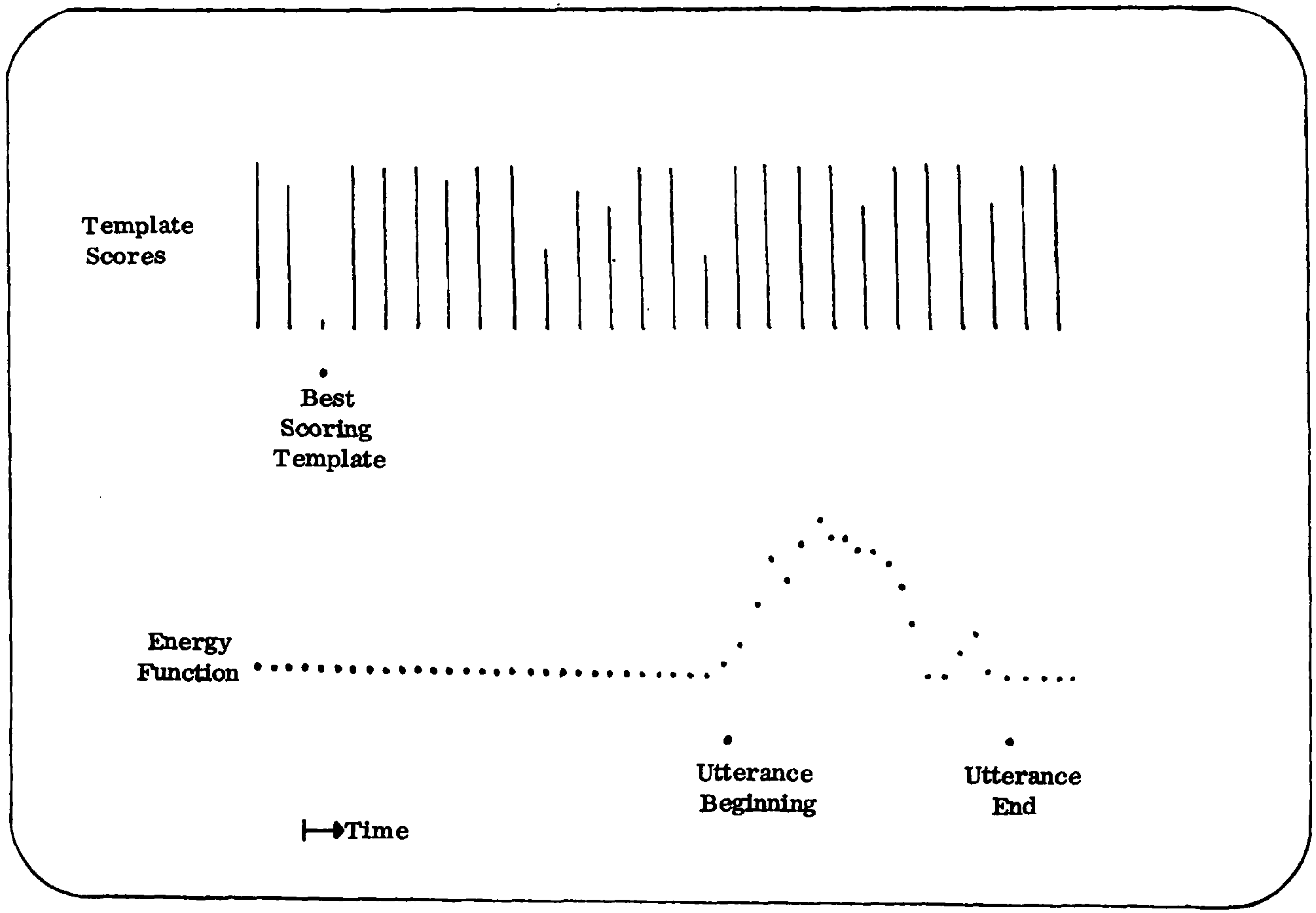


Figure 4.3 Realtime Graphic Display of Recognition Results

their locations on the display. The energy display is frozen when the end of the utterance is detected.

The upper half of the display shows a series of lines of varying lengths, one line for each resident template or vocabulary word. The height of the line represents the relative match score of the template. This part of the display is also updated throughout the recognition sequence and is also frozen at the end of the word. The shortest line represents the best matching template and is marked with a dot.

The second operator display is controlled by the PDP-11/60. At this terminal, the "recognize" command is executed and the recognized word is displayed. Moreover, it is this display which is used to configure and control the modes of the laboratory AWR system. From this terminal, for example, the user controls the settings of system variables and thresholds. By a single command, the user may activate corner pruning and template pruning.

4.4 Template Generation Software

Template generation or training is the process by which the AWR system vocabulary is generated for a specific speaker. Isolated words are input to the AWR system via a tape recorder or microphone. In this mode, much of the AWR system operates in the same manner as in the recognition mode. There are a few major exceptions. No principal component transformations are performed in the Quintrell for the BPF and DFT analyzers, and the dynamic programming algorithm does not operate. When the end of an utterance is detected, the speech parameters and energy function for the entire utterance are sent to the PDP-11/60 where they are stored on disk.

After a set of DFT or BPF templates have been produced by a given speaker, PDP software generates an eigenvector matrix from the source templates. This matrix is then employed to perform a linear transformation on each template, thus producing a new set of principal component templates, one for each of the original templates. The principal component templates are later used as reference templates in the Quintrell during the recognition mode.

For cepstral or mel cepstral analysis, no principal component transformation is required. In this case, the same source templates created in the template generation mode are used in the recognition mode.

4.5 Experimental Control Procedures

In the course of this contract, a series of word recognition experiments were conducted for each of the CCD analyzers. The laboratory AWR system was used in a slightly different fashion in the performance of these experiments. This use of the system is described below.

Vocabulary words were recorded on analog tape and processed separately by each CCD analyzer to produce recognition parameters for subsequent recognition experiments. This process was the same as the foregoing description of template generation. After templates had been generated for the entire vocabulary, however, the CCD analyzers and their associated software were no longer needed and were therefore inoperative during the recognition experiments.

In a typical recognition experiment, one repetition of the test vocabulary was sent from the PDP-11/60 to the Quintrell to serve as templates. For the BPF and DFT analysis techniques, the principal component transformation associated with that repetition was also sent to the Quintrell. Then the source template parameters for a word from another repetition of the test vocabulary were sent to the Quintrell to serve as the unknown utterance. The recognition software identified the unknown utterance and passed the identification back to the PDP-11/60, along with the best match scores for each of the resident templates. In like manner, all the words in each vocabulary repetition (excepting the repetition being used as templates) were sent to the Quintrell. The PDP-11/60 generated an experimental file containing recognition statistics. These statistics included all template scores for each utterance as well as a tally of the dynamic programming distance computations required during the course of the experiment.

Chapter 5: SPEECH RECOGNITION DATA BASE

During the course of this contract, numerous word recognition experiments were performed on ITTDCD's laboratory AWR system. In this chapter, the data base used for these experiments is defined, and pertinent template generation procedures are discussed. The results of the word recognition experiments are detailed in Chapter 6.

5.1 Data Base Content

The data base used in the recognition experiments is based on the two vocabularies shown in Table 5.1. The first is a 26 word phonetic vocabulary that has a distinctive word for each letter of the alphabet. The second is a 20 word cockpit vocabulary, consisting of the ten digits and ten control words that might be useful in an aircraft voice input application.

Table 5.2 describes the three data sets that were constructed from recordings of these two vocabularies. Data Set 1 consists of five repetitions of the 26 word phonetic vocabulary by each of four speakers, including two males and two females. Four new speakers (three males and one female) recorded five repetitions of the 20 word cockpit vocabulary to create Data Set 2. These same speakers also recorded five repetitions of the 26 word phonetic vocabulary, producing Data Set 3.

As described in Section 2.3, all experiments with the AWR system were performed in a speaker dependent manner by running all word recognition trials for a particular speaker against a set of word templates generated by the same speaker. Each vocabulary repetition was in turn used as a set of templates and compared against the other four repetitions as test utterances. This experimental paradigm yielded a total of 520 (5 * 4 * 26) trials per speaker for Data Sets 1 and 3 and 400 (5 * 4 *20) trials per speaker for Data Set 2.

Table 5.1 Vocabularies Used for Automatic Word Recognition Experiments

26 Word Phonetic Vocabulary

Adam	Henry	Otto	Victor
Baker	Ida	Peter	William
Charlie	John	Queen	X-ray
David	King	Robert	Young
Edward	Lewis	Susan	Zebra
Frank	Mary	Thomas	
George	Nancy	Union	

20 Word Cockpit Vocabulary

Zero	Five	Altitude	Clouds
One	Six	Heading	Descend
Two	Seven	Speed	Course
Three	Eight	Vertical	Frequency
Four	Niner	Horizontal	Kilometers

Table 5.2 Data Set Characteristics for Automatic Word Recognition Experiments

Data Set	Vocabulary	Speakers	Repetitions	Tests
1	26 Word Phonetic	MA (M)	5	520
		MB (M)	5	520
		FA (F)	5	520
		FB (F)	5	520
2	20 Word Cockpit	BB (M)	5	400
		BL (M)	5	400
		RS (M)	5	400
		WB (F)	5	400
3	26 Word Phonetic	BB (M)	5	520
		BL (M)	5	520
		RS (M)	5	520
		WB (F)	5	520

The experimental data base itself was created by processing the three data sets through the laboratory AWR system operating in the template generation mode. Data Sets 1 and 2 were processed three separate times by the system, once for each of the three CCD analyzers. Data Set 3 was only processed with the bandpass filter bank to provide additional test material for that CCD analyzer.

5.2 Endpoint Determination

In a preliminary examination of the templates, a potential problem was discovered. This problem involved the consistency of word beginnings and ends across CCD analyzers. The beginning and ending detection for each template had been determined by software algorithms, as described in Section 2.2. While the same algorithm was used in all cases, the energy function was computed independently for each CCD analyzer. This approach resulted in occasional inconsistencies in endpoint detection for the same utterance. An example of this problem is presented in Table 5.3, which contains five repetitions of the utterance "Frank" for each of the three CCD analyzers. Inspection of the energy profiles shows that the final consonant "k" is missing from the third and fourth repetitions of the DFT templates, and from the fourth repetition of the mel cepstral templates. The full utterance is contained in all five of the BPF templates. Note that each template was intentionally "padded" with the five frames preceding the starting point and eight frames after the end of the utterance, thus permitting the subsequent manual adjustment of endpoints where needed.

We reasoned that it was unfair to compare the recognition capabilities of the three analyzers with varying segments of speech for the same word. Thus, we decided to manually specify beginnings and ends on any utterances where the software determinations were inconsistent across CCD analyzers. Of the 2760 templates generated from Data Sets 1 and 2, less than 3 % required such manual specification, and no manual changes were made on Data Set 3 templates. It also should be noted that in a few cases, an incorrect

TABLE 5.3 ENERGY FUNCTIONS FROM THE THREE CCD ANALYZERS FOR FIVE REPITITIONS OF THE UTTERANCE "FRANK"

DFT TEMPLATES

FRAME	TAKE 1	TAKE 2	TAKE 3	TAKE 4	TAKE 5
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	1	0	5	0	2
5	S->202	S-> 17	S->264	S->142	S->164
6	225	2	293	256	192
7	266	209	246	247	275
8	178	247	252	290	253
9	138	301	287	265	206
10	119	286	227	271	172
11	104	210	196	294	156
12	76	213	203	250	134
13	59	170	151	219	72
14	12	115	76	214	42
15	0	108	E-> 4	145	10
16	0	25	0	103	0
17	0	9	0	28	0
18	0	0	0	E-> 0	0
19	35	0	0	0	0
20	24	0	2	0	2
21	E-> 7	15	1	1	10
22	0	30	0	6	E-> 0
23	0	E-> 3		0	0
24	0	0		0	0
25	0	0		0	0
26	0	0			0
27	0	0			0
28	0	0			0
29		0			0
30		0			
31					
32					
33					
34					
35					
36					

MEL CEPSTRAL TEMPLATES

FRAME	TAKE 1	TAKE 2	TAKE 3	TAKE 4	TAKE 5
0	0	0	0	0	0
1	1	0	0	0	0
2	4	0	1	0	0
3	7	0	1	0	0
4	1	0	4	2	2
5	S->136	S-> 8	S->135	S->148	S-> 91
6	222	17	229	221	175
7	264	127	253	275	197
8	210	127	288	287	225
9	219	188	305	289	252
10	197	261	319	339	197
11	251	272	294	294	196
12	183	255	314	281	159
13	80	272	251	310	139
14	24	231	190	248	82
15	2	235	49	193	20
16	0	181	0	127	2
17	0	108	0	8	1
18	0	23	0	E-> 0	0
19	65	0	0	0	0
20	30	0	3	0	1
21	21	0	12	3	28
22	E-> 0	24	E-> 0	6	E-> 0
23	0	59	0	0	0
24	0	E-> 6	0	0	0
25	0	1	0	0	0
26	0	0	0		0
27	0	0	0		0
28	0	0	0		0
29	0	0	0		0
30		0			
31		0			
32					
33					
34					
35					
36					

BPF TEMPLATES

FRAME	TAKE 1	TAKE 2	TAKE 3	TAKE 4	TAKE 5
0	5	11	3	0	0
1	11	36	0	1	0
2	18	0	26	2	2
3	11	15	26	0	0
4	15	20	23	13	3
5	S-> 60	S->163	S-> 49	S-> 43	S-> 51
6	83	111	111	67	85
7	127	252	223	111	138
8	120	124	350	175	245
9	261	248	395	316	353
10	362	381	459	429	430
11	444	432	482	457	429
12	439	453	488	482	421
13	419	459	459	483	387
14	384	439	432	496	367
15	348	433	394	483	325
16	325	400	353	459	283
17	290	373	309	421	243
18	213	325	199	386	167
19	119	260	141	336	110
20	89	158	70	297	66
21	27	105	20	213	13
22	5	60	9	132	6
23	160	16	23	72	0
24	257	110	156	24	85
25	200	156	57	7	80
26	141	122	E-> 39	82	57
27	108	93	21	54	E-> 23
28	E-> 38	50	16	E-> 38	6
29	21	E-> 22	0	21	0
30	3	17	1	6	5
31	25	23	0	2	0
32	6	7	0	5	0
33	0	27	0	0	0
34	18	0		0	0
35	17	18		0	
36		6			

endpoint was deliberately specified in a template to make the template consistent with missing speech parameters in a corresponding template for another CCD analyzer. While such specifications definitely lowered recognition rates somewhat, they also provided for a better comparison of the three CCD analyzers.

The foregoing discussion of "manual" versus "automatically" determined endpoints is intended to preface the next chapter, where experimental results are presented. For each set of experiments, the type of endpoints used is specified.

Chapter 6: RESULTS FROM LABORATORY WORD RECOGNITION EXPERIMENTS

This chapter provides the detailed results of the word recognition experiments performed in the course of this contract. Three major groups of experiments were conducted. The first series compared the word recognition accuracies of the DFT, Cepstral and BPF CCD analyzers. The second set of experiments evaluated the speed versus accuracy tradeoffs of four speed-up techniques using BPF speech parameters. The third group of experiments provided an additional performance evaluation of the Bandpass Filter AWR system.

Each group of experiments was designed and conducted with the intent of comparing alternative word recognition techniques and algorithms. None of the experiments were intended to demonstrate the maximum achievable accuracy of a realtime AWR system.

While all of the preceding chapters serve as background material for these experiments, three previous sections are especially relevant to the experimental framework: the experimental test paradigm discussed in Section 2.3, the experimental control procedures of Section 4.5, and the template endpoint discussion in Section 5.2.

6.1 Comparison Results for the CCD Analyzers

The CCD analyzers were evaluated by comparing their performance on the laboratory AWR system. An initial set of experiments was conducted with Data Set 1 to determine the relative performance of the cepstral and mel cepstral analyses. These results are given in Table 6.1.

Table 6.1 Cepstral and Mel Cepstral Word Recogniton Accuracies with Software Determined Endpoints

Data Set	Speaker	Trials	Cepstral	Mel Cepstral
1. 26 Word	MA (M)	520	91.0%	93.8%
Phonetic	MB (M)	520	80.2%	92.7%
Vocabulary	FA (F)	520	87.9%	94.0%
	FB (F)	520	94.2%	94.6%
	Total	2080	88.3%	93.8%

As anticipated, the mel cepstral analysis achieved higher word recognition accuracy (93.8%) than did the cepstral analysis (88.3%). This result is also confirmed on an individual speaker basis, though the comparison is quite close for speaker FB. Based on these results, the mel cepstral analysis technique was selected for further experiments and was compared with the DFT and BPF techniques.

Recognition experiments were then conducted for each of the three analyzers over Data Sets 1 and 2, with manually determined endpoints as described in Section 5.2. The results of the final comparison are given in Table 6.2.

The total accuracy over the eight speakers and two vocabulary sets clearly shows that the BPF technique outperformed the mel cepstral technique, and that both of these techniques are better than the DFT technique. It is significant that this rank order is also generally true on an individual speaker basis. In addition, it is very encouraging that such high accuracies were obtained in this first attempt to use CCD analyzers. For example, the BPF system correctly recognized 97.5% of the words tested in Data Set 1, and 99.7% of the words tested in Data Set 2.

Table 6.2 DFT, Mel Cepstral, and Bandpass Filter Word Recognition Accuracy with Manually Determined Endpoints

Data Set	Speaker	Trials	DFT	Mel Cepstrum	BPF
1. 26 Word Phonetic Vocab.	MA (M)	520	95.8	97.5	98.8
	MB (M)	520	92.5	93.8	98.8
	FA (F)	520	88.5	91.5	92.7
	FB (F)	520	89.4	92.9	99.4
	Sub Total	2080	91.5%	93.9%	97.5%
2. 20 Word Cockpit Vocab.	BB (M)	400	99.0	99.3	100.0
	BL (M)	400	99.0	99.0	99.5
	RS (M)	400	99.8	99.3	99.8
	WB (F)	400	98.0	99.8	99.5
	Sub Total	1600	98.9%	99.3%	99.7%
	Total	3680	94.7%	96.3%	98.4%

Note that most of the errors in the BPF trials were attributed to one female speaker, "FA". All other speakers had recognition rates exceeding 98.8%. Speaker FA also showed the poorest performance with the DFT and mel cepstral analyzers, although for these devices she was responsible for a lesser percentage of the total errors. The problems encountered with speaker FA become clear when listening to her various repetitions of certain words. She drastically changed her pronunciation of the words "Adam," "Otto," and "Peter" on specific repetitions. Such a speaker demonstrates the need for multiple templates for certain words in order to enhance recognition accuracy on a given vocabulary.

Since the three speech parameter types (DFT, cepstrum, and BPF) are considered essentially equivalent in theory, why did their respective word recognition accuracies vary so consistently? Two explanations are offered. First, both the DFT and mel cepstral analyzers have considerably lower frame rates than does the BPF device. Even with variable frame rate encoding, the BPF system generates about 50 frames per second, compared to 39 frames per second for both the DFT and mel cepstral analyzers. Secondly, both the DFT and mel cepstral implementations are handicapped by as much as a 15% error in the CCD generated estimate of the magnitude of each spectral point. This is a result of the method used in the Reticon CZT chip for approximating the spectral magnitude.

6.2 A Comparison of LPC and BPF Results

Since the Bandpass Filter was the most effective of the CCD analyzers, we were interested in how it might compare to an LPC analyzer. Recognition experiments were therefore conducted on ITTDCD's PDP-11/60 computer, using software generated autocorrelation coefficients and resulting LPC parameters for Data Set 1. The amplitude data which accompanied the Berkeley generated autocorrelation coefficients was found to be in error. Consequently, it was necessary to manually determine the word beginnings and ends on all utterances of Data Set 1. In order to conduct a fair comparison with the BPF analyzer, the word beginnings and ends were likewise made consistent on Data Set 1 for the BPF parameters. The results of 2080 recognition trials conducted on each of these data types are presented in Table 6.3.

Table 6.3 Word Recognition Results for BPF and Software Generated LPC with Data Set 1 (2080 Trials)

Analysis Type	Accuracy
BPF	98.5%
LPC	99.5%

A recognition accuracy of 99.5% was achieved on the LPC coefficients, compared to 98.5% on the BPF. It should be noted that the frame rate on the software generated LPC coefficients was 100 frames per second, compared to approximately 50 frames per second on the variable frame rate encoded BPF parameters. This difference may explain the higher accuracy of the simulated LPC analyzer.

6.3 Speed/Accuracy Tradeoff Results for the BPF System

The results of the comparison experiments presented in Section 6.1 showed the Bandpass Filter to be the superior CCD analyzer for word recognition. Therefore, BPF outputs were chosen as the appropriate parameter base for a study of four speed-up techniques. The speed-up techniques were designed to reduce data rates, template storage requirements, and dynamic programming computations. The speed/accuracy experiments measured both the degree of speed-up and the accompanying effect on recognition accuracy.

These experiments were conducted on the BPF parameters of Data Sets 1 and 2, the same data used in the comparison experiments of Section 6.1, where the beginnings and endings of certain utterances had been manually specified. The bandpass filter results of Table 6.2 serve as a baseline for measuring the accuracy tradeoff of various speed-up techniques.

In order to measure the degree of speed up for these techniques, the number of microcycles required by each component of the recognition algorithm was estimated. (On the Quintrell processor, a microcycle is 225 nanoseconds.) A formula was then devised to estimate the number of microcycles required by the recognition algorithm for each frame of unknown speech. Assuming 16 bandpass filter coefficients and an average word length of 25 frames, the devised formula is:

$$MC = t\ (\ 4860 + (1-d)125c - 3200d\ )\ ,$$

where

MC = number of microcycles per frame of unknown speech
c = number of principal components
t = average number of active templates
d = percentage of distance calls eliminated by corner pruning

Thus for the baseline BPF comparison experiments on Data Set 1 (Table 6.2), c is 10, t is 26, and d is zero. In this case the formula yields MC = 159,000 for the number of microcycles per unknown frame. For Data Set 2, the baseline microcycle count is 122,000. In the remainder of this chapter, computational load refers to the percentage of microcycles required compared to these baseline figures for computational load.

The remainder of this section presents the results of the various speed-up experiments. The speed-up techniques themselves are described in Section 2.1.2. At the end of this section, the results are summarized in graphic form for convenient reference.

6.3.1 Variable Frame Rate Encoding

Variable frame rate encoding achieves a reduction in the data rate by eliminating frames in which there is little change in the bandpass filter coefficients. Various thresholds for a frame by frame distance metric were tested to determine the appropriate threshold for cutting the data rate by approximately 50 percent, from 100 frames per second to 50 frames per second. The selected threshold was then used with variable frame rate encoding to generate bandpass filter templates for Data Sets 1, 2, and 3. Thus variable frame rate encoding was an inherent technique in all bandpass filter experiments in this study. It alone produced a speed-up factor of four, as the size of both templates and unknowns are halved. A 50% reduction in template storage is also achieved by this technique.

No attempt was made to measure the recognition accuracy tradeoff for variable frame rate encoding. Such a measurement would involve storage and computational costs beyond the scope of this contract.

6.3.2 Number of Principal Components

Principal component analysis involves a linear transformation which maps the 16 bandpass filter coefficients into an orthogonal space of the same or a fewer dimensions. The eigenvectors which are employed in this transformation are ordered so that the first orthogonal coefficient has the maximum variance, the second coefficient has the second most variance, and so on. In this study, a set of eigenvectors is associated with each repetition of a vocabulary by a given speaker. Thus in Data Sets 1 and 2, there are five repetitions by eight speakers, resulting in 40 sets of eigenvectors. For a given eigenvector set, the amount of variance in each orthogonal dimension can be computed from the eigenvalues. These variances were averaged over the 40 sets and are presented in Table 6.4, along with the accumulated variance for successive sets of components.

Table 6.4 Percentage of Variance Accounted for by Principal Components for BPF Parameters from Data Sets 1 and 2

Principal Component	Variance (%)	Cumulative Variance (%)
1	58.4	54.4
2	15.7	74.1
3	9.8	83.9
4	4.4	88.3
5	3.2	91.5
6	2.2	93.7
7	1.4	95.1
8	1.1	96.2
9	0.9	97.1
10	0.7	97.8

A series of experiments was carried out to measure the effectiveness of BPF recognition with different numbers of principal components. Ten principal components had been used in the experiments discussed in Section 6.1. Further experiments were performed using seven, five, and three principal components on Data Sets 1 and 2. The results are presented in Table 6.5.

Table 6.5 Word Recognition Results with Varying Principal Components for BPF Parameters from Data Sets 1 and 2 (3680 Trials)

Number of Components	Recognition Rate %	Computational Load %
10	98.4	100.0
7	98.1	93.6
5	97.8	89.4
3	96.5	85.3

The results indicate that the BPF system degrades only slightly when using seven principal components. Principal component reduction yields a modest reduction in computational load. This speed up is attained by reducing the number of multiplies required in the principal component linear transformation and by reducing the number of coefficients involved in the dynamic programming distance metric.

The primary savings of principal component reduction is in storage required for templates. For example, templates containing five principal components require only half the storage of templates with ten principal components.

6.3.3 Corner Pruning

Corner pruning is an effective technique for eliminating many frame to frame comparisons in the dynamic programming algorithm. The corner pruning bandwidth is defined as the number of horizontal frames to which the dynamic programming alignment path is limited. For the baseline experiments, there was no such bandwidth limitation. Corner pruning experiments were designed and performed on Data Sets 1 and 2 for bandwidths of 7, 11, 15, and 19 frames. The number of dynamic programming distance computations was recorded during these experiments so that the computational load could be accurately measured. The results of these experiments are presented in Table 6.6.

Table 6.6 Word Recognition Results with Corner Pruning for BPF Parameters from Data Sets 1 and 2 (3680 Trials)

Corner Pruning Bandwidth	Recognition Rate %	Computational Load %
None	98.4	100.0
19	98.3	73.9
15	98.1	65.8
11	97.3	57.2
7	92.9	47.2

The results show corner pruning to be an effective speed up technique at the proper bandwidth. At a bandwidth of 19 frames, over one quarter of the computations can be eliminated with almost no decline in accuracy. At narrower bandwidths, further speed up occurs, but recognition accuracy also suffers.

If corner pruning is used, recognition errors tend to occur when the unknown word is spoken at a much faster or slower rate than the appropriate template for the word. That is, corner pruning errors occur when there is a significant disparity in length between the unknown and the appropriate template, in which case the bandwidth limitation prevents the proper dynamic programming alignment. This disparity is more likely to happen on longer words. Recognition results might therefore be improved if the corner pruning bandwidth were varied according to the length of the template.

6.3.4 Template Pruning

Template pruning is a technique whereby unlikely templates can be eliminated from the dynamic programming process at some point prior to the end of the unknown utterance. A template is pruned when its partial match score exceeds the the best (lowest) partial match score plus a threshold. Four threshold values were eventually chosen and experiments were conducted on Data Sets 1 and 2. The number of active (unpruned) templates was recorded during these experiments so that the reduction in computational load could be measured. The results of these experiments are presented in Table 6.7.

Table 6.7 Word Recognition Results with Template Pruning for BPF Parameters from Data Sets 1 and 2 (3680 Trials)

Template Pruning Threshold	Recognition Rate %	Computational Load %
None	98.4	100.0
150	98.4	55.2
100	98.4	41.4
75	98.0	33.8

The results show template pruning to be clearly the most powerful speed-up technique, capable of reducing computations by 60 percent with no decrease in accuracy. During most recognition trials, only two or three templates remained active throughout the recognition process.

6.3.5 Combined Template and Corner Pruning

Template and corner pruning proved to be such effective speed-up techniques that the evaluation of a combination of these techniques seemed worthy of pursuit. The results of three additional experiments combining both template and corner pruning are presented in Table 6.8.

Table 6.8 Word Recognition Results with Corner and Template Pruning for BPF Parameters from Data Sets 1 and 2 (3680 Trials)

Template Pruning Threshold	Corner Pruning Bandwidth	Recognition Rate %	Computational Load %
None	None	98.4	100.0
150	19	98.4	40.8
100	19	98.3	30.5
100	15	98.0	27.5

The combined pruning results match figures which could have been predicted from the individual corner and template pruning experiments, since the reductions in computational load are equivalent to the product of template pruning reduction and corner pruning reduction. Table 6.8 shows, for example, that 69.5% of the computational load can be eliminated with only 0.1% decrease in recognition accuracy.

6.3.6 Summary of Speed/Accuracy Tradeoff Results

A summary of the results obtained in this speed/accuracy tradeoff study is presented graphically in Figure 6.1. The horizontal dimension gives the percentage reduction in computational load, as well as the equivalent speed-up factor, while the vertical dimension shows the added error. Results are graphed separately for each of the speed-up techniques, and for two combinations of corner pruning and template pruning. Since the upper right hand corner of the graph represents the most speed-up with the

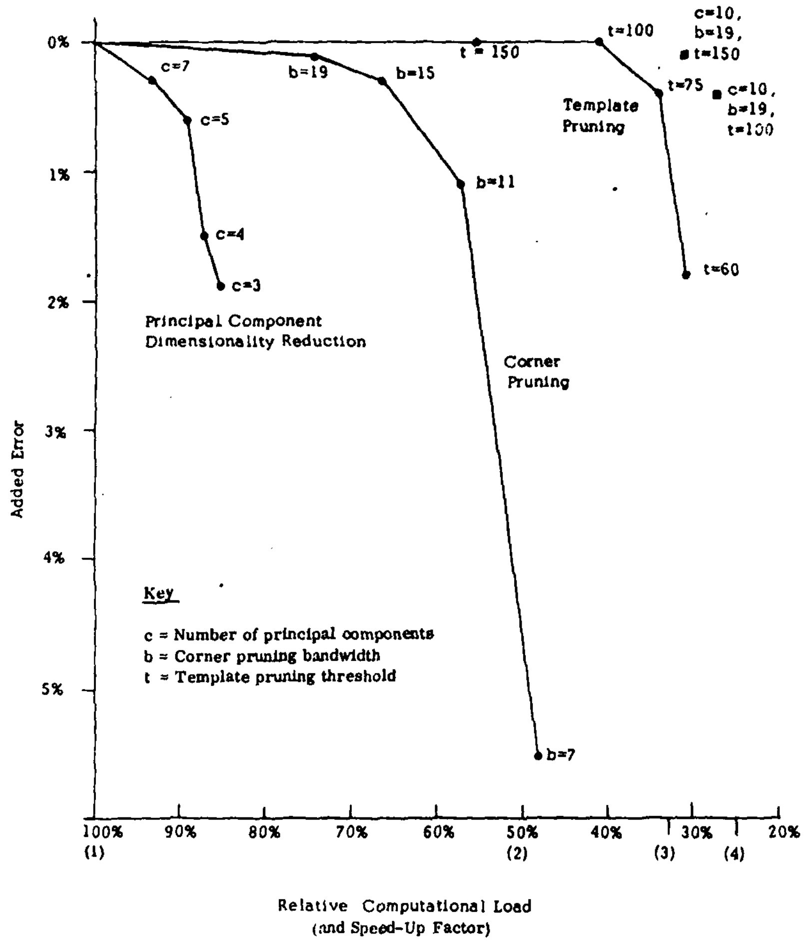


Figure 6.1 A Summary of Speed/Accuracy Trade-off Results

least decrease in accuracy, the figure clearly shows that template pruning is the most effective single speed-up technique. The figure also indicates that the inclusion of conservative corner pruning further improves overall performance. While the variable frame rate technique is not graphed in the figure, it was inherent in all BPF experiments and was responsible for a speed-up factor of four.

6.4 Additional Word Recognition Results for the BPF system

Because of the superior recognition accuracy shown by the Bandpass Filter CCD analyzer in the comparisons presented in Section 6.1, it was decided to gather additional performance statistics on the BPF system. Of particular interest was the performance of the system with automatically determined endpoints and with an extended vocabulary.

6.4.1 Results with Automatic Endpoint Detection

The further evaluation of the bandpass filter AWR system was accomplished by conducting additional recognition experiments on Data Sets 1, 2, and 3. Ten principal components were used in all recognition trials. Corner and template pruning were inoperative. Data Sets 1 and 2 had already been evaluated in the CCD analyzer comparison experiments with manually determined endpoints. In these overall performance experiments, however, endpoint detection was performed automatically by software algorithm.

The overall performance statistics for the laboratory BPF system with automatically determined endpoints are presented in Table 6.9. Average word recognition accuracies of 97.0%, 99.4%, and 98.5% were obtained for Data Sets 1, 2, and 3.

The results for Data Set 2 show that the contract performance goal (at least 99% accuracy on a 20 word vocabulary) was surpassed. The slight discrepancies between these results for Data Sets 1 and 2 and those in Table 6.2, are due to the differences between manual and automatic endpoint detection.

Table 6.9 Bandpass Filter Word Recognition Accuracies with Automatically Determined Endpoints

Data Set	Speaker	Trials	Accuracy
1. 26 Word Phonetic Vocab.	MA (M)	520	98.7
	MB (M)	520	98.1
	FA (F)	520	91.7
	FB (F)	520	99.4
	Sub Total	2080	97.0%
2. 20 Word Cockpit Vocab.	BB (M)	400	100.0
	BL (M)	400	98.5
	RS (M)	400	99.8
	WB (F)	400	99.5
	Sub Total	1600	99.4%
3. 26 Word Phonetic Vocab.	BB (M)	520	98.3
	BL (M)	520	99.2
	RS (M)	520	97.9
	WB (F)	520	98.3
	Sub Total	2080	98.5%
	Total	5760	98.2%

For the phonetic alphabet vocabulary (Data Sets 1 and 3), the most commonly missed words in these experiments are presented in Table 6.10. The word "Baker" heads the missed words list, being responsible for a total of 15 recognition errors from 160 total trials. The overall recognition rate on this word was still above 90%. "Baker" was missed at least once by five of the eight speakers. The most commonly confused words were the "David/Baker" combination. "David" was recognized as "Baker" ten different times by four different speakers.

6.4.2 Results with a Larger Vocabulary

In order to assess the performance of the bandpass filter AWR system on a larger vocabulary, experiments were conducted on the combined vocabularies of Data Sets 2 and 3. The merging of these data sets resulted in a 46 word vocabulary with five repetitions by each of four speakers, three males and one female. The 46 words include the 26 word phonetic alphabet list and the 20 word cockpit list. For each of the five repetitions of the resulting 46 word vocabulary by each of the four speakers, a new principal component matrix was computed.

Because of data memory limitations in the Quintrell processor, it was necessary to use only five principal components in the combined vocabulary experiments. To fairly evaluate the recognition performance on 46 words, the results must be compared to individual experiments on Data Sets 2 and 3 that were also performed with five principal components. Such an experiment had already been performed on Data Set 2 as one of the principal component series discussed in Section 6.3.2. A similar experiment therfore was performed on the 26 word vocabulary of Data Set 3 using five principal components.

Table 6.11 presents the results for a larger vocabulary. To provide a convenient reference, the table shows the averaged recognition accuracies for Data Sets 2 and 3 treated separately. These results are given for both ten and five principal components, and indicate that reducing the number of principal components used in the recognition process lowers overall accuracy by 0.3%. The last row in Table 6.11 shows that 97.7% correct

Table 6.10 Most Commonly Missed Words from the Phonetic Vocabulary of Data Sets 1 and 3 (160 Trials per Word, 4160 Total Trials, 96 Total Errors)

Vocabulary Word	Number of Misses	Number of Speakers	Confused with (Times)	
Baker	15	5	David	(7)
			Edward	(6)
			Others	(2)
David	12	5	Baker	(10)
			Others	(2)
Peter	12	3	Union	(8)
			Victor	(4)
Charlie	10	2	John	(9)
			Others	(1)
Victor	9	3	Baker	(7)
			Others	(2)
Adam	8	5	Ida	(3)
			X-ray	(3)
			Others	(2)
Otto	8	5	Ida	(5)
			Others	(3)

recognition resulted for the enlarged 46 word vocabulary. This compares to 98.6% averaged recognition accuracy for the two data sets treated separately, a decrease of 0.9%.

Table 6.11 Bandpass Filter Word Recognition Results for a Larger Vocabulary (3680 Trials, No Corner or Template Pruning)

Experimental Conditions	Vocabulary Size(s)	Principal Components	Recognition Accuracy %
Data Sets 2 & 3 Separate; Results Averaged	20 & 26 Words	10	98.9
Data Sets 2 & 3 Separate; Results Averaged	20 & 26 Words	5	98.6
Data Sets 2 & 3 Merged	46 Words	5	97.7

An additional 35 errors occurred over the 3680 trials of the enlarged vocabulary experiment. It should be noted that only 13 of the 35 errors were "across vocabularies", that is, words in the phonetic vocabulary being confused with words in the cockpit vocabulary, and vice versa. This indicates that the remaining 22 new errors were the result of the new principal component matrices computed for this experiment. All of these matrices were generated from the BPF coefficients of the 46 word vocabulary and thus provide a linear transformation for a more variable range of sounds than the principal components matrices derived from the 20 or 26 word vocabularies. Based on these results, we have hypothesized that for a larger vocabulary, more principal components may be required to achieve the same level of recognition accuracy.

An additional experiment on the same data was performed with corner pruning and template pruning. The corner pruning bandwidth was set at 19 and the template pruning threshold at 100. A recognition accuracy of 97.3% was achieved. The drop in accuracy is slight (0.4%), but greater than that of similar experiments with smaller vocabularies. The reduction in computational load is estimated at 75% for this experiment, corresponding to a speed-up factor of four.

Chapter 7: PROJECTIONS FOR LOW COST AUTOMATIC WORD RECOGNITION SYSTEMS

The previous chapters in this report give an affirmative answer to the question: Can CCD devices be used to generate speech recognition parameters that are useful for accurate low cost speech recognition? This chapter addresses a second basic question: What would be the cost for a realtime AWR system, using a CCD analyzer and current microprocessor technology?

7.1 The Design of a Microprocessor Based AWR System

In order to develop meaningful cost estimates, a low cost microprocessor architecture was designed to implement the realtime AWR system discussed in the previous chapters. The system uses speech parameters generated by the CCD analyzers as input. The microprocessor architecture was evaluated in terms of cost and complexity for solving various isolated word recognition problems using three different microprocessors: the 8-bit Intel 8085A, the 16-bit Motorola MC68000, and a 16-bit configuration of the AMD 2901A. These devices represent three classes of commonly available microprocessors.

The design analysis presented here is based primarily on a component analysis of the realtime Quintrell AWR system. However, since the realtime system was not completed until late in the contract, speed-up/accuracy estimates of the corner pruning and template pruning for the microprocessor study were based on recognition tests performed on the PDP-11/60 AWR system reported in Chapter 2. These tests indicated that about 60% of the dynamic programming matrix could be eliminated by corner pruning, and 30% of the templates could be eliminated by template pruning without significantly lowering performance. Results for the realtime laboratory AWR system reported in Chapter 6 showed that less of the dynamic programming matrix could be eliminated by corner pruning, but twice the number of templates could be be eliminated by template pruning without significantly lowering performance. The net result was a somewhat greater reduction in computational load than that reported in Chapter 2. Thus, the results of the computational analysis given here for the microprocessor design would improve somewhat if the newer speed-up/accuracy estimates were incorporated.

Figure 7.1 shows a block diagram of the complete microprocessor AWR system. The system was designed to use the CCD BPF analyzer. This choice, however, is reflected in the code only by the inclusion of the variable frame rate encoding algorithm and the principal component transformation. Only minor changes would have to be made if another CCD analyzer was selected. For example, the CCD mel cepstral analyzer (as developed for this study) does not use the variable frame rate encoding algorithm, but would require a linear transformation based on the mel cosine curves.

There are two modifications in this system relative to the AWR system used for the evaluation of the CCD BPF analyzer. First, only five coefficients are produced by the principal component transformation, instead of the ten used in the laboratory evaluation. This modification halves template storage requirements and increases recognition speed. It was shown in Chapter 6 that using five coefficients rather than ten lowered the recognition performance by only 0.5%. The reason for such small performance degradation is that the first five coefficients carry about 92% of the variance of the speech data, and coefficients 6 through 10 include only about 6% of the variance.

The second modification is to lower the precision of the eigenvectors in the principal component transformation from 16 bits to 8 bits. This reduces storage requirements for the matrix and reduces the cost, size, and power consumption of the multiplier needed for the principal component transformation. Though no tests were conducted in this mode, the modification should have little effect on the performance of the system since the eigenvectors serve only as a constant weighting function for the original 16 coefficients to obtain the reduced set. Empirical evidence indicates that the range of eigenvectors can be adequately represented in eight bits.

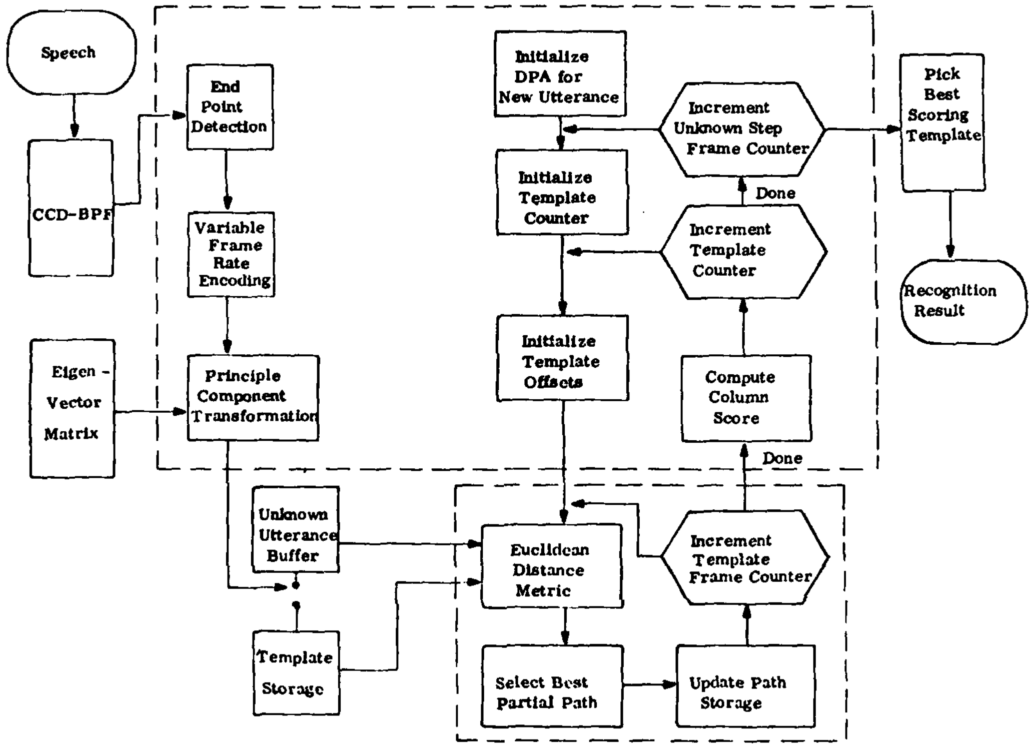


Figure 7.1 Block Diagram of the Microprocessor AWR System

7.2 Microprocessor Architectures

In determining the suitability of microprocessors for an AWR system, two different architectures were designed and evaluated. The first divides the total processing requirements between one master microprocessor and several slave microprocessors. The second approach uses several microprocessors that are ganged together in parallel.

The performance estimates for the microprocessors were derived by code translation from the Quintrell processor assembly language level code for the laboratory AWR system. The critical areas (distance computation and the principal component transformation) were translated in detail to the 8085A, MC68000, and 2901A assembly languages. The less critical areas were estimated by multiplying Quintrell execution times by the ratios obtained from detailed benchmarks. Approximate throughput ratios relative to the Quintrell (excluding multiplications) are given for these estimates in Table 7.1.

Table 7.1 Approximate Throughput Ratios (Excluding Multiplications) for Several Processors

Processor	Throughput Ratio
Quintrell	1.0
8085A	0.1 (8-bit operations)
MC68000	0.45
2901A	1.5

The microprocessors must be augmented with a hardware multiplier/accumulator to perform the principal component transformation. The slave processors compute the squaring within the distance calculation by table lookup. Memory size estimates were obtained from the code translation and rounded up.

7.2.1 Master-Slave Architecture

In the first architecture configuration, the two regions outlined in Figure 7.1 indicate how the processing would be divided between master and slave microprocessors. The computationally intensive inner loop of the dynamic programming recognition algorithm (the smaller region in Figure 7.1) is performed by a slave processor for several templates in realtime. The required number of slaves depends on the number of word templates to be recognized. In addition, the number of slaves that can be controlled by one master is limited. If a large number of templates are required, several master-slave systems would work in parallel with the templates distributed equally amongst them. The CCD analyzer output is distributed to all processors simultaneously. Figure 7.2 shows this architecture.

The master-slave architecture was evaluated for the Intel 8085A and the Motorala MC68000 microprocessors. Figure 7.3 presents a block diagram of the 8085A architecture configuration. The master processor consists of an 8085A CPU, 2 K bytes of PROM program memory, 4 K bytes of RAM memory for template storage (20 templates) and working area, a USART for serial communications, and interrupt control and bus interface circuitry. It is assumed that the master processor generates the timing for the system. The serial I/O is for communicating the recognition results and other data to (or from) a TTY, CRT, host processor, or other compatible device.

The slave processors are simpler. Each slave consists of an 8085A CPU, a 1.5 K byte program memory, and 1 K byte RAM memory for template storage and working space. Each slave processor can handle just over two templates in realtime, and each master processor can handle 10 slave processors.

Each master processor is assumed to have 10 miscellaneous SSI and MSI components, which are not shown in the figure. Each slave processor is is assumed to have five miscellaneous SSI and MSI components.

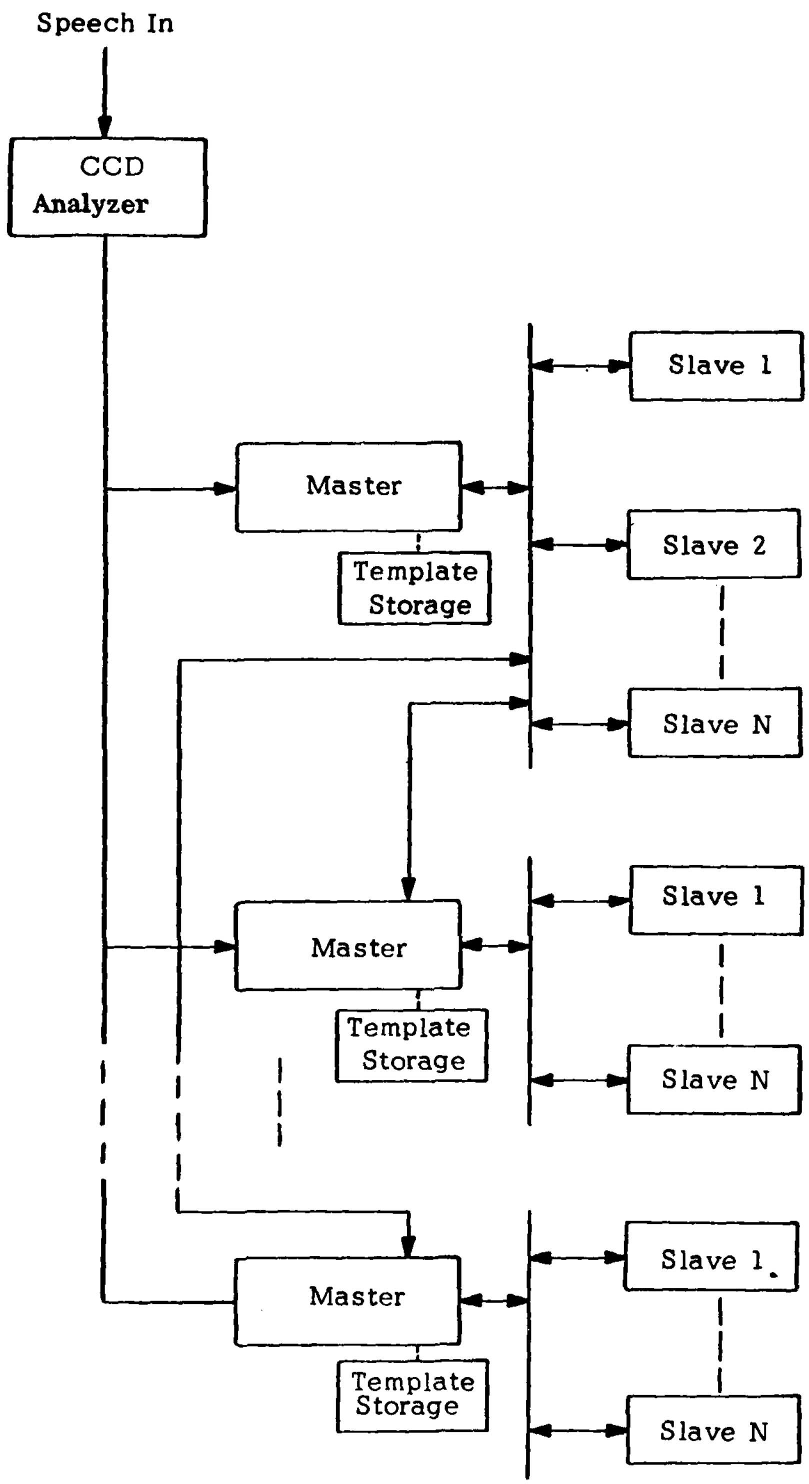
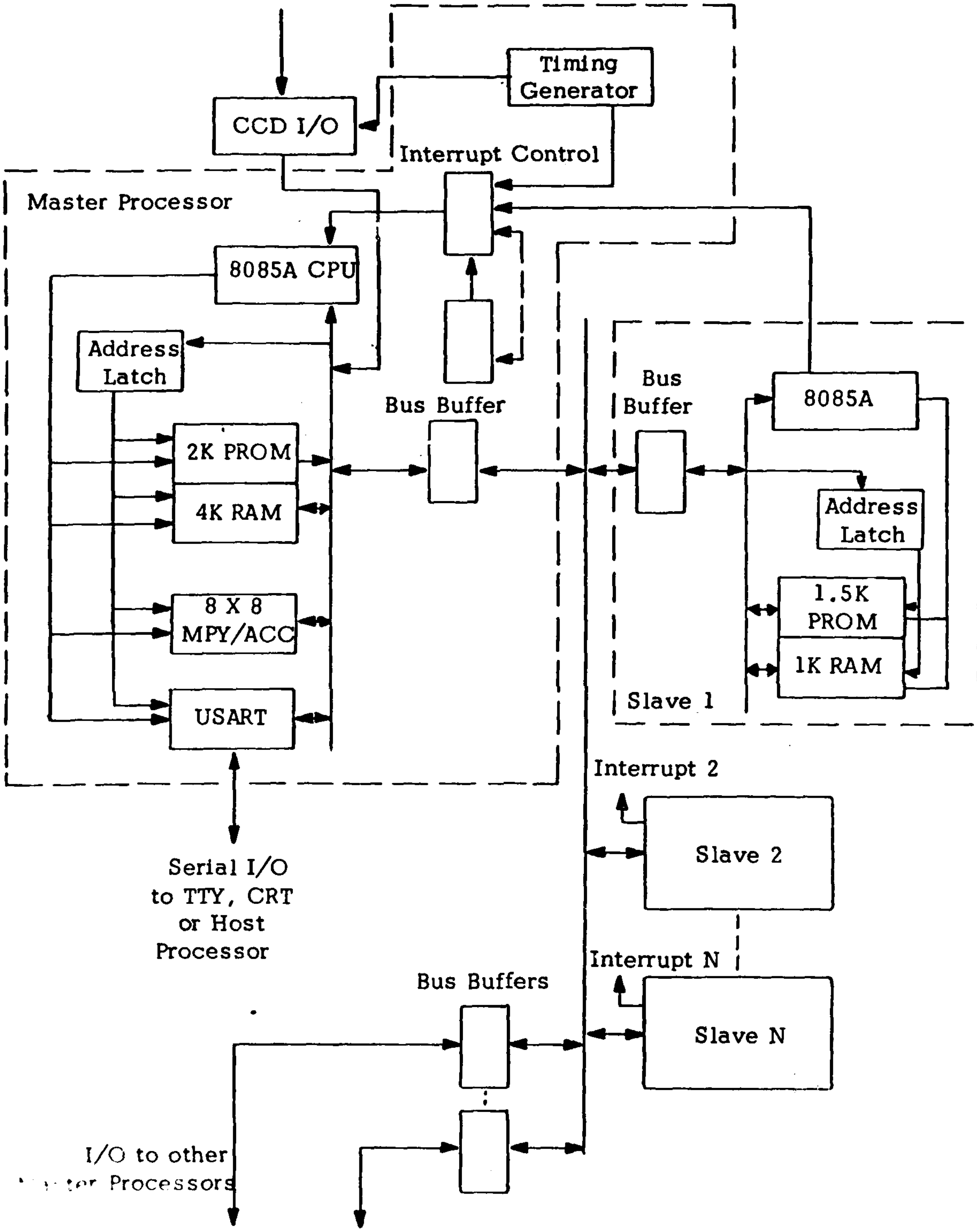


Figure 7.2 Master - Slave Microprocessor AWR Architecture



7.3 Intel 8085A Design for the Master – Slave AWR Architecture

Communication between master and slaves is initiated by interrupts, with each slave processor having a dedicated interrupt in the master. Communication is between master and slaves only, that is, there is no direct interchange between slave processors.

Figure 7.4 is a block diagram of the Motorola MC68000 architecture configuration. The elements of the master and slave processors are very similar to that in the 8085A implementation. The memory sizes are somewhat larger because the processing capacity is greater — each slave can process 10 templates in realtime, compared to two for the 8085A. Communication in this architecture is the same as in the 8085A implementation.

7.2.2 Ganged Architecture

The second architecture that was investigated employs the more powerful AMD 2901A in a bit slice configuration. The processing capacity of the 2901A is such that a master-slave subdivision would not be appropriate for vocabularies of up to several hundred words. If many word templates are needed, a number of processors are ganged together as shown in Figure 7.5, and the templates are divided amongst them. During recognition, the CCD outputs are distributed to all processors simultaneously. At the end of the utterance, Processor 1 makes the final recognition decision based on its own results and on those of the other processors. This final decision does not add significant computational load for the vocabulary sizes considered. Figure 7.6 illustrates the architecture configuration for one 2901A microprocessor. The design has separate program and data memory, a hardware multiplier/accumulator (8-bit by 8-bit), and a 2901A connected in a 16-bit configuration. The high throughput capacity allows each processor to compare 38 templates in realtime.

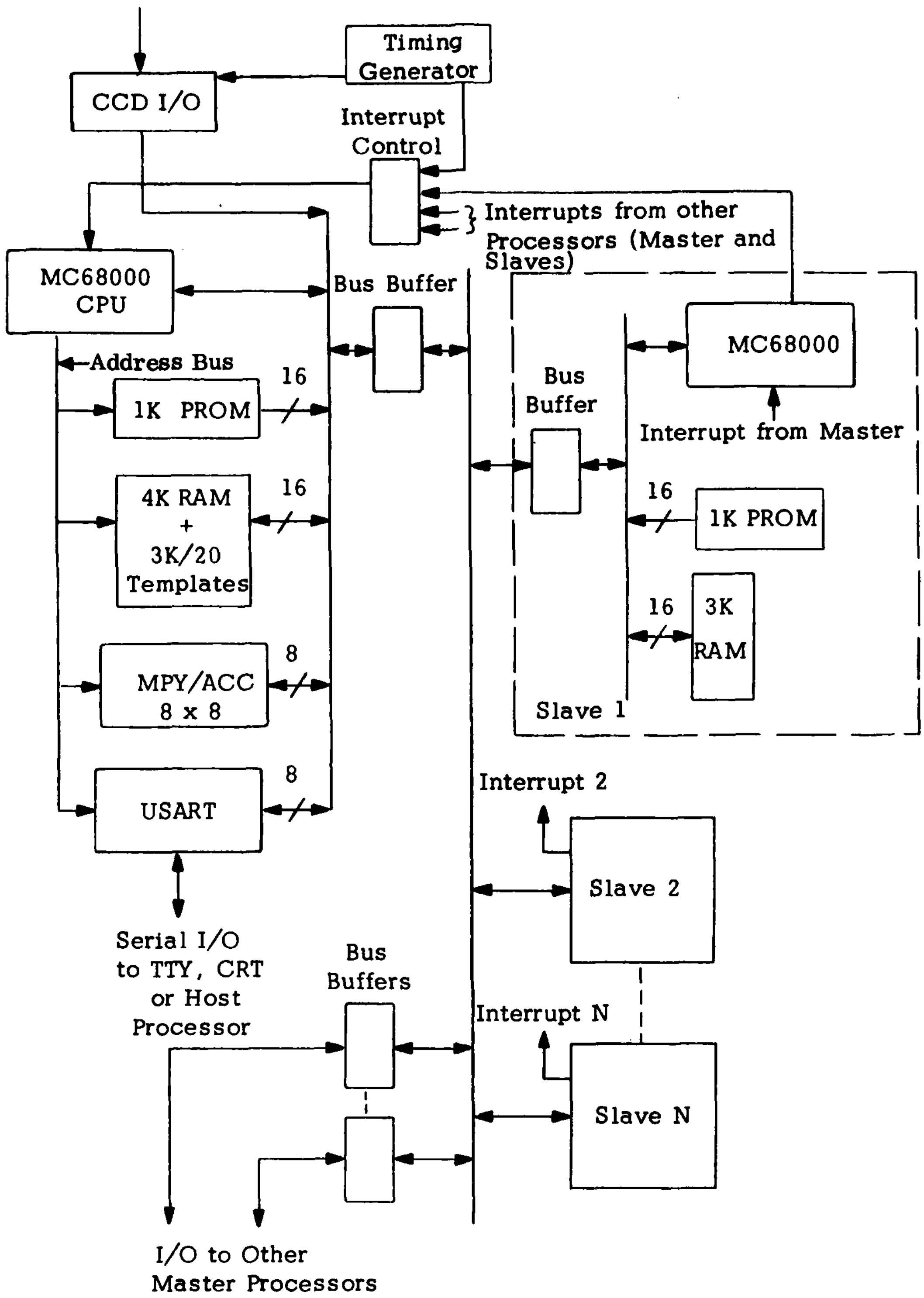


Figure 7.4 Motorola MC68000 Design for the Master-Slave AWR Architecture

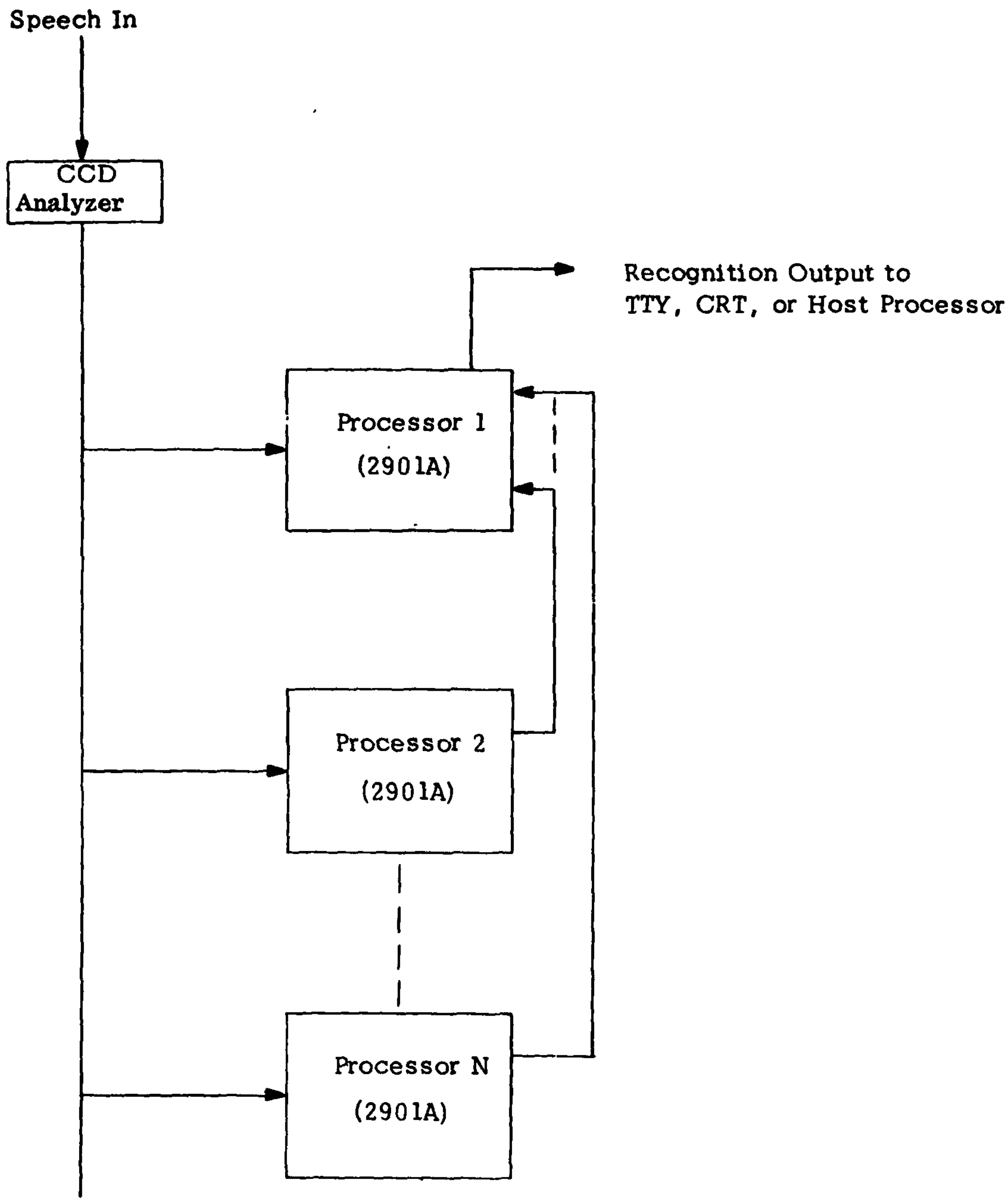


Figure 7.5 Ganged Microprocessor AWR Architecture Based on the AMD 2901A

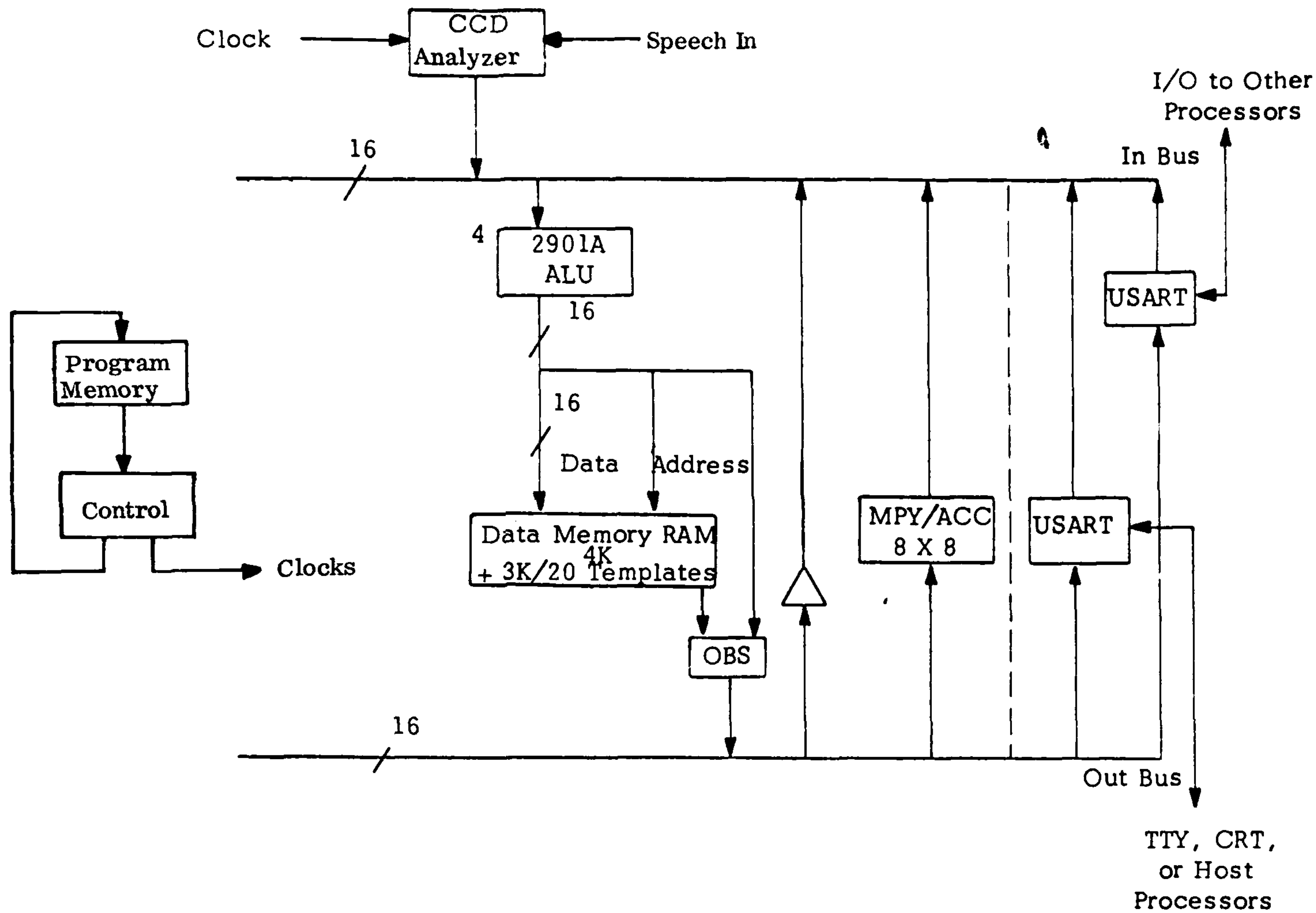


Figure 7.6 Design for One 2901A Processor in the Ganged AWR Architecture

7.2.3 Special Purpose Hardware

Another consideration for the development of a low cost AWR system is whether certain functions could be performed more economically with special purpose hardware. The master-slave architecture was examined to investigate this possibility for the "minimum" function in the dynamic programming algorithm. It was determined that special purpose hardware using MSI components would not be effective in this situation, because the number of additional components required would not be offset by a sufficient increase in throughput.

An LSI implementation of the "minimum" function could be cost effective in high volumes. This might be especially true if it incorporated other features, such as the capability to add the distance to the selected minimum value. Custom LSI implementations of special purpose hardware, however, were felt to be beyond the scope of this analysis and were therefore not pursued further.

7.3 Cost Analysis for a Microprocessor AWR System

To determine cost projections for a microprocessor AWR system, estimates must be made for both the CCD analyzers and the microprocessor implementations. Such projections are more difficult for the CCD analyzers, since new LSI chips would most likely need to be developed. Cost estimates for the microprocessor portion of the AWR system are more reliable because standard components were used in the design.

7.3.1 CCD Analyzer Hardware Costs

There is very limited data on which to base CCD cost estimates. Reticon currently markets a line of CCD analog processing chips, two types of which were used in the speech analyzers for this study. The DFT/Cepstral analyzer is based on a single chip Chirp-Z transformer (R5601), and the Bandpass Filter analyzer is constructed with six third-octave filter chips (R5604) and a single octave chip filter chip (R5606). Texas Instruments is developing a single chip version of a complete filter bank, including rectifiers, low pass filters and multiplexed A/D. Dr. Broderson at Berkely has also projected the feasibility of single chip autocorrelator which could be used with a single chip microprocessor to perform LPC anaylsis. To estimate the cost of future CCD speech analyzers, therefore, we have assumed that appropriate single chip CCD devices would be available for each type of analyzer, and have based their costs on the current 100 quantity selling price of Reticon chips of approximately equal complexity. Specifically excluded from the these estimates are the costs for chip development, which could easily exceed several hundred thousand dollars per device.

Table 7.2 summarizes the CCD analyzer cost projections. It shows that each of the four analyzer types has a $300 CCD chip as the main processing component, along with hardware to perform analog preemphasis and anti-aliasing. The LPC analyzer also includes a $45 single chip microprocessor. In addition, the LPC, DFT, and Cepstral analyzers contain a $10 A/D or log A/D chip. Quantity projections for total costs range from $306 for the BPF analyzer, to $361 for the LPC analyzer. These differences are probably insignificant given the assumptions on which the estimates are based. What is significant, however, is the fact that high quality speech analyzers using CCD components should be available in the future for only several hundered dollars.

7.3.2 Microprocessor Hardware Costs

The microprocessors and architectures were compared by cost and complexity for recognition tasks with different numbers of word templates. The results are based on an average length template of 0.6 seconds (i.e. 30 frames after variable frame rate encoding; the word "frequency" is typically about 0.6 seconds long). The results also reflect the goal that processing is done in realtime, so that the identity of a spoken utterance is available as soon as the end of the utterance is detected.

The cost estimates are based on the cost of IC's plus $1.00 per watt to cover power supply costs. IC's were priced from current vendor quotes in quantities of 100. The cost estimates do not include packaging, testing, and related tasks, because these factors are highly dependent on quantity, the efficiency of the manufacturer, environmental requirements,

Table 7.2 Cost Projections for CCD Speech Analyzers

DEVICES	LPC	DFT	CEPSTRAL	BPF
A/D	10.00	10.00	-	-
LOG A/D	-	-	10.00	-
ANALOG PREEMPHIS AND ANTI-ALIASING	6.00	6.00	6.00	6.00
SINGLE CHIP MICRO-PROCESSOR (8748)	45.00	-	-	-
CCD CHIP	300.00	300.00	300.00	300.00
TOTAL	361.00	316.00	316.00	306.00

and other conditions.

Table 7.3 shows a comparison of the three architecture configurations for three vocabulary sizes, assuming 60% corner pruning and no template pruning. The 8085A is clearly the least efficient. The 8085A approach is not cost effective because each slave can only handle two templates in realtime. The resulting large number of CPU's requires replication of program memory. Also, use of data memory is less efficient and many components are needed for interprocessor communication. With less effective corner pruning, as the Quintrell experiments suggest, each slave can handle only one template, and this design becomes even less attractive.

The 68000 and 2901A cost comparison is fairly close. A clear advantage of the 2901A implementation is that there is no master/slave partitioning. The communication between master and slave processors is in two directions and involves establishing bus control and command and data interchange. The 2901A processors simply report recognition results at the end of an utterance. Program development cost, documentation, testing, and maintenance should be simpler for the single CPU design of the 2901A.

The same basic recognition hardware can process more vocabulary templates by taking advantage of template pruning. For example, 30% template pruning enables the effective vocabulary size to be increased by 30%, with only a small increase in hardware cost due to the requirement for additional template storage memory. Table 7.4 shows the results of combining 60% corner pruning with 30% template pruning for the more attractive MC68000 and 2901A designs. Somewhat better results would be obtained if the speed-up/accuracy estimates from the Quintrell experiments were used. While corner pruning was less effective in those experiments, the additional template pruning more than compensated, yielding a net improvement in speed-up/accuracy performance.

Another method for increasing the number of templates that can be processed by the same basic hardware is to relax the requirement for realtime response. For example, a 0.6 second recognition response time doubles the amount of processing time avaiable, and therefore doubles the

Table 7.3 Cost Comparison of Three Microprocessor Devices for a Realtime AWR System (60% Corner Pruning; No Template Pruning)

Vocabulary Size	Intel 8085A				Motorola MC 68000				AMD 2901 A		
	IC's	M	S	$	IC's	M	S	$	IC's	P	$
20	135	1	10	1084	50	1	2	780	80	1	895
100	675	5	50	5420	194	1	10	2674	249	3	3050
300	2025	15	150	16260	582	3	30	8022	671	8	8412

M = Number of Master Processors
S = Number of Slave Processors
P = Number of Processors

Table 7.4 Microprocessor Cost Projections for a Realtime AWR System with 30% Template Pruning (and 60% Corner Pruning)

Vocabulary Size	Motorola MC 68000				AMD 2901 A		
	IC's	M	S	$	IC's	P	$
26	56	1	2	849	82	1	976
130	224	1	10	2915	255	3	3294
390	672	3	30	8745	688	8	9104

Table 7.5 Microprocessor Cost Projections for an AWR System with a 0.6 Second Response Time (60% Corner Pruning; 30% Template Pruning)

Vocabulary Size	Motorola MC 68000				AMD 2901 A		
	IC's	M	S	$	IC's	P	$
52	76	1	2	1076	87	1	1178
260	346	1	10	4395	270	3	3900
780	1038	3	30	13185	768	8	12344

vocabulary size of a given system. In some applications of an AWR, such a delay in response would not be prohibitive. It should be noted, however, that this method also requires a corresponding increase in template storage memory which increases the overall cost of the system by a small amount. The net result of permitting a 0.6 second delayed response and increasing the cost of template storage memory is shown in Table 7.5. With the 2901A, such a relaxation of response time allows the vocabulary size to be doubled, with only a 20% - 30% resulting increase in cost.

7.4 Overall AWR Cost Projections

Based on the preceding analysis, Table 7.6 presents a summary of the overall cost projections for a microprocessor AWR system with a CCD speech analyzer. The table contains data for the CCD Bandpass Filter analyzer, which performed better than the DFT and Cepstral analyzers in realtime laboratory tests. It also uses the cost projections for the 2901A microprocessor design with a 0.6 second response time, the configuration that appears to be the most attractive for the vocabulary sizes being considered. As Table 7.6 indicates, hardware costs for a complete system should range between about $1,500 for a 52 word vocabulary, to about $12,700 for a 780 word vocabulary.

Table 7.6 Overall Cost Projections for a Microprocessor AWR System with a CCD Speech Analyzer

Vocabulary Size	CCD BPF Analyzer	2901A Microprocessor	Total Hardware Cost
52	$306	$1,178	$1,484
260	$306	$3,900	$4,206
780	$306	$12,344	$12,650

Chapter 8: CONCLUSIONS AND RECOMMENDATIONS

During this study, ITTDCD evaluated the feasibility of using Charge Coupled Devices (CCD'S) and microprocessors to reduce the cost and complexity of Automatic Word Recognition (AWR) systems. Three speech analysis techniques were implemented using currently available CCD hardware. These included a Bandpass Filter (BPF) analyzer, a Discrete Fourier Transform (DFT) analyzer, and a Cepstral analyzer. For each of these CCD analyzers, software was developed to make the respective speech parameters more suitable for realtime word recognition. ITTDCD then incorporated the CCD hardware and software into a realtime laboratory AWR system and employed this system in a performance comparison of the three CCD based speech analysis techniques. The laboratory AWR system was further used as a test vehicle for experiments measuring the effectiveness of various word recognition speed-up methods. Finally, ITTDCD designed and evaluated architectures for microprocessor based versions of the realtime AWR system and formulated cost projections for such systems.

ITTDCD has drawn a number of conclusions from these development activities and associated experimental results. The major conclusions are discussed in the following section, while recommendations for future investigations are presented in Section 8.2.

8.1 Conclusions

The major conclusion that has resulted from this study is that a combination of CCD devices and microprocessors can provide an effective, low cost Automatic Word Recognizer. Details supporting this conclusion are summarized below.

1. CCD speech analyzers can provide speech parameters which are useful for accurate word recognition in realtime. Recognition accuracies exceeding 99% can be achieved on a 20 word vocabulary.

2. Of the CCD speech analysis techniques compared, the Bandpass Filter analyzer provides the best parameters for isolated word recognition. Over 3680 trials, the BPF analyzer produced less than half as many recognition errors as the second best technique, the mel cepstrum analyzer. Futhermore, the BPF analyzer achieved a higher recognition accuracy than the other CCD analyzers for seven of the eight speakers used in the comparison experiments.

3. The second best CCD analyzer, the mel cepstrum technique, was clearly superior to the third CCD analyzer, the DFT.

4. Mel cepstral analysis is superior to cepstral analysis as a word recognition technique.

5. All four of the speed-up algorithms which were studied are worthwhile applications for an efficient realtime AWR system.

   a. Variable frame rate encoding is an integral part of the BPF technique and provides significant data rate and template storage reductions.

   b. Template pruning is a powerful speed-up technique, capable of reducing recognition computations by 60 percent with no decline in recognition accuracy.

   c. Principal component reduction provides template storage savings and a modest computational speed-up.

   d. Corner pruning is an effective speed-up technique which can lower computations by 25% with little loss in accuracy.

6. Of the three microprocessor architectures evaluated as potential low cost AWR systems, the Intel 8085A approach is clearly the least efficient and least cost effective. While the AMD 2901A and Mototola MC68000 designs are comparable from a cost standpoint,

AD-A084 743 ITT DEFENSE COMMUNICATIONS DIV NUTLEY N J F/G 9/2
SOLID STATE AUDIO/SPEECH PROCESSOR ANALYSIS.(U)
MAR 80 A R SMITH, B P LANDELL, G VENSKO F30602-78-C-0359
UNCLASSIFIED RADC-TR-80-75 NL

2 OF 2
AD
A084743

END
DATE
FILMED
6-80
DTIC

the 2901A is preferable from a performance standpoint, particularly with regard to throughput and simplicity.

7. Hardware cost projections for an AWR system featuring an AMD 2901A architecture and a Bandpass Filter CCD analyzer should range between $1,500 for a 52 word vocabulary, to about $12,700 for a 780 word vocabulary. These costs do not include custom chip development, detailed hardware design, construction or testing.

8.2 Recommendations for Future Investigations

ITTDCD recognizes that many aspects of CCD analyzer and microprocessor based AWR systems are deserving of further research and development. These aspects include additional improvements to the AWR algorithms themselves, the development and customer evaluation of a deliverable low cost AWR system, and the extension of the AWR algorithms and hardware design to a continuous speech recognition system.

Certain algorithms employed in the word recognition process might be refined through further experimentation and analysis. Among these are algorithms for detection of word boundaries (beginnings and ends) and for variable frame rate encoding. Principal component matrices (eigenvectors) deserve further study with respect to the number of components versus vocabulary size. The possibility of a speaker independent principal component transformation for a specific vocabulary should also be explored. In addition, better methods for generating vocabulary templates should be developed that utilize clustering and averaging techniques. These methods should result in less sensitivity to inter- and intra-speaker variablility in the pronunciation of the vocabulary words.

Since such encouraging results were obtained with respect to word recognition accuracy and hardware cost projections, we recommend the development of a deliverable AWR system that is based on the ideas formulated and tested here. To proceed with such a development, the AWR system characteristics would have to be specified with respect to vocabulary size, vocabulary subsetting, accuracy, and response time. Then a detailed system design would be completed and the AWR system constructed and tested at the ITTDCD laboratory. This system could be delivered to the Air Force for additional testing and evaluation in an Air Force laboratory or operational environment.

Another promising area for further activity is the application of the low cost AWR hardware and software concepts to the problem of recognizing natural continuous speech. Recent experiments at ITTDCD have shown that the dynamic approach to word matching can effectively locate words embedded in conversational speech, without requiring the words to be separated by pauses. These general concepts could be extended and enhanced for continuous speech recognition to properly handle word boundary coarticulation and other variablility effects. The CCD analyzers, microprocessor architectures, and dynamic programming software should significantly improve the prospects for accurate and affordable continuous speech recognition systems.

REFERENCES

[1] Sakoe, H., and Chiba, S., "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-26, pp. 43-49, Feb. 1978.

[2] Harman, H., Modern Factor Analysis, Chicago: University of Chicago Press, 1967.

[3] Pols, L., "Real-Time Recognition of Spoken Words," IEEE Trans. on Computers, Vol. C-20, No. 9, pp. 972-978, Sept. 1971.

[4] Itakura, F., "Minimum Prediction Residual Applied to Speech Recognition," IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-23, pp. 67-72, Feb. 1975.

[5] Davis, S. B., and Mermelstein, P., "Evaluation of Acoustic Parameters for Monosyllabic Word Identification, "Journal Acoust. Soc. Am., Vol. 64, Suppl. 1, pp. S180-S181, Fall 1978, (abstract).

[6] Atal, B., "Effectiveness of Linear Prediction Characteristics of the Speech Wave for Automatic Speaker Identification and Verification," Journal Acoust. Soc. Amer., Vol. 55, pp. 1304-1312, June 1974.

[7] Gray, A., and Markel, J., "Distance Measures for Speech Processing," IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-24, pp. 380-391, Oct. 1976.

[8] Rabiner, L., Levinson, S., Rosenberg, A., and Wilpon, J., "Speaker Independent Recognition of Isolated Words Using Clustering Techniques," Conference Record of the 1978 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, Cat. No. 79CH1379-7 ASSP, pp. 574-577, April 1978.

[9] Klatt, D., "A Digital Filter Bank for Spectral Matching," Conference Record of the 1976 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, Cat. No. 76CH1067-8 ASSP, pp. 573-576, April 1976.