Audio Coding and Identification for an Interactive Television

Application

by

Gregory S. Sheets

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

IN

ELECTRICAL ENGINEERING

APPROVED:

[Signatures]

A. A. (Louis) Beex, Chairman

[Signatures]

H. F. VanLandingham

John S. Bay

May 1996
AUDIO CODING AND IDENTIFICATION FOR AN
INTERACTIVE TELEVISION APPLICATION

by

Gregory S. Sheets

Committee Chairman: Dr. A. A. (Louis) Beex

Electrical Engineering

(ABSTRACT)

This thesis describes the development of the audio coding and identification algorithms for an Interactive Video Data Service (IVDS) system. The overall purpose of the IVDS system that is discussed in this thesis is to provide the user with a wireless return path to the host site. The information being broadcast from the host site can vary. However, it is assumed here that the host site is broadcasting some type of advertisement or commercial, in which case the user's return path is for communicating an interest to the host site in obtaining something of value. The host site is made aware of the interest of the user when a transmission from the user is received. After the user's transmission has been obtained, decoded, and analyzed the host site takes an appropriate course of action.

Our particular IVDS system must be noninvasive to the existing broadcast system
and have no physical contact with the television at the user end of the system. Our system must also allow the user to make a selection at virtually any time. The performance and feasibility of such a system is studied. The necessary number of bits per user transmission, speed of commercial identification, and the success rate of the commercial identification process are tradeoff factors in our IVDS application. We examine these tradeoff factors and discuss the effect various approaches to solving our problem have on them.

The correlation coefficient is used as a distance measure in our IVDS application and compared on the basis of feasibility to other distance measures that have recently been used in the related problems of speech recognition and speaker identity verification. The performance of a few distance measures that are based on the inner product are studied through various simulations.

The performance of the 'optimal' system that is developed as a result of the various simulations is acceptable for 'small scale' usage. The performance can be optimized for various criteria and depends on the distance measure used and some of the parameters of the proposed system.
ACKNOWLEDGMENTS

I would like to thank my advisor Dr. A. A. (Louis) Beex for his patience and diligent work in revising my thesis, as well as for his helpful suggestions and directions that I needed to finish this thesis. I would also like to thank Dr. Hugh VanLandingham and Dr. John Bay for serving on my committee. I want to acknowledge the Center for Wireless Telecommunications for their financial support under the IVDS project.

I wish to thank my lab mates in the DSP Research Lab. It has been a pleasure to work with them all. I would like to thank my parents for being supportive throughout all of the years of my education. I want to thank my fiancée Linda for standing by me throughout the last couple of years and encouraging me when I needed it.

Finally and most importantly, I would like to thank my Lord and Savior Jesus Christ to whom I am eternally indebted.
# TABLE OF CONTENTS

1.0 INTRODUCTION .................................................................................................................. 1
  1.1 Problem Statement ............................................................................................................. 1
  1.2 Purpose of This Thesis ...................................................................................................... 3
  1.3 Overview ......................................................................................................................... 4

2.0 GENERAL APPROACHES .................................................................................................... 5
  2.1 Assumptions Made About the Problem ............................................................................ 5
    2.1.1 Channel Assumptions ............................................................................................... 5
    2.1.2 Input Assumptions ..................................................................................................... 8
  2.2 Coding Approaches ......................................................................................................... 10
    2.2.1 Waveform Coders ..................................................................................................... 10
      2.2.1.1 Time Domain Waveform Coders ........................................................................... 11
      2.2.1.2 Frequency Domain Waveform Coders ................................................................. 13
    2.2.2 Vcoders .................................................................................................................... 14
      2.2.2.1 Vocoder Approaches .......................................................................................... 15
      2.2.2.2 Linear Predictive Vcoders .................................................................................. 16
  2.3 Identification Approaches ............................................................................................... 19
    2.3.1 Differences from Typical Identification Applications .............................................. 20
    2.3.2 Feature Selection ....................................................................................................... 21
    2.3.3 Measuring the Distance Between Feature Vectors ................................................... 22
      2.3.3.1 Euclidean Distance Measure .............................................................................. 22
      2.3.3.2 Cepstral Distance Measure ............................................................................... 23
      2.3.3.3 Itakura Distance Measure .................................................................................. 23
      2.3.3.4 Mahalanobis Distance Measure ........................................................................ 24
      2.3.3.5 Correlation as a Distance Measure ..................................................................... 25
      2.3.3.6 Analysis of Efficiency and Feasibility of the Measures ..................................... 33
  2.4 System Description ......................................................................................................... 34
    2.4.1 Preprocessing and Parameterization at the Host Side .............................................. 34
    2.4.2 Preprocessing and Parameterization at the User Side ............................................. 37
    2.4.3 Identification Technique ........................................................................................... 38
3.0. THEORY AND DETAILS OF SYSTEM .................................................43

3.1 Windows ..................................................................................43
  3.1.1 Window Selection .................................................................44
  3.1.2 Spectral-Temporal Resolution Trade-Off ..............................48

3.2 Preprocessing Filters ..................................................................49
  3.2.1 Bandpass Filter ..................................................................49
  3.2.2 Preemphasis Filter ...............................................................51

3.3 Order Estimation .......................................................................52

3.4 Optimal Coefficient Representation ........................................53
  3.4.1 Identification Performance Optimal Representation .............55
  3.4.2 Quantization Optimal Representation ....................................58

3.5 User Vector Calculation ............................................................59
  3.5.1 Number of User Vectors ......................................................60
  3.5.2 Spacing of User Vectors ......................................................60

3.6 Quantization of Coefficients ......................................................62

3.7 Coefficient Significance .............................................................63

3.8 Overlap of Frames in Database ................................................63

4.0 SIMULATIONS AND RESULTS ......................................................65

4.1 Test Conditions .........................................................................65

4.2 Standard Test Configuration .....................................................67

4.3 Tests Results and Analysis .......................................................70
  4.3.1 Windowing ..........................................................................71
  4.3.2 Preemphasis Filter ...............................................................74
  4.3.3 Bandpass Filter ....................................................................77
  4.3.4 Order Estimation ..................................................................81
  4.3.5 Overlap of Database Frames ................................................85
  4.3.6 Coefficient Representation ..................................................88
  4.3.7 Importance of Particular Coefficients in the User Vectors ....90
  4.3.8 Number of User Vectors ......................................................94
  4.3.9 Spacing of User Vectors ......................................................96
LIST OF FIGURES

Figure 1.1: High Level Block Diagram of an IVDS System ............................................. 1

Figure 2.1: Assumed Host-User Channel Model ............................................................ 8
Figure 2.2: Diagram of Inverse Filter ............................................................................ 18
Figure 2.3: Efficiency Comparison of Distance Calculations ..................................... 33
Figure 2.4: Selected Approach to Solving our IVDS Problem ................................ 35
Figure 2.5: Host Side Preprocessing and Parameterization Approach .................. 36
Figure 2.6: User Side Preprocessing and Parameterization Approach ................. 37

Figure 3.1: Entropy of $X$ for various probabilities $p$ ............................................. 56

Figure 4.1: Power Spectral Density of a Realization of Room Noise .................... 68
Figure 4.2: Standard Bandpass Filter Frequency Response ....................................... 69
Figure 4.3: IVDS System Performance for Various Windows ............................... 72
Figure 4.4: IVDS System Performance for Various Preemphasis Filters ................ 75
Figure 4.5: Bandpass Filter #2 Frequency Response .............................................. 78
Figure 4.6: Bandpass Filter #3 Frequency Response .............................................. 79
Figure 4.7: IVDS System Performance for Various Bandpass Filters ..................... 79
Figure 4.8: Maximum Correlation Difference for Various Model Orders - BPF Test #1. 82
Figure 4.9: Maximum Correlation Difference for Various Model Orders - BPF Test #2. 83
Figure 4.10: IVDS System Performance for Different Amounts of Frame Overlap .... 86
Figure 4.11: Max Correlation for Correct Commercial for Each Overlap ................ 88
Figure 4.12: Performance for Different Coefficient Representations ................. 89
Figure 4.13: IVDS Performance Using Different Parts for the User Vector ........... 92
Figure 4.14: IVDS System Performance for Different Numbers of User Vectors ... 94
Figure 4.15: IVDS Performance for Different Amounts of User Vector Spacing .... 97
Figure 4.16: IVDS Performance for Differing Numbers of Bits/Coefficient .......... 99
Figure 4.17: IVDS Performance for Different Distance Measures ......................... 101
Figure 4.18: Distributions of Odd Indexed Reflection Coefficients in the Database ... 103
LIST OF TABLES

Table 4.1: Maximum Correlation Difference Statistics for Windowing Tests .................. 73
Table 4.2: Maximum Correlation Difference Statistics for Preemphasis Tests .................. 76
Table 4.3: Maximum Correlation Difference Statistics for Bandpass Filter Test .............. 81
Table 4.4: Maximum Correlation Difference Statistics for (409-6736) Hz BPF ................... 84
Table 4.5: Maximum Correlation Difference Statistics for (804-7112) Hz BPF ................... 84
Table 4.6: Maximum Correlation Difference Statistics for Database Overlap Test ............. 87
Table 4.7: Maximum Correlation Difference Statistics for Coefficient Representations .......... 91
Table 4.8: Maximum Correlation Difference Statistics for Coefficient Importance Test .......... 93
Table 4.9: Maximum Correlation Difference Statistics for Number of User Vector Test 95
Table 4.10: Maximum Correlation Difference Statistics for User Vector Spacing Tests ....... 97
Table 4.11: Maximum Correlation Difference Statistics for the Quantization Tests .......... 100
Table 4.12: Maximum Correlation Difference Statistics for Distance Calculation Tests 102
Table 4.13: Statistics for Various Coefficients in Vectors in Database ......................... 102
Table 4.14: Maximum Correlation Difference Statistics for the 'Optimal' System .......... 104
1. INTRODUCTION

1.1. Problem Statement

A brief description of interactive video data service (IVDS) systems is given in this section, as well as a general overview of how such systems operate. A set of requirements is then described for a particular user-initiated, noninvasive application. A high level block diagram of an IVDS system is shown in Figure 1.1.

![High Level Block Diagram of an IVDS System](image)

Figure 1.1 - High Level Block Diagram of an IVDS System

The overall purpose of an IVDS system is to provide the user with a wireless return path to the host site. The information being broadcast from the host site can vary. However, it is assumed here that the host site is broadcasting some type of advertisement or commercial, in which case the user's return path is for communicating an interest to the
host site in obtaining a particular product, service, or information about a product. The host site is made aware of the interest of the user when a transmission from the user is received. After the user's transmission has been obtained, decoded, and analyzed, the host site notifies the appropriate company or organization of an order for a particular product by the particular user. Note that in the block diagram of Figure 1.1 only the return path of the IVDS system is shown.

An identification process needs to take place, the goal of which is to identify the correspondence of a user's transmitted parameters with a particular commercial. Because of the desire to keep the hardware costs to a minimum at the user side of the IVDS system, very high speed hardware at the user side is not an option. So, for computational reasons the identification is best done at the host site, as shown in Figure 1.1, because at the user site there is not enough processing power or memory available.

The design of the particular IVDS system of interest is subject to very specific constraints and requirements. For a user initiated system, the user needs to be able to order the product or information at any random time in the interval during which the commercial is being viewed by the user. Please note that this may be at the time of the actual broadcast, but it does not have to be; the commercial could have been recorded, for example on a video cassette recorder (VCR). A main requirement of the IVDS system is to be noninvasive, that is it has no physical connection or contact with the television. Therefore, the only access to the broadcast signal is through the audio from the speaker of the user's television, which is also illustrated in Figure 1.1. Another major requirement is
that there be no modification or coding of the television signal before it is transmitted.

There are many tradeoffs involved in finding solutions to the above IVDS problem. The necessary bits per transmission, speed of commercial identification, and commercial identification success rate are tradeoff factors in our IVDS application. The number of bits that can be used per user for identification of a commercial, due to practical implementation issues, was specified to be no more than 200. The speed of identification of a particular technique will be considered in terms of the number of computations per user per 30 second commercial so that one technique may be compared with another. Also, the performance in terms of commercial identification, due to the limited number of commercials in the database, was assumed to be acceptable only if the system could identify the proper commercial 100% of the time with some margin of safety in a reasonably noisy environment.

1.2. Purpose of this Thesis

This thesis deals with the research efforts aimed at developing a noninvasive, and user initiated, interactive television system. Transmitter and controller development are not considered here. Instead, this thesis focuses on the audio processing portion of the system. The other parts of the system will be referred to only as needed in the discussion of the audio processing development. It will be assumed that, due to proper modulation, channel coding, and repeater placement, the uncorrected bit errors resulting from noise in the return channel will be negligible.
1.3. Overview

One requirement of the system is that it use no more than 200 bits per user transmission to characterize the broadcast audio signal. Therefore, the coding process must involve deriving a small number of parameters from a section of audio at the user site, and implies a kind of coding of the broadcast audio signal. After the broadcast audio signal parameters have been transmitted and received, they need to be compared with a set of parameters that was derived from the original commercial recordings and stored in a database at the host site. The goal is to determine the segment of audio in the database that corresponds to the user transmitted parameters, which implies that a certain type of audio identification technique needs to be employed. In Chapter 2 several general approaches are discussed to the coding and identification of audio in the context of the problem that has been described in Section 1.1. Also, at the end of Chapter 2 a high level description of the selected approach is presented. The details and theory behind the selected approach are given in Chapter 3.

Test procedures and partial results will be given in Chapter 4. Conclusions will be presented in Chapter 5, which will include comments on feasibility as well as on future work that can be done in applications similar to our IVDS application.
2.0 GENERAL APPROACHES

In this chapter various audio coding approaches as well as audio identification approaches will be discussed in the context of the IVDS problem that was described in Section 1.1. The first section gives details about the assumptions made about certain aspects of the problem. In Section 2.2 we discuss various audio coding or compression techniques and their suitability for our application. In Section 2.3 we discuss two aspects of audio identification. These are feature selection and distance measurement. Various approaches are discussed and a comparison is made between them. The last section of this chapter contains a high level description of the approach that has been selected to solve our IVDS problem.

2.1. Assumptions Made About the Problem

An attempt was made to keep the number of assumptions about the IVDS problem to a minimum. However, because the solution to the problem must work for a variety of channels and inputs, certain assumptions needed to be made. The first part of this section discusses the assumptions made about the channel from the host side to the microphone of the user unit. The second part of this section discusses the assumptions made about the sound content of the inputs to the channel (the commercials).

2.1.1. Channel Assumptions

For the audio coding portion of the IVDS system, the channel is comprised of the
The RF communications path between the host side and the user’s television.

2. The electrical path through the television from the receiver to the speaker, which may also include the electrical path through the VCR, if one is used to record the signal for playback at a later time.

3. The acoustic path from the speaker of the television to the microphone of the user unit.

The broadcaster uses frequency modulation (FM) to broadcast the audio portion of the signal over the first portion of the channel (from the host side to the user’s television). The requirement by the FCC on the maximum deviation for peak modulation is ±25 kHz. The transmission is capable of producing an audio bandwidth of 50 to 15,000 Hz. The reception of a broadcast is affected by and will depend on the distance, structures, topology, and atmospheric conditions between the host side and the user’s television [1]. We have assumed that under normal operating conditions the noise introduced in the RF portion of the channel, in comparison to other portions of the channel, is negligible. We assume that the video component of the transmitted signal will be corrupted before the audio component, so that the user could not watch the television if the audio was significantly degraded.

The path through the television is dependent upon the type of television as well as when it was manufactured and will surely differ from television to television. Newer and
better models of televisions and VCRs will have a flatter frequency response than the older and cheaper models. We will assume that in the worst case scenario the frequency response of the channel from the receiver to the speaker of the television is flat up to 7.5 kHz, which is an important assumption as it influences the choice of the sampling rate for the IVDS user unit. The noise introduced by the television set includes thermal noise, shot noise and flicker noise [1], all of which are normally insignificant and definitely negligible in comparison to the noise introduced in the last part of the channel.

The last part of the channel is from the speaker of the television to the microphone of the user unit. There will usually be a considerable degradation of the signal-to-noise ratio (SNR) as the signal passes acoustically through the air in this portion of the channel. There are no real a priori assumptions that can be made about the character of the noise introduced in the acoustic portion of the channel. It is different from user side to user side and depends on the acoustic environment in which the user’s television is placed. The noise could be white and Gaussian in nature, interfering speech, background music, or a wide variety of other sounds. Most of the tests were conducted with a low pass type of noise that had a white Gaussian characteristic over the frequencies of interest.

As mentioned in Section 1.1, it is assumed that the proper channel coding, repeater placement, modulation technique and transmitter power will be used, so that the uncorrected bit errors induced by the channel from the user unit back to the host side will be negligible.
A model for the channel based upon the assumptions is now shown in Figure 2.1.

Figure 2.1 - Assumed Host-User Channel Model

2.1.2. Input Assumptions

The inputs can be considered to be the commercials that are broadcast from the host side. The commercials may be comprised of audio components that are very different in nature. The majority of components can in general be classified as either speech, music, or silence. It is impossible to know a priori how much of each of these components to expect and when in the commercial to expect them. In addition to this complication, there are certain audio components that will comprise some commercials that will not fall into any of the three categories listed. However, we assume that the commercial is mostly comprised of speech and/or music or sounds that can be modeled in a similar manner.

In the IVDS problem that is presented in Section 1.1, the commercials in the
database are seen as a finite set of realizations of random processes. It is assumed that the majority of these realizations are mainly composed of linear combinations of speech and music. Since speech and music are random processes, a linear combination of them is also a random process, but the resultant process has greater complexity. We assume the processes that we are dealing with are nonstationary, because of the dynamic nature of speech and music over a long term interval. However, when we are looking at short time periods of each realization we can assume it to be a realization of a process that is wide sense stationary and correlation ergodic, if the interval is small enough. We will assume the length of this small interval to be about 20 ms [2]. We can model the short term input to the channel as an autoregressive (AR) process. Speech has been successfully modeled as an AR process for years [3]. Voiced speech is especially well suited for the AR model, because of its spectrally 'peaky' nature. The spectrum of most music has stronger spectral peaks than speech and therefore also tends to fit the AR model. Therefore, overall we can reasonably assume that we are dealing with an AR process, since linear combinations of two AR processes will fit the AR model. This can be demonstrated using the linearity property of the z-transform. Due to the 200 bits per user transmission limitation, we will plan on a limited number of spectral peaks being present in the signal and therefore a limited number of speakers and/or instruments at any one time. This will be discussed in Section 3.3, which deals with the subject of order estimation.
2.2. Coding Approaches

There are many different techniques for coding wideband speech and/or audio [4, 5, 8] and speech confined to the 300-3000 Hz range [6]. The major difference between our IVDS application and the majority of speech and audio coding applications is the fact that most applications require resynthesizing of the coded speech and/or audio at the receiving end of the system. However, our IVDS application involves a matching of the coded audio with a particular segment of audio at the receiving end of the system. No synthesis at the receiving end is necessary.

There are two basic approaches to the coding of speech and audio. They are waveform coding and vocoding. These two approaches have different goals and are based on different theories. We will discuss both of these approaches, focusing on the advantages and disadvantages of each method in the context of the IVDS problem that was presented in Section 1.1. The discussion of these techniques will be kept general and an emphasis will be placed on the philosophies of each method as well as on the major advantages and/or disadvantages of each technique in the context of this application.

2.2.1. Waveform Coders

Waveform coding is a technique that is concerned with preserving the waveform of the source as accurately as possible. There are many coding approaches that fall into this general classification [9]. The more popular of these methods include Adaptive Delta Modulation (ADM), Adaptive Differential Pulse Code Modulation (ADPCM), Adaptive
Predictive Coding (APC), SubBand Coding (SBC), and Adaptive Transform Coding (ATC).

Waveform coding usually results in a high quality synthesized waveform at the receiving end at the cost of a higher bit rate. Waveform coders can be divided into two general categories. They are time domain waveform coders, such as ADM and ADPCM, and frequency domain waveform coders, such as SBC and ATC. In the following two sections we will discuss the disadvantages and advantages of these two types of waveform coders in the context of our IVDS problem.

2.2.1.1. Time Domain Waveform Coders

Time domain waveform coding is a technique that is concerned with accurately preserving the sampled waveform as a series of discrete amplitudes over a finite time interval. If this technique were used in our IVDS application, the speech samples themselves would be the features for the identification process. This is a major disadvantage because speech samples do not make good features for two reasons. The first reason deals with the restriction of the interval of the signal in the time domain over which the features are obtained. The second reason is because of the lack of orthogonality of the feature vectors in the database. We will now discuss these two disadvantages of using speech samples as coded features in this application.

First of all, because the sampling rate of the system is 16 kHz, a 20 ms segment of speech consists of 320 samples. ADPCM is capable of providing a bit rate in the range of
40 kbps down to 24 kbps in the usual applications, where the coded speech is
resynthesized on the receiving end of the system. As bit rates less than 24 kbps are
attempted, the quality quickly decreases [18]. At the coding rate of 24 kbps, we would
need

\[
\frac{24\text{bits}}{1\text{frame}} = \frac{480\text{bits}}{0.020\text{s}}.
\]

ADPCM would require more than double the maximum allowable number of bits to code
a section of audio that is just one phone long. Therefore, the segment of the signal over
which the features would be calculated would not be very long. The ADM technique
yields a lower bit rate but not low enough to be used in our application. It provides a rate
of about 1 bit per sample. This would require 320 bits per 20 ms frame, which is more
than the limit per user transmission. The unknown noise environment can contain some
strong impulsive noise that has a chance of occurring during a single 20 ms frame, which
makes the sampled data at that point of very little value. The host side would not have
much accurate information to identify the user's audio from. It would be advantageous to
choose features that can represent the commercials over long time intervals with as low a
number of bits as possible, even if the representation is not as accurate a representation
over the interval. There are other techniques that we can use to represent the signal with a
relatively low number of bits over a relatively long time interval. We will discuss the
techniques and the ways in which these techniques can be used for our IVDS application in the following sections.

Another problem is the fact that there may be many 20 ms frames in the database that are similar to one another (repeated occurrences of the vowel /a/, for example). This similarity makes it difficult to distinguish between them, especially when the user has a set of features from only one 20 ms frame of the commercial that, in addition, has been corrupted by noise and distortion. This problem of feature vector similarity will be discussed in greater detail in Section 2.3.1.

2.2.1.2. Frequency Domain Waveform Coders

This type of waveform coding includes SBC and ATC. Frequency domain waveform coding is a technique that involves decomposing the source signal into different frequency bands or subbands and coding the signal in each of these subbands individually. Waveform coding, such as (APCM), is done on either the time domain waveform or the frequency representation of the signal in each subband. Frequency domain waveform coding is better for us than time domain waveform coding, because it allows us to focus on the frequency band that we predict to contain the most information and/or the least noise. In this manner, we can reduce the number of bits used to code each frame of the signal, by coding only the important and/or least corrupted features of each frame.

For the variation of this technique that involves coding of the frequency domain, the number of bits can be reduced because we can use bandpass filtering to eliminate
corrupted and unimportant portions of the commercials. This will be discussed in detail in Section 3.2.1. The idea of subband coding is beneficial to us in our IVDS application. It enables us to increase the likelihood of an acceptable SNR by eliminating the subbands of the received signal that are likely, according to our assumptions about the channel and inputs to the channel, to have a low SNR. However, there is a technique that can produce coefficients that more concisely describe the spectral content of the signal and therefore further reduce the number of bits needed per frame. This technique is called linear prediction and is used in many types of vocoders [20, 21, 22, 23, 19]. This operation produces good features for our IVDS application. Many different transformations of these parameters are used successfully as features in many speech recognition algorithms [25]. The theory and benefits of linear prediction in the context of this application will be discussed in more detail in Section 2.2.2.2.

2.2.2. Vocoders

There are a number of different types of vocoders [2], [6]. Most vocoders are based on the model of human speech production that was developed in the early 1900s [17]. The goal of vocoders in general is to characterize individual frames of the sound obtained from the source and to reproduce the sound as accurately as possible at the receiving end of the system. Parameters are obtained from each frame of speech and can be viewed as coefficients of the short-term transfer function of the vocal tract. In Section 2.2.2.1 we will discuss a variety of vocoder approaches in the context of their application to our
IVDS problem. We will then describe the linear prediction vocoder in more detail in Section 2.2.2.2.

2.2.2.1. Vocoder Approaches

In this section we will describe some of the problems and benefits of some vocoder approaches. The channel, cepstral, and formant vocoders will be discussed. Linear predictive vocoders will be discussed in the next section.

Channel vocoders have an advantage that was noted in Section 2.2.1.2 in the discussion of the subband coder. Bandpass filters are used to focus on individual parts of the spectrum. This technique is advantageous in our IVDS application. However, we do not need to retain every section of the spectrum, as will be demonstrated in Chapter 4, and because we are coding audio and do not need to resynthesize the signal on the host side, calculating and transmitting voicing information does not really make sense.

In this application the cepstral vocoder will have similar problems as the frequency domain waveform coders, although the overall philosophy is quite different. The cepstral vocoder codes the signal in the quefrency domain in a manner similar to the way the frequency domain waveform coders code the signal in the frequency domain. This technique will generate too many features per 20 ms frame of audio.

The formant vocoder is similar to the channel vocoder except its goal is to estimate only the most dominant peaks in the spectrum. It attempts to model each of the spectral peaks as a two-pole filter. This method of estimating only the dominant peaks in the
spectrum is a good idea in our application, which allows us to focus on a few important features of the spectrum of the signal within each frame. By using fewer bits per frame we can use more frames and therefore take advantage of the long term nonstationarity of the commercials. The importance of exploiting the nonstationarity property of the commercials can be seen from the tests that are described in Sections 4.3.8 and 4.3.9. There is an accurate means for estimating spectral peaks. It is called linear prediction and will be discussed in the following section.

2.2.2.2. Linear Predictive Vocoder

Linear prediction has been used in the modeling of speech for many years [3]. Although our IVDS application involves signals that are composed of more than just speech, the method of linear prediction can still be used successfully to solve our IVDS problem, according to the assumptions in Section 2.1.2. Many vocoders use linear prediction (LP) to estimate the coefficients of the denominator polynomial in the system function of an all-pole process, which can be represented by the following equation.

\[
\hat{\Theta}(z) = \frac{1}{1 - \sum_{i=1}^{M} \hat{a}(i)z^{-i}}
\]  

(2.1)

However, the real process being modeled has poles and zeros, which can be represented by the following equation.
\[
\Theta(z) = \Theta_0 \frac{1 + \sum_{i=1}^{L} \hat{b}(i)z^{-i}}{1 - \sum_{i=1}^{R} \hat{a}(i)z^{-i}}
\]  

(2.2)

The system \( \Theta(z) \), which is a causal rational system, can be decomposed in the following manner:

\[
\Theta(z) = \Theta_0 \Theta_{\text{min}}(z) \Theta_{\text{ap}}(z)
\]  

(2.3)

where \( \Theta_0 \) is a constant, \( \Theta_{\text{min}}(z) \) is a minimum phase component, \( \Theta_{\text{ap}}(z) \) is an all-pass component. The minimum phase equivalent of the system is represented by (2.1). This is due to the fact that any first order polynomial of the form \( 1 - z_0 z^{-1} \) can be represented by an all-pole system function

\[
\frac{1}{1 + \sum_{k=1}^{\infty} z_0^k z^{-k}}
\]  

(2.4)

where (2.4) indicates a theoretically infinite number of terms. However, the all-pole model is chosen to have an order similar to the expected order of the source, which in practice for speech and most audio signals is much less than the frame length. When the order is chosen correctly and the prediction error sequence \( e(n) \) is orthogonal, where the setup is shown in Figure 2.2, then the estimates \( \hat{a}(i) \) can be calculated by

\[
\sum_{i=1}^{\infty} \hat{a}(i)r_s(\eta - i) = r_s(\eta), \quad \eta = 1, 2, \ldots, M
\]  

(2.5)
to perfectly model the minimum-phase component of $\Theta(z)$ [2]. The variable $r_s$ corresponds to the autocorrelation sequence that is derived from signal $s$. There are problems with underestimating the order of the model as well as overestimating it. The tests that are related to model order estimation are described in Section 4.3.4. The results of these tests are also given in Section 4.3.4. However, at this time it would be good to note and discuss other limitations of the linear predictive vocoder approach to extracting features from commercials.

![Diagram of inverse filter](image)

Figure 2.2 - Diagram of inverse filter

Note that the $\hat{a}(i)$ parameters that are extracted from the commercial can be used only to represent the minimum phase portion of the signal. Therefore, if the only difference between two signals is in the time domain, we will not be able to detect the difference and the best identification algorithm that we may develop would be unable to distinguish one frame from the other. Our assumption is that by acquiring several frames of the commercial that are displaced in time, as described in Section 2.4.2, we can overcome this problem.
The assumption of $e(n)$ being orthogonal is valid in the case of silence, unvoiced speech, and low pitch voiced speech. For the case of high pitch voiced speech, the glottal pulse does not get a chance to decay before the next pulse. This causes problems that are very much similar to the problems encountered in the case where the order is underestimated. The number of spectral peaks can vary greatly from commercial to commercial, and therefore at times we will be either underestimating the model order or overestimating it. The effects of this will be discussed further in Section 3.3.

2.3. Identification Approaches

In this section we will discuss several approaches to identification. One of the most basic and fundamental selections that must be made in the development of any identification algorithm is the selection of the features of the signal that are important and convenient and that can be used to compare between signals. This topic will be discussed in Section 2.3.1. The difference between the user's transmitted vector of features and the vectors in the database can be measured by determining the distance between them. Different methods for calculating this distance will be discussed in Section 2.3.2. The use of correlation for identifying signals will be discussed in Section 2.3.3.

In this section we will discuss the differences between the identification that must be done in our IVDS application and typical signal identification applications. The two main criteria for the selection of an identification algorithm in an application such as the one presented in Section 1.1 are the time needed to identify the signal and the success rate of
the algorithm. The goal would be to get the identification algorithm to operate correctly every time with the identification process causing a minimum of delay.

2.3.1 Differences from Typical Identification Applications

Much research has been done on a variety of techniques for speech recognition [2] and speaker verification [12], such as dynamic time warping (DTW), the Hidden Markov Model (HMM), and the Artificial Neural Network (ANN). However, these techniques are too computationally intensive for our IVDS application and HMMs and ANNs involve a potentially lengthy training procedure. Our IVDS application actually falls more under the category of ‘speaker’ (commercial) identity verification. In our IVDS application it will be unknown a priori whether the current ‘speaker’ is a single speaker, multiple speakers, and/or music. This difference from the usual applications, where only one speaker is involved, makes our IVDS problem more difficult. Other disadvantages include the fact that the data to be identified must be transmitted with a restricted number of bits, while the database is very large. Only 200 bits can be used per signal. This is much less than in the majority of current applications. Also, the acoustical environment around the user unit, as shown in Figure 2.1, could be very noisy and the statistical character of the noise that will be introduced along the broadcast channel, also shown in Figure 2.1, is unknown.

However, there is one major advantage that this application has over the traditional applications of speech recognition, speaker verification, and speaker identification. In our IVDS application the test signal (signal to be identified) is produced at very close to the
same speed as the reference signal (signal in the database). Some techniques, such as DTW, are frequently used to expand or compress the time scale of the test signal. However, this technique would be unnecessary and wasteful in this application. Another advantage is the fact that if no noise is present in the channel from the host side to the microphone of the user unit and the signal does not get distorted in the channel, the reference signal will be the test signal. So, during those times when the channel is quiet, we have a better situation to work with than in most applications, where the reference signal must be reproduced by a human vocal tract and is therefore going to be different.

2.3.2 Feature Selection

The decision about which features of the signal we should extract and use to represent the signal is of utmost importance in this application. It is not easy to restore important information that is removed in the analysis phase. The selection of features to represent the signal will influence the ultimate, overall success rate. There are many factors and restrictions that influence the feature selection. The most stifling restriction is the bit rate requirement. In the discussion of the various coding techniques we mentioned various possible features that could be used in this application. They included:

1. The speech samples themselves,

2. The Fast Fourier Transform results,

3. The complex or real cepstrum results,

4. The LP coefficients and their equivalent representations,
as well as a few others. The conclusion of the discussion in Section 2.2 was that from a
coding efficiency point of view the linear predictive coefficients were the best. The choice
of the representation of these coefficients is subject to the tests and results shown in
Section 4.3.6. The variance of each coefficient is of importance in the selection of a
proper representation for the coefficients. This topic will be discussed more thoroughly in
Section 3.4.

2.3.3 Measuring the Distance Between Feature Vectors

There are many types of distance measures for calculating the distance between two
vectors [2, 25, 27]. We will discuss a variety of these measures and comment on their
suitability for this application. Correlation can also be viewed as a distance measure.
However, we will delay discussion of this measure until Section 2.3.4.

2.3.3.1 Euclidean Distance Measure

One distance measurement is the Euclidean distance, which is used quite frequently
in the analysis of engineering problems. However, the Euclidean distance measure does
not perform well when the components of the vector are highly correlated [2], and the
coefficients in a vector of LP coefficients are very correlated, as can be seen from a study
of the Levinson-Durbin algorithm in Appendix A. Also, the Euclidean distance measure
will yield 'natural' results only if the representation of the vectors is based on an
orthonormal basis. Weighting the Euclidean distance measure helps to improve the
performance of this technique. Weighting involves multiplying by a matrix that
decorrelates the components of the feature vectors. The weighting philosophy is used in some of the measures discussed in the following sections.

2.3.3.2 Cepstral Distance Measure

The cepstral distance measure attempts the weighting approach that is mentioned in the previous section. The weighted cepstral distance is given by

$$d_{2*}[\tilde{c}(m), \tilde{c'}(m)] = \sqrt{\tilde{c}(m) - \tilde{c'}(m)}^T \Lambda^{-1} \tilde{c}(m) - \tilde{c'}(m), \quad (2.6)$$

where the $c'$s are the vectors of cepstral coefficients and $\Lambda$ is a diagonal matrix of the variances of the individual coefficients. The cepstral distance measure takes $2(D+1)^2 + 2D + 16D + 1$ operations per calculation, assuming that the square root can be performed by performing a table lookup procedure taking one instruction cycle and the inverse calculation takes about sixteen cycles per element on the diagonal. The length of the feature vectors is represented by $D$.

2.3.3.3 Itakura Distance Measure

This distance measure is the most popular method for calculating the similarity between LP vectors for speech. It was developed for use with DTW. The measure is calculated by

$$d_I[\hat{a}(m), \hat{b}(m')] = \log \frac{\beta^T(m') \hat{R}_I(m) \beta(m')}{\alpha^T(m) \hat{R}_I(m) \alpha(m)}, \quad (2.7)$$
where $\mathbf{\tilde{R}}_s(m)$ is the augmented autocorrelation matrix, and it is defined as

$$
\mathbf{\tilde{R}}_s(m) = \begin{bmatrix}
    r_s(0,m) & r_s^T(m) \\
    r_s^T(m) & R_s(m)
\end{bmatrix},
$$

(2.8)

where $r_s(m)$ represents the autocorrelation of the reference signal for the frame ending at time $m$. $R_s(m)$ is the autocorrelation matrix of the reference signal, which is defined as

$$
R_s = E\{SS^T\},
$$

(2.9)

where $s$ contains samples of the reference signal, and $E\{\}$ is the expected value operation.

Also, $\alpha(m)$ and $\beta(m')$ are defined as follows

$$
\alpha(m) = \begin{bmatrix} 1 & -\tilde{\alpha}^T(m) \end{bmatrix}^T, 
$$

(2.10)

$$
\beta(m) = \begin{bmatrix} 1 & -\tilde{\beta}^T(m') \end{bmatrix}^T, 
$$

(2.11)

where $\tilde{\alpha}^T(m)$ and $\tilde{\beta}^T(m')$ are the vectors of LP coefficients for comparison. The vectors $\tilde{\alpha}^T(m)$ and $\tilde{\beta}^T(m')$ do not necessarily need to be derived from the same frame of audio.

Hence one of the arguments is primed. Therefore, the measure takes $4(D+1)^2 + (D+1) + 8 + 1$, where the log calculation will be a table lookup that takes about one instruction cycle, and the division will take about eight instruction cycles using the iterative convergent divide algorithm [31]. The length of the vectors being compared
is $D$. The measure compares the prediction error of the test vector to that of the reference vector by calculating the ratio of the two quantities.

2.3.3.4 Mahalanobis Distance Measure

The Mahalanobis distance measure is also called the Itakura-Saito distance measure, and it can be written as

$$d_m[\hat{a}(m), \hat{b}(m')] = \frac{[\alpha(m) - \beta(m')]^T \hat{R}_x(m) [\alpha(m) - \beta(m')]}{\alpha^T(m) \hat{R}_x(m) \alpha(m)}$$  \hspace{1cm} (2.12)

The number of operations per distance measure is $2(D+1)^2 + (2D+1) + 8$, where $D$ is the length of $\hat{a}(m)$ and $\hat{b}(m')$. It takes about eight instruction cycles to perform the division using the iterative convergent divide algorithm [31].

2.3.3.5 Correlation as a Distance Measure

The identification algorithm that was developed for use in this application involves correlating the user’s feature vectors with those in the database and then comparing the correlation coefficients that are produced. The identification technique is described in detail in Section 2.4. The purpose of this section is to give a description of what the correlation coefficient represents as well as some of the theoretical benefits and problems that we can encounter when using this measurement in our IVDS application. As shown in Section 2.3.3.6, the distance measures discussed in this section are the fastest estimators of the ones discussed in Section 2.3.3. Therefore, we thoroughly examine them.
There are several different distance measures that we could use to compare one vector to another, some of which have already been discussed. However, in this thesis we simulate the use of only three different measures in our IVDS system. The inner product of the received vector and a set of vectors that might have been transmitted is usually calculated at the receiving end of digital communication systems to identify which signal out of a finite set of signals was transmitted [24]. The three distance measures that we will look at all use the inner product to calculate the angular distance between vectors in \( \mathbb{R}^D \), where \( D \) is the dimension of the vector space in which the vectors reside. However, some of these measures are more direct than others. One of these measures of the distance between the vectors \( \mathbf{x} \) and \( \mathbf{y} \) is given by

\[
\rho_1(\mathbf{x}, \mathbf{y}) = \frac{\left( \mathbf{x} - (\mu_x \otimes 1) \right)^H \left( \mathbf{y} - (\mu_y \otimes 1) \right)}{\left\| \mathbf{x} - \mu_x \right\| \left\| \mathbf{y} - \mu_y \right\|},
\]

where \( 1 \) is a vector of all ones with a length equal to the length of vectors \( \mathbf{x} \) and \( \mathbf{y} \).

Also, \( \mu_x \) and \( \mu_y \) are the means of the vectors \( \mathbf{x} \) and \( \mathbf{y} \) respectively and are defined as

\[
\mu_x = \frac{1}{D} \sum_{i=1}^{D} x(i)
\]

(2.14)

and

\[
\mu_y = \frac{1}{D} \sum_{i=1}^{D} y(i),
\]

(2.15)
where \( D \) is the number of components in vectors \( \mathbf{x} \) and \( \mathbf{y} \). Therefore, \( \mu_x \) and \( \mu_y \) are scalars. Equation (2.13) is equivalent to the standard definition of the correlation coefficient, and it can be proved that \( 0 \leq |\rho| \leq 1 \) [28]. The measure given in (2.13) allows us to measure the linear relationship between the bivariate pairs of random data \( (x(i), y(i)) \). Notice that in our IVDS application, where \( \mathbf{x} \) and \( \mathbf{y} \) are vectors of reflection coefficients (derived as shown in Appendix A), subtracting out the mean takes away any constant bias that the room noise might add to the vector of reflection coefficients. Kay has done experiments and documented the effects of noise on the autoregressive spectral estimator [11]. The results showed that, for a sinusoid in white noise and a constant model order, the radius of the poles of the model increased as the SNR decreased. Kay showed that the values of the reflection coefficients are proportional to the radius of the poles of the model. Therefore, we can see that white noise can generate a bias in our AR parameters and reflection coefficients. Therefore, at least in the case of white noise, subtracting the mean of the vector from each of the components in the vector as done in (2.13) should have some benefit.

A second measure \( \rho_2(\mathbf{X}, \mathbf{Y}) \) can be used to calculate the distance between the vectors \( \mathbf{X} \) and \( \mathbf{Y} \) and it can be defined as

\[
\rho_2(\mathbf{X}, \mathbf{Y}) = \frac{\mathbf{X}^T \mathbf{Y}}{||\mathbf{X}|| ||\mathbf{Y}||}.
\]

(2.16)

This measure does not subtract the mean of the vector from each of the components.
Equation (2.16) is equivalent to (2.13) if we let

$$X(i) = x(i) - \mu_x \quad \text{for } i = 1, 2, \ldots, D \quad (2.17)$$

and

$$Y(i) = y(i) - \mu_y \quad \text{for } i = 1, 2, \ldots, D \quad (2.18)$$

The distance measure in (2.16) is related to the cosine of the angle between the vectors $X$ and $Y$ in a $D$ dimensional vector space by

$$\rho_2(X, Y) = \cos(\Theta), \quad (2.19)$$

where $N$ is the number of components of each vector.

The correlation coefficient closest to one will be produced by the vector in the database that forms the smallest angle with the user's feature vector. Therefore, the vector in the database that produces the largest value of $\rho_2$ is in one sense closest to the user observation vector. Also, it can be said that a $\rho_1$ of 0 means that $\underline{x}$ and $\underline{y}$ are uncorrelated or in other words, they are not linearly dependent at all. However, a $\rho_1$ equal to 1 means that the vectors $\underline{x}$ and $\underline{y}$ are perfectly correlated or perfectly linearly dependent.

If we use (2.13) to calculate the correlation coefficient in this application, then $\|y - \mu_y\|$ and $\mu_y$ for each vector in the database could be calculated and stored at the host side and be ready for retrieval. Therefore, in terms of the complexity of this approach, the
calculation for each $\rho_1$ would consist of:

$\mu_\hat{z}$ takes $D$ additions + 1 multiply

$\|x - \mu_\hat{z}\|$ takes $D$ additions

+ $D$ multiply-accumulates (MACs)

+ 1 operation for the square root computation

$D$ additions for $x(i) - \mu_\hat{z}$

Numerator takes $D$ MAC operations

8 operations for the division calculation

So, the total number of instructions for the calculation of $\rho_1$ would be $5D + 10$, assuming that we use a lookup table for the square root calculation, which will only take one operation. However, if we desired extra precision we could use the iterative convergent divide algorithm, which takes about eight instruction cycles to perform the division [31].

In the calculation of $\rho_1$, we subtract the mean of each vector's components from each of the components of the corresponding vector. This is done to make sure that a constant additive factor between the user observation vector and the reference feature vector does not negatively affect the correlation measurement. Also, the correlation result is divided by the product of the standard deviations of each vector. The division is done to make sure that if one vector is proportional to another but has been amplified, the measurement will not be affected negatively. Both of these operations help to make the measurement robust against certain types of noise.

A disadvantage of the correlation coefficient is that it does not take into account
noise and distortion that can alter the feature vectors in a nonlinear fashion. Various types of distortion and noise, such as fading and interfering speech, that occur over the broadcast channel from the host side to the user are discussed in Section 2.1.1 and shown in Figure 2.1. Therefore, even if the user observation vector has a nonlinear dependence on one of the vectors in the database, the correlation coefficient will produce a low value, although the vectors may be quite dependent.

Although the idea of correlation is used in digital receivers, there are limitations to its usage here. The designer of a digital communication system has the freedom to choose the signals that the system will transmit, in order to have maximum distance properties. In the case of QPSK, this means that the signals in the vector space are orthogonal to one another. So, the noise has to be very large to change one signal into another. However, in our IVDS application there are many more vectors than dimensions in the vector space where the vectors reside, and therefore the vectors can not all be orthogonal to one another. As a matter of fact, the angles or distances between many of the vectors will be quite small. So, it will not take much noise to make one feature vector appear close to another.

We hope to circumvent this problem in two ways. We can choose the representation for our vectors that gives them the best distance properties for some distance measure. Also, we can use several frames over a period of time. By computing multiple feature vectors over a period of time, we are attempting to force the user's commercial to match the ones in the database over a longer period of time.
The vectors in the database can be organized as arrays and are concatenated to form one long array that can be represented as

\[
\tilde{y} = \left[ \frac{k(m_d) - c}{k(m_d + (1 - v)m_d) - c} \right] \ldots
\]

\[
\left[ k(m_d + (E - 1)(1 - v)m_d) - c \left[ k(m_d + E(1 - v)m_d) - c \right] \right],
\]

(2.20)

where \(v\) is the overlap of the frames in the database expressed as a fraction, and \(c\) is defined as

\[
c(i) = \frac{1}{E} \sum_{j=1}^{E} y(i, j) \quad i = 1, 2, \ldots D,
\]

(2.21)

\(E\) represents the index of the last vector in the database, \(m_d\) is the length of one frame in time, and \(k(m_d)\) represents the vector or array of reflection coefficients that end at time \(m_d\). The underscore under and arrow over \(y\) in (2.20) indicate that it is an array of arrays. The \(i^{th}\) component of the \(j^{th}\) vector or array in the database can be written as \(y(i, j)\), and the \(j^{th}\) vector in the database is referred to as \(\tilde{y}(j)\). Note that we are using the terms vector and array interchangeably to refer to the segments of coefficients that come from the same frame and are grouped together as a unit in the database \(\tilde{y}\).

We tested another measure of the distance between two vectors \(x\) and \(y\). It is a modified version of the measure in (2.13). This measure is given by
\[
\rho_3(x, y) = \frac{\sum_{i=1}^{N} ((x(i) - c(i))(y(i) - c(i)))}{\sqrt{\sum_{i=1}^{N} ((x(i) - c(i))^2)} \times \sqrt{\sum_{i=1}^{N} ((y(i) - c(i))^2)}},
\]

(2.22)

where \( x \) could be any user observation vector and \( y \) could be any database vector, and \( c(i) \) is defined in (2.21). It will be shown in Section 4.3.11 that (2.22) is not as accurate a measure in our IVDS application as (2.13), but that if we use a low number of bits per coefficient in our system, the use of (2.22) will increase the performance.

If we use (2.22) to calculate the correlation coefficient in this application, then the vector \( c \) could be stored at the host side of the system and \( \sqrt{\sum_{i=1}^{D} ((x(i) - c(i))^2)} \) for each vector in the database could be calculated and stored at the host side and be ready for retrieval. Therefore, in terms of the complexity of this approach, the calculation for each \( p \) would consist of:

\[
\sqrt{\sum_{i=1}^{D} ((x(i) - c(i))^2)} \text{ takes } D \text{ additions}
\]

+ \( D \) multiply-accumulates (MACs)

+ 1 operation for the square root computation

\( D \) additions for \( x(i) - \mu_x \)

Numerator takes \( D \) MAC operations

8 operations for the division calculation

So, the total number of instructions for the calculation of \( \rho_3(x, y) \) would be \( 4D + 9 \).
assuming that we use a lookup table for the square root calculation that takes about one instruction. The computational requirements of (2.13) and (2.22) will be displayed in the following section.

2.3.3.6 Analysis of the Efficiency and Feasibility of the Distance Measures

A comparison of the computational cost of the distance calculations that have been discussed in this section is shown in Figure 2.3.

Figure 2.3 - Efficiency Comparison of Distance Calculations

Figure 2.3 gives the approximate number of instructions that would be required to perform a distance calculation between two individual vectors as the length of the vectors is varied. We see that the two methods that are based on the inner product are capable of
calculating the distance much more quickly than the other methods, especially for large vectors. The numbers shown in Figure 2.3 are approximate and they do not include the time required for fetches from external memory components.

2.4. System Description

A block diagram of the simulation that we set up to test the selected approach to solving our IVDS problem is shown in Figure 2.4. The only portion of the channel from the host side to the user unit that was included in the simulations was the acoustic path from the television speaker to the microphone as shown in Figure 2.4. This is because of the assumptions in Section 2.1.1 that were made about the channel.

The host side has access to each of the commercials in an uncorrupted form and in their entirety. The processing of these commercials and their addition to the database does not have to be done in real time. There is much more time allowed for this computation than for similar computations on the user’s side of the system. The user unit must be able to do its processing in 15 to 30 seconds. This is because the user may want to respond to the next commercial being broadcast.

2.4.1. Preprocessing and Parameterization at the Host Side

The commercials are stored in memory at the host side at a particular sample rate.
The approach that is used to preprocess and parameterize each commercial on the host side of the system is shown in Figure 2.5.

The commercial, which we can call $\mathbf{x}$ (where the underline indicates an array), is first broken up into frames that are 20 ms in length and a window $w(n)$ is applied to each
\[
f_x(n; m) = x(n)w(m - n),
\]

where \( m \) is the end time of the frame and \( n \) is the time index within the specified frame.

Figure 2.5 - Host Side Preprocessing and Parameterization Approach

After the commercial has been segmented and windowed, preprocessing filters are applied to condition the signal before parameterization. A variety of bandpass and preemphasis filters were tested. The reflection coefficients for each of the frames are then computed using the autocorrelation method of linear prediction via Levinson recursion [7] and are placed sequentially in the database. For instance, the reflection coefficients corresponding to the frame ending at \( m \) will be placed first in the array corresponding to that particular commercial. Then the frame ending at \( m + \text{overlap} \) is placed next in the array with the first coefficient of this frame following the last coefficient of the previous
frame, even though this frame really overlaps the previous one in time.

2.4.2. Preprocessing and Parameterization at the User’s Side

The processing that is done on the user side of the system is shown in Figure 2.6. Although the same procedures are performed on the user's side of the system, they are performed in a different manner. Whenever the user selects a particular commercial, the user unit samples 20 ms of audio and then waits a predetermined amount of time, which

![Interval Length Diagram](image)

Figure 2.6 - User Side Preprocessing and Parameterization Approach

we will call the interval length $L$ and then again samples 20 ms of audio. This process continues until $N$ frames have been acquired, where $N$ is much smaller than the number of frames in the database. Each frame is windowed with the exact same window that was used on the host side of the system. The same pre-filters are used to condition the signal as were used on the host side of the system. Then reflection coefficients are obtained using the same techniques as on the host side of the system. The coefficients are then
quantized and of course transmitted to the host side of the system.

2.4.3. Identification Technique

When a user's transmission is received, the host side begins correlating the user's transmitted coefficients $\vec{k}(n) \ n = 1, 2, \ldots, D$, where $D$ is the dimension of the vector, with those in the database $k$ and averaging the results that are obtained for each of the $\vec{k}$'s to produce the correlation coefficients $\vec{\rho}$ for each of the commercials in the database. The arrays of coefficients $\vec{x}$ and $\vec{y}$ are arrays of coefficients that can be divided up into vectors of length $D$. The method by which $k$ and $\vec{k}$ are obtained is described in Section 2.4.1 and Section 2.4.2 respectively. The distance between each of the individual user observations and a corresponding vector in the database is calculated first and then the results are averaged. Using the first method of calculating the correlation coefficients given in (2.13) yields the coefficients $\vec{\rho}_1$, which are calculated according to

$$\vec{\rho}_1(j; k) = \frac{1}{N} \sum_{i=1}^{N} \rho_z \left( \vec{x}(i), \vec{y}(j + (i - 1)I) \right) \quad j = 1, 2, \ldots, lp \quad (2.24)$$

where $k$ is the index of the commercial in the database, $lp$ is given by (2.28), $N$ is the number of vectors that are obtained at the user side of the system, and $I$ represents the number of vectors in the database per time interval between the user observation vectors, as computed by
where \( N_L \) is the number of samples between the start of each frame that the user samples, \( N_m \) is the number of samples per frame, and \( \nu \) is the amount of overlap expressed as a fraction. The variable \( \tilde{x}(i) \) corresponds to the \( i^{th} \) array or vector in \( \tilde{x} \), which is defined as

\[
\tilde{x} = \left[ \tilde{k}(m_u) - \mu_{\tilde{x}(m_u)} \right] \left[ \tilde{k}(m_u + L) - \mu_{\tilde{x}(m_u + L)} \right] \ldots \left[ \tilde{k}(m_u + NL) - \mu_{\tilde{x}(m_u + NL)} \right]
\]  

(2.26)

where \( \tilde{y}(i) \) represents a particular vector in \( \tilde{y} \), which is defined as

\[
\tilde{y} = \left[ \tilde{k}(m_d) - \mu_{\tilde{y}(m_d)} \right] \left[ \tilde{k}(m_d + (1-\nu)m_d) - \mu_{\tilde{y}(m_d + (1-\nu)m_d)} \right] \ldots
\[
\left[ \tilde{k}(m_d + (E-1)(1-\nu)m_d) - \mu_{\tilde{y}(m_d + (E-1)(1-\nu)m_d)} \right] \left[ \tilde{k}(m_d + E(1-\nu)m_d) - \mu_{\tilde{y}(m_d + E(1-\nu)m_d)} \right]
\]  

(2.27)

where \( m_u \) is the ending time of the first user observation vector, \( m_d \) is the length of one frame in time, \( \nu \) denotes the amount of overlap expressed as a fraction, \( k(m_u) \) represents the vector or array of reflection coefficients that end at time \( m_u \), and

\[
\mu_{\tilde{x}(m_u)} = \frac{1}{D} \sum_{i=1}^{D} k(i; m_u)
\]  

(2.28)

where \( k(i; m_u) \) is the \( i^{th} \) component of the vector of reflection coefficients ending at time \( m_u \). It can be seen from (2.24) that the correlation coefficient is calculated from frame to
frame. The length of the array of correlation coefficients $\rho$ for a particular commercial $s$ will be

$$lp = \frac{\text{length}(s) - LN}{m(1 - \nu)} \quad (2.29)$$

where $\text{length}(s)$ is the length in time of the commercial that we are correlating against, $L$ is the length of the time interval between user coefficients, $N$ is the number of frames processed by the user unit, $m$ is the length in time of a frame, and $\nu$ is the amount of overlap expressed as a fraction. In Chapter 4 (2.24) is referred to as the standard distance measure, because it is used in the majority of the tests.

The second method of calculating the correlation coefficients yields the coefficients $\rho_2$, which are based on (2.22) and are calculated in the following manner:

$$\rho_2(j; k) = \frac{1}{N} \sum_{i=1}^{N} \rho_2(X(i), Y(j + (i - 1)i)) \quad j = 1, 2, \ldots, lp \quad (2.30)$$

where $k$ is the index of the commercial in the database, $lp$ is given by (2.29) and where $X$ and $Y$ are vectors in the arrays of vectors $\tilde{X}$ and $\tilde{Y}$ that are defined to be

$$\tilde{X} = \left[ [\tilde{k}(m) - c] [\tilde{k}(m + L) - c] \ldots [\tilde{k}(m + NL) - c] \right] \quad (2.31)$$

$$\tilde{Y} = \left[ [k(m) - c] [k(m + (1 - \nu)m) - c] \ldots \right.$$

$$\left. [k(m + (E - 1)(1 - \nu)m) - c] [k(m + E(1 - \nu)m) - c] \right] \quad (2.32)$$
The index of the last frame in the database is denoted by \( E \). The mean vector of all vectors in the database is given by \( \bar{c} \) and is defined in (2.21). Results that are produced by this second distance measure are given in Section 4.3.11.

In many of the figures in Chapter 4 we show graphs of the maximum correlation difference. This quantity is defined as

\[
\text{MAX\_CORR\_DIFF} = \max_j \left[ \bar{\rho}(j; \text{correct commercial}) \right] - \max_{k \neq \text{correct commercial}} \left[ \max_j \left[ \bar{\rho}(j, k) \right] \right],
\]

where \( j \) is the index that produces the highest correlation for the specified commercial \( k \). We want the maximum correlation difference to be as large as possible.

When the user's transmitted coefficients most closely match the database, we will obtain a maximum in the array \( \rho(k) \). The maximum correlation coefficient that is obtained for each commercial is saved. After the user's coefficients have been correlated with all of the commercials, the commercial that produced the highest correlation coefficient, if it is above a predetermined threshold, is identified as the commercial that the user selected. The correlation coefficient will vary in the range \(-1 \leq \rho \leq 1\). However, in theory it can only reach a value of +1 if there is no noise introduced in the channel and the minimum phase equivalent spectrum of the test signal exactly matches a frame in the database. Although the last condition could possibly happen, the first could not. In other words, in practice \( \rho \) will never reach a value of +1, so that the commercial that produces
the correlation coefficient closest to +1 will be identified as the user's selection.
3.0. THEORY AND DETAILS OF SYSTEM

Now that we have discussed several general approaches to solving various problems in our IVDS application and have presented a block diagram and description of the proposed approach, we will discuss the theory and details of the system level approach that we proposed in Section 2.4. Each section of this chapter corresponds to a particular part of the system that we described in Section 2.4. We will now describe in detail the theory and tradeoffs involved in the development of each individual processing step.

3.1 Windows

The windowing process is included in the ‘preprocessing’ blocks of Figure 2.4. It is actually the second preprocessing step. There is first an analog bandpass filter that is applied to the signal before it is broken into segments by the window. The bandpass filter is discussed in Section 3.2.1. In this application the commercials must be divided into frames before any further analysis or processing can be done because they are nonstationary signals, and many of the analysis techniques that are used in this application require the signal to be stationary. The following equation describes the way in which a window is applied to speech.

\[ f_x(n,m) = x(n)w(m-n), \]  

(3.1)

where \( x \) is the signal sequence, \( w \) is the window sequence, \( m \) is the ending sample of the frame in the original sequence \( x \), and \( n \) is the index in the frame. There are several
commonly used windows [13]. The choice of window is application dependent. We will
discuss the theory governing the selection of a window shape in Section 3.1.1 as well as
the tradeoff involved in choosing the length of the window in Section 3.1.2.

3.1.1 Window Selection

Although there are some theory-based rules for selecting a window, 'trial and error'
could be considered one of the best and most reliable techniques. In the simulations of the
proposed system a couple of different window types were tested. The effect of a
particular type of window on the performance of the system can be seen in the
experiments that are described in Section 4.3.1. Rectangular, Hamming, and Hanning
windows were tested in the simulation of the system. These windows are defined by the
following equations:

\[
\text{Rectangular: } w(n) = \begin{cases} 
1, & 0 \leq n \leq N, \\
0, & \text{otherwise},
\end{cases} \quad (3.2)
\]

\[
\text{Hamming: } w(n) = \begin{cases} 
0.54 - 0.46\cos\left(\frac{2\pi n}{N}\right), & 0 \leq n \leq N, \\
0, & \text{otherwise},
\end{cases} \quad (3.3)
\]

\[
\text{Hanning: } w(n) = \begin{cases} 
0.5 - 0.5\cos\left(\frac{2\pi n}{N}\right), & 0 \leq n \leq N, \\
0, & \text{otherwise},
\end{cases} \quad (3.4)
\]

Some of the theory-based rules that can be utilized in selecting a window will now
be described and discussed.
The windows described in (3.1)-(3.3) and most other windows that are commonly used are symmetric about their midpoint. This means that these sequences have a linear phase characteristic and their DTFTs can in general be written as

\[ W(\omega) = |W(\omega)| e^{-j\omega((N-1)/2)} \]  

(3.5)

It can be seen from (3.1) that there is a time domain distortion of the original signal \( x(n) \), due to the application of the window, and this distortion can be viewed in the frequency domain as demonstrated by

\[ F_x(\omega;m) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega - \theta)W(-\theta)e^{-j\omega \theta} d\theta, \]  

(3.6)

where (3.6) is simply the Fourier transform of (3.1). However, for the purposes of the current discussion, we assume that our window is positioned so that \( w(m-n) \) is centered on time \( n = 0 \) with \( m = (M-1)/2 \). This means that we are assuming that the time origin of the signal \( x(n) \) is at the midpoint of the window. This causes some loss of generality, but it is sufficient for our purpose of describing the principles involved in the selection of a window. By substituting (3.5) into (3.6), we get

\[ F_x(\omega;m) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega - \theta)|W(-\theta)|d\theta = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega - \theta)|W(\theta)|d\theta \]  

(3.7)

where we have taken advantage of the fact that \( W(\theta) \) is an even function. Therefore, we can see from (3.7) that the key, as far as accurately preserving the spectral content of the signal, is to choose a window that has the spectral property shown in the following
\[ |W(\theta)| \approx 2\pi \delta_x(\theta), \quad (3.8) \]

where \( |W(\theta)| \) represents the magnitude spectrum of the window and \( \delta_x(\theta) \) represents the spectrum of an analog impulse function. All commonly used windows have a similarly shaped magnitude spectrum that is lowpass in nature with a main lobe at a low frequency and several attenuated ‘sidelobes’ at the higher frequencies. The spectral distortion that is caused by each of these windows can be seen by observing the alteration caused by the convolution of the signal spectrum with the spectrum of the window. Therefore, using this result we can conclude that the rectangular window in (3.2) is the best of all possible windows at resolving the sharp details of the spectrum, because it most closely matches (3.5) at the peak for a given window length \( N \). Any other window will have more of a smoothing effect about the frequency of interest but include less of the noise that is far away from the frequency of interest in the spectral domain.

A tradeoff is involved in the selection of a window, because windows with narrow main lobes, such as the rectangular window, have less attenuated sidelobes and windows with wider main lobes have more attenuated sidelobes. There is no general analytical criterion which can make the choice of a window obvious for every application.

For this application the window selection is dependent on the amount and type of acoustic noise in the presence of the user. If there is little noise, the window with the most narrow main lobe would probably be the best choice. Keep in mind that the same
processing procedures are performed on both the user and the host sides of the system, as shown in Figure 2.4. If there was no noise added to the signal before it reached the user, the user unit would process the same signal in the same manner as done on the host side, i.e., derive the parameters in the same way, and transmit feature vectors that would be identical to the ones in the database. The signal would be distorted by the windowing process in exactly the same way on both sides of the system, and the type of window that was used would not matter at all.

The situation is entirely different if there is a lot of noise introduced into the signal before it reaches the user. When a lot of noise has corrupted the signal, a window with higher magnitude sidelobes will include a lot more noise when it is applied to a segment of audio. This would imply a rule that says, 'Select a window with more attenuated sidelobes, if there will be a lot of wideband noise introduced in the channel from the host side to the user.' The major disadvantage of the windows with more attenuated sidelobes is the fact that they have wider main lobes. This causes a smoothing of the signal's spectrum, and therefore a loss of resolution. If the noise is broadband and white in nature, then a rectangular window would probably be a bad choice, because it would corrupt the signal at each particular frequency by bringing in a lot of the noise from other parts of the spectrum. If the noise is relatively narrowband, then the choice of a rectangular window might be a good choice.

Rules that can be established in the selection of a window for this application would include:
1. If it is expected that there is no noise added to the signal before it reaches the user, the type of window does not really matter.

2. If it is expected that the noise will be narrowband, we would most likely choose a rectangular window.

3. The amount of tapering of the selected window should be directly proportional to the amount of noise, especially the amount of white noise, that we expect to be added to the signal before it is sampled by the user.

3.1.2 Spectral-Temporal Resolution Trade-Off

Another tradeoff in this application is the one between increasing and decreasing the window length. Making the window longer will provide a more accurate spectral representation, if the window includes only a quasi-stationary portion of the signal. Making the window length smaller will yield better temporal resolution and make it more likely that each frame will contain a quasi-stationary segment of the signal that includes no transitional boundaries between adjacent phonemes or musical notes. However, a shorter window length results in more frames in the database and therefore more correlations for the identification process, which will therefore require more time. The length of the window is inversely proportional to the time needed for the identification process. For example, if the window length is decreased by one half, the time required for the identification of the user's parameters will increase by a factor of two. One of the goals that was established, and given in Section 1.1, is to limit the time required for the
identification process on the host side. The decision process for choosing the window length will now be discussed.

We have assumed that the commercial will be comprised mostly of speech, which is normally considered to be quasi-stationary over time periods of 5-20 ms [6]. Therefore, the window cannot be any longer than 20 ms if each phoneme is to be adequately resolved in time. A length of 20 ms appears to be the best choice for the length of the window, because using it will require the least amount of time for identification, yet it will still allow us to resolve the phonemes and musical notes in the time domain.

3.2 Preprocessing Filters

There are two filters that fit into the preprocessing block of Figure 2.2 and are applied before the parameters are derived from the signal. The first filter is a bandpass filter that is discussed in Section 3.2.1. The second filter is the preemphasis filter, which is discussed in Section 3.2.2. The reasons why these filters are needed will be the focus of the discussion in this section.

3.2.1 Bandpass Filter

The bandpass filter is the first step in the preprocessing block that is shown in Figure 2.4. There were two reasons the bandpass filter is needed. The first reason is to eliminate the spectral components in the commercial that are above the cutoff frequency of the broadcast channel. The filter should be applied to the signal at both sides of the system

49
before the parameters are calculated so that the parameters that are obtained on the user’s side can be compared with those that were obtained on the host side.

The second reason for using a bandpass filter was to eliminate some of the strong low frequency noise that was described in Section 2.1.2. Some of the first formants in speech are around 300 Hz and they are usually the strongest ones [2]. Any interfering speech in the room will produce a lot of noise in the range 270-730 Hz, which can be considered the average range for the first formants of a variety of vowels in American English [2], and they need to be preserved for identification purposes on the host side. If the main source of noise is interfering speech and there is at least some music in the commercial, the system will probably work better with a low end cutoff frequency in the range 730-840 Hz.

By limiting the bandwidth, we are utilizing a philosophy that is often referred to as channel or subband coding in speech processing literature. If noise at the user side is coming from adjacent rooms, then only the lower frequencies will be able to propagate through structures such as walls. Therefore, an attenuation of the lower frequencies will help for this reason as well. Several bandpass filters were tested in the system simulations described in Section 4.3.3. The results in terms of performance are also listed there.

After bandpass filtering the signal, we can focus on the portion of the spectrum that has been altered the least in the broadcast channel that is shown in Figure 2.1. The filter could be implemented in practice as an analog filter and would be used to process the
signal before it is sampled. This would additionally provide the function of an anti-aliasing filter. However, it is implemented here as a digital filter because a codec is used to sample the signal and takes care of the aliasing problem ahead of time by sampling at a very high rate.

3.2.2 Preemphasis Filter

The preemphasis filter is applied to the signal after the window and the bandpass filter have been applied. It can be included in the preprocessing block of Figure 2.4 and has the following system function:

\[ P(z) = 1 - \mu z^{-1} \]  \hspace{1cm} (3.9)

where usually \( 0.9 \leq \mu \leq 1 \).

There are two advantages to using a preemphasis filter in this application. Preemphasis gives the higher frequency components of the audio a better chance of influencing the outcome of the parameterization process. Since the spectrum of speech is normally dominated by the lower frequency components, the application of a preemphasis filter will have a ‘whitening’ effect on the signal’s spectrum, and hopefully the higher frequency peaks in the spectrum will influence the coefficient values more.

There is a second reason that preemphasis is beneficial. It prevents numerical instability that can be caused by the linear prediction analysis of the signal. The numerical instability can occur in signals that are composed mostly of low frequency components and
manifests itself in the form of an ill-conditioned autocorrelation matrix because the signal is too predictable and the linear prediction model order is too high [14]. The preemphasis filter in (3.5) will have a ‘whitening’ effect on the spectrum of such a signal, and tend to eliminate ill-conditioning.

If anything can be assumed about the statistical nature of the noise, we can assume that it has a low pass type of characteristic. Therefore, preemphasis will emphasize the portion of the signal spectrum that is most likely to have a high SNR. Various types of preemphasis were tested and the experiments and results are described in Section 4.3.2.

3.3. Order Estimation

Due to the variety of commercials that can be used in this application, it is very difficult to choose a model order for the linear predictive analysis. If the order of the model is chosen too small, then the linear prediction analysis will not be successful, because in this case there are not enough poles to adequately represent the signal’s true spectrum. The linear prediction process will often result in the poles of the model being placed between resonant places in the true spectrum. This is called formant splitting in the context of speech processing.

If the order is chosen to be higher than that for the actual signal or signals which comprise the commercial or the composite of these signals, then the acoustic noise is included in the parameterization of the signal at the user side. This means that the results will be influenced more heavily by the noise than they should be.
There are estimation rules for applications that involve accurately resynthesizing the speech from a single vocal tract at the receiving end of the system [3]. However, because of our assumptions about the input signal in Section 2.1.2, we tried many possible model orders on what we considered to be a ‘typical’ commercial in an attempt to find a model order that performed well in this application. It is upon these assumptions and experimental data that is listed in Section 4.3.4 that we have estimated the order of the linear prediction model for this application.

3.4. Optimal Coefficient Representation

There are many ways in which the linear prediction coefficients can be represented. The properties of various representations of the linear prediction coefficients for speech coding applications have been studied numerous times and in a variety of ways [2, 15, 16]. We want the chosen representation to have good quantization properties and properties that will make the identification performance as good as possible. In other words, we want to have the best identification performance possible as well as the fewest number of bits to transmit. These goals are contrary to one another. Therefore, the decision of which coefficient representation to use presents us with yet another tradeoff.

We have done system simulations with a variety of different representations of the LP parameters. These representations include:

1. the autoregressive (AR) parameters (denominator coefficients of the AR model),

2. the cepstral coefficients,
3. the reflection coefficients,

4. the inverse sine (IS) coefficients,

5. the log-area ratio (LAR) coefficients.

The tests that were done with each of these representations of the LP parameters are described in Section 4.3.6, and the results of the tests are also given in Section 4.3.6. The AR parameters $\hat{a}(i;m)$ are calculated from a segment of speech by performing the autocorrelation method of linear prediction on a frame of speech samples using the Levinson-Durbin algorithm, which is given in Appendix A. The reflection coefficients $k(i;m)$ can be determined from this algorithm as well. The cepstral coefficients $c(i;m)$ are derived from the AR parameters by making use of the relationship in the following equation [26].

$$c(n,m) = \sum_{k=1}^{\text{ORD}} \left( \frac{ap(k;m)}{n} \right)^n \quad 1 < n < \text{ORD} \quad (3.10)$$

where the $ap's$ are the roots of the denominator of the AR model, the $c's$ are the cepstral parameters, and $\text{ORD}$ is the AR model order. The cepstral coefficients actually represent the cepstrum of the impulse response of the AR model. The LAR coefficients $g(i;m)$ are derived from the reflection coefficients by (3.11).

$$g(i;m) = \tanh^{-1} k(i;m) = \frac{1}{2} \log \left( \frac{1 + k(i;m)}{1 - k(i;m)} \right), \quad 1 \leq i \leq M, \quad (3.11)$$
The IS coefficients $\sigma(i; m)$ can be calculated if the reflection coefficients are known by using (3.12).

$$
\sigma(i; m) = \frac{2}{\pi} \sin^{-1}(k(i; m)), \quad 1 \leq i \leq M,
$$

(3.12)

In this section we discuss the optimal choice of a representation of the LP parameters from two perspectives. We first discuss the choice with the goal of trying to select the representation that provides maximum identification performance. The second perspective is with the goal of trying to choose the representation that has optimal properties for quantization.

3.4.1. Identification Performance Optimal Representation

The optimal choice of the representation of the LP parameters is the representation that maximizes the information and minimizes the redundancy the most that is contained in each vector of coefficients. The information of interest to us is the angle between the feature vectors. It is this information that we use to distinguish the commercial that the user was watching from all of the others in the database, as shown in Section 2.4.3. The coefficients of the feature vector are random variables in this application. Entropy is a measure of disorder in a random variable. This quantity can also be viewed as the expected information in a random variable [30]. Therefore, if we maximize the entropy, we maximize the expected information.

Given a discrete random variable $X$, with alphabet $A$ and probability mass function
\( p(x) = \Pr\{X = x\}, x \in \mathcal{X} \) the entropy of the random variable is given by (3.13).

\[
H(X) = -\sum_{x \in \mathcal{X}} p(x) \log(p(x))
\] (3.13)

In the case where

\[
X = \begin{cases}
1 & \text{with probability } p, \\
0 & \text{with probability } 1 - p,
\end{cases}
\] (3.14)

we have

\[
H(X) = -p \log(p) - (1 - p) \log(1 - p) = H(p)
\] (3.15)

\( H(p) \) is plotted in Figure 3.1 for a variety of probabilities of the random variable \( X \).

![Figure 3.1 - Entropy of X for various probabilities p](image)

From the graph in Figure 3.1 we see that the optimal distribution for all of the random variables in the feature vector is a flat one. Therefore, we want to use the representation
in which each coefficient has an equally likely probability of taking on any of the values in a given range. This means that the angle between the feature vector and some reference vector should be equally likely to be any angle in the range $-\frac{\pi}{2} \leq \Theta \leq \frac{\pi}{2}$. We also want the mutual information between coefficients to be minimized. This is very important in minimizing the redundancy that we must transmit. The mutual information $I(X;Y)$ between two random variables $X$ and $Y$ is given by (3.16).

\[
I(X;Y) = \sum_{x \in A} \sum_{y \in B} p(x,y) \log \left( \frac{p(x,y)}{p(x)p(y)} \right),
\]

(3.16)

where $B$ is the alphabet of $Y$ and $p(x,y)$ is the joint probability mass function of $X$ and $Y$. Assuming in this case that $X$ and $Y$ are the final transmitted representations of two of the LP coefficients, the mutual information as given in (3.16) is a measure of the information that the knowledge of one of the coefficients will convey about the other. Theoretically, we could calculate the mutual information between any two of the coefficients in a feature vector. We want to minimize it. This can be done by minimizing $p(x,y)$.

In practice it is not possible to reduce all of the mutual information quantities to zero for any representation of the LP coefficients, because the linear predictive process by which the coefficients are originally derived is a recursive one. Therefore, each component of a vector depends to a certain extent on all of the other components but especially on the component immediately preceding it in the derivation process, as can be
seen in Appendix A. This does not change the fact that the LP coefficients or transformations thereof are the best currently known set of features to use for identification in our IVDS application. Their merits are noted in Section 2.2.2.2.

It may be possible to develop a better set of features, that could be derived from a segment of speech samples, by using a different initial optimization criterion in the development. However, the computation of such features may not be as convenient, nor as efficient from a coding efficiency perspective as the LP parameters. We have studied the performance of a variety of popular transformations of the LP parameters in our IVDS application and the results of the system simulations can be seen in Section 4.3.6. Our choice is made based upon these results.

3.4.2. Quantization Optimal Representation

It would be good to note at this point that, from a quantization point of view, it is very helpful to know the boundaries or range of a particular representation of the LP parameters. If we can determine a range for a particular representation of the coefficients, then we can efficiently quantize the coefficients in a uniform manner, which is very convenient, if we expect the best representation to have a flat or close to flat probability mass function. The only two representations that allow us a sure knowledge of the range of the distribution are the reflection coefficients and the inverse sine coefficients. So from a strict quantization point of view, the reflection coefficients and the inverse sine coefficients appear to be the best two representations, especially because their magnitudes
are bounded by one.

If another representation can be used to perform the identification better, we can transmit the reflection coefficients and perform the proper transformation at the host side to obtain the other representation. However, this can be done only at the cost of increasing the identification time, which we aim to avoid.

3.5. User Vector Calculation

A block diagram of the method that will be used to calculate the user vectors is shown in Figure 2.4. It is described in Section 2.4.2. Some of the reasons for this approach and the philosophy behind it are given in Chapter 2. The way in which the user vectors are to be calculated can be seen in Figure 2.4. Experiments were performed where all of the variables in the system were kept the same and the number of the vectors and the spacing of the intervals between the vectors were allowed to change individually. A description of the tests in which the number of user vectors was allowed to vary is given in Section 4.3.8. A description of the tests in which the spacing of the intervals was allowed to vary is given in Section 4.3.9. In this section we first discuss some of the principles involved in selecting the number of user vectors to compute and transmit in our IVDS application. Then, we discuss the issues related to the spacing between the user vectors in time.
3.5.1 Number of User Vectors

Theoretically, it is better in terms of identification performance to increase the number of user observation vectors calculated at the user side and transmitted to the host side. This is because it forces the commercial to match in more places. However, there are two important reasons why an increase in the number of user vectors is detrimental to the system and the specifications thereof. First of all, we are limited to 200 bits per user transmission. Secondly, using more observation vectors from the user side results in more distance calculations at the host side of the system, as well as in a longer identification time at the user end of the system. Therefore, we cannot use as many user observation vectors as we would like, and we are presented with another tradeoff. As a result, we have made an attempt to ascertain how important it is to transmit a large number of observation vectors from the user side as well as to determine how much an extra vector will improve the performance of the system. Section 4.3.8 describes the tests that were conducted, gives the results, and then analyzes the results and discusses the impact of them on the choice of how many user vectors we should use in an optimal system.

3.5.2 Spacing of User Vectors

Theoretically, long time intervals between the user vectors would be good, because it forces the user’s selected commercial to match with the one in the database for a longer period of time. It is more likely that two arbitrary commercials can match for a short period of time than for a long period of time. For instance, a commercial may have silent
parts that are more than several frames long and possibly even up to a second or two in length. This can pose a problem when multiple commercials fall into this category. If there is little noise at the user side, there will be minimum distances found in multiple commercials and a decision can not be made concerning which of these commercials the user was really watching when they made the selection. If there is a lot of noise at the user side, then the SNR is low and the noise at the user side will resemble one of the commercials more closely than the others, but the likelihood that it will be the correct commercial will be very low.

Another reason why long time intervals between the collection of frames for a particular user’s selection within a commercial is good is because it lessens the likelihood that the observed user vectors will have been influenced by the same type of noise. For instance, the noise at the user end may be impulsive or of short duration. If this is the case and the frames from which the user vectors were obtained are adjacent in time to one another, then it is very likely that both vectors could be corrupted by the same pulse of noise. However, if the frames from which the vectors were obtained are separated by a great distance in time, the likelihood that both frames will be corrupted by the noise is smaller.

One limiting factor in the length of time between each user observation is the fact that we do not know how long each commercial will last. Also, we have no way of knowing how far from the end of the commercial the user will make the selection, and the user may wish to respond to two different commercials that are broadcast consecutively.
If the commercials are short relative to the span of time over which we are collecting frames, we may not have enough time to sample the second of the two commercials that the user wanted to select. If the span of time over which we obtain the frames is too long, it could also contribute to identification errors. The user may select the first commercial. However, if the span of time over which we obtain frames is too long, we may collect more frames from the second commercial, which would possibly cause the system to falsely identify the user's selection as the second commercial. Section 4.3.9 describes the tests that were conducted, gives the results, and then analyzes the results and discusses their impact on the choice of what the spacing should be between the user vectors in an optimal system.

3.6. Quantization of Coefficients

The Loyd-Max algorithm or even some type of vector quantization could be advantageously applied if the distribution of the components of the vectors is known [17, 18]. However, because we do not know the distribution of the components of the vector for every possible commercial that could be added to the database, we can not effectively use this technique. Therefore, we will simply try to quantize the coefficients in a uniform manner. The optimum number of bits necessary to represent the components of each user vector for our application was determined from the results of the series of test simulations that are described in Section 4.3.10.
3.7. Coefficient Significance

Certain of the coefficients that are derived in the linear prediction analysis are more important in the identification of the commercial than others. The best subset of components of the vectors to retain for usage in our IVDS application are the ones that have the greatest amount of entropy individually and the least amount of mutual information between each other. An optimal decision could be made by the system if the entropy for each coefficient and mutual information for each subset of coefficients could be calculated. This is not feasible, because of the dynamically changing database. Over the course of a year or so, the database could change 100%, as old commercials are taken out and new commercials are added. We tested a few different subsets of the coefficients from the feature vectors. This is described in Section 4.3.7. These simulations were performed in an effort to see which of the coefficients worked best for our IVDS application.

3.8. Overlap of Frames in Database

It should be easy to see by examining the diagram in Figure 2.3 that more overlap of the frames in the database will yield better performance. This is because it will be more likely that the start of the frames from which the user’s coefficients are calculated will be closer to the starting times of the frames in the database, resulting in higher correlation. Experiments were performed with the frames in the database overlapped by differing amounts for different tests. The extent to which the overlap of the frames in the database
affects the performance of the system can be seen in the results of the experiments that are described in Section 4.3.5.
4.0. SIMULATIONS AND RESULTS

In this chapter we describe the experiments that were performed, list the results that were obtained, and provide an analysis of these results. The goal of the experiments was to determine some optimum value for a particular parameter of the system or to determine an optimum way of processing the signal at a particular stage of the proposed system that is given in the block diagram of Figure 2.4. By varying one of the system parameters while holding others constant, and then noting the variation in the performance of the system, we were able to achieve our goal.

In Section 4.1 we describe some of the test conditions that existed when the commercial was sampled by the ‘user’, as well as some of the characteristics of the sounds in the commercials that comprised the database. In Section 4.2 we describe a standard test configuration and give the standard parameter values and techniques for each individual processing step. The standard parameter values and techniques are the values and techniques that are used in the majority of the tests. In each section of Section 4.3 we first make a note of the parameters and techniques that are different from the standard configuration. Then we give the results of the section's set of tests and analyze them.

4.1. Test Conditions

The purpose of this section is to describe the conditions that existed at the time when the ‘user’ sampled the commercial. For the tests that are discussed in Sections
4.3.1-4.3.11, the conditions are described in the following list:

**Acoustic Environment:**

4 Computers Running in Close Proximity

Hard Surfaces in Room (Multi-path Propagation)

**Signal Output from Audio Source:**

Volume Control on Source set to 7

One Speaker Used (Mono output)

**Microphone Positioning:**

Approximately 1.80m (6ft) from the Speaker Device

Pointed Directly at the Speaker Device

**Commercial Database:**

Each commercial was approximately 15 seconds in length

Commercial #1 - music, man’s voice, high pitched and distorted voice

Commercial #2 - woman’s voice only

Commercial #3 - man’s voice, music

**User Selections:**

The commercial was first sampled continuously from beginning to end with the setup as described. Then, each 'user' was given a totally random starting time somewhere in the
first 12 seconds of the commercial. The commercial that was selected by the user in each case is commercial number one.

**Noise Characteristics and SNR:**

The power spectral density of a realization of the noise can be seen in Figure 4.1. A 1024 point FFT was calculated for a 1024 sample frame in the center of the noise sequence. We estimated the SNR for the tests included in this chapter by analyzing a realization of the noise using the following equation

\[
SNR = 10\log_{10}\left(\frac{\sigma_{s+n}^2 - \sigma_n^2}{\sigma_n^2}\right)
\]  

(4.1)

where \(\sigma_{s+n}^2\) is the variance of the realization of signal plus noise that was received at the microphone and \(\sigma_n^2\) is the variance of a different realization of only noise that was received at the microphone from the same noise sources at the same volume levels as before. The frame used in the computation was from three seconds into the commercial to thirteen seconds into the commercial. Therefore, it was ten seconds long. The SNR was calculated from (4.1) to be 31.4480 dB. Figure 4.1 shows that in this case the noise does not have the power spectral density of a white process. It has sort of a low pass characteristic, that is somewhat similar to what we would expect from interfering speech.

### 4.2. Standard Test Configuration

We now describe the standard test configuration. The standard distance measure is
the one that is given in (2.24). We chose to use (2.24), except in the last section in which we use a variety of distance measures. The standard bandpass filter is a Kaiser windowed FIR design with an order of 183 and a Beta of 8. The frequency response of the standard bandpass filter is shown in Figure 4.2. A 20 ms rectangular window was the standard window used in our IVDS application. Preemphasis was performed on every frame that was in the database as well as every frame obtained at the user side of our IVDS system. The standard amount of preemphasis is given by

\[ H(z) = 1 - 0.95z^{-1}. \] (4.2)

An autoregressive model of order ten was originally used to model the commercial, so a tenth order linear predictive analysis was performed on each frame using the
autocorrelation method via the Levinson-Durbin algorithm, which is given in Appendix A.

![Figure 4.2 - Standard Bandpass Filter Frequency Response](image)

Unless otherwise noted, the user observation vectors and reference feature vectors in the database are comprised of the odd indexed reflection coefficients that were derived from the LPC(10) analysis via the Levinson-Durbin algorithm. Therefore, the standard number of coefficients per feature vector is five. The standard number of frames of audio that were obtained at the user side of our IVDS system is eight and the standard spacing of these frames in time was one quarter of a second. The total number of bits used to
quantize each of the coefficients, unless otherwise noted, is five. The standard amount of overlap in the database is 75%. The following table summarizes the standard IVDS system configuration:

- **AR Model Order**: 10
- **Parameterization**: odd indexed reflection coefficients
- **Overlap in Database**: 75%
- **Number of Bits per Coefficient**: 5
- **Number of User Vectors**: 8
- **Spacing of User Vectors**: 1/4 second
- **Bandpass Filter**: Passband - (409-6736) Hz. Kaiser windowed FIR(183; β = 8)
- **Preemphasis**: 1 − 0.95z⁻¹, applied to every frame
- **Window**: 20ms rectangular
- **Distance Measure**: Given by (2.24)

Therefore, the total number of bits that get transmitted for each user selection when using the standard configuration is \(\left(\frac{5 \text{ bits}}{\text{coeff}}\right)\left(\frac{5 \text{ coeff}}{\text{vector}}\right)(8 \text{ vectors}) = 200 \text{ bits}\).

### 4.3. Test Results and Analysis

This section is broken into sections that give the results of the sets of tests that we performed in an attempt to optimize particular techniques or parameters that correspond to a particular portion of our IVDS system. Each section describes the ways in which the parameters or techniques differed from the standard test configuration that is given in Section 4.2. Also, the results are presented in a form that facilitates a comparison of the different techniques or parameter values used in each individual part of our IVDS system.
The results are then analyzed via descriptive statistics, which are tabulated for each test set. A negative result in the difference calculation of (2.33) means that the system identified the wrong commercial. If the mean of the maximum correlation difference for a large number of users is negative, then the system probably identified the wrong commercial the majority of the times. The actual number of times our IVDS system fails will depend on the distribution of the maximum correlation differences for the users. The first of two important statistics is the one that tells us how many standard deviations the mean of the users' maximum correlation difference is from failure. The second is the one that tells us how close the worst maximum correlation difference is from failure. Each part of this section contains tests that are aimed at helping us to obtain an optimization of a particular block or part of one of the blocks in the diagram in Figure 2.4.

4.3.1. Windowing

In the following set of tests we applied different types of windows to the commercial after it had been filtered with the standard bandpass filter. The other techniques and parameters used in the set of tests included in this section were identical to that of the standard configuration and were kept the same throughout this set of tests, so that comparisons could be made between the performance of various windows in our IVDS system.

The graph in Figure 4.3 shows the maximum correlation difference, which is given by (2.33). The maximum correlation difference is the difference between the maximum
correlation value produced by the commercial that the particular ‘user’ was truly watching and the maximum correlation value obtained in the correlation of the user observation vectors with all of the other commercials in the database. The plot is made for all of the ‘users’ for each of the different windows.

We see from Figure 4.3 that the performance of our IVDS system with a particular window in terms of the maximum correlation difference depends on the time at which the user starts sampling the commercial.

![Figure 4.3 - IVDS System Performance for Various Windows](image)

The rectangular window is the most resistant to the variation in the time offset of the user's frames with the database frames, as can be seen from the tightly clustered data.
points in Figure 4.3 and the standard deviation in Table 4.1.

Table 4.1 - Maximum Correlation Difference Statistics for Windowing Tests

<table>
<thead>
<tr>
<th>Window Type</th>
<th>mean $\mu$ of Max_Cor_Dif in Figure 4.3</th>
<th>st dev $\sigma$ of Max_Cor_Dif in Figure 4.3</th>
<th>$\frac{\mu}{\sigma}$</th>
<th>closest to failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rectangular</td>
<td>0.0856</td>
<td>0.0451</td>
<td>1.8995</td>
<td>0.0318</td>
</tr>
<tr>
<td>Hamming</td>
<td>0.0708</td>
<td>0.0562</td>
<td>1.2615</td>
<td>-0.0120</td>
</tr>
<tr>
<td>Hanning</td>
<td>0.0698</td>
<td>0.0552</td>
<td>1.2663</td>
<td>-0.052</td>
</tr>
</tbody>
</table>

The other windows can provide better results for certain selection times. However, on the average the rectangular window is ‘safer,’ as we see from Figure 4.3 and by noting that the mean of the maximum correlation difference data points is more standard deviations from failure when the rectangular window is used, as shown in Table 4.1. It is important to note that the distribution of the data points in Figure 4.3 is not necessarily Gaussian. This makes it more difficult to calculate the probability of error by examining only the mean and standard deviation of the data points in Figure 4.3. However, in general it is true that if the mean correlation difference is more standard deviations from negativity, then there tends to be less chance for error.

The window selection philosophy was discussed in Section 3.1.1. We should choose a window with more tapering if we expect the noise to have a flatter spectrum than shown.
in Figure 4.1. The window application in the frequency domain for a commercial that has been corrupted by white noise can be thought of as the convolution of the spectrum of the rectangular window with the corrupted signal will include more noise in the resulting frame than a more tapered window that does not have the high sidelobes in the frequency domain. However, we will assume that Figure 4.1 is close to the typical spectrum for the acoustic noise encountered at the user end of the system. Therefore, we should use the rectangular window over the others because it does provide greater resolution, as discussed in Section 3.1.1.

4.3.2. Preemphasis Filter

We now look at the effect of the preemphasis filter. In this set of tests the only difference from the standard configuration is that all of the reflection coefficients that were derived in the standard manner were used, not just the odd ones, and the standard preemphasis filter was either present or absent. The graph in Figure 4.4 shows the maximum correlation difference, which is given by (2.33). The maximum correlation difference is the difference between the maximum correlation value of the commercial that the particular ‘user’ was truly watching and the maximum correlation value obtained in the correlation of the user observation vectors with all of the other commercials in the database. The plot is made for all of the ‘users’ for the cases of presence and absence of the standard preemphasis filter.

The results of the preemphasis testing are shown in Figure 4.4. We see that our
IVDS system performs much better when the standard preemphasis filter is used. The results are as expected and a possible explanation for the improved performance of the system using the preemphasis filter is that it flattens the distribution of the resulting coefficients, which in most cases increases the information each coefficient can convey to the host side of our IVDS system. It also emphasizes the part of the spectrum that has a higher SNR prior to parameterization.

The improvement caused by the addition of a preemphasis filter to the system in this case is substantial. Note from Table 4.2, that the mean of the maximum correlation
difference without preemphasis is much less than the mean for the case where preemphasis is applied. In Table 4.2 we can see from the statistics that the mean of the system with preemphasis is much better than without. The result for each individual user can be seen in Figure 4.4.

Table 4.2 - Maximum Correlation Difference Statistics for Preemphasis Tests

<table>
<thead>
<tr>
<th>Preemphasis</th>
<th>mean $\mu$ Max_Cor_Dif in Figure 4.4</th>
<th>st dev $\sigma$ Max_Cor_Dif in Figure 4.4</th>
<th>$\frac{\mu}{\sigma}$</th>
<th>closest to failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Preemphasis</td>
<td>0.0431</td>
<td>0.0144</td>
<td>2.9931</td>
<td>0.0201</td>
</tr>
<tr>
<td>No Preemphasis</td>
<td>-0.0043</td>
<td>0.0155</td>
<td>0.2774</td>
<td>-0.0377</td>
</tr>
</tbody>
</table>

However, the mean of the maximum correlation distances is 2.9931 standard deviations away from negativity when the preemphasis filter is used. There is therefore a low probability of error for our IVDS system with the standard preemphasis included under the given test conditions. Therefore, it is easy to see that the preemphasis filter is very useful and necessary in our IVDS application.

The amount of preemphasis could probably be varied to obtain even better results.
However, the optimal amount of preemphasis will vary from commercial to commercial, because of the differences in spectral content. The preemphasis will probably want to be chosen in such a way as to allow the peaks in the higher frequency portion of the spectrum that are local maximums in amplitude to be boosted so that they are equal with the global peaks. If this could be done in every case, the maximum correlation differences, according to the comparative results in Figure 4.4, would be better. However, we have no way of knowing a priori how great the magnitude of the higher portion of the spectrum is at the user end for every commercial, because each commercial is corrupted by noise of an unknown nature. Therefore, we are left to make an educated guess. The amount of preemphasis that we have used in this experiment is similar to that which is commonly used for speech applications [2].

4.3.3. Bandpass Filter

We now look at the effect of three different bandpass filters. Here, the only difference from the standard configuration is that all reflection coefficients were used, not just the odd ones, and the bandpass filter that was used on the commercial was either bandpass filter #1 (the standard bandpass filter), bandpass filter #2, or bandpass filter #3. Where bandpass filter #2 and bandpass filter #3 are defined in the following manner.

Bandpass filter #2 has a passband of (282-5518) Hz and is a Kaiser windowed FIR design with an order of 183 and beta of 8. The frequency response of the filter is shown in Figure 4.5. Bandpass filter #3 has a passband of (804-7112) Hz, which was chosen
according to the discussion given in Section 3.2.1, and is a Kaiser windowed FIR design, because the Kaiser window produces a large amount of attenuation in the stopband for a given order.

![Figure 4.5 - Bandpass Filter #2 Frequency Response](image)

The order used in this application was 183 and beta of 8 was used. Figure 4.6 shows the frequency response. The graph in Figure 4.7 shows the maximum correlation difference, which is given by (2.33). The maximum correlation difference is the difference between the maximum correlation value of the commercial that the particular 'user' was truly watching and the maximum correlation value obtained in the correlation of the user observation vectors with all of the other commercials in the database. The plot is made for all of the 'users' for all three bandpass filters.
Figure 4.6 - Bandpass Filter #3 Frequency Response

Figure 4.7 - IVDS System Performance for Various Bandpass Filters

Figure 4.7 shows that the performance of our IVDS system is dependent on the
region of the audio spectrum that we choose to use for identification. However, we do need to be careful in interpreting these results. As we discuss in Section 4.3.4, the optimal model order for the bandpass filter that has the passband (409-6736) Hz is twelve. We are using a model order of ten in this set of tests. So, the improvements that one filter makes over another could be due to the fact that the signal fits the autoregressive model order of ten better after being bandpass filtered over that particular range of frequencies. However, the improvement by a bandpass filter may also be due to the fact that the filter passes the portion of the spectrum that has the highest SNR. Our goal in the selection of a bandpass filter is to increase the maximum correlation difference by selecting a portion of the spectrum that has the highest SNR, while keeping the complexity of the resulting signal within the order of our autoregressive model.

We see from Table 4.3 that the best bandpass filter for the model order of ten and this particular realization of signal plus noise is the one with the passband range of (804-7112) Hz. In this case the mean of the maximum correlation difference is 3.4527 standard deviations from failure (the theoretical breakdown point of the correlation approach). For the (282-5518) Hz bandpass filter, the mean of the maximum correlation difference is 1.7568 standard deviations from failure. For the case where the bandpass filter has a passband of (409-6736) Hz, the mean of the maximum correlation difference is 2.9906 standard deviations away from failure.

We made an attempt to find the optimal AR model order for the bandpass filter with the passband (409-6736) Hz as well as the filter with the passband (804-7112) Hz. The
corresponding tests and the results are described in the following section (Section 4.3.4).

Table 4.3 - Maximum Correlation Difference Statistics for Bandpass Filter Test

<table>
<thead>
<tr>
<th>Passband of BPF</th>
<th>mean $\mu$ Max_Cor_Dif in Figure 4.7</th>
<th>st dev $\sigma$ Max_Cor_Dif in Figure 4.7</th>
<th>$\frac{\mu}{\sigma}$</th>
<th>closest to failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>(409-6736) Hz</td>
<td>0.0431</td>
<td>0.0144</td>
<td>2.9906</td>
<td>0.0201</td>
</tr>
<tr>
<td>(282-5518) Hz</td>
<td>0.0272</td>
<td>0.0155</td>
<td>1.7568</td>
<td>0.0028</td>
</tr>
<tr>
<td>(804-7112) Hz</td>
<td>0.0469</td>
<td>0.0136</td>
<td>3.4527</td>
<td>0.0242</td>
</tr>
</tbody>
</table>

4.3.4. Order Estimation

We have run two sets of tests in an effort to estimate the optimal model order for the realization of signal plus noise that is described in Section 4.1. In the first set of tests the signal plus noise has been put through the standard bandpass filter before any parameterization is done, and in the second set the signal plus noise has been put through bandpass filter #3 before any parameterization is done. The standard bandpass filter is described in Section 4.2 and bandpass filter #3 is described in Section 4.3.3. We will call the first test set the ‘BPF test #1’, and it was done using the standard test configuration with varying model orders. The model used is an autoregressive one, where the coefficients are calculated using the autocorrelation method of linear prediction via the
Levinson-Durbin algorithm, and the order is allowed to equal 10, 12, 14, and 16. We will call the second test set 'BPF test #2', and it was done using the standard test configuration except with bandpass filter #3, 50% overlap in the database instead of 75%, and varying model orders. BPF test #2 also uses an autoregressive model, where the coefficients are calculated using the autocorrelation method of linear prediction via the Levinson-Durbin algorithm, and the order is allowed to equal 10, 12, 14, and 16.

The graphs in Figures 4.8 and 4.9 show the maximum correlation difference, which is given by (2.33). The maximum correlation difference is the difference between the maximum correlation value of the commercial that the particular 'user' was truly watching and the maximum correlation value obtained in the correlation of the user observation vectors with all of the other commercials in the database. The plot is made for all of the 'users' for all three bandpass filters.

![Graph](image-url)

Figure 4.8 - Maximum Correlation Difference for Various Model Orders - BPF Test #1
Figure 4.9 - Maximum Correlation Difference for Various Model Orders - BPF Test #2

The statistics associated with the graph in Figure 4.8 are shown in Table 4.4. The mean and variance of the results from Figure 4.9 for each of the model orders are shown in Table 4.5. The 'optimal' model order is equal to ten or twelve for the case where the commercial is bandpass filtered with the filter having the passband of (409-6736) Hz, and much higher in the case where the commercial is bandpass filtered by the filter having the passband of (804-7112) Hz.

In the case of the first bandpass filter, the ratio $\frac{\mu}{\sigma}$ is better for an order of twelve. However, some of the outliers in the graph in Figure 4.8 are closer to the breakdown point than in the case where the model order is chosen to be ten. Therefore, an order of ten is possibly the best choice for the first bandpass filter. However, we can see that the
Table 4.4 - Maximum Correlation Difference Statistics for (409-6736) Hz BPF

<table>
<thead>
<tr>
<th>Order</th>
<th>mean $\mu$ Max_Cor_Dif in Figure 4.8</th>
<th>st dev $\sigma$ Max_Cor_Dif in Figure 4.8</th>
<th>$\frac{\mu}{\sigma}$</th>
<th>closest to failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.0780</td>
<td>0.0346</td>
<td>2.2543</td>
<td>0.0309</td>
</tr>
<tr>
<td>12</td>
<td>0.0940</td>
<td>0.0402</td>
<td>2.3383</td>
<td>0.0310</td>
</tr>
<tr>
<td>14</td>
<td>0.0865</td>
<td>0.0471</td>
<td>1.8365</td>
<td>0.018i</td>
</tr>
<tr>
<td>16</td>
<td>0.0856</td>
<td>0.0451</td>
<td>1.8980</td>
<td>0.0318</td>
</tr>
</tbody>
</table>

Table 4.5 - Maximum Correlation Difference Statistics for (804-7112) Hz BPF

<table>
<thead>
<tr>
<th>Order</th>
<th>mean $\mu$ Max_Cor_Dif in Figure 4.9</th>
<th>st dev $\sigma$ Max_Cor_Dif in Figure 4.9</th>
<th>$\frac{\mu}{\sigma}$</th>
<th>closest to failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.0808</td>
<td>0.0381</td>
<td>2.1207</td>
<td>0.0360</td>
</tr>
<tr>
<td>12</td>
<td>0.0776</td>
<td>0.0341</td>
<td>2.2757</td>
<td>0.0062</td>
</tr>
<tr>
<td>14</td>
<td>0.0918</td>
<td>0.0369</td>
<td>2.4878</td>
<td>0.0186</td>
</tr>
<tr>
<td>16</td>
<td>0.0881</td>
<td>0.0341</td>
<td>2.5876</td>
<td>0.0068</td>
</tr>
</tbody>
</table>
frequency range that is selected by the second bandpass filter may be able to give us better results in terms of identification as the model order is increased beyond sixteen.

We see from Table 4.5, with the second bandpass filter we never reached an ‘optimum’ performance in terms of the model order for the set of tests that was conducted. We cannot have a model order that is too large, because it restricts us on the number of vectors that we can use, and as we show in Section 4.3.8, the number of user observation vectors should not be reduced below eight. Also, we show in Section 4.3.10 that the number of bits per coefficient should not be reduced below five. This means that we cannot have any more than five coefficients per user observation vector.

Therefore, we have two choices. We could either find another bandpass filter over the range of (804-7112) Hz that would produce a signal that has a lower order of complexity, or we could simply use the first bandpass filter with an order of ten or twelve.

4.3.5. Overlap of Database Frames

We now look at the effect of varying the amount of overlap of the database frames. The only difference in this set of tests from the standard configuration is that the amount of overlap is varied. The first test uses a standard amount of overlap (75%) and the second test uses 50% overlap of the frames in the database. The graph in Figure 4.10 shows the maximum correlation difference, which is given by (2.33). The maximum correlation difference is the difference between the maximum correlation value of the commercial that the particular ‘user’ was truly watching and the maximum correlation.
value obtained in the correlation of the user observation vectors with all of the other commercials in the database. The plot is made for all of the ‘users’ for the each amount of database frame overlap.

Figure 4.10 - IVDS System Performance for Different Amounts of Frame Overlap

In a way the results from this test were not as expected, and in a way they were. As discussed in Section 3.8, we would expect the system to perform theoretically better with more overlap in the database. According to Figure 4.10 and Table 4.6, the system with 50% overlap of the frames in the database performed at least as good, if not better than
the system with 75% overlap. Neither system produced an error. The worst results of both systems were at about the same level in the graph in Figure 4.10. The mean of the maximum correlation distances for both systems is about the same number of standard deviations from the breakdown point.

Table 4.6 - Maximum Correlation Difference Statistics for Database Overlap Tests

<table>
<thead>
<tr>
<th>Amount of Overlap</th>
<th>mean $\mu$ Max_Cor_Dif in Figure 4.10</th>
<th>st dev $\sigma$ Max_Cor_Dif in Figure 4.10</th>
<th>$\frac{\mu}{\sigma}$</th>
<th>closest to failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>75%</td>
<td>0.0856</td>
<td>0.0451</td>
<td>1.8995</td>
<td>0.0318</td>
</tr>
<tr>
<td>50%</td>
<td>0.0885</td>
<td>0.0426</td>
<td>2.0759</td>
<td>0.0273</td>
</tr>
</tbody>
</table>

However, we can see in Figure 4.11 that the system with the 75% overlapped database produces either the same or greater maximum correlation per user for the correct commercial than that obtained for the same user by the system with the 50% overlapped database. The gain in the correlation of the correct commercial seems to be offset by the fact that the wrong commercials produce higher correlations as well. In the majority of the cases it appears to be at least equal to if not greater than the gain that is achieved by the overlap. This is most likely due to the fact that there are more reference feature vectors in the database that is 75% overlapped, and therefore, there is a better chance that one of the vectors in a wrong commercial will be closer to the user’s observation vector.
The main disadvantage to more vectors in the database is the fact that it increases the identification time by a factor that is directly proportional to the amount of additional overlap. Systems with even less than 50% database overlap may produce acceptable results.

4.3.6. Coefficient Representation

We now look at the effect of using various coefficient representations. Here, the only difference from the standard configuration is that each of the coefficients is quantized
to a 64 bit floating point number for all of the tests in this set, which is not really much quantization at all, and the representations of the coefficients is varied. Reflection coefficients, autoregressive parameters, cepstral coefficients, log-area ratio coefficients, and inverse sine coefficients were used in one test each. The graph in Figure 4.12 shows

![Graph showing correlation differences for different representations](image)

**Figure 4.12 - Performance for Different Coefficient Representations**

the maximum correlation difference, which is given by (2.33). The maximum correlation difference is the difference between the maximum correlation value of the commercial that the particular 'user' was truly watching and the maximum correlation value obtained in the correlation of the user observation vectors with all of the other commercials in the database. The plot is made for all of the 'users' for each of the coefficient representations.
From Figure 4.12 and Table 4.7 it is easy to see that the reflection coefficients, LAR coefficients, IS coefficients, and cepstral coefficients yield comparable identification performance. It can also be seen that the AR parameters are not suitable for use in our IVDS application. In the analysis of these results, we use the same criterion in making a selection as before.

The first criterion is the number of standard deviations the mean is from zero in each case. The best parameterization to use is the reflection coefficients as seen from Table 4.6. However, performance of the LAR and IS parameterizations are very close to the reflection coefficient parameterization in terms of the maximum correlation difference.

The other performance measure that is of interest to us is how close to zero the outliers of the distribution are on the zero side of the mean. From the graph in Figure 4.9, we can see that the outliers closest to causing an error in the identification process are at about the same place. Therefore, we will conclude that the reflection coefficients may be slightly better than the IS and LAR coefficients, are definitely better than the cepstral coefficients and are much better than the AR parameters for our IVDS application.

4.3.7. Importance of Particular Coefficients in the User Vectors

We now look at the effect of using various sets of the LPC(10) coefficients. The only difference in this set of tests from the standard configuration is that differing sets of the coefficients that were derived from the frames via LPC(10) in the standard manner were used. All of the coefficients that were derived were used in the first test. Only the
### Table 4.7 - Maximum Correlation Difference Statistics for Coefficient Representations

<table>
<thead>
<tr>
<th>Coefficient Representation</th>
<th>Mean $\mu$ Max_Cor_Dif in Figure 4.12</th>
<th>Std Dev $\sigma$ Max_Cor_Dif in Figure 4.12</th>
<th>$\frac{\mu}{\sigma}$</th>
<th>Closest to Failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reflection Coefficients</td>
<td>0.0431</td>
<td>0.0144</td>
<td>2.9931</td>
<td>0.0201</td>
</tr>
<tr>
<td>AR Parameters</td>
<td>0.0024</td>
<td>0.0029</td>
<td>0.8276</td>
<td>-0.0037</td>
</tr>
<tr>
<td>Cepstral Coefficients</td>
<td>0.0378</td>
<td>0.0134</td>
<td>2.8209</td>
<td>0.0151</td>
</tr>
<tr>
<td>LAR Coefficients</td>
<td>0.0461</td>
<td>0.0164</td>
<td>2.8110</td>
<td>0.0204</td>
</tr>
<tr>
<td>IS Coefficients</td>
<td>0.0448</td>
<td>0.0150</td>
<td>2.9867</td>
<td>0.0212</td>
</tr>
</tbody>
</table>

Odd indexed coefficients were used in the second simulation, and only the even indexed coefficients were used in the third. The graph in Figure 4.13 shows the maximum correlation difference, which is given by (2.33). The maximum correlation difference is the difference between the maximum correlation value of the commercial that the particular 'user' was truly watching and the maximum correlation value obtained in the correlation of the user observation vectors with all of the other commercials in the database. The plot is made for all of the 'users' for all, odd indexed, and even indexed reflection coefficients, where 'odd' in Figure 4.13 implies the usage of the odd indexed coefficients of the vector,
'even' implies usage of the even indexed coefficients of the vector, and 'all' implies usage of all of the vector coefficients.

![Graph showing correlation difference for different parts of the user vector](image)

**Figure 4.13 - Performance Using Different Parts for the User Vector**

An interesting aspect of these experiments is that when all of the coefficients are used the variance of the results from user to user decreases dramatically. It is also interesting to note that the odd coefficients perform much better than the even coefficients and the maximum correlation difference results in Figure 4.13 are almost as good as the results that were obtained by using all coefficients. In one way the odd coefficients look
like they perform better than the case where all coefficients were used. To answer the question of why it is so, one needs to pursue further research in this area. In particular, their worst results are better than the worst results obtained by using all coefficients, as can be seen in Figure 4.13. However, it is important to note that the mean of the odd coefficient test results is fewer standard deviations from zero than the case where all of the coefficients were used, as seen in Table 4.8.

Table 4.8 - Maximum Correlation Difference Statistics for Coefficient Importance Tests

<table>
<thead>
<tr>
<th>Coefficients Used</th>
<th>mean $\mu$ Max_Cor_Dif in Figure 4.13</th>
<th>st dev $\sigma$ Max_Cor_Dif in Figure 4.13</th>
<th>$\frac{\mu}{\sigma}$</th>
<th>closest to failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>0.0431</td>
<td>0.0144</td>
<td>2.9906</td>
<td>0.0201</td>
</tr>
<tr>
<td>odd</td>
<td>0.0856</td>
<td>0.0451</td>
<td>1.8995</td>
<td>0.0318</td>
</tr>
<tr>
<td>even</td>
<td>0.0336</td>
<td>0.0247</td>
<td>3.4629</td>
<td>-0.0244</td>
</tr>
</tbody>
</table>

Therefore, more testing should be done to make it clear exactly which combination of coefficients is best and why. However, it is clear that from a transmission compression point of view, there is a big price to pay for including the even coefficients. It doubles the number of bits per transmission. This price is too high. The extra bits that are obtained by reducing the number of coefficients to five per vector can be used more judiciously in other ways.
4.3.8. Number of User Vectors

We now look at the effect of using different numbers of user vectors. Here, the only difference from the standard configuration is that the number of user observation vectors that were obtained and used in each test was different. Ten, eight, and six vectors were used in one test each. The graph in Figure 4.14 shows the maximum correlation difference, which is given by (2.33). The maximum correlation difference is the difference between the maximum correlation value of the commercial that the particular ‘user’ was truly watching and the maximum correlation value obtained in the correlation of the user observation vectors with all of the other commercials in the database. The plot is made for all of the ‘users’ for each number of user observation vectors. We see from Figure 4.14 or Table 4.9 that increasing the number of user observation vectors will increase the identification performance, just as we expected.

![Figure 4.14 - IVDS System Performance for Different Numbers of User Vectors](image)
Table 4.9 - Maximum Correlation Difference Statistics for Number of User Vector Tests

<table>
<thead>
<tr>
<th>Number of Intervals</th>
<th>mean $\mu$ Max_Cor_Dif in Figure 4.14</th>
<th>st dev $\sigma$ Max_Cor_Dif in Figure 4.14</th>
<th>$\frac{\mu}{\sigma}$</th>
<th>closest to failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.1098</td>
<td>0.0519</td>
<td>2.1156</td>
<td>0.0362</td>
</tr>
<tr>
<td>8</td>
<td>0.0856</td>
<td>0.0451</td>
<td>1.8980</td>
<td>0.0318</td>
</tr>
<tr>
<td>6</td>
<td>0.1313</td>
<td>0.2601</td>
<td>0.5048</td>
<td>-0.0456</td>
</tr>
</tbody>
</table>

The data shows that increasing the number of user observation vectors from six to eight has more of an impact on the performance of the system than increasing from eight to ten user vectors. The increase from six to eight user vectors has two comforting effects on the performance. The first is that the system fails three fewer times with eight user vectors than with six of them. The second is that the mean of the maximum correlation distances in Figure 4.10 is more than one standard deviation from the breakdown point.

Even the increase in the number of user vectors from eight to ten increases the performance. However, if we are to keep the transmission per user under 200 bits, we need to keep the number of user vectors at nine or less. The main reason is that, as we show in Sections 4.3.7 and 4.3.10, it is judicial to use at least 5 coefficients per vector and 4 bits per coefficient.

Depending on how many commercials are expected in the database and how robust
the system is to be against the effects of noise, we may want to increase the number of user vectors to ten or more. This experiment shows that the system is very sensitive to a variation in the number of user observation vectors, and increasing the number of user vectors is an efficient way to improve the performance.

4.3.9. Spacing of User Vectors

We now look at the effect of using a different spacing between the user observation vectors in the database. Here, the only difference from the standard configuration is that only six user observation vectors were used, and the spacing of the vectors was one half of a second in one test and one quarter of a second in the other test. The graph in Figure 4.15 shows the maximum correlation difference, which is given by (2.33). The maximum correlation difference is the difference between the maximum correlation value of the commercial that the particular ‘user’ was truly watching and the maximum correlation value obtained in the correlation of the user observation vectors with all of the other commercials in the database. The plot is made for all of the ‘users’ for each amount of spacing.

We see from Figure 4.15 that increasing the spacing in time between the user observation vectors will increase the identification performance, just as we expected. The improvements can also be seen in Table 4.10.

Increasing the amount of spacing in time between the user observation vectors increases the performance in two ways. The first is that the system fails one less time with
Figure 4.15 - IVDS Performance for Different Amounts of User Vector Spacing

Table 4.10 - Maximum Correlation Difference Statistics for User Vector Spacing Tests

<table>
<thead>
<tr>
<th>Spacing of Intervals</th>
<th>mean $\mu$ Max_Cor_Dif in Figure 4.15</th>
<th>st dev $\sigma$ Max_Cor_Dif in Figure 4.15</th>
<th>$\frac{\mu}{\sigma}$</th>
<th>closest to failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2 sec</td>
<td>0.0750</td>
<td>0.0594</td>
<td>1.2625</td>
<td>-0.0048</td>
</tr>
<tr>
<td>1/4 sec</td>
<td>0.1313</td>
<td>0.2601</td>
<td>0.5050</td>
<td>-0.0456</td>
</tr>
</tbody>
</table>
1/2 second spacing than with 1/4 second spacing. The second is that the mean of the maximum correlation distances is more standard deviations away from the breakdown point.

The tradeoff involved in increasing the spacing between the user observation vectors is discussed in Section 3.5.2. According to the results that are shown in Figure 4.11 and Table 4.8, we should increase the spacing between the user vectors to the maximum allowable amount.

4.3.10. Quantization of Coefficients

We now look at the effect of quantizing the coefficients to differing numbers of bits. Here, the only difference from the standard configuration is that the number of bits per coefficient was 12 in the first test, 5 in the second, 4 in the third, and 3 in the last. The graph in Figure 4.16 shows the maximum correlation difference, which is given by (2.33). The maximum correlation difference is the difference between the maximum correlation value of the commercial that the particular 'user' was truly watching and the maximum correlation value obtained in the correlation of the user observation vectors with all of the other commercials in the database. It is the difference between the maximum correlation of the commercial that the particular 'user' was truly watching and the maximum correlation of the commercial in the database closest to that one. The plot is made for all of the 'users' for each number of bits per coefficient.
Figure 4.16 - IVDS Performance for Differing Numbers of Bits/Coefficient

Reducing the allocation of the number of bits per coefficient by one does not degrade the identification performance of the system very much until we reach 4 bits per coefficient. Also, in the case where there are few quantization levels, it is possible for us to have a few user observation vectors that have all components equal, which is a problem because the result is a variance of zero for that user vector and, according to (2.18), this produces a correlation coefficient of the form $\frac{0}{0}$. We will refer to this problem as the 'zero variance problem'. The result of a correlation coefficient equal to infinity is a corresponding difference calculation that is indeterminate, which is obviously in error. The difference we are referring to is the difference between the maximum correlation of the user coefficients with the correct commercial minus the correlation of the user coefficients with the closest of the wrong commercials, as shown in the graph in Figure 4.16. We see that for 3 bits
per coefficient in Figure 4.16, the difference calculation produces an infinite result for users 8, 15 and 20 by noting that the graph of this quantity disappears for these users. We have simply indicated that the results are 'not available' in Table 4.11. Therefore, we can see the importance of judiciously selecting the number of bits to represent the coefficients. There is a somewhat larger drop in performance in going from 5 bits per coefficient to 4 bits per coefficient than in going from 12 bits per coefficient to 5 bits per coefficient. So, a wise choice may be 5 bits per coefficient, because it appears that the performance begins dropping after any further reduction, and this choice will keep the likelihood of the zero variance problem small.

Table 4.11 - Maximum Correlation Difference Statistics for Quantization Tests

<table>
<thead>
<tr>
<th>Bits/Coefficient</th>
<th>mean $\mu$ Max_Cor_Dif in Figure 4.16</th>
<th>st dev $\sigma$ Max_Cor_Dif in Figure 4.16</th>
<th>$\frac{\mu}{\sigma}$</th>
<th>closest to failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>not available</td>
<td>not available</td>
<td>not available</td>
<td>0.0330</td>
</tr>
<tr>
<td>4</td>
<td>0.0765</td>
<td>0.0541</td>
<td>1.4140</td>
<td>0.0124</td>
</tr>
<tr>
<td>5</td>
<td>0.0856</td>
<td>0.0451</td>
<td>1.8980</td>
<td>0.0318</td>
</tr>
<tr>
<td>12</td>
<td>0.0879</td>
<td>0.0479</td>
<td>1.8354</td>
<td>0.0236</td>
</tr>
</tbody>
</table>
4.3.11. Distance Measures

In the set of tests that are discussed in this section we used the standard test configuration except with 50% overlap instead of 75% and we used different distance measures for each test. The distance measure used in the first test was based upon (2.24). In the second test we used (2.30), and in the third test we used (2.16). The graph in Figure 4.17 shows the maximum correlation difference, which is given by (2.33). The maximum correlation difference is the difference between the maximum correlation value of the commercial that the particular ‘user’ was truly watching and the maximum correlation value obtained in the correlation of the user observation vectors with all of the other commercials in the database. The plot is made for all of the ‘users’ for each of the different distance measures.

Figure 4.17 - IVDS Performance for Different Distance Measures
Table 4.12 - Maximum Correlation Difference Statistics for Distance Calculation Tests

<table>
<thead>
<tr>
<th>Distance Measure</th>
<th>mean $\mu$ Max_Cor_Dif in Figure 4.16</th>
<th>st dev $\sigma$ Max_Cor_Dif in Figure 4.16</th>
<th>$\frac{\mu}{\sigma}$</th>
<th>closest to failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>(2.24)</td>
<td>0.0885</td>
<td>0.0426</td>
<td>2.0759</td>
<td>0.0273</td>
</tr>
<tr>
<td>(2.16)</td>
<td>0.0473</td>
<td>0.0230</td>
<td>2.0613</td>
<td>0.0134</td>
</tr>
<tr>
<td>(2.30)</td>
<td>0.1127</td>
<td>0.1017</td>
<td>1.1076</td>
<td>-0.1458</td>
</tr>
</tbody>
</table>

From Figure 4.17 and Table 4.12 we see that the standard distance measure is better than the other two distance measures. However, as we noted in the previous section (Section 4.3.10), we are unable to quantize the coefficients of the user vectors to fewer than 3 bits each, because of what we referred to as the 'zero variance problem'. The 'zero variance problem' occurs because the distribution of the coefficients contained in the user vectors have very close to the same distribution and therefore have a high probability of all being equal for a particular vector. The statistics of the coefficients are given in Table 4.13 and the distributions are shown in Figure 4.18.

Table 4.13 - Statistics for Various Coefficients in Vectors in Database

<table>
<thead>
<tr>
<th>Coefficient Number</th>
<th>mean</th>
<th>std. dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.3949</td>
<td>0.2806</td>
</tr>
<tr>
<td>3</td>
<td>-0.1666</td>
<td>0.1919</td>
</tr>
<tr>
<td>5</td>
<td>-0.1197</td>
<td>0.1736</td>
</tr>
<tr>
<td>7</td>
<td>-0.0765</td>
<td>0.1623</td>
</tr>
<tr>
<td>9</td>
<td>-0.0899</td>
<td>0.1120</td>
</tr>
</tbody>
</table>
Figure 4.18 - Distributions of Odd Indexed Reflection Coefficients in the Database

In Section 4.3.8 we saw that increasing the number of user vectors significantly increased the performance of our IDVS system in terms of the maximum correlation difference. The distance measure that is given by (2.30) allows us to quantize the coefficients of the user vectors to fewer than 4 bits, which consequently allows us to increase the number of user vectors. If we use 3 bits per coefficient, then we can use thirteen user vectors, at five coefficients per vector. We ran a test in which we used the distance measure (2.30) instead of the standard distance measure. Other differences from the standard configuration were that we used 3 bits per coefficient and thirteen user observation vectors. We show the maximum correlation difference in Figure 4.19 for each of the users and the statistics for the user’s maximum correlation difference in Table 4.14.
The results in Figure 4.19 and Table 4.14 show that this final test configuration is the best that we have simulated at least in terms of the maximum correlation difference. Once again, note that the major advantage to this configuration is the added number of user observation vectors. Another advantage is the fact that the vectors are gathered in this test over a longer period of time. Therefore, this configuration forces the commercial to
match over a longer period of time. The result of having an extra number of observation vectors that are spaced over a longer period of time is improved performance in terms of the maximum correlation difference, as we see in Figure 4.19 and Table 4.14.
5.0. CONCLUSION

In this chapter we summarize the conclusions that are obtained from the results of each of the sets of experiments that were described in Chapter 4. We use these conclusions to construct an ‘optimal’ system that we will evaluate in terms of performance and feasibility. Then we discuss some of the directions that this research could take in the future that demonstrate the promise of providing an enhanced system that has better performance and/or is more feasible.

5.1 The 'Optimal' System

5.1.1 Summary of Conclusions

In reviewing the tests in which the distance measure given by correlation coefficient is used we see that the parameters and techniques that are the best choices for our IVDS application include the standard configuration except with the database frames overlapped by 50% in time instead of 75%. The reason for the choice concerning the overlap is the fact that the increase in performance that is obtained with the use of 75% overlap is not enough to justify slowing down the identification by approximately 33%. We also see that the correlation coefficient is a better distance measure for our IVDS application in terms of the maximum correlation difference than the altered form of this calculation, where we subtract out the mean of all of the vectors in the database from each vector, only when the number of bits per coefficient is higher than three or four. Also, because of the zero
variance problem, which we discussed in Section 4.3.10, the correlation coefficient breaks down at three bits per coefficient and is no longer a valid distance measure. However, in Section 4.3.11 we see that it is possible to use three bits per coefficient if we use as our distance measure the altered version of the correlation coefficient that rarely produces a zero vector. This enables us to include more user observation vectors in the identification process, and increases the performance of our IVDS system in terms of the maximum correlation difference. Therefore, the 'optimal' system for our IVDS application in terms of the maximum correlation difference would consist of the parameters and techniques that are given in the following listing:

Autoregressive Model Order of ten using autocorrelation method
Parameterization: odd numbered reflection coefficients
Overlap in Database: 75%
Number of Bits per Coefficient: 3
Number of User Vectors: 13
Spacing of User Vectors: 1/4 second
Bandpass Filter: Passband - (409-6736) Hz. Kaiser windowed FIR(183; β = 8)
Preemphasis: 1 – 0.95z^{-1}, applied to every frame
Window: 20ms rectangular
Distance Measure: Equation (2.30)

The system parameters and techniques in this list are based upon the analysis of all of the results collected and they are the most nearly optimal ones that we have simulated in terms of the maximum correlation performance. These parameters and techniques comprise the best solution to our IVDS problem that we have found. However, certain
changes, such as overlapping the database frames in time by 50%, will not decrease the system performance in terms of the maximum correlation difference very much, as seen in Section 4.3.5, and will serve to make our IVDS system more feasible. We discuss the feasibility of such a system in the Section 5.1.3.

5.1.2 Performance of the 'Optimal' System

The graph in Figure 5.1 shows the maximum correlation difference, which is given by (2.33). The maximum correlation difference is the difference between the maximum correlation value of the commercial that the particular 'user' was truly watching and the maximum correlation value obtained in the correlation of the user observation vectors with all of the other commercials in the database.

![Graph showing correlation difference](image)

Figure 5.1 - Performance of the 'Optimal' IVDS System
The mean of the data points in Figure 5.1 is 0.2180 and the standard deviation is 0.0803, so the ratio $\frac{\mu}{\sigma}$ is equal to 2.7158, and the closest data point to being in error is at approximately 0.0972.

The figure shows that the system works every time with some margin of safety in each case. This system transmits exactly 195 bits for each selection that is made by any given user, as can be seen in the following calculations.

$$\left( \frac{13}{\text{vectors}} \right) \left( \frac{5}{\text{coefficients/\text{vector}}} \right) \left( \frac{3}{\text{bits/\text{coefficient}}} \right) = \frac{195}{\text{bits/\text{user selection}}}$$

(5.1)

The individual correlations of the user observation vectors with each commercial in the database can be seen in Figure 5.2. Note that Figure 5.2 shows that each of the users selected commercial number one, which is the correct result. The solid line connects the maximum correlations of the wrong commercials that are the highest for each user.

5.1.3 Feasibility of the 'Optimal' System

We will now consider the feasibility of the technique that we developed as a solution for our IVDS problem. We will assume that we have a machine that can perform one multiply accumulation (MAC) or one sum in $10^{-8}$ seconds. We will also assume that we have forty commercials that are fifteen seconds in length in the database. The number of operations that it takes to calculate the distance between one user vector and a vector in
the database was given in Section 2.3.3.5 and it is given by

\[
\text{number of operations per correlation} = 4D + 9, \quad (5.2)
\]

where \( D \) is the dimension of the two vectors that we are calculating the distance between.

For our 'optimal' system \( D = 5 \). This gives us \( 4 \times 5 + 9 = 29 \) operations per distance.

![Graph showing maximum correlations for each user per commercial](image)

**Figure 5.2 - Maximum Correlations for Each User per Commercial**

calculation, and as we see from (2.30), we need to perform \( N \) of these. In our 'optimal' system \( N = 13 \). Therefore, we need to perform \( 29 \times 13 = 377 \) of these calculations per shift of the user vectors through the database (for each \( \tilde{\rho}_2(j, k) \)). We need to calculate \( i \rho \) of
\( \tilde{\rho}_2(j,k) \) per commercial, where \( I_p \) is given by (2.28). Therefore, we need to perform

\[
\frac{15 - 0.25 \ast 13}{0.020 \ast (1 - 0.5)} = 1175 \text{ operations per commercial in the database for each user}
\]

transmission that is received. We are assuming that we have forty commercials in the database so we need to perform \( 1175 \ast 377 \ast 40 = 17.719 \ast 10^6 \) operations per user. With a processor that can perform one of these operations every \( 10^{-8} \) seconds, we can process one user's received transmission in \( 17.719 \ast 10^6 \ast (10^{-8}) = 0.1772 \) seconds. So, every 15 minutes we can handle \( \frac{15 \ast 60}{0.1772} \approx 5079 \) user's responses.

In this system the user needs to be informed that they can not make a selection within the last 3.25 seconds of the commercial. This is somewhat of a limitation. However, the speed of identification is also a limiting factor to the feasibility of our IVDS system. However, there are tradeoffs possible near the 'optimal' system. We can optimize for speed of identification by lowering the number of user observation vectors that are included in the identification process. However, we need to make sure that we do not exceed 200 bits per user transmission, or allow the maximum correlation difference to reach zero.

In conclusion, we have shown that the solution that we have proposed for our IVDS application works for a limited number of users and database commercials as well as for the test conditions that are described in Section 4.1. It is a good solution for small scale applications, defined as cases where usage is not heavy and the 'commercials' have a
shorter length in time.

5.2 Future Work

While the system described Section 5.1.1 is based upon the analysis of all of the results we have collected so far and it is the most nearly optimal system in terms of the maximum correlation performance that we have simulated, it is not the truly optimal system and more testing needs to be done to determine exactly what the optimal system would be for solving our IVDS problem. Additional testing could be done to determine the result of using even fewer bits per coefficient in our system. Also, further testing could be done to optimize the other parameters for the distance measure of (2.33). More testing could be done in an effort to determine if (2.33) is better than (2.28) for larger databases and more noise.

Much research can still be done for our IVDS application, both in the direction that this thesis has taken in solving the problem as well as in other directions. Extensive testing with the ‘optimal’ system with larger databases and varying room noises would go a long way towards ascertaining robustness of our IVDS system. If increased performance is required, a different parameterization could possibly yield better results. A large amount of entropy of the vector angles implies a minimum amount of mutual information between the vector components. The more the feature vectors in the database are spread out in space in terms of the angles between them, the more robust the system will be to room noise.
The optimal choice of an identification technique in terms of performance in the presence of noise is the one that excludes all information except that which is orthogonal to the noise. In terms of improving the identification performance with a different identification technique, cumulants or higher order statistical techniques could focus in on the nonlinear relationships between the two vectors that are being compared. It is exactly the nonlinear dependencies between vectors that the correlation coefficient ignores. However, distance calculations using other more complicated measures will take more time, as is shown in Figure 2.3.

If we use vector quantization we can increase the amount of information that can be transmitted. It may even improve performance of the system. However, the additional improvement may be limited by the size of the database and how much it is expected to change over the course of time. Vector quantization also adds to the identification time at the host side of the system, because each vector index that is transmitted must be identified with a set of vector components before the correlation coefficients can be calculated, which takes some extra time. Also, the vector quantization process at the user end will take time and there will be a limit to the number of vector indices that can be produced at the user end per frame of audio.

A mark to strive for in this application is to reach a point at which the performance is good enough that a threshold for the maximum correlation can be set. Two types of thresholds could be set, depending on the performance that is achieved. If the performance is somewhat better, a threshold could be set so that, if the maximum
correlation with all of the commercials is below the threshold, then we could know that
the user was not watching any of the commercials in the database. The second threshold
could be set fairly close to one so that, if any single correlation reaches that point, we
would know immediately that the user selected that commercial. Hence, the identification
speed would be increased, because we would not have to look at any more commercials
for that user.

Another, totally different, approach to our IVDS problem would be to insert a code
into the commercial and have the user sample the commercial while the code is being
played, which yields several advantages with only one cost. First, if we send a commercial
identification number from the host side of the system to the user side, then the only
'identification' that needs to be done is in the reception of the code at the user end, which
would alleviate the computational burden at the host side of the system. Another
advantage is that by having the ability to insert a code, we have the freedom to choose
whatever method of communicating the commercial identification information we want, as
long as the sound of it is not objectionable to the user. Many communication schemes use
a finite set of signal vectors which have very good geometric properties, much better than
the properties of the signal vectors that are obtained at arbitrary times from segments of
arbitrary portions of audio. Therefore, we would expect better performance. Also, by
adding a code that the user must receive before transmitting to the host side, we gain the
ability to control the usage of the system. We can solicit a product only at times when we
are ready to process the user's requests, which may at times be advantageous.
The additional control over the usage of the system, and improvements in performance and identification time come at the cost of causing deterioration of the sound of the commercial. The major tradeoff is making the code loud enough to be able to detect it accurately without making it objectionable to the user, as well as in making the detection robust against room noise.
APPENDIX A

The Levinson-Durbin Algorithm

The Levinson-Durbin algorithm as applied to the frame \( n \in [m-N+1, m] \) of a signal \( x \) in order to obtain the autoregressive parameters \( A(i) \) for \( i=1,2,...M \) and the reflection coefficients \( K(i) \) for \( i=1,2,...M \), where \( M \) is the order of the linear predictive analysis, is given by the following listing of code. It should be noted that \( R \) is the autocorrelation matrix, which is defined as

\[
R = \begin{bmatrix}
    r_{xx}[0] & r_{xx}[-1] & r_{xx}[-2] & \cdots & r_{xx}[-(N-1)] \\
    r_{xx}[1] & r_{xx}[0] & r_{xx}[-1] & \cdots & r_{xx}[-(N-2)] \\
    r_{xx}[2] & r_{xx}[1] & r_{xx}[0] & \cdots & r_{xx}[-(N-3)] \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    r_{xx}[N-1] & r_{xx}[N-2] & r_{xx}[N-3] & \cdots & r_{xx}[0]
\end{bmatrix}
\]

\( r_{xx}[k] \) is the autocorrelation of \( x \) at lag \( k \).

real\( (Z) \) indicates the real part of the variable \( Z \).

imag\( (Z) \) indicates the imaginary part of the variable \( Z \).

conj\( (Z) \) indicates the complex conjugate of the variable \( Z \).
\( \rho_0 = R(1) \)
\( AA(1,1) = -R(2)/R(1) \)
\( K(1) = AA(1,1) \)
\( \rho(1) = (1-(\text{real}(AA(1,1)).^2+\text{imag}(AA(1,1)).^2))*R(1) \)
\( A(1) = AA(1,1) \)

if M is equal to 1 then stop here, otherwise continue

for k=2 to M
  \( B = -R(k+1) \)
  for I=1 to (k-1)
    \( B = B-AA(I,k-1)*R(k+1-I) \)
  end
  \( AA(k,k) = B/\rho(k-1) \)
  \( K(k) = AA(k,k) \)
  for I=1 to (k-1)
    \( AA(I,k) = AA(I,k-1)+AA(k,k)*\text{conj}(AA(k-I,k-1)) \)
  end
  \( \rho(k) = (1-(\text{real}(AA(k,k)).^2+\text{imag}(AA(k,k)).^2))*\rho(k-1) \)
end

The AR parameters \( A(1) - A(M) \) are given by \( AA(1,M) \), \( AA(2,M) \) ... \( AA(M,M) \)
REFERENCES


VITA

Gregory Sheets was born in Huntington, West Virginia on May 11, 1970. He received his Bachelor of Science degree in Electrical Engineering from West Virginia Institute of Technology in December of 1992. He has co-oped seven terms with Ericsson Inc. working on hardware and software design for mobile and portable radios, as well as some software design for testing cellular telephones. In the Fall of 1993, he entered graduate school at Virginia Polytechnic Institute and State University in the pursuit of a Master’s degree. In January of 1995 he began working in the Digital Signal Processing Research Lab at Virginia Polytechnic Institute and State University, where his research efforts were aimed at developing efficient ways to code and identify audio for an interactive wireless television application. He will obtain his Master’s degree and begin work at Ericsson Inc. in Research Triangle Park, North Carolina in May of 1996. He is a member of Tau Beta Pi and Eta Kappa Nu honor societies.