University of Alberta

Implementation and Evaluation of an Accurate Real-Time Voiceband Signal Classifier

by

Deepak Prasad Sarda

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Science.

Department of Electrical and Computer Engineering

Edmonton, Alberta, Canada

Spring 1999

0-612-47157-8

University of Alberta

Library Release Form

Name of Author: Deepak Prasad Sarda

Title of Thesis: Implementation and Evaluation of an Accurate Real-Time Voiceband Signal Classifier

Degree: Master of Science

Year this Degree Granted: 1999

Deepak P Sarda

Deepak Prasad Sarda
15 Amber Crescent
St. Albert, Alberta
CANADA T8N 2J2

Date: Jan 26, 1999

V. 80[illegible]

UNIVERSITY OF ALBERTA

October 15, 1998

Bill M. Boland
Executive Editor
The New York Academy of Sciences
2 East 63rd Street
New York, New York 10021

Dear Dr. Boland,

I wish to request a letter of permission to include a version of the paper "Winrow, C.J., Kassam, A., Miyata, K.S., Marcus, S.L., Hunter, J., Capone, J.P., and Rachubinski, R.A. (1996) Interplay of the peroxisome proliferator activated-activated receptor and thyroid hormone receptor signaling pathways in regulating peroxisome proliferator-responsive genes. Ann. N.Y. Acad. Sci. **804**:214-230." as a chapter in my Ph.D. thesis. In addition to being archived at the University of Alberta, the thesis will be transferred to microfilm and stored at the National Library of Canada. Please contact me if you require any further information.

Thank you for your assistance in this matter.

Sincerely,

Christopher Winrow

Department of Cell Biology and Anatomy
University of Alberta
6-50 Medical Sciences Building
Edmonton, Alberta
Canada
T6G 2H7

Tel. (403) 492-7407
Fax (403) 492-0450
e-mail: cwinrow@anat.med.ualberta.ca

[illegible]
BY THE [illegible]
SCIENCE[illegible]
OBTAIN[illegible]
PUBLISHED [illegible]

Bill Boland 11-4-98
BILL BOLAND (DATE)

UNIVERSITY OF ALBERTA

October 15, 1998

Journal Editorial Office
The Journal of Biological Chemistry
9650 Rockville Pike
Bethesda, MD 20814
U.S.A.

Fax (301) 571-1824

To Whom It May Concern:

I wish to request a letter of permission to include a version of the paper "Winrow, C.J., Capone, J.P., and Rachubinski, R.A. (1998) Crosstalk between orphan nuclear hormone receptor RZRα and peroxisome proliferator activated-activated receptor α in regulation of peroxisomal hydratase-dehydrogenase gene. J. Biol. Chem. (in press)" as a chapter in my Ph.D. thesis. In addition to being archived at the University of Alberta, the thesis will be transferred to microfilm and stored at the National Library of Canada. Please contact me if you require any further information.

Thank you for your assistance in this matter.

Sincerely,

Christopher Winrow

Department of Cell Biology and Anatomy
6-50 Medical Sciences Building
University of Alberta
Edmonton, Alberta
Canada
T6G 2H7

Tel. (403) 492-7407
Fax (403) 492-0450
e-mail: cwinrow@anat.med.ualberta.ca

PERMISSION GRANTED
contingent upon obtaining that of the author

for the copyright owner
THE AMERICAN SOCIETY FOR BIOCHEMISTRY & MOLECULAR BIOLOGY

OCT 15 1998

TOTAL P.02

UNIVERSITY OF ALBERTA

October 15, 1998

Gene Expression
Cognizant Communication Corporation
3 Hartsdale Road
Elmsford, N.Y. 10523-3701
U.S.A.

Fax (914) 592-8981

To Whom It May Concern:

I wish to request a letter of permission to include a version of the paper "Winrow, C.J., Marcus, S.L., Miyata, K.S., Zhang, B., Capone, J.P., and Rachubinski, R.A. (1994) Transactivation of the peroxisome proliferator activated-activated receptor is differentially modulated by hepatocyte nuclear factor-4. *Gene Expression* 4:53-62." as a chapter in my Ph.D. thesis. In addition to being archived at the University of Alberta, the thesis will be transferred to microfilm and stored at the National Library of Canada. Please contact me if you require any further information.

Thank you for your assistance in this matter.

Sincerely,

Chris Winrow

Christopher Winrow

Department of Cell Biology and Anatomy
University of Alberta
6-50 Medical Sciences Building
Edmonton, Alberta
Canada
T6G 2H7

Tel. (403) 492-7407
Fax (403) 492-0450
e-mail: cwinrow@anat.med.ualberta.ca

Permission Granted by:
Cognizant Communication Corporatio

Title: GENE EXPRESSION
Volume # 4 Page # 53-62
Date: 10/15/98 Fee: NONE
Reference to the orginal source of Publication is required.

Speak to Frances   OCT 1998

From: Chris Winrow <CWINROW@anat.med.ualberta.ca>
To: ES_OXF1.SSERV(Permissions)
Date: 15 October 1998 5:16pm
Subject: Re: Letter of Permission

Dear Dr. Rothwell,

I wish to request a letter of permission to include a version of the paper by Winrow et al. (1995) Mol. Cell. Endocrinol. 111:175-179. as a chapter in my Ph.D. thesis. In addition to being archived at the University of Alberta, the thesis will be transferred to microfilm and stored at the National Library of Canada. Please contact me if you require any further information.

- Molecular + Cellular Endocrinology

Thank you for your assistance in this matter.

Sincerely,

Christopher Winrow

Department of Cell Biology and Anatomy
6-50 Medical Sciences Building
University of Alberta
Edmonton, Alberta
Canada
T6G 2H7

Tel. (403) 492-7407
Fax (403) 492-0450
e-mail: cwinrow@anat.med.ualberta.ca

Thesis

We hereby grant you permission to reprint the material specified overleaf free of charge in both **print and electronic format**, subject to the following conditions:

1. If any part of the material to be used (for example, figures) has appeared in our publication with credit or acknowledgement to another source, permission must also be sought from that source. If such permission is not obtained then that material may not be included in your publication/copies.

2. Acknowledgement to the source is given as follows:
For Journals: "Reprinted from Journal title, Volume number, Author(s), Title of article, Page No's., Copyright (Year), with permission from Elsevier Science"
For Books: "Reprinted from (Author/Title), Copyright (Year), Pages No., with permission from Elsevier Science"

3. Reproduction of this material is confined to the purpose for which permission is hereby given and is granted for non-exclusive world **English** rights only.

For future permissions, please contact:
Subsidiary Rights Department
Elsevier Science
PO Box 800
Oxford OX5 1DX
UK
Email: permissions@elsevier.co.uk
Fax: (+44) 1865 853333

(for Elsevier Science)

Should your thesis be published commercially, please reapply for permission

This includes permission for the National Library of Canada to supply single copies, on demand, of the complete thesis.

Should your thesis be published commercially, please reapply for permission.

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled Implementation and Evaluation of an Accurate Real-Time Voiceband Signal Classifier submitted by Deepak Prasad Sarda in partial fulfillment of the requirements for the degree of Master of Science.

Dr. B. F. Cockburn, Supervisor

Dr. W. D. Grover, Internal Examiner

Dr. W. Pedrycz, Internal Examiner

Dr. I. Nikolaidis, External Examiner

Date: Jan. 26, 1999

Dedicated to my loving parents
Bala and Susila

Abstract

This thesis describes the implementation and evaluation of an accurate real-time voiceband signal classifier for use on T1 trunks in the telephone network. The classifier is implemented using a standard PC, with a T1 and DSP card. The classifier is trained to recognize a total of 12 signal classes. These include 4 data modem classes, 4 facsimile classes, random binary, FSK signalling, ringback, and a class containing 12 DTMF tones. The signal data is first segmented and then classified using both linear and quadratic discriminant functions. A total of 11 feature variables are used by the discriminant functions including the first 10 values of the normalized ACS, and the normalized second-order central moment. A third hybrid discriminant function was also evaluated that based its decisions on the results of the LDFs and the QDFs. The measured classification accuracy for all classes approaches 100%, for either the linear and hybrid discriminant functions and a segment size of 2052 samples. The classifier was evaluated experimentally at four field trials. During the field trials many different T1s were monitored, with a variety of different signal mixes and traffic patterns. From the resulting experimental databases, busy hour and pie chart graphs were generated which showed the classification of the traffic into the 12 different classes.

Acknowledgements

First, I would like to thank my supervisor Dr. Bruce F. Cockburn for all of his patience, support, and guidance that he has shown over the past few years. I would also like to thank him for giving me the opportunity to work on the signal classifier project, and for providing financial support. I would also like to thank TR*Labs* for providing financial support and for the opportunity to work on the signal classifier project.

A number of people from many different organizations made contributions that were significant to this research work. This section attempts to acknowledge these individuals and their efforts. I would like to thank:

- Jeremy Sewall, former TR*Labs* graduate student, for his work on the initial classification algorithms that are used in the signal classifier.
- Dr. Wayne Grover, Dr. Ioanis Nikolaidis, and Dr. Witold Pedrycz for reading and examining this thesis.
- Mr. Fred Halsey, TELUS Industrial Representative for TR*Labs*, for introducing us to the appropriate contact personnel at TELUS to arrange for the first and second field trials.
- Mr. Roy Lapp, TELUS V-P Engineering, for his approval for TELUS's co-operation with the signal classifier project.
- Mr. Harold Hoffmann, TELUS Manager Data Networks and Technical Support, for his approval for cooperation with TELUS maintenance personnel.
- Mr. Paul Holomis, TELUS Maintenance Engineering, for this help in organizing the first and second field trials, and his continued support and interest shown for the signal classifier project.
- Mr. Del Roes and Mr. Nigel Russell, TELUS Maintenance Engineering, for their technical help in setting up the telephone equipment at the first field trial.
- Mr. Bill Bartel, TELUS Manager, for his approval to conduct the second field trial at the TELUS Toll Building in Edmonton.
- Mr. Dennis Holyk, Technician TELUS Toll Building, for his technical help given throughout the second field trial.

- Mr. Ken Thiessen, and Mr. Mike Hocken, TELUS Mobility, for their approval to conduct the third and fourth field trials.

- Mr. Dan Kenway and Mr. Bruce Ramage, VisionSmart, for their help in organizing the third field trial at TELUS Mobility.

- Mr. Rocky Panizzon, Mr. Stephen Lau, and Mr. Doug Sadler, TELUS Mobility Technicians, for their technical help given at the third and fourth field trials. You were all so busy, but you always found time to help us out.

- Dr. Ron Kube, Dr. Bill Armstrong, Mr. Monroe Thomas, and Mr. Kyle Palmer, Dendronic Decisions Limited, for their initial analysis of the data using the ALN discrimination methods and for providing an educational license for their ALN software to TR*Labs*.

- Mr. Allan Pollard, Manitoba Telephone Systems Engineer, for initially proposing the signal classifier project and for his continued interest and support.

- Mr. Peng Mok, TRLabs graduate student, for his continued help in the understanding of the principles of digital communications and for his programming expertise.

- Mr. David Clegg and Mr. Tim Friesen, TR*Labs* employees, for their help in debugging and fixing a hardware problem with the DSP card. Thanks a lot guys. I don't think we would have found the problem without you.

- My wife, life-long partner, and closest friend Romina for all of her understanding and support.

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

Chapter 1

1.0 Introduction

This chapter briefly outlines the problem of signal classification in the Public Switched Telephone Network (PSTN). The main motivations and potential benefits of this research work are also given.

1.1 Signal Classification in the PSTN

Historically, telephone companies (telcos) have dealt with mainly voice calls within the PSTN. Thus networks have been designed to best handle the statistical patterns of voice calls. Increasingly, however, the PSTN is being used to transport much more than just voice calls as shown in Figure 1. Most businesses and many homes now have fax machines for transmitting scanned and encoded images. With the explosive growth of the Internet, many computer users access electronic services and "surf the web" via data modem (modulator/demodulator) connections through the PSTN. With the increasing usage of modems and fax machines, the original network design parameters are no longer valid. If telcos could distinguish and measure the usage patterns of these different signal types, they would be in a better position to provision their network to efficiently and profitably handle the changing mixture of network traffic.

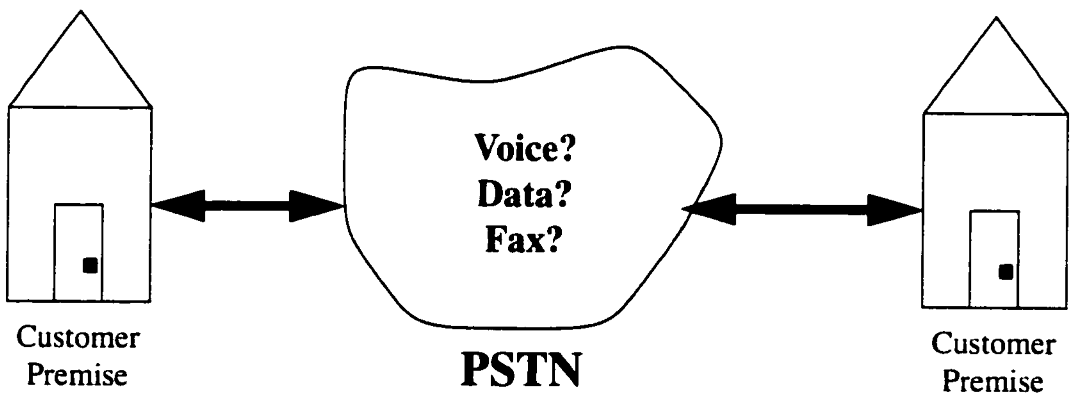


Figure 1: PSTN traffic.

1.2 Potential Benefits of Signal Classification

The ability to characterize traffic on the PSTN should benefit both telcos and customers. If telcos could classify the traffic, they could modify their billing rates to better reflect demand. They could also encourage customers to purchase special lines for their specific needs. For example, if telcos are certain that customers are using voice lines for mainly Internet usage, they could encourage the customers to purchase much faster Internet access services such as ADSL (Asymmetric Digital Subscriber Line). This would divert Internet traffic off of the PSTN, which is advantageous to the telcos, and customers would be getting services that would be better suited to their specific needs. Telcos could also apply different compression algorithms to different signal types if they can distinguish between voice and different classes of non-voice traffic. For example, if a signal is classified as voice, then the telcos could consider applying aggressive lossy compression algorithms; however, if the signal is data then only lossless compression algorithms would be acceptable.

1.3 Presently Available Signal Classifiers

Commercial signal classifiers have already been available for several years, however these units tend to have several drawbacks. First, the equipment is bulky and expensive. Second, some classifiers only differentiate between voice and non-voice traffic. Third, some classifiers need to monitor the entire call to catch the initial setup information communicated between two modems. Finally, the accuracy of the equipment has been found wanting by one of the telco sponsors of TR*Labs* (Telecommunications Research Laboratories). New attempts to approach the voiceband signal classification problem led to the M.Sc. work of Jeremy Sewall.

The proposed classifier is based on algorithms developed by Jeremy Sewall during his work as a masters student at TR*Labs* and the University of Alberta. Figure 2 shows the basic structure of the algorithms developed by Sewall. These algorithms involve computing 11 feature variables, namely the first 10 autocorrelation sequence lags and the second order-central moment, and then using discriminant analysis to make a decision as to which signal class a finite segment of observed data most likely belongs. The algorithms are very simple, accurate, and easily computed for up to 24 simultaneous digital voice channels using a modest 40 MHz Digital Signal Processor (DSP) [31].

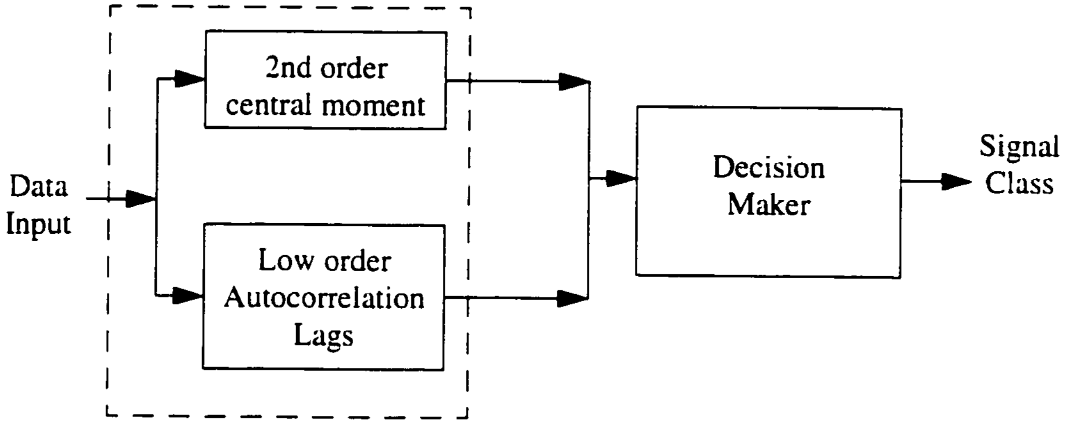


Figure 2: Classification algorithm structure.

1.4 Research Summary

This thesis re-examines and improves the algorithms developed by Sewall, and evaluates them experimentally in a field trial using a prototype classifier. A testbed classifier was developed on a Personal Computer (PC) under the MS-DOS Operating System (OS). The classifier was taken on several field trials to verify its performance for real signals in a real PSTN. The experience of the field trials caused us to add new signal classes to the ones proposed by Sewall. The data gathered by the field trials allowed us to retrain the classifier to improve its accuracy, and to help us better understand the limitations of the classifier. As well as the classifier studies, off-line numerical work was done to evaluate the ability of Quadratic Discriminant functions and Adaptive Logic Networks (ALNs) to classify the signal types. Also, the feature variables were analyzed to determine which subset of variables would yield the most accurate classifications for each method used.

1.5 Thesis Organization

This thesis contains eight chapters. The next chapter provides additional background information followed by a brief description of the research infrastructure. This will be followed with a discussion on the implementation and evaluation of the classifier, and finally the results from both field trials and off-line simulations. This is followed by a chapter that gives a list of recommendations. Our concluding remarks include directions for potential future research.

Chapter 2

2.0 Background

This chapter reviews research previously conducted by Nevio Benvenuto and Jeremy Sewall, which is the prior work most relevant to this thesis. This chapter will also briefly review technical and theoretical background information that is required in later chapters.

2.1 Previous Work on Voiceband Signal Classification

The signal classification problem is not a new problem and considerable previous research has been done in this area. Some researchers have simply looked at distinguishing between speech and non-speech. Others have looked at classifying different types of voiceband data. The methods employed, in many cases, are very similar including the use of feature variables which are then passed on to some type of decision maker. Popular feature variables include zero crossings and short-time energies, both of which provide information of the signal's power spectrum. If signals have different power spectra they should be easier to classify. One straightforward method would be to compute the spectrum of a signal, and then use a template matching method to classify it into one of the candidate classes. This process is computationally intensive and complicated if many different signals need to be classified in parallel. For a more complete discussion of the previous research work done please refer to [2].

Commercial classifiers are presently available from: CTel (Compression Telecommunications Corporation, Germantown, MD), Tellabs (Lisle, IL), AT&T, and MPR Teltech (Burnaby, BC) [2]. The CTel classifier (NET-MONITOR System 2432) classifies calls into three general categories: voice, data, and facsimile. DSP's are used to perform traffic classification, tone detection, demodulation, and spectral analysis. The Tellabs Digital Channel Occupancy Analyser (DCOA) provides features very similar to the CTel classifier. Using the DCOA, traffic analysis for 10 channels requires 1.1 seconds. The AT&T Voice/Data Call Classifier classifies calls into voice or voiceband data. The voiceband data category is subclassified into high, medium, and low bit rate connections. The MPR Teltech classifier is called a Service Discrimination Unit (SDU). The SDU is capable of monitoring up to 16 channels, out of 24, on a T1 in real-time. It uses four Motorola DSP56001 DSPs (24 MHz clock), and is controlled by two Motorola 68HC11 processors.

The Tellabs DCOA was tested by a TR*Labs* telco sponsor, and was found to have an accuracy of only approximately 72% [2]. The other systems have not been tested, nor have their accuracies been reported by the manufacturer. Also, many of the classifiers rely on knowledge of the call boundaries. In these cases the classifier most likely relies on call set-up information present at the start of modem and facsimile calls. Also, many of the units are quite large and expensive.

The next section will look at the relevant research work of Benvenuto, followed by the work of Sewall.

2.1.1 Benvenuto's Classification Methods

Benvenuto considered the problem of distinguishing between speech and data [1], different voiceband data signals [3], and different modem types and bit rates [4]. In [1] Benvenuto was able to correctly classify between speech and voiceband data with a misclassification rate of about 1% using an observation signal segment size of 32 ms. This was accomplished by calculating two discriminant variables, the autocorrelation at lag 2 ($\tilde{R}_\gamma(2)$), and the central second-order moment ($\tilde{\eta}_2$), of the envelope of a complex low-pass signal. A block diagram of this classifier is shown in Figure 3. The classification method is to first take the input signal samples and to derive a complex sequence, $\gamma(n)$, by first performing quadrature and in-phase demodulation using a

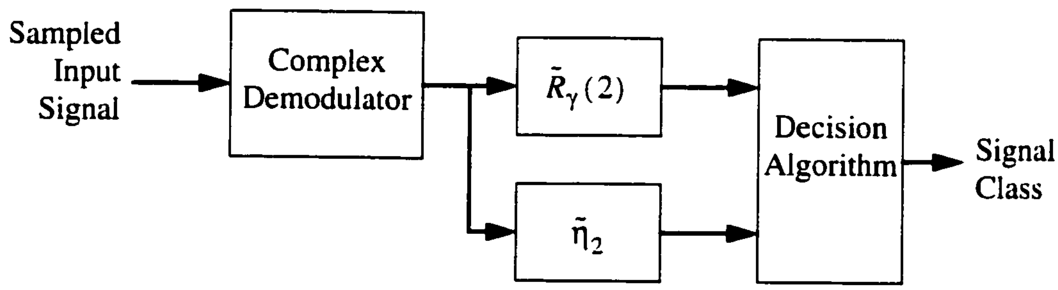


Figure 3: Block diagram of a Benvenuto classifier.

mid-passband carrier at 2 KHz and then low pass filtering. The mixing frequency of 2 KHz was chosen as an approximation to the unknown actual carrier frequency. This results in a discrete time complex baseband signal. The discrete time autocorrelation at lag k and at time n, using a window consisting of N consecutive samples, is given by equation (1).

$$R_{\gamma}(k) = \frac{1}{N}\sum_{i=1}^{N} \gamma(i+k)\gamma^{*}(i) \qquad \text{(Eq 1)}$$

Note that this is the biased estimate of the autocorrelation sequence because the summation is divided by the total number of terms, N, not the total number of non-zero terms, $N-|k|$, available at time N [5]. The normalized central second-order moment is given by

$$\tilde{\eta}_2 = \frac{m_2}{m_1^2} - 1, \qquad \text{(Eq 2)}$$

where m_1 and m_2 are defined in equation (3) and equation (4) respectively.

$$m_1 = \frac{1}{N}\sum_{i=1}^{N} |\gamma(i)| \qquad \text{(Eq 3)}$$

$$m_2 = \frac{1}{N}\sum_{i=1}^{N} |\gamma(i)|^2 \qquad \text{(Eq 4)}$$

The normalized central second-order moment is in fact simply the variance of the signal normalized by the square of the mean value of the signal. This can be shown by first looking at the variance of a random signal [6]

$$\sigma_x^2 = E\left(X^2\right) - E(X)^2, \qquad \text{(Eq 5)}$$

and substituting m_2 for $E\left(X^2\right)$, m_1 for $E(X)$, and finally dividing by m_1^2 (the signal mean squared). Next, the signal amplitude $|\gamma(n)|$ is compared against a minimum threshold, γ_{Th}. If $|\gamma(n)| > \gamma_{Th}$, then the next step is to compute the signal energy over a small window containing L samples. If $|\gamma(n)| < \gamma_{Th}$ then the window of samples is discarded and the next window is checked. This continues until a satisfactory window of $N \geq L$ consecutive samples is accumulated. These calculations will provide the discriminant variables needed by the decision maker.

Benvenuto found that when $\tilde{R}_d(2) > 0$ and $\tilde{\eta}_2 < 0.3$, the signal was most likely data, otherwise it should be classified as speech. This provides a simple means of discriminating between speech and data, as illustrated in Figure 4. Using this technique Benvenuto was able to achieve a very high classification accuracy using 32 ms windows.

Benvenuto's work was the starting point for the algorithms developed by Sewall, which will be discussed in the next section.

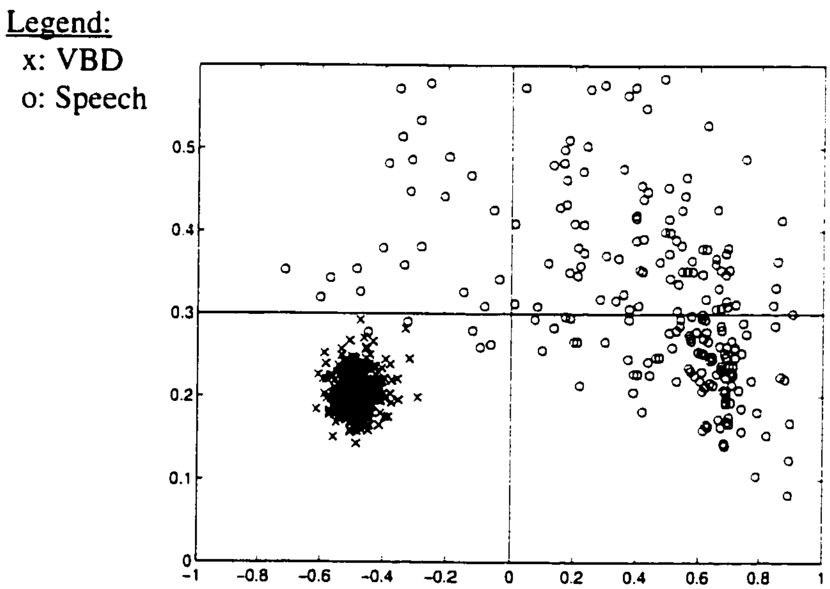


Figure 4: Scatter plot for one VBD signal and one speech signal [2].

2.1.2 Sewall's Classification Algorithms

Sewall's improved algorithms were able to distinguish between a greater number of signal types, including speech and several types of voiceband data, with often near-perfect accuracy. The main changes to Benvenuto's method were (1) to analyze the passband signal directly rather than an approximately demodulated baseband signal, (2) to use an unbiased estimate of the Autocorrelation Sequence (ACS), and (3) to use up to ten low-order lags of the ACS as discriminant variables.

Sewall found little advantage in performing complex demodulation in order to classify a baseband signal. Instead his algorithm performed full-wave rectification on the passband signal, where rectification can be accomplished by simply stripping the sign bit from the Pulse Code Modulated (PCM) sample. This simplified operation results in a non-zero value for the normalized central second-order moment, which still retains useful information about the signal. This simplification does not change the equation for $\tilde{\eta}_2$, but the definitions for m_1 and m_2 must change as noted in equations (6) and (7), where $\hat{d}(i)$ denotes the real-valued, full wave rectified, passband signal segment under test.

$$m_1 = \frac{1}{N}\sum_{i=1}^{N}\hat{d}(i) \qquad \text{(Eq 6)}$$

$$m_2 = \frac{1}{N}\sum_{i=1}^{N}\hat{d}(i)^2 \qquad \text{(Eq 7)}$$

The estimate of the ACS needs to be rewritten as shown in equation (8). Note that this is the unbiased estimator for the ACS [5]. The unbiased estimator will simply divide by the number of nonzero terms, $N-|k|$, rather than the total number of samples, N.

$$R_d(k) = \frac{1}{N-|k|}\sum_{i=1}^{N-|k|} d(i+k)\,d(i) \qquad \text{(Eq 8)}$$

So why did Sewall consider the ACS in his search for effective signal features? Benvenuto had already shown that the central-second order moment is useful when discriminating between speech and non-speech [1]. In addition, the ACS had been shown to be useful when discriminating between voice and many different types of voiceband data [2]. The spectral characteristics of a stochastic signal, which are very helpful for distinguishing different signal types, are obtained by computing the Fourier transform of the autocorrelation function [6]. This means that the information that is present in

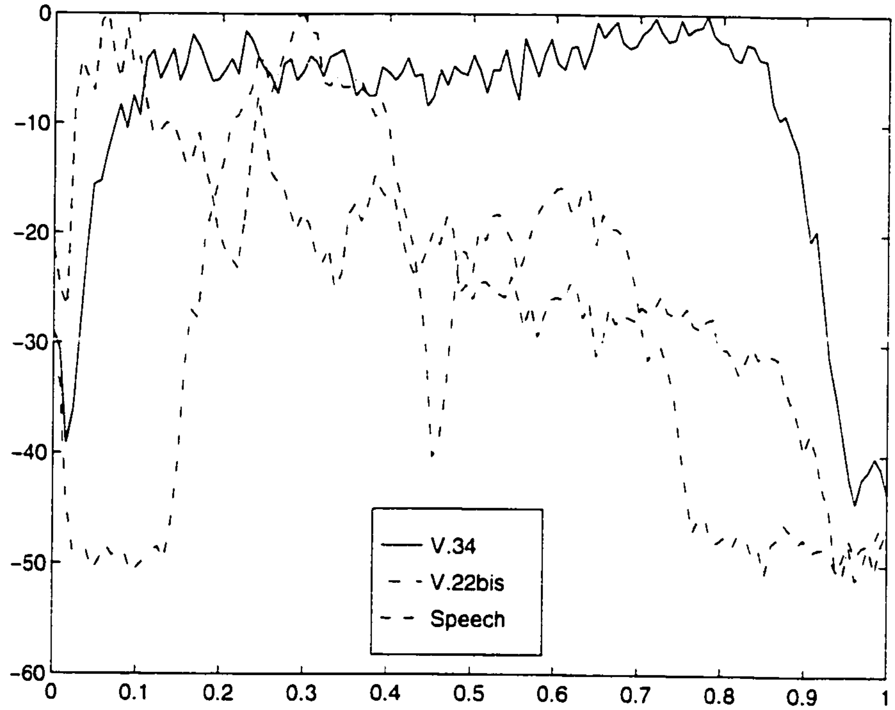


Figure 5: Power spectral densities of three voiceband signals [2].

the Power Spectral Density (PSD) function of a signal is also available in the ACS.

Figure 5 shows the PSD characteristics of three different voiceband signals sampled at 8000 Hz. The plot shows that the three signals have very different spectral characteristics. Figure 6 shows the first 20 values of the ACS for the same signals. Clearly the ACS are quite different for each signal, and the differences are most evident in the lower order lags. For lags much greater than 12 the ACS values for most voiceband signals tend to converge to zero. Hence Sewall decided to consider only the first ten values of the ACS (one to ten) as discriminant variables. By ignoring all but the first ten values of the ACS, only an estimate of the PSD is exploited by the classifier. But we will show later that the loss in information does not reduce the achievable classifier accuracy.

Using the central second-order moment and lag values of one to ten, Sewall's method can distinguish between the classes shown in Table 1.

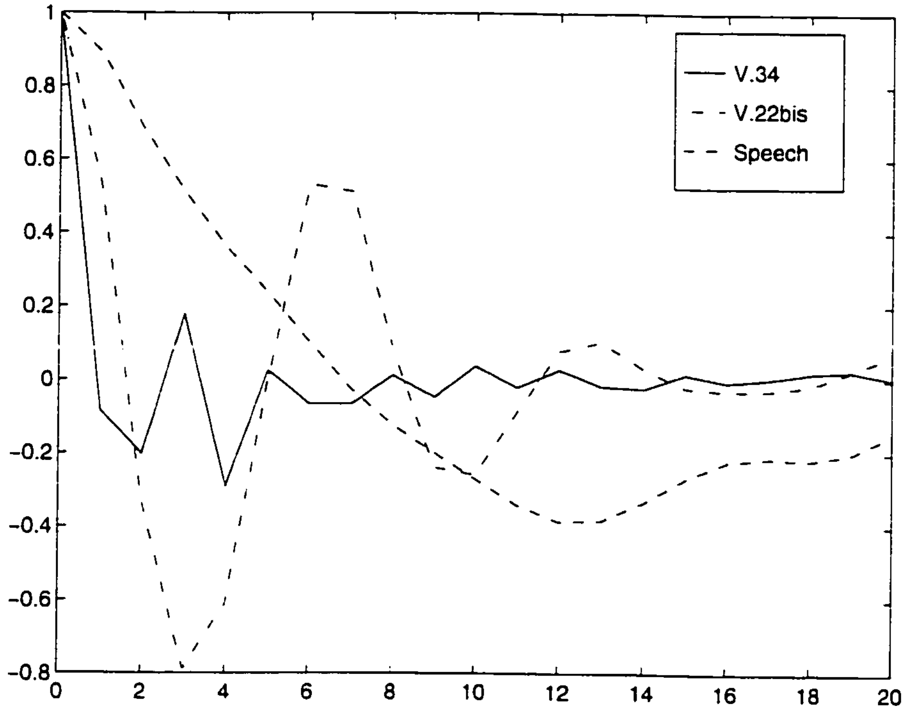


Figure 6: Autocorrelations of three voiceband signals [2].

Table 1: VBD, FAX, and speech subclassification classes.

Group No	Signals included
1	V.22 and V.22*bis* forward channels
2	V.22 and V.22*bis* reverse channels
3	V.34 at speeds greater than 14.4 Kbps
4	V.29 at all speeds
5	V.32, V.32*bis*, and V.17 at speeds greater than 2400 bps
6	V.27*ter* at 4800 bps
7	V.27*ter* at 2400 bps
8	Speech
9	Random PCM samples

Sewall used both linear and pseudo-quadratic discriminant functions to classify the signals based on the 11 feature variables. Sewall used the term "pseudo-quadratic" when referring to the method used by SPSS software version 6.1 when the "Separate-groups" covariance matrix option is selected. This option produces decision rules that operate on the values of the discriminant functions, not the original variables. For this reason pseudo-quadratic discrimination is not equivalent to true quadratic discrimination [7][8] but it has similar performance. The relevant theory will be reviewed later.

2.2 Public Switched Telephone Network (PSTN)

Many aspects of the PSTN will be briefly reviewed and discussed in this section. Figure 7 illustrates the signal path through the PSTN from one subscriber to

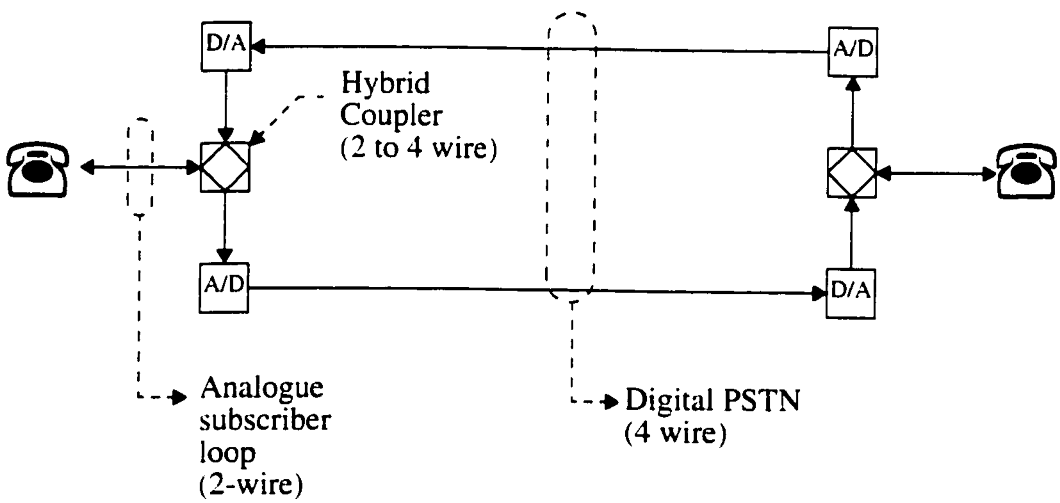


Figure 7: Customer to customer model.

another. An outgoing analogue signal originates at a telephone set at a subscriber's premise and is carried to the Central Office (CO) by a two-wire subscriber loop. Both the transmitted and received signals are superimposed together in the loop medium. Once at the CO, a so-called hybrid coupler separates the signal into the transmitted and received signals. This results in a four wire connection: one pair for the transmitted signal and one for the received signal. After the coupler, the transmitted signal is passed through an anti-aliasing filter (not shown) and then converted to digital form by an Analogue-to-Digital (A/D) converter. In practice the transmit-direction A/D and receive-direction D/A (Digital-to-Analogue) converters are implemented together in a so-called CODEC (Coder/Decoder) integrated circuit. Once at the destination end, the signal is converted into an analogue signal and the hybrid coupler does the four to two-wire conversion. The signal is then carried on another subscriber loop to the final destination. Two good reference books that discuss all aspects of the PSTN are [9] and [10].

2.2.1 Subscriber Loop

The subscriber loop is a twisted pair of copper wires that makes the electrical connection between the customer premise and the CO. All types of signals such as voice, data, and fax are carried on this one twisted pair. The subscriber loop introduces a variety of distortions on the signals, including frequency-dependent attenuation distortion and envelope delay distortion.

2.2.2 Central Office

Once the analogue signal reaches the CO, the Hybrid coupler separates the transmitted and received components, thus converting the signal from two to four-wire form. After this is done the transmitted signal is filtered down to the 300 Hz to 3400 Hz standard voiceband and then converted to a digital signal. For the PSTN the sampling rate is 8000 samples/second. After the signal is sampled each value is encoded as an 8-bit byte using either the μ-law or A-law PCM non-linear encoding methods. These encoding methods use non-linear amplitude compression, called companding, to ensure a roughly equal Signal-to-Noise Ratio (SNR), despite quantizing noise, at all signal amplitudes. A-law companding is used in European countries, while μ-law companding is used in North America. For a complete μ-law encoding/decoding table refer to Appendix B.

Each 8-bit PCM codeword comprises a 4-bit quantization code field, a 3-bit segment code field, and a sign bit. The quantization code makes up the lower order bits, and the segment code makes up the upper order bits. The sign bit (MSB) is set to 1 for negative and 0 for positive. Thus 00000000_2 and 10000000_2 are both considered to be 0. Before each encoded sample is transmitted through the network, all bits are actually inverted [10]. The decoding circuitry must therefore first invert all bits again before decoding each received μ-law encoded sample. Once the analogue signal is converted to digital form, no signal loss will occur during transport through the network until the signal is converted back and transmitted to the destination (assuming no bit errors). The digitizing process will of course add noise due to quantization, however this noise source is usually very small when compared to other potential line impairments over analogue connections.

After the PCM samples are encoded, the signal is switched through the PSTN to the destination. Once at the destination, the signal is converted back to analogue form and the hybrid coupler does the 4 to 2-wire conversion. The signal then goes over another subscriber loop to reach the destination.

2.2.3 Line Impairments

Line impairments can be introduced in the analogue parts of the network, including the subscriber loops and the hybrid couplers. A practical complication with the subscriber loop is that each one has different characteristics. For example, each loop will be of different length, and thus have different attenuation properties. Hybrid couplers need to interface electrically with each of these subscriber loops, and the resulting impedance mismatches can in turn be the source of many impairments.

Sewall demonstrated that the classification algorithms are capable of high accuracy even in the presence of severe line impairments [2]. For completeness some of the typical line impairments will briefly be discussed next.

2.2.3.1 Echo Delay

There are two types of echo delay: talker echo and listener echo. Talker echo occurs when a person is talking on the phone and hears their voice after a finite delay. This is most noticeable when talking to somebody over the long distance network. Listener echo is when the person will hear the same thing, but after a finite delay. Again this is most common in long distance phone calls. The reason for this is the impedance mismatch between the hybrid coupler and the subscriber loop, which causes the signal to be partially reflected. It is most noticeable in long distance calls because of the time delay incurred when the signal goes from the source to the destination.

2.2.3.2 Attenuation Distortion

Attenuation distortion is introduced while the signal is travelling down the subscriber loop. It is the result of the uneven amplitude-frequency response of the copper pair. This results in higher frequency signal components being attenuated more then lower frequency components [9]. Attenuation distortion would not be as much of a problem if all frequency components were to be subjected to the same loss.

2.2.3.3 Phase Distortion

Phase distortion is caused by the finite transmission delay a signal will experience in the system. It is not a problem if all of the spectral components are delayed by the same amount, but when different frequency components have a different delay, the phase structure of the received signal will be distorted. This distortion is measured by a parameter called *envelope delay distortion*.

The above sections have described how a call is carried through the network. The next section will briefly introduce how a call is initially set up and taken down. Signalling in the network can both help and hinder the problem of designing a signal classifier, as we shall see later.

2.2.4 Signalling

Signalling in telephone networks can be broken into two categories: supervisory and information bearing as shown in Figure 8. Examples of supervisory signals are request for service (going off-hook), ready to receive address (dial tone), call alerting (ringing), and call termination (going on-hook). Basically, supervisory signals convey status or control of network elements. Information-bearing signals convey information such as called party address, calling party address, and toll charges. Both of these signalling categories can be further sub-divided into in-channel and common channel signalling. In-channel signalling uses the same medium that the call would use to carry

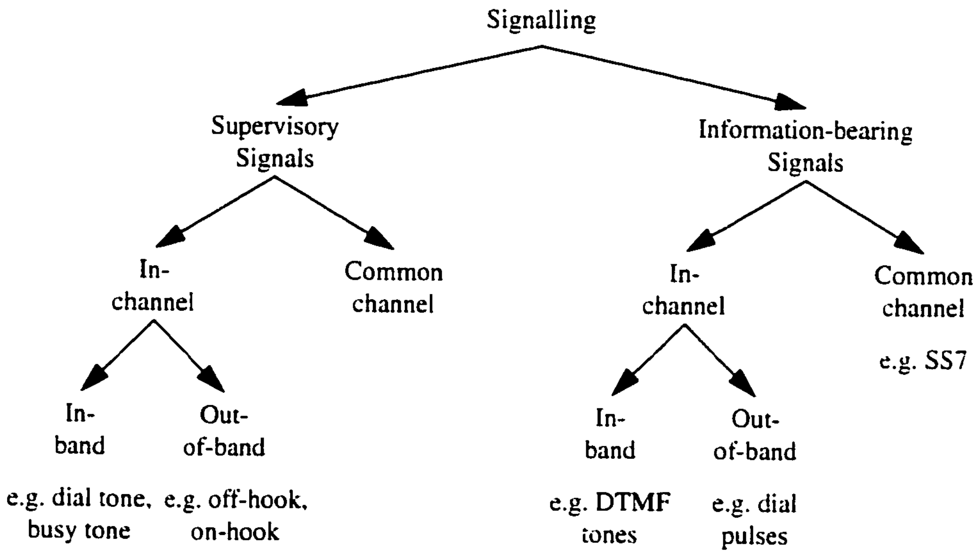


Figure 8: Signalling in telephone networks.

the signalling information. Common channel signalling uses entirely separate resources designed to carry only signalling information.

2.2.4.1 In-Channel Signalling

In-channel signalling can be further broken down into in-band and out-of-band signalling. In-band uses the same band of frequencies that is used to transmit voice, whereas out-of-band uses frequencies that are outside the voice band. An example of in-band signalling are dial tone, busy tone and DTMF tones. An example of out-of-band signalling occurs when a handset goes off-hook. This results in the flow of direct current in the line.

Robbed bit signalling is another example of in-band signalling in multiplexed digital trunks. It was used commonly in the PSTN before common channel signalling was widely implemented, however it is reportedly still used in certain older installations. Robbed bit signalling steals the least significant bit in the sixth and twelfth frame of a Superframe on a T1 line [21] [19]. The information from the robbed bits can be exploited to determine off-hook and on-hook conditions [22]. This would be extremely helpful in determining call boundaries; however, as mentioned earlier, it is not widely used and is being phased out by common channel signalling.

2.2.4.2 Common Channel Signalling

Common channel signalling uses dedicated digital channels to convey signalling information over a parallel network. Figure 9 shows how a call is set-up and released using SS7 (CCITT Signaling System No. 7) signalling. Note that the only portion of the call set-up signalling is carried in the voice-band. This includes the ringback signal and the actual conversation or voice-band data. The call setup sequence does not actually tie up any voiceband transmission circuits until the called party handset starts ringing. If the called party was on the phone (and did not have call waiting) the calling party would receive a busy signal. This busy signal would not go through the voice circuit. Instead of tying up resources, SS7 signalling will connect a source of busy signal to the calling party at the originating switch. On some long distance circuits, network in-band resources are not actually reserved until the called party actually answers their phone. In this case the ringback signal would again be generated locally at the calling party's switch.

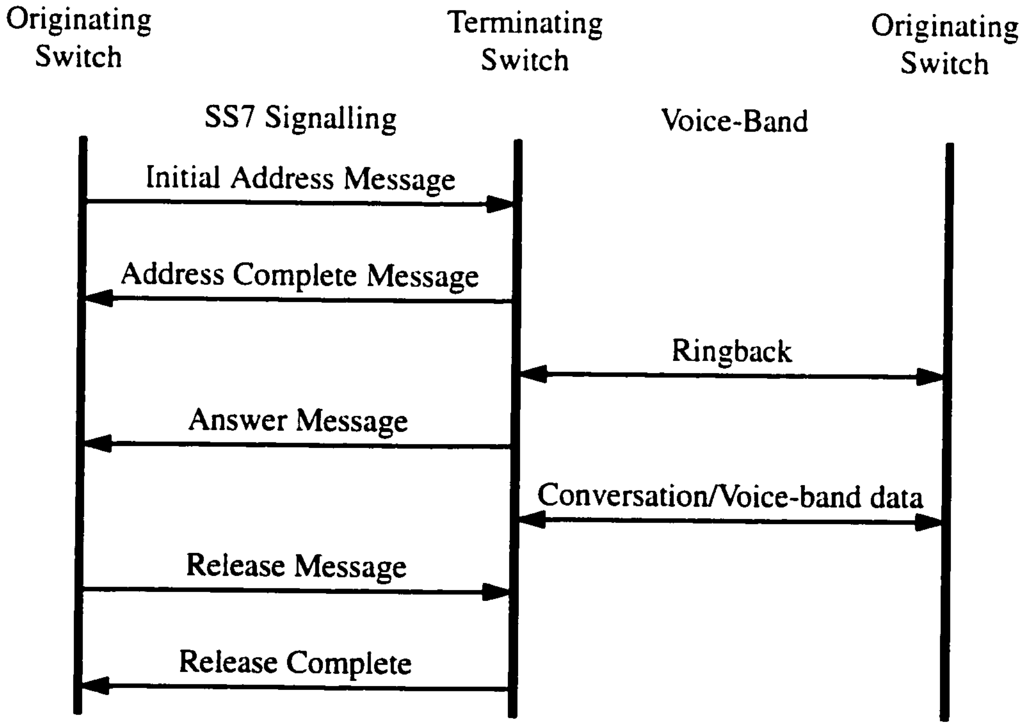


Figure 9: SS7 call set-up and release message sequence.

2.2.5 Traffic Analysis

It is important to understand traffic characteristics to properly provision a network to ensure that acceptable levels of service are provided to all customers. Empirical traffic data is essential to developing accurate mathematical models. A brief description of the models and network parameters used to provision the PSTN will now be discussed.

The total traffic actually attempting to access a network is called the *offered traffic*, and the traffic actually transported by the network is called the *carried traffic*. These two values will not necessarily be the same because of the possibility of *call blocking*. Blocking occurs when a call cannot be completed because of inadequate network resources at that particular time. This outcome is usually relayed to the customer in the form of a fast busy tone. It is desirable to deploy network resources in such a way as to decrease the probability that a call will be blocked. The holding times and busy hour are important network characteristics that are used by telephone companies to predict congested conditions that might lead to call blocking.

The *holding time* of a call is defined to be the length of time that a call occupies a two-way voiceband traffic path [9]. For speech traffic the average holding time value has been determined to be around 2.5 minutes [10]. The *busy hour* refers to the traffic volume over one continuous hour for which the traffic volume is the greatest. To measure the busy hour one would need to measure the carried traffic on a representative sample of voice circuits of a particular switch over a period of time.

It is important to realize that these traffic characteristics can be very different for different signal types. For example, Internet connections using modems have average holding times of considerably more then 2.5 minutes. Figure 10 shows the average holding time of Internet calls, and Figure 11 shows the average number of overflows (blocked calls) per line [11]. An overflow occurs when a call cannot connect to the destination and the caller receives a busy tone. Both graphs show a strong dependence on the time of day. The average holding time during the entire day is 30 minutes, and during the daytime is 18 minutes. This is considerably more than the 2.5 minutes average for speech calls.

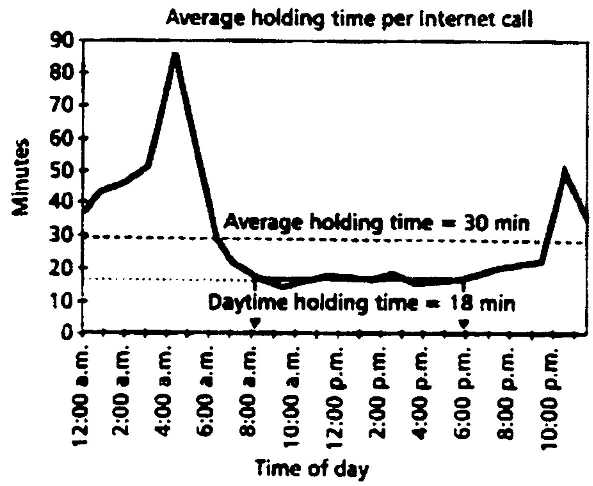


Figure 10: Average holding time per Internet call [11].

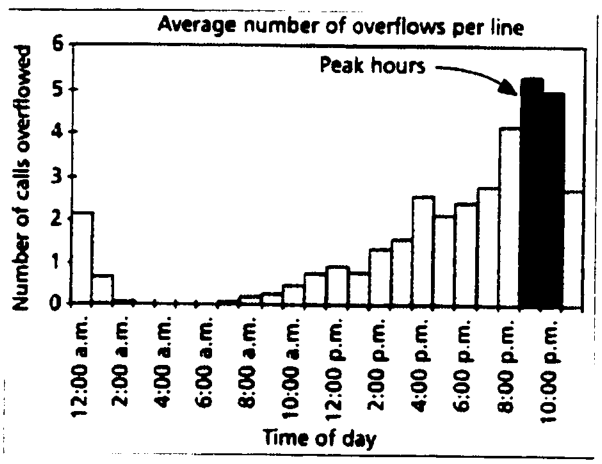


Figure 11: Average number of overflows per line [11].

The growing number of Internet users will increase the offered traffic to the network, thus network resources need to be increased appropriately if blocking is to be kept the same as for a speech-only network. The busy part of the day, for the data calls, occurs later in the evenings from 8 - 10 pm, whereas the busy hour for voice calls is typically from 10 - 11 am for a typical working day [9]. Network resources need to be re-allocated to account for these two busy hours. The data for both graphs were gathered from two telcos, who monitored 11 different Internet Service Providers (ISP) [11].

2.3 Discriminant Analysis

Discriminant analysis is a statistical technique which allows one to study the differences between two or more groups of objects with respect to several characteristic or feature variables simultaneously [12]. Discriminant analysis is used widely in many different areas of research such as the social sciences, medicine, and economics. Simply put, discriminant analysis permits the automatic classification of items belonging to one group into distinct groups or classes based on the values of the feature variables. To use the technique, one needs to first determine the individual classes, feature variables, and finally the discrimination method.

2.3.1 Discriminant Groups or Classes

The classes into which unknown data observations must be allocated need to be *mutually exclusive*, meaning that an observation cannot belong to more than one class [12]. Discriminant variables must be chosen appropriately to resolve effectively between two or more desired classes. Also, as the number of classes increases, the number of computations required to resolve the classes usually increases.

2.3.2 Discriminant Variables

The discriminant feature variables, henceforth called variables, are used to resolve the classes. It is important to include the most effective variables to maximize the classification accuracy. For example, discriminant variables should be chosen so no variable is a linear combination of the other [12]. Intuitively this makes sense since including a functionally dependent variable does not add any more information. One example of a feature variable could be a threshold value. One of two classes could be chosen depending on whether a feature variable value is above or below a certain threshold value. In practice we will see that six variables leads to maximum classification accuracy for our problems.

2.3.3 Discrimination Method

Given a set of variable inputs, the decision maker outputs the class that the current data item should be allocated to. This section will discuss the different decision makers evaluated in this thesis. First, we give definitions of some important statistical parameters.

The sample mean $\bar{x}$ of a data set with N data observations x_i of a variable x is given by equation (9) [15].

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad \text{(Eq 9)}$$

The variance for variable x is given by equation (10) [15].

$$Var(x) = S_x^2 = \frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N-1} \qquad \text{(Eq 10)}$$

An equivalent and more common way to evaluate the variance is shown in equation (11) [15].

$$Var(x) = S_x^2 = \frac{\sum_{i=1}^{N} x_i^2 - N\bar{x}^2}{N-1} \qquad \text{(Eq 11)}$$

The variance gives a normalized measure of how much the observation data is scattered about the mean. The covariance is shown in equation (12) [8].

$$Cov(x, y) = R = \frac{\sum_{i=1}^{N} x_i y_i - \bar{x}\bar{y}N}{N-1} \qquad \text{(Eq 12)}$$

The covariance gives a measure of how statistically correlated two data sets are. If the two variables, x and y, are statistically independent, then the Cov(x,y) = 0 [15]. Note that the Cov(x,x) is just the variance of x, i.e. Var(x) [15]. The covariance matrix of a random vector v is denoted by Cov[v] and is defined by [14]:

$$Cov[v] = \begin{bmatrix} Var(v) & Cov(v_1, v_2) & \dots & Cov(v_1, v_m) \\ Cov(v_2, v_1) & Var(v_2) & \dots & Cov(v_2, v_m) \\ \vdots & \vdots & & \vdots \\ Cov(v_m, v_1) & Cov(v_m, v_2) & \dots & Var(v_m) \end{bmatrix}. \qquad \text{(Eq 13)}$$

An important property of the covariance matrix is its symmetry about the main diagonal.

The following variables will be used to describe the linear and quadratic discriminant functions. Assume that there are p feature variables and g classes. Further assume that:

- x is a column vector of length p containing the values of the one or more variables for this observation.
- μ_i is a column vector of length p containing the means of the variables calculated from the observations in class i.
- R_i is the covariance matrix calculated from the observations in class i.
- R_p is the pooled covariance matrix.
- π_i is the prior probability than an observation is in class i.

The *pooled covariance* is obtained by taking the element-wise average of the separate group covariance matrices, R_i.

2.3.3.1 Linear Discriminant Functions

Linear discriminant functions (LDF) are used to determine which class an observation belongs to by minimizing the squared distance, also called the Mahalanobis distance, to the mean value of the expected observations for each class. The squared distance is given by:

$$d_i^2(x) = (x-\mu_i)'R_p^{-1}(x-\mu_i), \qquad \text{(Eq 14)}$$

where R_p^{-1} denotes the matrix inverse of R_p, and where $(x-\mu_i)'$ denotes the transpose of $(x-\mu_i)$ [13]. Expanding terms and simplifying gives:

$$d_i^2(x) = -2\left[\mu'_i R_p^{-1} x - \frac{1}{2}\mu'_i R_p^{-1}\mu_i\right] + x' R_p^{-1} x. \qquad \text{(Eq 15)}$$

The linear term inside the square brackets is called the *linear discriminant function* (LDF) for class *i*. The larger the linear discriminant function, the smaller the squared distance. An observation is classified into the class that results in the largest value for the linear discriminant function, which also results in the smallest squared distance. The bracketed part in equation (15) can be rewritten as follows:

$$g_i(x) = A_i x + B_i \qquad \text{(Eq 16)}$$

$$A_i = \mu'_i R_p^{-1} \qquad \text{(Eq 17)}$$

$$B = -\frac{1}{2}\mu'_i R_p^{-1}\mu_i \qquad \text{(Eq 18)}$$

The coefficients A and B are sometimes called the Fisher linear discriminant function coefficients [7]. Using the LDF to do classification is statistically optimal under the assumptions that the individual group covariance matrices, R_i, are all equal, and that the feature variables all have a multivariate normal distribution [18]. If this assumption is not true, then the consequence of using the LDF anyway will be sub-optimal classification accuracy [12]. In practice, using the LDF while violating the conditions slightly still leads to acceptably accurate classifications for many problems.

2.3.3.2 Quadratic Discriminant Functions

Quadratic Discriminant Functions (QDF) do not require that the individual group covariance matrixes, R_i, be equal. The squared distance from the observation to the mean is given by [13]:

$$d_i^2(x) = (x - m_i)' R_i^{-1} (x - m_i) + \ln|R_i|. \qquad \text{(Eq 19)}$$

Expanding terms results in the following equation:

$$d_i^2(x) = (-2)\,\mu'_i R_i^{-1} x + x' R_i^{-1} x + m_i' R_i^{-1} m_i + \ln|R_i|. \qquad \text{(Eq 20)}$$

The last two terms of the equation can be computed and combined into one constant for each class i. The first two terms are a function of the p-element observation vector x, and thus they must be calculated for each observation x. The QDF method is useful in classification if the separate group covariance matrices are very different. The price paid is increased computational complexity. Note that using the QDF yields statistically optimal classifications only if the feature variables have a multivariate normal distribution [18].

2.3.3.3 Linear and Quadratic Decision Rules

To classify a new observation into a class, the linear or quadratic discriminant function first needs to be evaluated for each class. Then a *decision rule* is used to produce the class decision. The decision rule that applies for LDFs is shown in equation (21) [17], for all distinct class indices l and j in the possible range $1, 2, 3, \ldots, i$.

$$u_{lj}(x) = g_l(x) - g_j(x) > \ln\pi_j - \ln\pi_l \qquad \text{(Eq 21)}$$

Prior probabilities π_j and π_l are the probabilities of obtaining observations from classes j and l, respectively. If the prior probabilities are all equal, then an observation x is classified into the class l which has the largest value of u_{lj}. For QDFs equation (21) is still valid; however, the values from equation (20) must first be negated before evaluating equation (21).

2.3.3.4 Pseudo-Quadratic Discriminant Functions

The term *pseudo-quadratic* was coined by Sewall in [2] to describe the quadratic discriminant functions used by SPSS. Since SPSS uses the discriminant function values to classify observations, and not the original variable values, the classifications do not follow a statistically optimal rule [7]. However, the decisions obtained using the functions and their covariance matrices are often not too different from those obtained using covariance matrices for the original variables [7]. For a complete description of this method please refer to [7] and [8].

2.3.3.5 Adaptive Logic Networks

Adaptive Logic Networks (ALNs) are classification methods produced by a general classification software package developed by Dendronic Decisions Limited of Edmonton, Alberta, Canada. The technique combines aspects of both linear and quadratic discrimination techniques. Linear discrimination can be viewed as placing hyperplane decision boundaries to optimally partition clusters of observations in the N-dimensional feature variable space. Quadratic discrimination, on the other hand, uses ellipsoids to form the decision boundaries. ALNs use piecewise linear methods to develop yet more flexible boundaries between classes. The computation of these boundaries is determined by several user-defined variables during a proprietary training phase [16]. The first step in classifying a new observation is to determine which linear segment in each variables's domain needs to be evaluated. This is done efficiently with the help of a decision tree. Once the relevant linear segment has been determined, it is a matter of evaluating an equation for each group.

2.4 Signal Types

The classification algorithms developed by Sewall are able to classify between many different signal types that are present in the PSTN. Those signal types will be briefly described along with others that we added to make the set of classes more complete.

2.4.1 Speech

Speech is a very complex signal type. The characteristics of speech change depending on the sex, age, regional accent, mood, etc. of an individual. The standard voiceband on the telephone network, as noted earlier, ranges from 300 Hz - 3400 Hz [9]. Speech signal power is concentrated typically in the lower parts of the voiceband, whereas for modulated data the signal power is typically uniformly distributed about a mid-band carrier frequency. An additional complication in speech is the presence of silent intervals, which vary in frequency and duration depending upon many factors, such as the age and sex of the talkers, and the nature of the call.

2.4.2 V.17

The V.17 recommendation defines voiceband data signals that are intended for high-speed facsimile applications [23]. The majority of fax machines currently in service use the older V.29 standard. V.17 is commonly available in new fax machines, and most fax modems available for PCs. This standard will gain in popularity as it operates at a maximum bit rate of 14 400 bps (bits per second), whereas the older V.29 standard operates at a maximum bit rate of 9 600 bits/s. The basic properties of the V.17 standard are shown in Table 2. Note that four possible operating modes are specified. The three slower "fall-back" modes are intended for use over degraded connections. The carrier frequency is 1800 Hz, with Quadrature Amplitude Modulation (QAM) at 2400 symbols per second, using half duplex transmission. All bit rates use trellis coding (TC) at all data rates to gain additional protection against bit errors in the received signal.

Table 2: V.17 properties.

Bit Rates (bps)	Constellation Size
14 400	128 point - TC
12 000	64 point - TC
9600	32 point - TC
7200	16 point - TC

2.4.3 V.22

This recommendation allows for data rates of up to 1200 bps with full duplex operation. Given the affordability of high-speed modems, this standard was, for the most part, obsolete until the advent of point-of-sale terminals. Point-of-sale terminals are used at retail locations to process purchases made by debit and credit cards. Interac Direct Payment (IDP) transactions alone in Canada topped one billion in 1997 (not including credit card purchases) [32]. The point-of-sale terminals utilize lower bit rate modem protocols such as V.22 and V.22*bis*. Thus, instead of disappearing from use, these protocols are increasingly used on the PSTN and hence this class cannot be ignored by any useful classifier. V.22 transmits data at 600 baud using Differential Phase Shift Keying (DPSK) as the modulation method. Two carrier frequencies of 1200 and 2400 Hz are used, one for transmitting and one for receiving. Table 3 shows the two different possible modes of operation.

Table 3: V.22 properties.

Bit Rates (bps)	Constellation Size
1200	4 Point
600	2 Point

2.4.4 V.22*bis*

This recommendation, originally intended to replace V.22, allows for data rates of up to 2400 bps using QAM and a symbol rate of 600 baud at each data rate. The transmitted and received signals are separated by using two carrier frequencies: one at 1200 Hz, and one at 2400 Hz. As mentioned above, this recommendation is commonly used to complete IDP or credit card transactions over the PSTN at retail stores.

Table 4: V.22*bis* properties.

Bit Rates (bps)	Constellation Size
2400	4 Point
1200	2 Point

2.4.5 V.27*ter*

This recommendation was developed for facsimile applications. It supports data rates of 2400 bps and 4800 bps using DPSK modulation at both data rates. The carrier frequency is 1800 Hz for both the transmit and receive signals. Most facsimile machines use the faster V.29 standard; however, V.27*ter* is usually available as a fall-back mode in the event that the communicating facsimile machines are unable to connect using the faster V.29 standard.

Table 5: V.27*ter* properties.

Bit Rates (bps)	Constellation Size	Symbol Rate
4800	8 point	1200 baud
2400	4 point	1600 baud

2.4.6 V.29

The V.29 recommendation is a facsimile standard which supports data rates of up to 9600 bps. The carrier frequency is 1700 Hz, and the symbol rate is 2400 baud. This standard supports both full and half duplex modes of operation, while using combined amplitude and phase modulation. This standard is, at present, the most commonly used facsimile standard.

Table 6: V.29 properties.

Bit Rate (bps)	Constellation Size
9 600	16 Point
7 200	8 Point
4 800	4 Point

2.4.7 V.32

V.32 allows for data rates at up to 9600 bps, in full duplex mode, using a symbol rate of 2400 baud. The carrier frequency is 1800 Hz for both signal directions, and echo cancellation techniques must be incorporated to separate the transmitted and received signals. V.32 uses QAM and trellis coding techniques as well as a 16-point constellation. The 16-point constellation is an alternative to trellis coding, used only for 9600 bps connections. The properties for V.32 are summarized in Table 7.

Table 7: V.32 properties.

Bit Rate (bps)	Constellation Size
9 600	32 Point - tc or 16 Point
4 800	4 Point

2.4.8 V.32*bis*

The V.32*bis* recommendation improves upon the V.32 recommendation by supporting bit rates of up to 14400 bps in full duplex mode. V.32*bis* uses a symbol rate of 2400 baud, and a carrier frequency of 1800 Hz. QAM modulation is used at all bit rates and trellis coding techniques are used at all but the 4800 bps data rate. V.32*bis* must also incorporate echo cancellation techniques to separate the transmitted and received signals.

Table 8: V.32*bis* properties.

Bit Rate (bps)	Constellation Size
14 400	128 Point - tc
12 000	64 Point - tc
9 600	32 Point - tc
7 200	16 Point - tc
4 800	4 Point

2.4.9 V.32 terbo

V.32 terbo is not an ITU (International Telecommunication Union) standard; rather, it is a proprietary modem protocol developed by AT&T. V.32 terbo increased the maximum bit rate to 19200 bps from the previous maximum of 14400 bps attainable using the V.32*bis* standard. At the time that was a feature that could be sold to customers, resulting in many modem manufacturers supporting this protocol. Unfortunately, information regarding the symbol rate, carrier frequency, modulation, and coding methods does not appear to be in the public domain as it is a proprietary protocol. We believe that since this is an extension of the V.32*bis* standard, it is most likely using many of the same parameters as V.32*bis*. This protocol did not gain in popularity once the V.34 recommendation was passed by the ITU, which allowed for data rates of up to 33600 bps. It should be noted that many modems are still capable of connecting using V.32 terbo; however, such connections are unlikely if the modems also support the superior V.34 standard.

2.4.10 V.34

The V.34 recommendation is by far the most complex voiceband data standard. Trellis coding and QAM are used for all bit rates, which vary from 2400 bps up to 33600 bps. V.34 supports both full and half-duplex modes of operation. The carrier frequency and baud rates are both variable depending upon line conditions as measured during an elaborate initial training sequence. The constellations are all subsets of a

Table 9: V.34 carrier frequencies and supported baud and bit rates.

Symbol Rate	Low Carrier (Hz)	High Carrier (Hz)	Bit Rates (bps)
2400	1600	1800	2 400 - 21 800
2743	1646	1829	4 800 - 26 600
2800	1680	1867	4 800 - 26 400
3000	1800	2000	4 800 - 29 000
3200	1829	1920	4 800 - 31 400
3429	1959	1959	4 800 - 33 800

1664-point superconstellation. In practice, connecting at 33600 bps requires rarely occurring ideal line conditions. From personal experience, even the 28 800 bps data rate is difficult to negotiate. Most newer modems support the V.34 standard since, as well as high bit-rate modes, the recommendation also provides an extensive series of fall-back modes down to 2400 bps that can be used in increasingly impaired line conditions. For details on the V.34 standard please refer to [27].

2.4.11 V.90

The ITU adopted the V.90 standard on February 6, 1998 [33]. This standard incorporates two competing standards in use prior to the official V.90 agreement. The V.90 standard supports download speeds at up to 56000 bps, and upload speeds at up to 33600 bps [28] [29]. V.90 obtains the higher download bit rate by using techniques that ensure greater control over the digital PCM codes that are transmitted in the downstream direction. This greater control is achieved after the downstream transmitter recovers the sampling timing and then uses that timing to ensure that precise analogue voltages are presented to its local CODEC's coder circuit. The end result is that the downstream transmitter gains access to the downstream path as if it were a digital pipe at 56 kbps. Pulse amplitude modulation (PAM) is used as the analogue modulation method over the two subscriber loops. The entire upload path still uses the older V.34 standard. Thus, within the PSTN, one direction of a V.90 call will look like a V.34 modem, and the other direction will look like a purely digital 56 Kbps data connection. For a detailed explanation of V.90 refer to [28] and [29].

2.4.12 Ringback

Ringback is a return signal to the calling subscriber telling them that the dialed party's set is being rung [9]. The actual signal sent to the calling subscriber's handset is a 20 Hz signal at a level of 86 Vrms for ringer excitation [10]. In North America the signal is applied repeatedly for two seconds and then removed for four seconds [30]. Ringback is useful for identifying the call boundaries of monitored voice channels; however, when calls are transferred from one person to another (typically when calling large companies), the ringback signal may again be generated. In this circumstance the call did not end, and a new call begin, even though the ringback signal was again briefly present. Caution must therefore clearly be exercised when using ringback to identify the start of calls.

2.4.13 FSK Signalling

Frequency Shift Keying (FSK) signalling appears within facsimile calls. It is used initially to negotiate connection parameters and, then at the end of the transmission, to signal the end of the connection. It is also used between transmitted pages to indicate the end of one page and the start of a new page. FSK signalling is thus used to convey control information between the two modems.

2.4.14 DTMF Tones

Dual Tone Multi Frequency (DTMF) tones are used to communicate the dialed number from a telephone handset to the CO. Each digit is encoded using two frequencies, a row and a column frequency. The row and column frequencies are shown in Figure 12 [30]. The twelve signals in the first three columns are commonly found on all telephone handsets. The signals in the last column, labelled "ABCD", have apparently been used in private (e.g. military) networks [30]. The DTMF tones must be present for at least 40 ms and must have a 40 ms gap between tones to ensure accurate decoding [30].

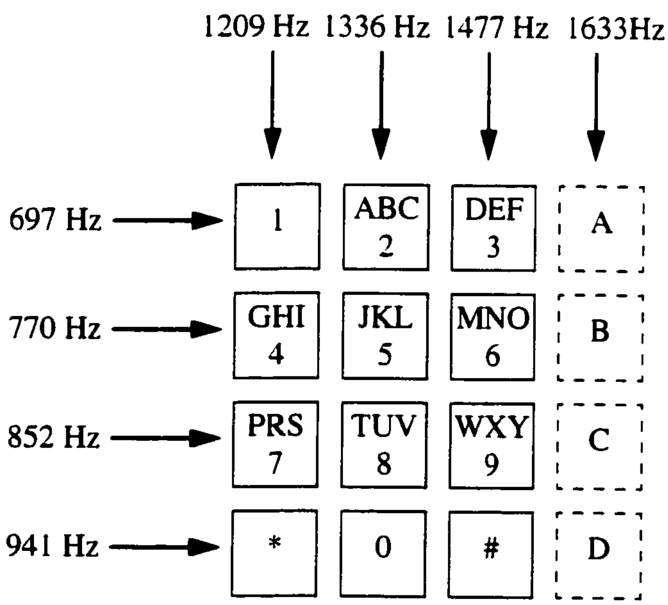


Figure 12: DTMF tone frequencies.

2.4.15 Binary Traffic Types

The binary class was created to encompass the many different types of digital traffic commonly found on the PSTN that appear to an observer to be random bit streams. For example, companies may lease channelized T1s from telcos for strictly digital data transmission. The protocols used could be proprietary, or they could follow a standard such as frame relay. In either case this would be considered to be binary traffic if the bit streams appear sufficiently random. Another example of signals that belong to this class is the in-band signalling carried by T1s. In North America, signalling is generally carried by a separate parallel packet switched network and not by voice circuits. In some circumstances, however, this separate signalling network is not available, such as at remote base stations in wireless networks. In these cases, one reserved channel in a T1 could be used to carry this signalling information. Another class of binary traffic is the V.90 downlink mentioned in the previous section.

2.5 T1

T1 is a physical layer digital communications standard that supports the transmission of digitized voice and data at a rate of 1.544 million bits per second (Mbps) [20]. T1 was first introduced in the 1960s and was initially used by telephone companies who wished to reduce the number of telephone cables in large metropolitan areas by multiplexing many voice channels onto each T1 connection. In Europe the corresponding trunk standard is the E1, which has a 2.048 Mbps data rate. Digital trunks, such as the T1 and E1, permit many calls to be multiplexed together on one physical connection. Within switches and COs, such trunks form a convenient and efficient unit for handling connections. The following discussion briefly describes the organization of a T1, the different framing formats, and different coding techniques. For a more in depth discussion of T1 please refer to [19] and [20].

2.5.1 T1 organization

A channelized T1 line has 24 separate time slots for voice channels that are multiplexed onto a single line using Time Division Multiplexing (TDM). Each channel

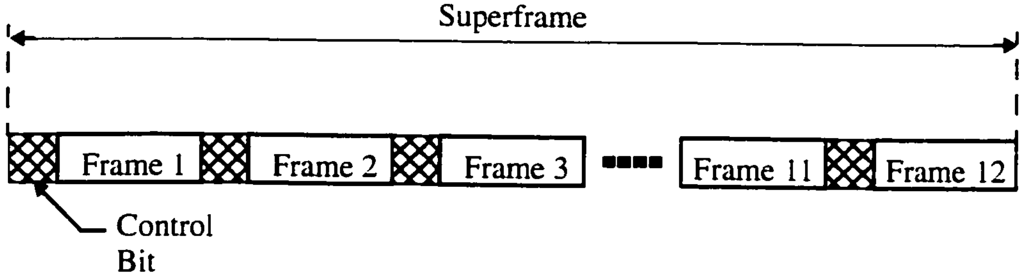


Figure 13: The T1 superframe.

provides 64 thousand bits per second (Kbps) to transmit an analogue signal that is sampled at 8000 samples per second, with 8 bits per sample. In North America, these samples are the PCM μ-law encoded. Each 64 Kbps channel is also commonly referred to as a Digital Signal-Level Zero (DS-0). In addition to 24 time multiplexed DS-0's, framing bits are inserted into the bit stream at a rate of 8 Kbps. The framing bits are used at the receiver to allow the DS-0's and other signalling information to be recovered. The resulting signal has a total bit rate of 1.544 Mbps. A channelized T1 is also commonly referred to as a Digital Signal-Level One (DS-1).

2.5.2 T1 Framing Formats

Two framing formats are used on T1 lines: the SuperFrame (SF) and the Extended SuperFrame (ESF) formats. A SF is a data structure involving 12 consecutive frames, with each frame containing one sample from each of the 24 channels. The frames are organized using the so-called D4 framing pattern that is present in the framing pattern [19]. The D4 framing pattern contains control bits, when combined, form a twelve bit control word for each 12-frame SF. The odd bits in the control word, called the terminal frame or FT bits, mark frame and superframe boundaries. The even bits in the control word, called signalling frame or FS bits, identify the frames carrying signalling information. To enable the sharing of signalling bits by all frames, D4 framing uses "robbed bit" signalling. In this technique, the least significant bit in all DS-0s in the sixth and twelfth frames is reserved for signalling information (and the corresponding data bits are discarded). A new superframe is recovered from a T1 every 1.5 ms.

The ESF frame format is very similar to the SF format except that the ESF for-

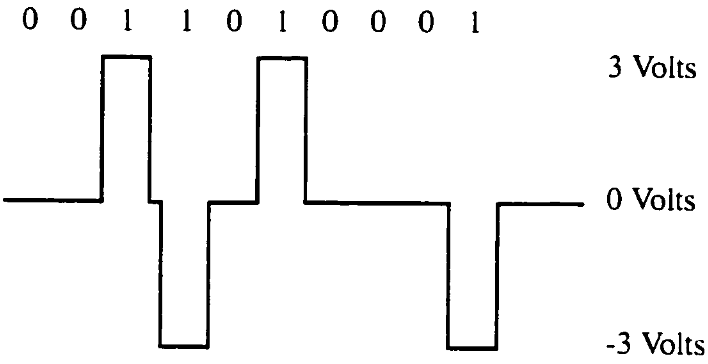


Figure 14: Alternate mark inversion.

mat is composed of 24 frames, each containing a sample from each of the 24 channels. The ESF was introduced to permit the testing of a line without requiring it to be taken out of service. The control word contains a total of 24 bits. Six bits in the control word are reserved for a Cyclic Redundancy Check (CRC) code, twelve bits are reserved for use by the transmitting and receiving equipment at either end of the link, and six bits are used to manage signalling and framing. The remainder are used for the evaluation of the circuit performance. A new ESF is recovered every 3 ms.

2.5.3 T1 Line Coding Techniques

The logical 1s and 0s are transmitted electrically using Alternate Mark Inversion (AMI) signalling. In AMI a 1 is coded by a pulse, and a 0 with the absence of a pulse. Subsequent 1s are encoded using alternating positive and negative pulses. See Figure 14. The AMI format is considered bipolar because of the positive and negative pulses. It is critical that the signal maintain a certain amount of ones density so that equipment along the line has something to frame on to regenerate the signal. To accomplish this, a coding technique known as Bipolar with 8-Zero Substitution (B8ZS) is widely used. This coding format ensures that 8 consecutive zeros are never transmitted down the line by inserting logical 1s into the bitstream. This gives equipment a pulse to frame on to, even while transmitting long streams of logical 0s. Our classifier will be shielded from the details of line coding by a standard T1 interface circuit.

2.5.4 Fractional T1

Fractional T1 was introduced to allow customers to lease a fixed part of a T1 circuit. This saves money as the customer does not need to pay for leasing an entire T1 line. This service allowed customers to purchase T1 bandwidth in 64 Kbps (DS-0) increments. As far as a signal classifier is concerned, a leased fractional T1 data service will appear like several (often adjacent) DS-0 channels carrying binary traffic.

2.6 Higher Bit Rate Trunks

Modern telephone networks make extensive use of higher bit rate trunks than T1s and E1s. In fact, trunk signals are organized into synchronous digital hierarchies, with multiple trunk tributaries at the same nominal bit rate being multiplexed into even higher bit rate trunks. Thus a DS-2 trunk consists of four multiplexed DS-1's, and a DS-3 consists of seven multiplexed DS-2s (or 28 DS-1s).

A T3 line is capable of carrying 28 T1s, or 672 DS-0s. The total bit rate of a T3 is 44.736 Mbps. The term, T3, is actually the unofficial name used for the official term, DS-3. T3s would typically be used as long distance trunks by telcos, large companies and larger ISPs. Tables 10, 11, and 12 list the various digital hierarchies used in North America, Japan, and Europe [9]. Note that "Level 4" hierarchy was never accepted as a standard although proprietary schemes were developed.

Table 10: Higher-level PCM multiplex comparison for North America.

	Level 1 (DS-1)	Level 2 (DS-2)	Level 3 (DS-3)	Level 4 (DS-4)
No. voice channels	24	96	672	4032
Bit rate (Mbps)	1.544	6.312	44.736	274.176

Table 11: Higher-level PCM multiplex comparison for Japan.

	Level 1	Level 2	Level 3	Level 4	Level 5
No. voice channels	24	96	480	1440	5760
Bit rate (Mbps)	1.544	6.312	32.064	97.728	400.352

Table 12: Higher-level PCM multiplex comparison for Europe.

	Level 1 (E-1)	Level 2 (E-2)	Level 3 (E-3)	Level 4 (E-4)	Level 5 (E-5)
No. voice channels	30	120	480	1920	7680
Bit rate (Mbps)	2.048	8.448	34.368	139.264	560.0

In addition to the synchronous digital hierarchies described, there exists a more recent hierarchy of fully synchronous signals that were developed for digital signal transmission over optical fibre. These signals belong to the very similar Synchronous Optical Network (SONET) and Synchronous Digital Hierarchy (SDH) standards. In North America it is common to package DS-3 signals into 51.84 Mbps OC-1 signals in the SONET hierarchy. Multiple OC-1s are then synchronously multiplexed into the 622.08 Mbps OC-12 and 2.49 Gbps OC-48 SONET connections that form the back-bone of the present long-distance network.

With respect to our signal classifier, we restricted our investigations to T1 trunks as they are expected to remain a major unit of transmission in telephone exchanges and wireless base station networks. Access to T1 signals is widely available on switching equipment patch panels using inexpensive bantam jack cables.

Chapter 3

3.0 Research Infrastructure

The bulk of our research was completed at the Edmonton laboratory of TR*Labs*. Field trials were conducted in cooperation with TELUS Corporation (a TR*Labs* Industrial Sponsor) at their maintenance engineering lab and at the main toll building in downtown Edmonton. Additional field trials were conducted at base stations in Edmonton and Calgary in the wireless network of TELUS Mobility. Various different hardware and software tools were utilized to build and improve the prototype voiceband signal classifier. All of these items will be discussed in the following sections.

3.1 Voiceband Signal Classifier Prototype Specifications

The prototype voiceband signal classifier is simply a PC with specialized off-the-shelf hardware and custom software running under the MS-DOS operating system. The PC platform was chosen because it is affordable and commonly available, with many third party vendors developing both hardware and software that is compatible with the PC architecture.

3.1.1 PC Hardware Specifications

The PC host system has a generic 486 PCI (Peripheral Component Interconnect)/ISA (Industry Standard Architecture) bus motherboard with a 486 DX4 CPU (Central Processing Unit) clocked at 100 MHz. Refer to Appendix C for detailed specifications. This computer is considered inadequate for most current consumer and engineering applications; however, it has proven to be powerful enough to serve as an initial prototype platform running under MS-DOS. The specialized hardware that plugs into the ISA bus includes a T1 interface card and a DSP card.

3.1.2 T1 Interface Card

The T1 interface card was purchased from GL Communications Inc. (Gaithersburg, MD). It is a super-T1 (rev. 1) card that plugs into the 8 MHz ISA bus of a PC.

This card frames on an incoming T1 signal and extracts data from all 24 channels. The data is then stored in a memory area that is shared between the T1 card and the PC, thus allowing the PC to read the T1 data. The T1 card is able to accommodate various framing and line coding formats, making it very versatile. It is also capable of generating a hardware interrupt to the PC to signal the arrival of new data. For example, if the framing format is SF, interrupts are generated every 1.5 ms (12 PCM codes per voice channel). If the framing format is ESF, then interrupts will be generated every 3.0 ms (24 PCM codes per voice channel). The T1 card comes with various software packages that are useful for testing and accessing the features of the card. Sample code was also provided by GL Communications, which was used as a framework to develop custom software to properly initialize and control the T1 card.

3.1.3 DSP Card

The DSP card is a Tiger 30 card from DSP Research Incorporated (Sunnyvale, CA) with a Texas Instruments TMS320C30 40 MHz floating point DSP chip on board. The card also plugs into the ISA bus of a PC. The DSP is used to compute the feature variables and evaluate the linear and quadratic discriminant functions. Incoming T1 data is passed to the DSP card from the PC via on-board external PC-DSP shared memory. This card is also capable of generating and acknowledging interrupts to and from the PC, which is very useful in real-time applications. The DSP card also comes with specific software library functions for initializing and controlling the DSP, as well as an optimizing C compiler for the TMS320C30. Detailed specifications for the TMS320C30 DSP are provided in Appendix C.

3.2 Software

Various different software packages were utilized. This section has been broken down into two sections: the software development tools, and the application software packages.

3.2.1 Software Development Tools

Borland C++, version 4.5 for Windows 3.1, was used to develop applications on the PC to run under MS-DOS. Borland TASM version 4.0 was also used to compile code written in assembly language. The Borland compiler accepts code written in ANSI (American National Standards Institute) C. It also includes library functions that are useful for MS-DOS applications.

The library functions of the database package CodeBase® version 6.0, by Sequiter Software Inc. (Edmonton, AB), were used to generate the database file that stores the classification vectors. CodeBase supports many different database formats, and is available in both C and C++ versions. We opted to use the dBase format because of its simplicity and wide support among other database products.

All of the code that runs on the DSP was developed using the Texas Instruments TMS320C3X/4X ANSI C compiler and linker, version 4.6. This compiler includes optimizing features that improve the run-time performance for the specific DSP processor. Several library functions provided by DSP Research were used to interface our software with the DSP card and processor.

Library functions (LIBALN 1.1) and sample source code provided by Dendronic Decisions Limited were used to perform off-line classification studies using the ALN classification method. To recompile the source code and to use the library functions we used Microsoft's Visual C++, a Windows 95 32-bit C++ compiler.

3.2.2 Applications

SPSS® version 6.1.2 for Windows 3.11 is a statistical software package that was used to perform discriminant analysis. SPSS was used to compute both linear and pseudo-quadratic discrimination functions given training data sets.

MATLAB® (Matrix Laboratory) version 4.2 is a mathematical package that incorporates numerical analysis, matrix computation, signal processing, and graphics. MATLAB was used extensively to perform classification simulations, and to compute quadratic discrimination functions from training data.

3.3 Test Equipment

The Electrodata Ez-tester model TTS 3-EZ was borrowed from TELUS to test the T1 interface at TR*Labs*. This hand-held unit is capable of generating a T1 signal in various formats using different test patterns as data. The test set verified that the T1 card was accurately locking onto the T1 signal, and extracting data from all 24 channels. It also verified that the customized software was actually transferring data from the T1 line to the DSP card. The test set was also generally useful for testing the real-time capabilities and limitations of the prototype.

3.4 Computer Resources

The computing resources include the prototype voiceband signal classifier and various UNIX (Uniplexed Information and Computing System) workstations. The UNIX workstations were used to perform intensive simulations, mainly using MATLAB.

3.5 Sample Data Files

A multilingual speech database was obtained from The Center for Spoken Language Understanding (CSLU) (Beaverton, OR). This database contains telephone speech samples from over 2000 speakers representing 22 different languages. Both fluent continuous speech and fixed vocabulary utterances are included, which vary in length from 3 seconds to 1 minute. This database was useful for our project because it contains extensive telephone speech recordings of languages other than English.

Chapter 4

4.0 Prototype Voiceband Signal Classifier Implementation

The classifier is implemented using a standard PC, running MS-DOS, a T1 interface card, and a DSP card, as shown in Figure 15. MS-DOS was chosen because it has predictable and reliable real-time behaviour. Although MS-DOS is not normally considered a real time OS, it has been proven to be adequate for this particular application. MS-DOS allows the programmer to directly control hardware devices without the need for a device driver running in a protected mode. Device driver development requires extensive knowledge of details of the underlying hardware, which is sometimes difficult to obtain from third party vendors. Device driver development can greatly prolong development time. Since MS-DOS is not a multitasking OS, the single thread of execution has predictable interrupt response behaviour, making it easier to construct a real-time system. Also, many of the library functions and utility programs that came with the T1 and DSP card work exclusively under MS-DOS.

All of the above reasons made MS-DOS an easy choice. However, development under MS-DOS does have its disadvantages. For example, developing a Graphical User Interface (GUI) is difficult under MS-DOS because it is not a Windows-type OS. Some modern GUI conveniences were provided, however, such as pull-down menus and mouse-activated operation. As a result of the limitations of MS-DOS, a Windows NT implementation is being developed as part of an on-going follow-up project.

The T1 card is required to frame on an incoming T1 signal and to extract 8-bit PCM data for each of the 24 voice channels. The DSP card is used to compute the feature variables, to evaluate the LDFs and QDFs, and finally to send classification vectors to the PC. The PC passes 8-bit PCM data from the T1 card to the DSP card and processes the classification vectors returned by the DSP. Processing the classification vectors involves storing the classification vectors into a database, as well as possibly updating a real-time display. The PC is also capable of performing off-line busy hour and pie chart queries from existing database files.

4.1 Overall System Architecture

Figure 15 shows the overall system architecture of the voiceband signal classifier. Data from each of the 24 channels of a T1 are extracted using the T1 interface card. Once enough samples are gathered for each channel (12 for SF, 24 for ESF) the T1 card generates an interrupt to the PC. The PC ISR (Interrupt Service Routine) acknowledges the interrupt by copying data from the PC-T1 shared memory to a FIFO (First-In First-Out) buffer that is shared between the PC and DSP. The PC then generates an interrupt to the DSP card. The DSP ISR responds by copying the data from the FIFO buffer into an internal circular buffer. A circular buffer is required to provide elastic data storage during the discriminant function computation. If a circular buffer is not used then incoming data will be lost while the DSP is busy computing the classification decisions for the previous batch of data. Data is then copied from the circular buffer to compute the feature variables. Data samples will temporarily back up in the

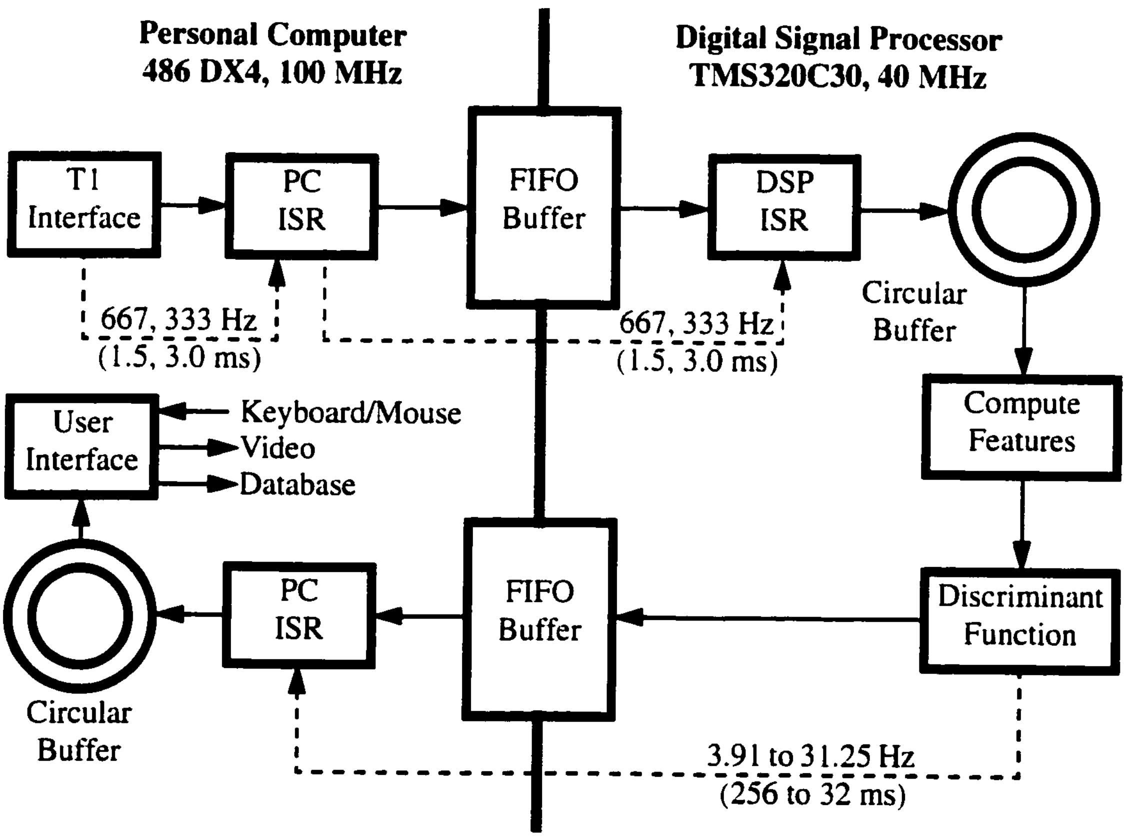


Figure 15: Voiceband signal classifier system architecture.

circular buffer when the DSP is busy evaluating the discriminant functions. Once the LDF and QDF have been evaluated, a class is selected for each of the 24 channels. The classes assigned to each channel are called classification vectors. The classification vectors are then copied into another shared PC-DSP FIFO buffer and then the DSP generates an interrupt to the PC to let the PC know that new vectors are available. The PC then copies the classification vectors into a circular buffer, again to ensure that no data loss will occur when the PC is temporarily unable to attend to the data. The GUI then extracts the classification vectors from the circular buffer and displays the results on the video monitor (if a real-time display is being viewed by the user), and stores them into a database.

4.1.1 PC Implementation

A prototype of the GUI software was first developed by a group of students at the University of Alberta as a group term project for a software engineering course (CMPE 313). They built a working interface that could read in a standard dBASE IV file and then display the classification vectors on a real-time graph. As the classifier generates classification vectors, they are displayed on the graph in the form of bars which are updated in real-time. They also added busy hour and pie chart query features. The busy hour graph is used to display the busy hour for a given 24 hour period, and the pie chart graph displays the breakdown of the data according to class. Extensive modifications were then required to extend the prototype to provide additional functionality, to communicate with the actual DSP and T1 cards, and finally to operate in real time with no data loss. Figure 16 shows an actual screen-shot of the GUI. Five pull down menus are provided so that the user can easily change options using a mouse. Keyboard input is only required when changing numerical value settings (e.g. power threshold).

Figure 17 shows an actual screen shot of the real-time display. The real-time display presents the recent classification decisions for all 24 channels of a T1 with time advancing to the right. A maximum of two classification methods can be displayed for each channel (represented by the coloured bars). In this case, the lower bar, for each channel, represents the linear filtered results (discussed later), and the upper bar represents the hybrid filtered results. This particular real-time display shows the classification results of a data file generated for the 1998 TR*Labs* Technology Forum. For a complete description of all pull-down menu options please refer to Appendix A.

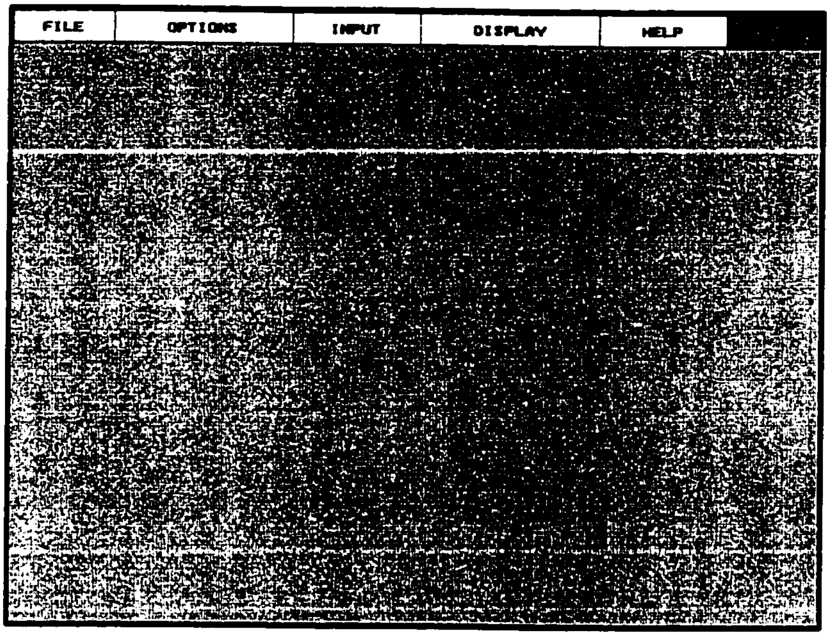


Figure 16: Signal classifier GUI.

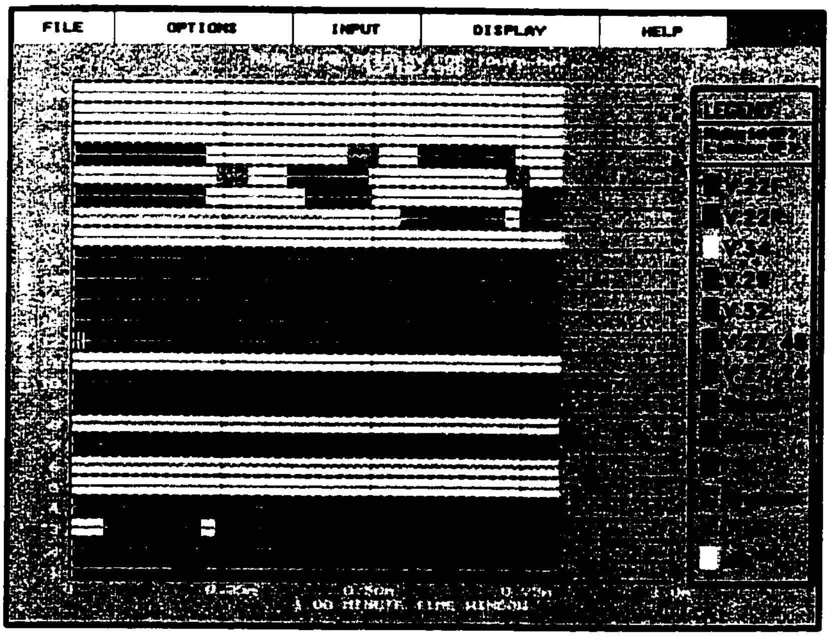


Figure 17: Signal classifier real-time display.

4.1.2 DSP Implementation

The code for the DSP was also developed by a second group of students in the same software engineering course. The code they developed ran under MS-DOS and read in PCM data from a file and displayed the numerical classification vectors on the screen. The prototype code had to be debugged and then ported to the actual DSP hardware. Once the DSP implementation was tested, further optimizations were required to ensure that all 24 channels could be processed in parallel and in real time. Figure 18 shows a functional view of the algorithms implemented on the DSP.

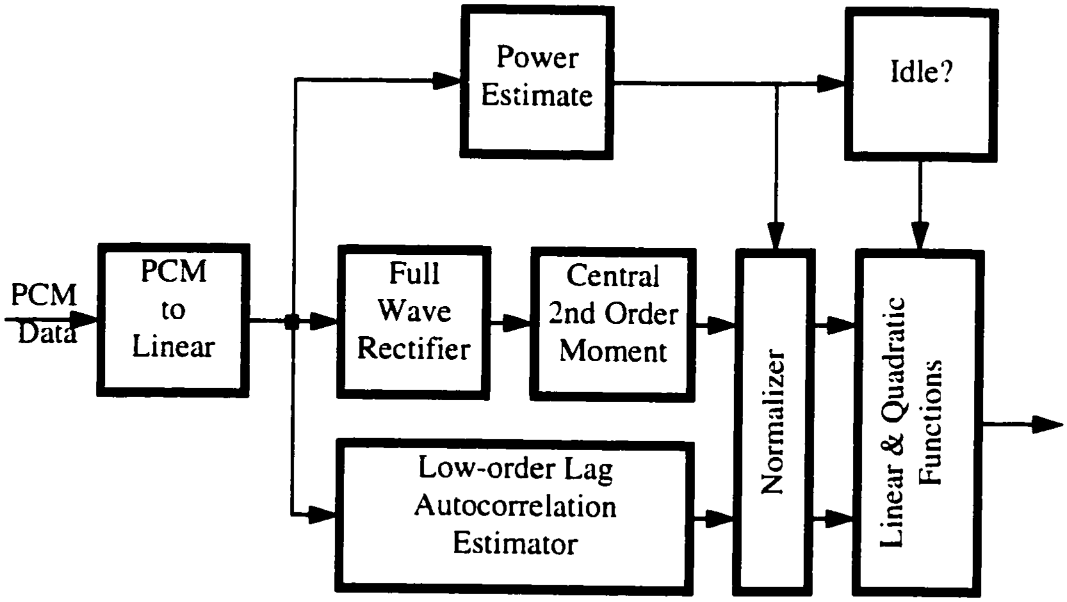


Figure 18: Functional view of DSP algorithm.

PCM data is received from the PC by the DSP and then μ-law decoded. Next the feature variables are calculated; namely, the power, the central second-order moment (normalized w.r.t. power), and the first 10 values (normalized w.r.t. power) of the ACS. Before the central second-order moment is calculated, the signal passes through a full-wave rectifier. Once enough samples have been processed, the feature variables are normalized and the LDFs and QDFs are evaluated. The central second-order moment is first evaluated using equations (2), (6), and (7), and the ACS lags are evaluated by using equation (8); normalization involves dividing by the average power in the particular segment. If the power in the particular signal segment is less than the programmed threshold, then the channel is considered idle and the classification decisions of the LDF and QDF are vetoed.

4.2 File Formats

Two file formats are worth mentioning at this point: the format of pre-recorded telephone calls and the format of the database generated by the prototype voiceband signal classifier.

4.2.1 Recorded Data File Format

Data files recorded by Sewall were stored in binary MAT-file format (generated by using the "save" command in MATLAB). This data can be read by opening the file as a binary file, skipping past a 30-byte MATLAB header, and finally reading the PCM μ-law encoded data as a packed sequence of unsigned bytes (8 bits). These data files can also be read back into MATLAB directly by using the "load" command.

Data files that I recorded can be read in the same manner as described for Sewall; however, my new data files cannot be read back directly into MATLAB using the "load" command as these files do not have the correct MATLAB header information.

Data files from the multilingual speech database, obtained from CSLU, are stored in what CSLU calls a binary NIST .wav format. This format is not the same as the Windows .wav file format. The NIST .wav format incorporates a 1024-byte header followed by unsigned 8 bit PCM μ-law encoded data.

The prototype voiceband signal classifier can read the data files recorded by both Sewall and Sarda, but unfortunately not those from CSLU. A list and description of all recorded telephone files is located in Appendix D.

4.2.2 Database File Format

The prototype voiceband signal classifier is capable of reading and writing database files in dBASE IV file format. This format was chosen because it is an open standard, and is supported by many commercial database and spreadsheet programs. The classifiers adds entries into the database only when the classification vectors change for a given channel and classification method. The classifier also adds entries into the database during a synchronization phase, in which entries are made for every channel and classification method. This synchronization phase is performed once every 15 minutes, and is necessary to speed up future database queries. The information stored for each database entry is as follows:

• Channel:	The channel for which the entry applies. Valid values range from 0 to 23.
• Classification Vector:	The classification vector returned by the DSP. Valid values range from 0 to 23, with 0 representing silence.
• Number of Vectors:	The number of classification vectors returned for this entry.
• Segment Size:	The segment size used for the classification.
• Classification Method:	The classification method used. Valid entries are LR (linear raw), QR (quadratic raw), HR (hybrid raw), LF (linear filtered), QF (quadratic filtered), or HF (hybrid filtered).
• Variables Used:	Which variables were used by the discriminant functions in the classification? Presently, the only valid entry is "ALL", meaning that all variables are used in the classification.
• Starting Date:	The starting date for the entry, in the format MMDDYYYY.
• Starting Time:	The starting time for the entry in 24 hour format, without a ":" between the hour and minutes (e.g. HHMM).
• Starting Seconds:	The starting seconds (0 to 59).
• Synchronization Point:	Was this entry made as part of a synchronization phase (1 for TRUE and 0 for FALSE)?

4.3 Real Time Issues

The GUI running on the PC is able to keep up, in real time, with the processing of the classification vectors generated by the DSP. The circular buffer used to store the incoming classification vectors is generally only needed when the advancing horizontal bars on the real-time graph (described later) reach the right edge of the screen. When this happens, the entire graph shifts leftwards to the midpoint, where the bars resume their advance to the right. The circular buffer is also required when many database updates are scheduled together in a burst. This is not likely to occur in a real system (except during a synchronization phase) because, over the 24 channels, the signal classes do not generally change simultaneously.

4.3.1 Real Time Limitations

The algorithms running on the DSP are able to process data in real time for a segment size of 1020 samples or greater. If a segment size of 252 or 516 is selected, the DSP cannot keep up with the incoming data and starts losing data. This limit is postponed if fewer than 24 channels are monitored and if the LDFs and QDFs are not both being evaluated. The main reason of this limitation has to do with the frequency at which the LDF and QDF are calculated. For the 1020 segment size, the LDFs and QDFs are only calculated about 8 times per second, but for the 252 and 516 segment

Table 13: Frequency of the classification vectors depending on the segment size.

Segment Size		Frequency of Classification Vectors
No. of Samples	Equivalent Time	
252	31.5 ms	31.75 Hz
516	64.5 ms	15.50 Hz
1 020	127.5 ms	7.84 Hz
2 052	256.5 ms	3.99 Hz
4 092	511.5 ms	1.96 Hz
8 196	1.0 s	1.00 Hz

Table 13: Frequency of the classification vectors depending on the segment size.

Segment Size		Frequency of Classification Vectors
No. of Samples	Equivalent Time	
16 380	2.0 s	0.50 Hz
32 772	4.1 s	0.24 Hz
65 532	8.2 s	0.12 Hz

sizes the LDFs and QDFs are calculated about 16 and 32 times per second, respectively. These additional computations cannot be completed in real time for all 24 channels. This can be seen in figures 19, 20, and 21. These plots show how the buffer count, in the DSP circular buffer, changes during different stages in the classification process. To ensure no data loss, the discriminant function calculation and the backed-up feature variable calculations must be completed before then next LDF and QDF calculation (shown by the vertical dashed lines). If this does not occur, the buffer count will con-

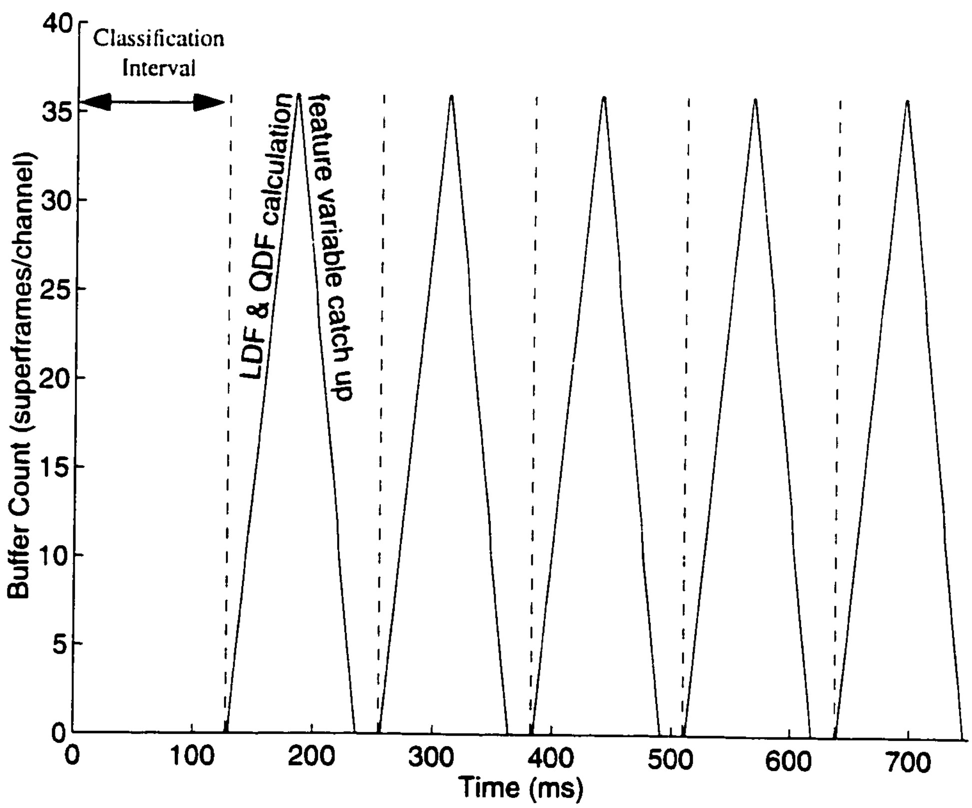


Figure 19: Buffer count using a 1020 segment size.

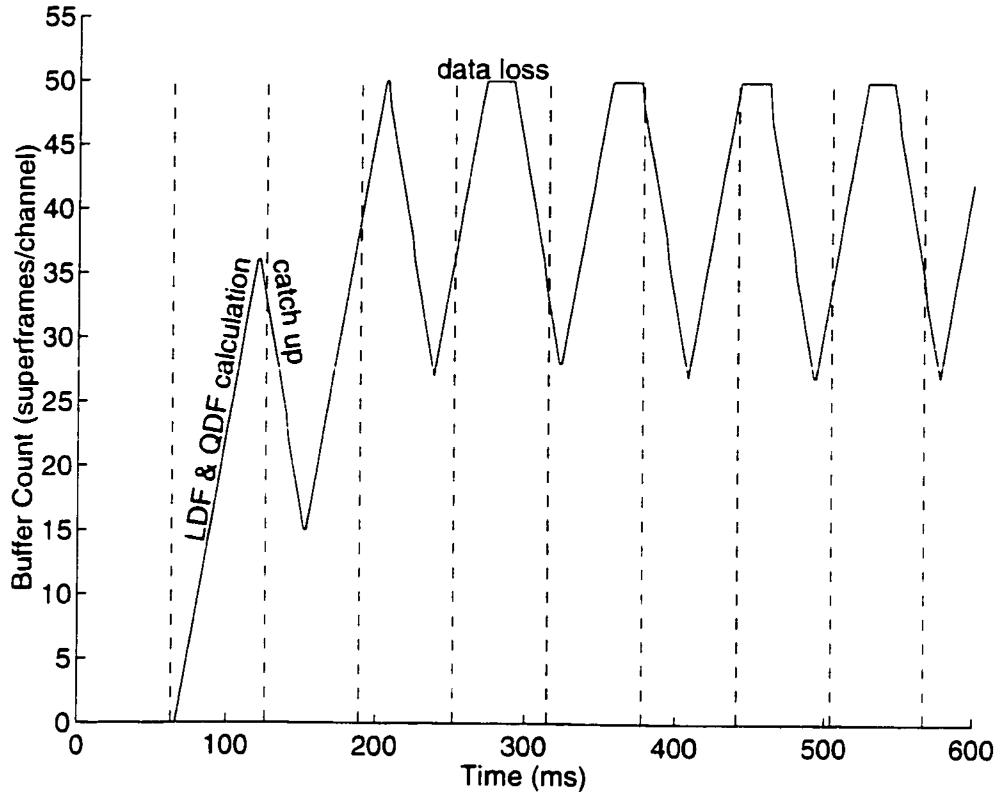


Figure 20: Buffer count using a 516 segment size.

tinue to increase until it exceeds full capacity resulting in a loss of data. For example, for the 1020-sample segment size, the ramping up and down of the buffer count occurs just before the next LDF and QDF calculation. The cycle continues with the beginning of each discriminant function calculation beginning with a buffer count of zero. For the 516 segment size there is enough time to complete the LDF and QDF calculation, but not enough real time for the feature variable catch up stage, resulting in an increase of the buffer count and finally in the loss of data. This is also true when the segment size is 252, the only difference being that there is not even enough time to compute the LDF and QDF calculation before the next classification decision time arrives.

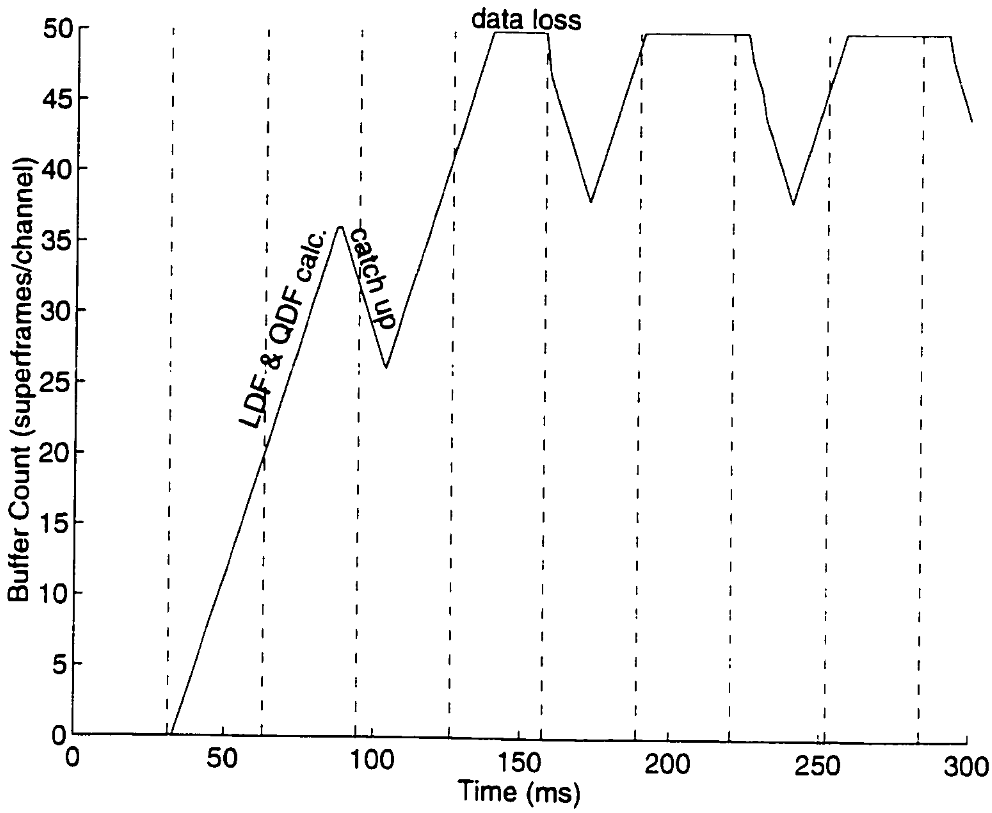


Figure 21: Buffer count using a 252 segment size.

In conclusion, the DSP is only able to classify data in real time if the segment size is greater than 1020 samples, and the LDF and QDF are being evaluated. The discriminant functions were evaluated using 23 classes and 11 feature variables for the LDF, and 6 classes and 11 feature variables for the QDF.

4.3.2 Computational Analysis

There are three stages in the classification process: the DSP ISR for incoming T1 data buffers, the feature variable calculation, and the discriminant function evaluation. Each of these stages differs in its computational requirements, as discussed below. All of our timing values were determined experimentally using counters inserted in various sections of the code.

The ISR stage does not burden the DSP as much compared with the other stages of the classification process. The ISR simply copies data from the shared PC-DSP FIFO buffer into the DSP circular buffer. This takes about 7% of the DSP's time (i.e.

2.8 MIPs) between superframe interrupts (1.5 ms). The ISR is executed by the DSP with a higher priority than other routines; however, ISR handling is delayed if a critical section of code is being processed. A critical section occurs when a group of instructions must be processed without being interrupted. In the classifier the only critical section in the DSP code contains the instructions which update the pointers and flags associated with the circular buffer. This is a critical section because, if this section is interrupted, the interrupting code could corrupt the circular buffer data structure.

The feature variable computation stage is computed once new data arrives. The data is processed 12 samples at a time for each channel (one superframe), and takes about 68% of the DSP's time (i.e. 27.2 MIPS) between superframe interrupts. It is important that this stage be computed efficiently because it directly affects how quickly the buffer gets cleared before the next discriminant function evaluation stage (feature variable catch-up).

The evaluation of the discriminant functions imposes a sudden load at the end of each segment. From the above figures it can be seen that the buffer count swells to a maximum value of 36 during this stage. Since the buffer count increments once every 1.5 ms, this count corresponds to an approximate time of 54 ms.

The actual number of multiply and accumulates required for the LDF and QDF, for N classes and J feature variables, are given by equations (22) and (23) respectively, and are derived from equations (16) and (20).

$$\text{Computations for LDF} = N(J+1) \text{ Multiply and Accumulates} \qquad \text{(Eq 22)}$$

$$\text{Computations for QDF} = N\left(J^2 + 2J + 2\right) \text{ Multiply and Accumulates} \qquad \text{(Eq 23)}$$

By reducing the number of classes, N, and the number of feature variables, J, the number of computations required reduce thus making real time classification at segment sizes of less than 1020 samples possible.

One can obtain an approximate limit on the computational load of the discriminant function evaluation (assuming 23 classes and 11 feature variables) as follows. The DSP just barely keeps up at the 1020 segment size. The upper limit on discriminant function calculation is thus (40 MIPS)*(100%-7%-68%) = 10 MIPS. Clearly this load is inversely proportional to the segment size. Therefore we have,

$$\left(\frac{8000}{1020}\right)M \leq 10 \text{ MIPS}, \qquad \text{(Eq 24)}$$

where M is a constant of proportionality. Thus the load of the discriminant function evaluation is upper bounded by:

$$\left(\frac{8000}{\text{Segment Size}}\right)(1.275) \text{ MIPS}. \qquad \text{(Eq 25)}$$

If the number of feature variables were now reduced from 11 to 6, the computational load on the DSP is reduced (it will be shown later that using six variables results in a higher classification accuracies for both the LDFs and QDFs). The computations required to complete the feature variable calculation stage and discriminant function evaluation stage are both reduced by approximately 45% and 60%, respectively. Note that the computations saved for the feature variable calculation stage is only valid if the same 6 variables are used for both the LDFs and QDFs. With these computational savings it is likely that the classifier can handle a segment size of 516 samples without losing any data samples. Additional computational savings are likely needed to handle a segment size of 252 samples.

Chapter 5

5.0 Prototype Voiceband Signal Classifier Evaluation

After building and testing the prototype voiceband signal classifier at TR*Labs*, field trials were necessary to test and evaluate the classifier in the PSTN. This was accomplished by holding a first field trial at the TELUS Maintenance Engineering Laboratory, and additional field trials at the TELUS toll building and at the Bonnie Doon TELUS Mobility base station.

5.1 TELUS Maintenance Engineering Lab

The intent of the first field trial was to essentially "get our feet wet" and to perform initial system-level verification with access to the PSTN in a lab environment. Using signal data collected during the first field trial, the discriminant functions were re-trained in preparation for further field trials with live traffic. Also, software defects were discovered and fixed, and additional functionality was built into the GUI.

The TELUS maintenance engineering lab contained all of the equipment which was necessary to monitor the network in the 4-wire digital section, as seen in Figure 22. A standard 2-wire analogue subscriber loop linked the 9th floor lab with the PSTN. The analogue signal first went through a hybrid coupler to separate each direction of the signal. It was then PCM μ-law encoded by a first channel bank and finally multiplexed

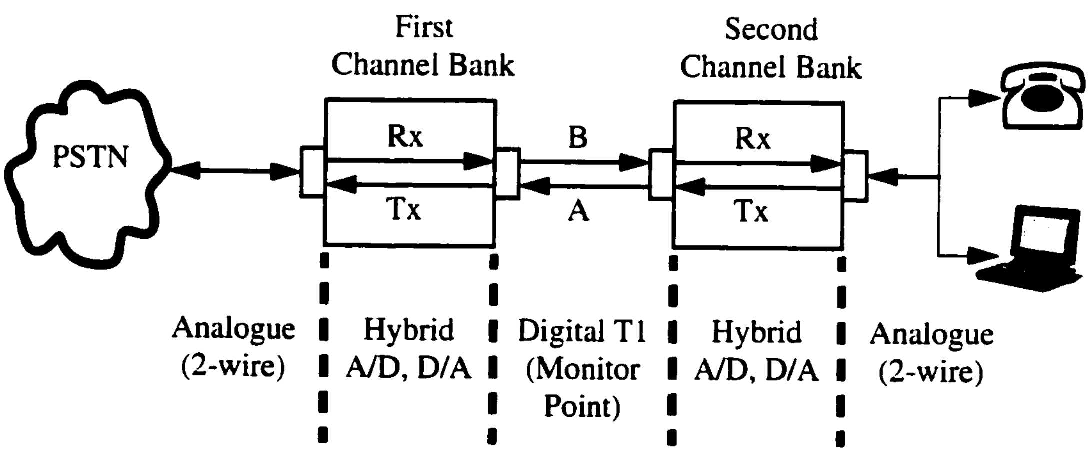


Figure 22: Equipment setup at the TELUS Maintenance Engineering Lab.

onto channel 1 of a T1. This T1 was an input into a second channel bank. The second T1 extracted the data from each channel of the T1, PCM μ-law decoded the data, and then finally combined both directions of the signal for transmission onto another analogue subscriber loop. The roles of each channel bank are reversed for signals transmitted from the lab. This setup models the setup of a typical CO, which is where the classifier is intended to operate.

The programming and setup of all equipment were completed by TELUS maintenance staff. They hooked up the telephone handset to transmit and receive data on the channel 1 of both T1 lines; one T1 for each direction. Both T1s were wired to a DSX-1 patch panel with two ports, labelled ‘A’ and ‘B’. It is from this panel that the classifier monitored the traffic. A ringing signal detector (not shown in Figure 22) was also added to generate a ringing signal to the telephone handset to allow it to receive incoming calls. A list of the equipment and all services used at the maintenance engineering lab is provided in Appendix C.

To generate samples of the various signal classes, we used an Apple Powerbook 150 computer and a USR Sportster modem. This allowed all but the "Binary" and "Voice" signal classes to be generated locally using the modem. All data modem signal samples were generated by dialling into local ISPs, and all fax samples were generated by dialing local fax machines. Faxes were also received from local fax machines and long distance faxback services. Different data rates were obtained by forcing the modem connection bit rate using the AT Command Set. A list of the AT commands used is given in Table 14. The speech samples were generated by having various individuals call the TELUS lab and read text from a book or newspaper. A few duplex conversations were also recorded. A complete listing and description of all data files recorded is given in Appendix D.

Table 14: AT commands used for the USR 33.6 modem [35] [37].

Command	Description
+++	Escapes to on-line-command mode
AT&N1 - AT&N16	Sets connect speed from 300 bps to 33 600 bps

Table 14: AT commands used for the USR 33.6 modem [35] [37].

Command	Description
ATI6 & ATI11	Returns diagnostic information for current connection
ATS33=0 - ATS33=63	Disable/Enable symbol rates for V.34 protocol

5.2 TELUS Downtown Toll Building

The goal of the second field trial was to test the system in the PSTN and to gather call statistics on real T1 lines. The T1 lines included both long distance inter-city trunks as well as local trunks originating from different neighbourhoods in and around the city of Edmonton.

The TELUS Toll Building was an ideal place to test the classifier with real, live traffic. The toll building routes traffic that originates from different sources. For example, long distance trunks from cities such as Toronto, Vancouver and Calgary are available, along with many different local trunks. This field trial lasted for about 3 weeks, which gave us plenty of time to monitor many different types of trunks. Since the traffic being monitored was live traffic, no recordings of individual calls were made. Instead the classification vectors were captured and stored into a database. A detailed description of all database files is provided in Appendix D. Figure 23 shows how the classifier was set up at the toll building.

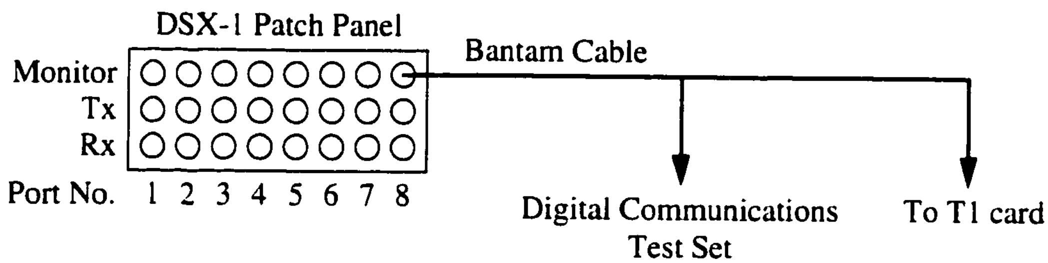


Figure 23: TELUS Toll Building setup.

A connection was made from the "Monitor" port on a DSX-1 patch panel using a standard bantam cable. A Digital Communications Test Set was connected in parallel with the classifier. This test set was used to verify the electrical connection and framing functionality to the DSX-1 patch panel. It was also useful for independently verifying the type of traffic being carried on the T1 because it had the ability to select and audibly play the signal on any channel. The test set was strictly used as a research tool to verify the rough operation of the classifier on individual T1 channels. A list of all equipment used at this location is provided in Appendix C.

5.3 TELUS Mobility Bonnie Doon Base Station

The goal of the third and fourth field trials was to test the system with mostly wireless traffic, and to evaluate filtering techniques that we developed to improve the accuracy of the classifier. TELUS Mobility was also interested in determining the amount of fax and data traffic that was being carried by their analogue wireless network. These field trials took place at the Bonnie Doon TELUS Mobility base station. This base station routed and switched traffic that originated from all base stations in northern Alberta. These include base stations serving oil and gas facilities as well as local communities. This diversity resulted in T1s that had very different traffic patterns. The classifier setup was almost identical to the setup at the TELUS Toll Building.

The third field trial lasted for five days in October 1998. During this field trial two T1s were monitored: one T1 terminated in Quigley, Alberta, and the other T1 terminated in Worsley, Alberta. Each T1 was monitored for approximately two days. While monitoring the Worsley T1, the local base station experienced a power bump, resulting in the database being corrupted for this T1. A software defect was also discovered during this field trial which limited the speech filter window to 5 seconds. This defect was noted and fixed before the fourth field trial.

The fourth field trial lasted for ten days in November 1998 and involved monitoring a T1 to Peace River, Alberta. This T1 was monitored for the ten days consecutively without any intervention, which demonstrated the stability of the classifier. During this field trial the filters (described later) appeared to be working as desired.

Chapter 6

6.0 Evaluation of the Signal Classifier

This chapter describes the results of off-line simulations and on-site field trials involving various versions of the prototype classifier. This chapter is organized into sections which show results obtained before the first field trial at the TELUS maintenance lab, after the first field trial, after the second field trial at the Edmonton toll building, and finally after the third and fourth field trials at the Bonnie Doon base station.

6.1 Sewall's Results

Sewall's algorithms classified a signal into one of nine possible signal classes as, shown in Table 1. The training data for these algorithms included both recorded and simulated calls covering each of the nine classes. A complete list and description of all the recorded and simulated data files is provided in Appendix D. For comparison purposes, Tables 15 and 16 show the percent classification accuracies using Sewall's evaluation data. The "Predicted Class" is the class assigned to the segmented data, and the "Actual Class" is the class that the segmented data really belongs to. This method of presenting the classification accuracies will be consistently used throughout this section. Each table will also note the discrimination method, and the segment size, N, used to generate the classification accuracies. The segment size represents the number of continuous samples taken from the data stream for each classification vector. Please note that all feature variables are used in all tables unless stated otherwise.

Table 15: Sewall percent classification accuracy (N=2052, LDF).

	Predicted Class									
	Class	1	2	3	4	5	6	7	8	9
Actual Class	1	100.00	-	-	-	-	-	-	-	-
	2	-	100.00	-	-	-	-	-	-	-
	3	-	-	100.00	-	-	-	-	-	-
	4	-	-	-	98.27	1.73	-	-	-	-
	5	-	-	-	13.74	86.26	-	-	-	-
	6	-	-	-	-	-	99.83	0.17	-	-
	7	-	-	-	-	-	-	100.00	-	-
	8	-	-	2.65	0.24	-	-	-	95.42	1.69
	9	-	-	-	-	-	-	-	-	100.00

Table 16: Sewall percent classification accuracy (N=2052, QDF).

Class	1	2	3	4	5	6	7	8	9
1	100.00	-	-	-	-	-	-	-	-
2	-	100.00	-	-	-	-	-	-	-
3	-	-	99.92	-	-	-	-	0.08	-
4	-	-	-	100.00	-	-	-	-	-
5	-	-	-	-	100.00	-	-	-	-
6	-	-	-	-	-	100.00	-	-	-
7	-	-	-	-	-	-	100.00	-	-
8	-	-	-	-	-	-	-	99.76	0.24
9	-	-	-	-	-	-	-	-	100.00

When using QDFs the overall classification accuracy is almost 100%, but when using LDFs, the classification accuracy, over all classes, is only 97.75%. The majority the classification errors, when using LDFs, is due to classes 4, 5, and 8. The reasons for relatively high misclassifications in these classes will be discussed later. Using QDFs consistently result in a higher classification accuracy, but compared with LDFs the QDFs are more computationaly expensive. A segment size of 2052 was chosen because it appears to be a good compromise between classification accuracy and precision. Again, this will be discussed in the later sections. Both tables were generated using approximately half of the available signal data as the training set, and the remaining half as the test set. A training set is used to compute the discriminant functions (i.e., the coefficients for the LDFs and QDFs). A test set is then used to measure the classification accuracy of the discriminant functions. Allocation of data into the training and test sets was determined using a random number generator. The data was first partitioned into 2052 segments, and then the random number generator determined which segment would be included in the training and test sets. The exact number of 2052 segments used for both the training and test set is shown in Table 17.

Table 17: Train and test set counts for Sewall's data.

Class	Training Set Count	Test Set Count	Total Segments
1	1179	1214	2393
2	728	696	1424
3	1125	1223	2348
4	897	982	1879
5	2225	2162	4387
6	608	575	1183
7	471	499	970
8	395	415	810
9	491	483	974

6.2 Results of First Field Trial

After building a fully functional prototype classifier, a first field trial was arranged at the TELUS Maintenance Engineering Lab. The classifier, in general, performed very well with signals of known classes recorded at the lab. Samples of real signals covering all but one of Sewall's nine classes were collected. Attempts were made to connect at all different combinations of bit rates, baud (one symbol per second) rates, and carrier frequencies that are specified by each modem standard. The only signal class that was not generated was the random 64 kbps binary. Even though voice samples were generated and collected, they were not actually used later on after the arrival of the multilingual speech database from CSLU.

The real-time display on the GUI allowed for immediate verification of the classified signal class. From visual inspection, the classifier appeared to perform very well. Certain classes did have accuracy problems, and these classes were noted for further study. The next section describes the initial results, and the corresponding improvements that were made to the classifier.

6.2.1 Initial Results

The data collected at the Maintenance Engineering Lab needed to be processed to remove unwanted information from the data files. For example, all data calls have an initial negotiation phase which needs to be omitted from the signal training sets. We decided that the classifier would not be trained to recognize training tones as one or more separate classes. This was done to minimize the number of classes and to avoid having to deal with the complexity of the relatively brief training signal intervals. Figure 24 illustrates the typical call structure for voice, data, and facsimile calls. For voice calls, only the portions of the call that contained clear speech samples were processed, with silence thus largely removed. For data calls the initial negotiation phase needed to be removed, and for facsimile calls the initial negotiation phase and the FSK signalling information was removed. This processing was necessary to ensure that the training and test sets contain only known samples of the signals that are to be classified. The initial results, before re-training the classifier using the processed data, are shown in Table 18.

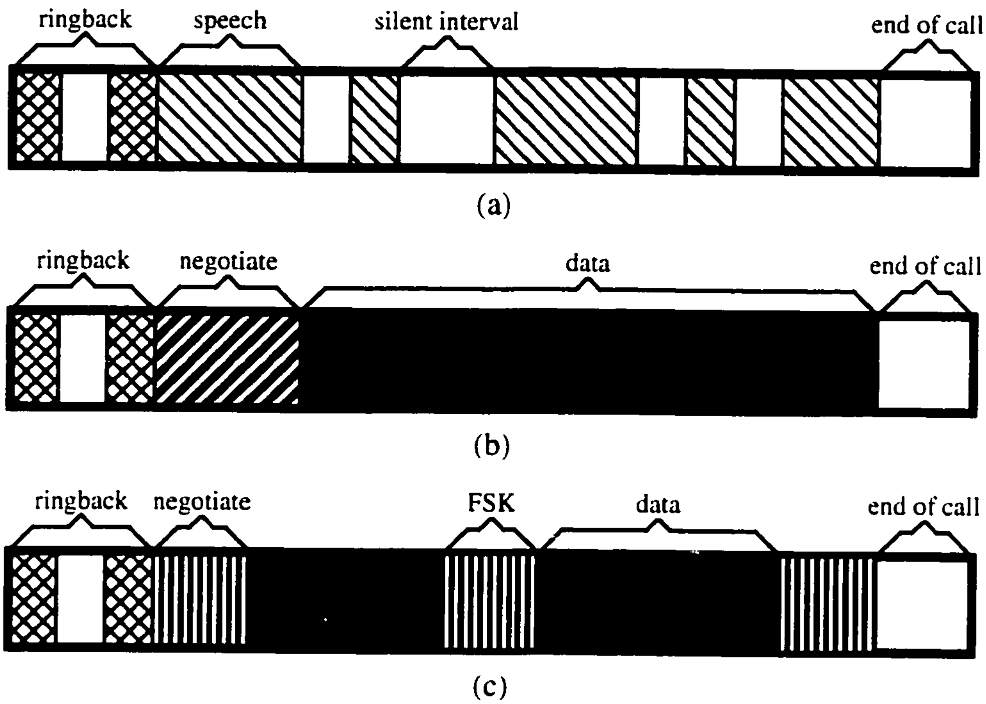


Figure 24: Typical call structure for (a) voice, (b) data, and (c) facsimile calls.

Table 18: Percent classification accuracy using Sewall's data to train and Sarda's to test (N=2052, LDF).

Class	1	2	3	4	5	6	7	8	9
1	100.00	-	-	-	-	-	-	-	-
2	-	81.11	-	18.89	-	-	-	-	-
3	-	-	71.19	13.41	15.36	0.01	0.02	0.01	-
4	-	-	-	91.66	8.34	-	-	-	-
5	-	-	-	2.30	97.70	-	-	-	-
6	-	-	-	-	0.12	99.81	0.07	-	-
7	-	-	-	-	-	4.40	95.59	0.01	-
8	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-	-

This table was generated using Sewall's data to train the classifier, while testing on the data collected at the field trial. No values are shown for class 9 because this signal type (binary) was not generated at the lab, and no numbers are shown for class 8 (voice) because only a relatively small sample was taken and this sample was dropped from consideration shortly after the arrival of the Multilingual Speech Database CD-ROM. Overall, the classifier performed well (above 90% averaged over all classes), with the exception of classes 2 and 3. Table 19 shows the classification accuracy using the QDF with the same training and test sets.

Table 19: Percent classification accuracy using Sewall's data to train and Sarda's to test (N=2052, QDF).

Class	1	2	3	4	5	6	7	8	9
1	91.56	-	-	-	-	-	-	8.44	-
2	-	-	-	-	-	-	-	100.00	-
3	-	-	46.05	1.35	-	-	-	52.60	-
4	-	-	-	84.94	1.63	-	-	13.43	-
5	-	-	-	-	88.00	-	-	12.00	-
6	-	-	-	-	1.15	72.76	-	26.09	-
7	-	-	-	-	-	96.77	0.70	2.53	-
8	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-	-

Surprisingly the classifier performed rather poorly when using QDF. Signals of class 2 are classified as speech (class 8), and class 7 are assigned to class 6. Also, class 3 (V.34) was frequently misclassified as speech. This result was very surprising because Sewall generally found that the QDFs gave higher accuracies than LDFs. From these initial results it appeared that there was no benefit to using QDFs over LDFs to classify the signals.

So far the training set has been completely composed of Sewall's data, and the test set has been composed of data collected at the field trial. The next few results use a combination of Sewall's data and the data collected at the field trial as the training and testing sets.

Table 20: Percent classification accuracy using Sewall's and Sarda's data to train and test (N=2052, LDF).

Class	1	2	3	4	5	6	7	8	9
1	100.00	-	-	-	-	-	-	-	-
2	-	100.00	-	-	-	-	-	-	-
3	-	-	88.30	11.54	0.13	-	0.01	0.02	-
4	-	-	0.31	89.61	10.08	-	-	-	-
5	-	-	-	4.69	95.29	0.02	-	-	-
6	-	-	-	-	-	99.02	0.98	-	-
7	0.01	-	-	-	-	1.09	98.90	-	-
8	-	0.24	8.03	-	-	-	-	91.24	0.49
9	-	-	-	-	-	-	-	-	100.00

After re-training the classifier a number of improvements are quite evident as shown in Table 20. The classification accuracies for classes 2, 3, and 7 have greatly increased. Accuracies for classes 4, 5, and 6 did go down, but only very marginally. Classes 4 and 5 appear to be difficult for the classifier to separate. The majority of misclassifications for class 4 are incorrect assignments to class 5 (10.08%). This is a problem that Sewall observed [2] (but to a lesser extent), as shown in Table 15. The only major difference between these two signals is that class 4 (V.29 fax) is half-duplex QAM with a 1700 Hz carrier, while class 5 (V.32 modem) is full-duplex QAM with an 1800 Hz carrier [2]. These similar carriers and bit rates imply similar PSDs. This hunch is confirmed by the plots in Figure 25. The quadratic results are much better after re-training the classifier as shown in Table 21. Clearly QDF accuracies are quite sensitive to the training conditions. The problem classes noted for the linear case do not appear to be a problem in the quadratic case.

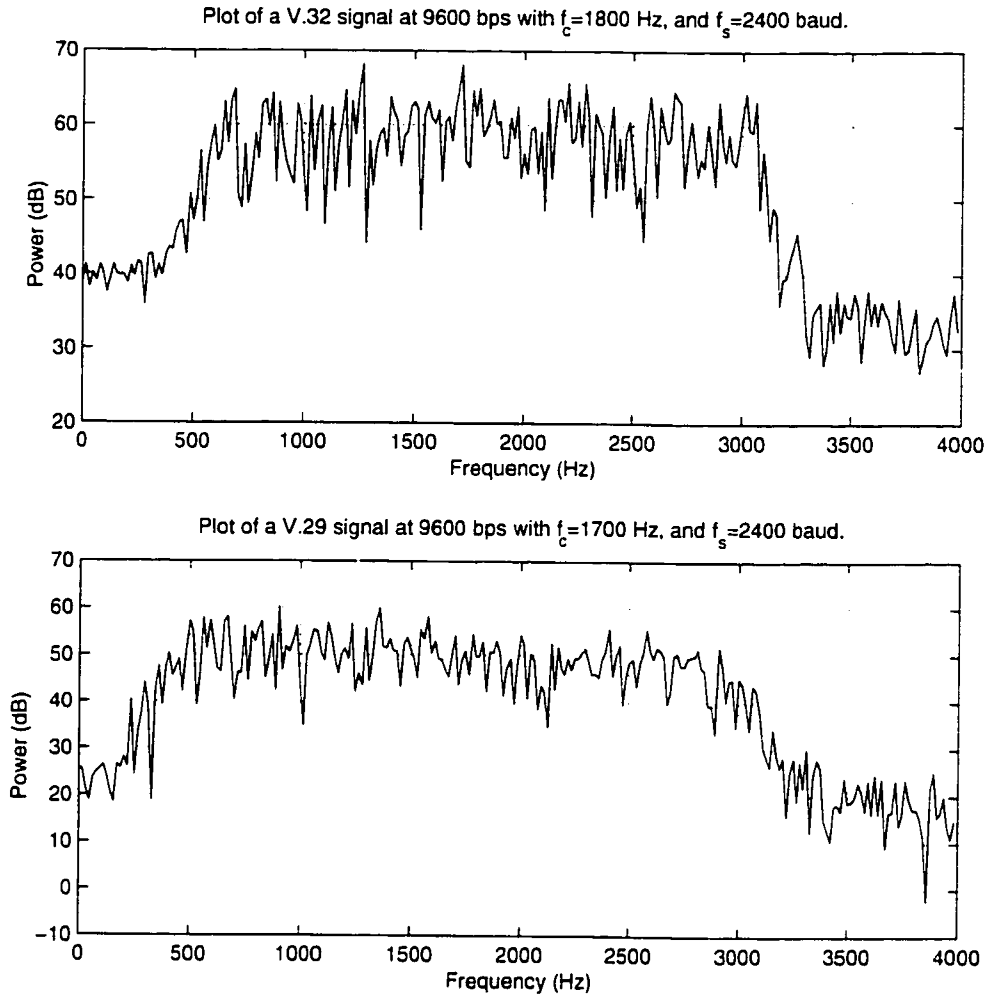


Figure 25: PSD plots for a V.32 signal and a V.29 signal.

Table 21: Percent classification accuracy using Sewall's and Sarda's data to train and test (N=2052, QDF).

Class	1	2	3	4	5	6	7	8	9
1	100.00	-	-	-	-	-	-	-	-
2	-	100.00	-	-	-	-	-	-	-
3	-	-	98.97	0.96	-	-	-	0.07	-
4	-	-	0.02	98.84	1.14	-	-	-	-
5	-	-	0.02	-	99.98	-	-	-	-
6	-	-	-	-	0.18	99.82	-	-	-
7	-	-	-	-	-	0.28	99.68	0.04	-

Table 21: Percent classification accuracy using Sewall's and Sarda's data to train and test (N=2052, QDF).

Class	1	2	3	4	5	6	7	8	9
8	-	-	-	-	-	-	-	100.00	-
9	-	-	-	-	-	-	-	-	100.00

6.2.2 Results after Increasing the Number of Classes

After analyzing the structure of the three general groups of calls (voice, data, and fax), it was determined that adding new classes would be useful. In facsimile calls, FSK signalling is used by the connecting modems to exchange control information. FSK signalling is also used to communicate information at the page breaks in a fax call. A classifier that has an FSK class can thus determine the number of pages sent for each fax call. Another complication is that both V.32*bis* and V.17 belong to the same class. V.32*bis* is a data modem standard that connects at a maximum bit rate of 14.4 kbps. V.17 is a fax standard, also with a maximum connect speed of 14.4 kbps. However, using FSK signalling the two signal types can be separated because FSK signalling would not be present in a V.32*bis* connection.

Another new class was ringback. Ringback is the faked ringing signal that is heard by the calling party to signal that the called party's handset is being rung. To train and test the classifier, recordings of the ringback signal were made by calling various local numbers. Ringback is, in most cases, carried back over the PSTN using the same channel resources that would carry the information of the ensuing call. For some long distance calls, the ringback signal might not actually be present over the bulk of the connection in the PSTN so that resources are not tied up unnecessarily before the called party picks up the handset. Instead the ringback signal would be generated at the local switch. Only when the conversation portion of the call begins would PSTN channel resources be required to carry the call. In all cases monitored during field trials, both local calls and long-distance calls, the ringback signal was in fact present.

Finally, the DTMF signalling tones were added producing 12 new classes. Samples of the DTMF tones were generated by simply pressing and holding each of the 12 buttons on the handset. When a calling party dials up a call, the DTMF digits are not usually carried by the PSTN in-band beyond the local CO. Instead, the digits are intercepted by digit collection circuits and translated into messages carried separately by the packet-switched SS7 signalling network to route the call. However, it is still common to use the digits on a handset once a call has been setup. For example, many companies have menu-driven messaging systems requiring the input of DTMF digits from the calling party. In these situations the DTMF digits will be carried in-band by the PSTN. For simplicity and privacy reasons, all of the 12 DTMF classes were combined and reported as one class in the prototype voiceband signal classifier.

In addition to the three new classes, the recent appearance of a new modem standard required some changes to the classifier. In early 1998 the ITU finally released the V.90 standard. This standard allows for bit rates of up to 56 kbps from a digital source of data (e.g. an ISP) to a customer connected via a conventional analogue local loop (i.e. the downlink) [28][29]. The uplink uses the 33.6 kbps V.34 standard which falls into the existing class 3 category. Within the network, a V.90 downlink appears to be a purely random binary data stream (class 9).

The enlarged new list of classes is shown in Table 22. Note that V.32 terbo does not belong to any class, because specific information regarding this protocol could not be obtained (it is proprietary) and a recording using this protocol was never made. The classification accuracies using the expanded signal classes are listed in Tables 23 and 24 below for both LDFs and QDFs.

Table 22: Expanded list of signal classes.

Group No.	Signals Included
1	V.22 and V.22*bis* forward channels
2	V.22 and V.22*bis* reverse channels
3	V.34 & V.90 uplink
4	V.29 all speeds
5	V.32, V.32*bis*, and V.17 at speeds greater than 2400 bps
6	V.27*ter* at 4800 bps
7	V.27*ter* at 2400 bps

Table 22: Expanded list of signal classes.

Group No.	Signals Included
8	Speech
9	Random PCM samples & V.90 downlink
10	FSK signalling
11	Ringback
12	DTMF tones for 0, 1, 2, ..., 9, *, #

Table 23: Percent classification accuracy using Sewall's and Sarda's data to train and test (N=2052, LDF).

Class	1	2	3	4	5	6	7	8	9	10	11	12
1	90.64	-	-	-	-	-	-	-	-	-	-	9.36
2	-	100.00	-	-	-	-	-	-	-	-	-	-
3	-	-	87.70	12.14	0.14	-	0.01	-	-	-	0.01	-
4	-	-	0.17	89.22	10.61	-	-	-	-	-	-	-
5	-	-	-	5.19	94.81	-	-	-	-	-	-	-
6	-	-	-	-	-	98.99	1.01	-	-	-	-	-
7	-	-	-	-	-	1.05	98.95	-	-	-	-	-
8	-	0.25	5.99	-	-	-	-	89.02	0.50	-	2.49	1.74
9	-	-	-	-	-	-	-	-	100.00	-	-	-
10	-	-	-	-	-	-	-	-	-	100.00	-	-
11	-	-	-	-	-	-	-	-	-	-	100.00	-
12	-	-	-	-	-	-	-	-	-	-	-	100.00

Overall the classification accuracy over all classes remains about the same, except for class 1. By adding three additional classes the LDF accuracy for class 1 has dropped by 9.36% to 90.65%, with all of the misclassifications being assigned into class 12. For the quadratic case, accuracy over all classes remained at the same high levels, even with classes 1 and 12.

Table 24: Percent classification accuracy using Sewall's and Sarda's data to train and test, (N=2052, QDF).

Class	1	2	3	4	5	6	7	8	9	10	11	12
1	100.00	-	-	-	-	-	-	-	-	-	-	-
2	-	100.00	-	-	-	-	-	-	-	-	-	-
3	-	-	99.16	0.78	-	-	-	0.06	-	-	-	-
4	-	-	0.02	98.58	1.40	-	-	-	-	-	-	-
5	-	-	0.04	0.02	99.94	-	-	-	-	-	-	-
6	-	-	-	-	0.17	99.83	-	-	-	-	-	-
7	-	-	-	-	-	0.40	99.59	0.01	-	-	-	-
8	-	-	-	-	-	-	-	100.00	-	-	-	-
9	-	-	-	-	-	-	-	-	100.00	-	-	-
10	-	-	-	-	-	-	-	-	-	100.00	-	-
11	-	-	-	-	-	-	-	-	-	-	100.00	-
12	-	-	-	-	-	-	-	2.22	-	-	-	97.78

6.2.3 Problem Classes

As noted earlier, Sewall observed a high misclassification rate between classes 4 and 5. This weakness re-appeared in the accuracy results after re-training the classifier with the new data. However, the classification accuracy of class 3 (V.34) fell from 100% to 87.7%. V.34 requires that the connecting modems be able to negotiate at various possible symbol rates, data rates, and carrier frequencies (refer to Table 9). These different combinations were tested at the first field trial by forcing the modem, using the AT command set, to negotiate unusual connection parameter combinations. A listing of the various baud rates, bit rates, and carrier frequencies recorded is listed in Appendix D. If the modems are not forced then, according to our measurements, land line to land line connections will tend to default to a symbol rate of 3429 baud and a carrier frequency of 1959 Hz. This carrier frequency is at the centre of the passband to accommodate the broadest possible passband spectrum at the high symbol rate of 3429 baud. Figure 26 shows PSD plots of three different V.34 signals. When connecting at the default carrier frequency f_c and symbol rate f_s, the entire available spectrum is efficiently filled with signal power. In the 1600 Hz and 2000 Hz plots the available passband is not filled as aggressively because the symbol rate has been reduced from 3429 Hz to 2400 Hz and 3000 Hz, respectively.

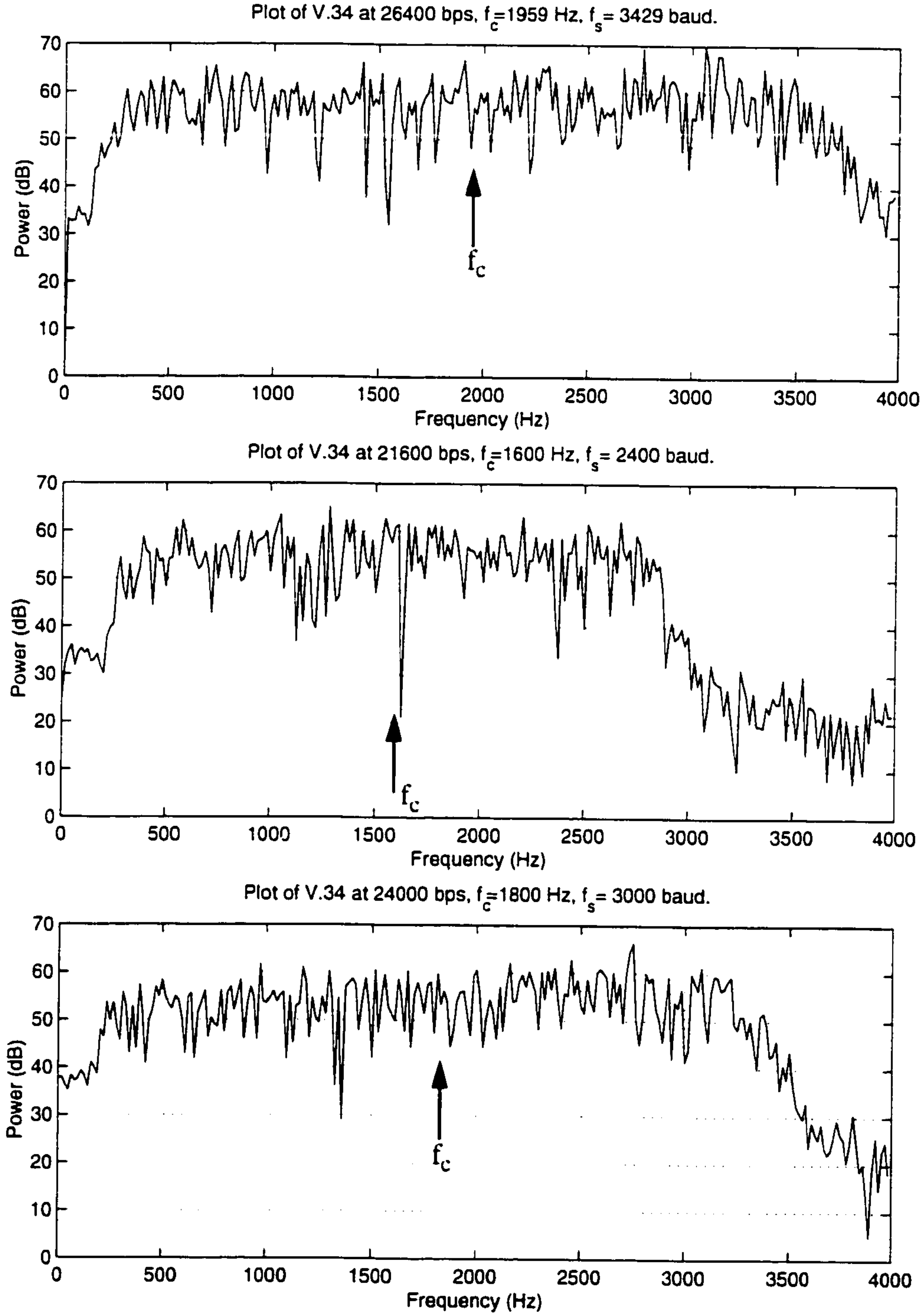


Figure 26: PSD comparison of different V.34 connection modes.

After noting the difficulties encountered when classifying uncommon V.34 connection modes, new experiments were done using only the common V.34 calls to train and test the classifier. Table 25 shows the results using only the common V.34 signal settings ($f_c = 1959\ \mathrm{Hz}$, $f_s = 3429\ \mathrm{baud}$) to train and test the classifier. Class 3 (V.34) accuracy has increased from 87.7% to almost 100%. This raises a difficult decision: can unusual V.34 modes be ignored? During the first field trial we noticed that the connecting modems will choose the default carrier and symbol rate, with the data rate varying

Table 25: Percent classification accuracy using Sewall's and Sarda's data to train and test (N=2052, LDF, Std V.34).

Class	1	2	3	4	5	6	7	8	9	10	11	12
1	90.36	-	-	-	-	-	-	-	-	-	-	9.64
2	-	100.00	-	-	-	-	-	-	-	-	-	-
3	-	0.01	99.90	-	0.04	-	0.02	0.01	-	-	0.02	-
4	-	-	-	88.27	11.73	-	-	-	-	-	-	-
5	-	-	-	6.75	93.23	0.02	-	-	-	-	-	-
6	-	-	-	-	-	98.94	1.06	-	-	-	-	-
7	-	-	-	-	-	1.16	98.83	-	-	-	-	0.01
8	-	0.26	3.34	0.51	-	-	-	90.49	1.03	-	2.31	2.06
9	-	-	-	-	-	-	-	-	100.00	-	-	-
10	-	-	-	-	-	-	-	-	-	100.00	-	-
11	-	-	-	-	-	-	-	-	-	-	100.00	-
12	-	-	-	-	-	-	-	-	-	-	-	100.00

depending (apparently) on line conditions. By issuing the appropriate AT commands our V.34 modem could be forced to connect at an unusual different symbol and carrier frequency, even with good line conditions. These other modes would presumably be used automatically if the training sequence found that the available passband was substandard. Even if unusual signal settings are present, the classification accuracy after training using only the common V.34 modes is still above 87%.

6.2.4 Multilingual Speech Database

To increase the number of speech samples used in the training and testing sets, we used the Multilingual Speech Database obtained from CSLU. Ten files were selected from the English speech data files. Out of the ten, five speakers were male and five were female. Each speaker recites many items such as their location, sex, age, etc. Each speaker is also asked to speak about something for 1 minute. These "story" files were of suitably long duration, roughly 1 minute, and contained a fair number of good speech samples (not silence). All files used from this database are listed in Appendix D. Only the English speech files (files with the "EN" prefix) were used to train and test the classifier (other languages are considered later).

Table 26: Percent classification accuracy using Sewall's and Sarda's data to train and test (N=2052, LDF, Std V.34, Incl. EN).

Class	1	2	3	4	5	6	7	8	9	10	11	12
1	89.44	-	-	-	-	-	-	-	-	-	-	10.56
2	-	100.00	-	-	-	-	-	-	-	-	-	-
3	-	-	99.90	-	0.04	-	0.03	0.01	-	-	0.02	-
4	-	-	-	85.63	14.37	-	-	-	-	-	-	-
5	-	-	-	11.56	88.40	0.04	-	-	-	-	-	-
6	-	-	-	-	-	98.90	1.10	-	-	-	-	-
7	-	-	-	-	-	1.20	98.79	-	-	-	-	0.01
8	-	0.25	1.97	1.35	-	-	-	91.63	0.49	-	1.72	2.59
9	-	-	-	-	-	-	-	-	100.00	-	-	-
10	-	-	-	-	-	-	-	-	-	100.00	-	-
11	-	-	-	-	-	-	-	-	-	-	100.00	-
12	-	-	-	-	-	-	-	-	-	-	-	100.00

As shown in Table 26, the classification rates using LDFs range from 85% - 100%. Again classes 4 and 5 appear to be hard to separate. Other difficulties include confusion between class 1 (V.22*bis* forward) and class 12 (DTMF tones). Speech was also misclassified into classes 2, 3, 4, 9, 11, and 12. Using QDFs the accuracies are even better. Classes 4 and 5 are much more reliably resolved, and class 1 signals are no longer misclassified into class 12. Also, by using QDFs the accuracy for speech has increased from 91.63% to 100%. The classification accuracy for class 12 (DTMF tones) has unfortunately fallen from 100% to 95.03%. Most of these were caused by

Table 27: Percent classification accuracy using Sewall's and Sarda's data to train and test (N=2052, QDF, Std V.34, Incl. EN).

Class	1	2	3	4	5	6	7	8	9	10	11	12
1	100.00	-	-	-	-	-	-	-	-	-	-	-
2	-	100.00	-	-	-	-	-	-	-	-	-	-
3	-	-	99.90	-	-	-	-	0.10	-	-	-	-
4	-	-	-	98.80	1.20	-	-	-	-	-	-	-
5	-	-	-	0.02	99.96	-	-	0.02	-	-	-	-
6	-	-	-	-	0.04	99.96	-	-	-	-	-	-
7	-	-	-	-	-	0.12	99.85	0.03	-	-	-	-
8	-	-	-	-	-	-	-	100.00	-	-	-	-
9	-	-	-	-	-	-	-	-	100.00	-	-	-
10	-	-	-	-	-	-	-	-	-	100.00	-	-
11	-	-	-	-	-	-	-	-	-	-	100.00	-
12	-	-	-	-	-	-	-	4.97	-	-	-	95.03

misclassifications of DTMF digits "4" and "8" into class 8 (speech).

Table 28 shows the results obtained using the ALN algorithms. Overall these accuracies are better than the LDF results and are comparable to the QDF results. The parameters used by the ALN software to generate these results are outlined in Appendix E.

Table 28: Percent classification accuracy using Sewall's and Sarda's data to train and test (N=2052, ALN, Std V.34, Incl. EN).

Class	1	2	3	4	5	6	7	8	9	10	11	12
1	99.96	0.04	-	-	-	-	-	-	-	-	-	-
2	-	99.86	0.09	0.05	-	-	-	-	-	-	-	-
3	0.02	0.01	99.91	0.02	-	0.02	-	0.01	0.01	-	-	-
4	-	-	-	98.75	1.25	-	-	-	-	-	-	-
5	-	-	-	0.13	99.85	0.02	-	-	-	-	-	-
6	-	-	-	-	-	99.96	0.04	-	-	-	-	-
7	-	-	-	-	-	-	99.99	-	0.01	-	-	-
8	-	0.25	0.25	-	-	0.25	1.85	95.06	1.11	0.62	0.49	0.12
9	-	-	-	-	-	-	-	-	100.00	-	-	-

Table 28: Percent classification accuracy using Sewall's and Sarda's data to train and test (N=2052, ALN, Std V.34, Incl. EN).

Class	1	2	3	4	5	6	7	8	9	10	11	12
10	-	-	-	-	-	-	-	-	0.22	99.78	-	-
11	-	-	-	-	-	-	-	-	-	-	100.00	-
12	-	-	-	-	-	-	-	-	-	-	-	100.00

6.2.5 Effects of Varying Segment Sizes

Thus far all off-line classification experiments have been done using a fixed segment size of 2052 samples, which is equivalent to almost four classification vectors per second (4 Hz). Changing the segment size changes the number of classifications vectors calculated per second. Table 29 shows the average accuracies, taken over all classes, of using linear, quadratic, and ALN discrimination methods with varying segment sizes.

Table 29: Accuracy for varying segment sizes and discrimination methods.

Clas s	Linear (%)				Quadratic (%)				ALN (%)			
	2052 4 Hz	1020 8 Hz	516 16 Hz	252 32 Hz	2052 4 Hz	1020 8 Hz	516 16 Hz	252 32 Hz	2052 4 Hz	1020 8 Hz	516 16 Hz	252 32 Hz
1	89.44	80.48	72.31	62.31	100.0	100.0	100.0	100.0	99.96	99.91	99.93	99.96
2	100.0	100.0	99.73	97.67	100.0	100.0	100.0	99.89	99.86	99.80	100.0	99.93
3	99.90	99.88	99.76	95.83	99.90	99.89	99.85	96.27	99.91	99.90	99.85	97.57
4	85.63	83.27	82.20	73.96	98.80	96.78	98.19	87.95	98.75	98.61	98.11	91.81
5	88.40	83.44	81.77	76.46	99.96	98.88	98.38	90.38	99.85	99.49	99.08	93.40
6	98.90	96.46	91.29	81.38	99.96	99.82	99.60	96.53	99.96	99.95	99.76	98.60
7	98.79	95.80	90.56	83.62	99.85	98.75	96.81	92.79	99.99	99.95	99.96	99.56
8	91.63	87.45	81.99	78.73	100.0	99.57	99.64	98.68	95.06	98.17	97.96	97.50
9	100.0	100.0	99.79	98.70	100.0	100.0	100.0	100.0	100.0	98.83	99.89	99.77
10	100.0	100.0	100.0	99.98	100.0	99.80	100.0	99.73	99.78	99.60	99.95	99.83
11	100.0	100.0	100.0	100.0	100.0	100.0	99.45	99.87	100.0	78.70	99.17	99.61
12	100.0	100.0	100.0	100.0	95.03	100.0	100.0	100.0	100.0	100.0	100.0	99.89
Total	96.06	93.90	91.62	87.39	99.46	99.46	99.33	96.84	99.43	97.74	99.47	98.12

Generally, as the segment size increases, the classification accuracy also increases. A larger segment size allows more information about the signal to be considered by the classifier before generating a classification vector. For LDFs, the accuracy averaged over all classes ranges from 96% to 87% for segment sizes falling from 2052 to 252 samples. The largest drops in accuracy occur in classes 1, 4, 5, 6, 7, and 8. The classification accuracy for QDFs falls from 99% to 97%, with largest drops appearing in classes 4, 5, and 8. Using the ALN method the classification accuracy only falls from 99% to 97%, with the largest drops occurring in classes 4 and 5. Overall the QDF and ALN methods did not differ significantly in average accuracy (~2%). However, when using the LDF method the accuracy fell 10% as the segment size was shortened from 2052 to 252.

Additional simulations were conducted by further increasing the segment length to determine if the classification accuracy would improve to 99% over all classes while using LDFs. The data used to generate the classification accuracy values for the 2052 sample (4 Hz) segment length were used to generate the data to be used for the 4092 sample (2 Hz) segment length. This was done by taking the values of each corresponding feature variable and then simply averaging them. The data for the 1 Hz and 1/2 Hz were then obtained similarly.

Table 30: Accuracy using larger segment sizes and LDFs.

Class	Linear (%)		
	0.5 Hz	1 Hz	2 Hz
1	99.40	99.26	97.49
2	100.0	100.0	100.0
3	99.94	99.88	99.94
4	91.45	90.53	89.00
5	99.44	94.62	92.29
6	99.84	99.92	99.66
7	99.57	99.43	99.48
8	100.0	99.48	96.44
9	100.0	100.0	100.0

Table 30: Accuracy using larger segment sizes and LDFs.

Class	Linear (%)		
	0.5 Hz	1 Hz	2 Hz
10	100.0	100.0	100.0
11	100.0	100.0	100.0
12	100.0	100.0	100.0
Total	99.14	98.51	97.85

Using a segment length of 16416 samples (~ 1/2 Hz) the classification accuracy over all classes improves from 96.06% (using a 2052 segment size) to 99.41%. The classes which showed the most improvements were classes 1, 5, and 8.

6.2.6 Effects of Classifying Non-English Speech

Three languages other than English were investigated. Ten "story" files from the multilingual speech database were selected from the available: Japanese, French, and German files. The exact files used in this simulation are listed in Appendix D. The test set was composed of the processed data (with silent intervals removed) from these three languages. Our goal here is to observe how our classifier (trained with only English speech) performs when classifying non-English speech. The results are shown in Table 31.

Table 31: Accuracies For non-English speech files (N=2052, LDF).

Class	English (%)	Japanese (%)	French (%)	German (%)
1	-	-	-	-
2	0.25	-	-	0.14
3	1.97	0.92	0.91	0.54
4	1.35	0.31	0.34	0.27
5	-	-	-	-
6	-	-	-	-
7	-	-	1.03	-
8	91.63	91.72	89.00	94.73

Table 31: Accuracies For non-English speech files (N=2052, LDF).

Class	English (%)	Japanese (%)	French (%)	German (%)
9	0.49	-	0.11	-
10	-	-	-	-
11	1.72	6.75	7.3	3.38
12	2.59	0.31	0.91	0.95

For English speech the classification accuracy for recognizing speech was 91.63%, which is very close to the non-English speech results. French was recognized as speech with an accuracy just below 89%, while German was recognized with 95% accuracy. The accuracy for Japanese was intermediate between French and German.

6.2.7 Effects of Changing the Feature Variables

All of the simulations performed thus far have been done using all of the feature variables, namely, the first 10 values of the ACS and the normalized central second-order moment. These variables were chosen by Sewall because the first 10 values of the ACS appeared to him to capture much of the relevant signal information present in the PSD, and the normalized central second-order moment was shown to be useful when separating speech from various voiceband data types [2]. By using these variables, the overall accuracy of the classifier appears quite high; however, for maximum accuracy a classifier may not necessarily require all of these variables. As we will report below, using subsets of discriminant variables from the 11 available can actually increase the accuracy. It is important to choose the combination of feature variables that are best, in a statistical sense, at separating all of the classes. Such a combination tends to include feature variables that are effective at separating easily confused classes. Problems with even two easily confused classes can quickly degrade the accuracy of a classifier. Also, the optimum subset of variables is not necessarily the same for each classification method.

By reducing the number of variables, the number of computations required by both the LDF and QDF is reduced. For N classes and J feature variables the number of computations required by LDFs and QDFs is given by Equations (22) and (23), respectively. Using these equations, the number of basic computations (multiply and accumulates) saved by reducing the number of feature variables, J, from J_2 to J_1 can be derived as shown in Equations (26) and (27).

$$\text{Computations saved for LDF} = N(J_2 - J_1),\ J_2 > J_1 \qquad \text{(Eq 26)}$$

$$\text{Computations saved for QDF} = N\left(J_2^2 - J_1^2\right) + 2N(J_2 - J_1),\ J_2 > J_1 \qquad \text{(Eq 27)}$$

Figure 27 shows how the number of computations increases for increasing numbers of feature variables. As expected, when using LDFs, the change follows a linear curve; when using QDFs, the increase in the number of computations is quadratic. Consequently, if there is not enough real-time to perform classification using a desired segment size, the number of variables can always be reduced. This will reducc the number of computations, but may produce a drop in classification accuracy. Clearly it is important to determine which subset of variables provides the best overall accuracy.

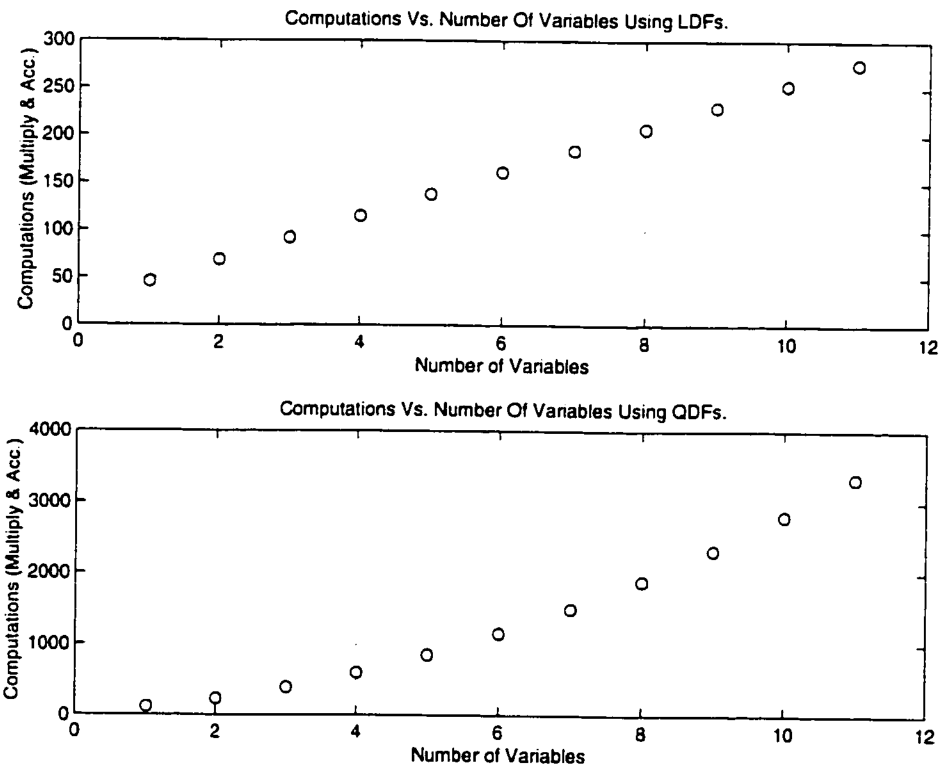


Figure 27: Computations using different numbers of feature variables (23 classes).

6.2.7.1 Optimum Feature Variable Sets

Using the data that Sewall collected, off-line classification experiments were conducted to determine optimal sets of variables. This involved determining both the number of feature variables to use and, specifically, which ones to select. Various simulations were initially done by hand-picking subsets of variables. The variables were selected partly based on the rankings generated by the "stepwise mode" option in SPSS. This mode allows forward selection and backward elimination of the variables at each step in the selection process [36]. This results in a table that ranks the variables according to their effectiveness in separating the classes. Sewall performed these tests and generated ranking tables when discriminating between all classes, speech and non-speech classes, and all non-speech classes. Variable selection was also based on using variables that were best at separating the problem classes.

As mentioned above, various simulations were done while hand-picking variables. N2 was selected because it ranked the highest when discriminating between speech and non-speech. Rd2, Rd4, and Rd5 were selected because they were ranked the highest when classifying between various non-speech signals. Rd8 was selected because it was ranked very high when classifying between classes 4 and 5 (which are hard to separate). Finally, Rd1 was also selected because it had an above average ranking in two of the three simulations. Also, from the plots of the ACS of various signals, it is apparent that the ACS lags have the greatest magnitudes (and thus greater "signal-to-noise" ratios) in the low order lags. Using this combination of discriminants with Sewall's data resulted in an average classification accuracy of 97.85%. This is a slightly higher accuracy than when using all variables (96.02%).

But is this the optimum subset of variables? Would the accuracy increase if even more different variables were selected? If more than six, but fewer than 11, variables are used, would the accuracy increase? These questions are difficult to answer because of the number of different combinations of discriminants that can be used. The total number of combinations for n elements in the set, and k elements in the subset is given by equation (28) [14].

$$c(n,k) = \frac{n!}{k!\,(n-k)\,!} \qquad \text{(Eq 28)}$$

Using 11 variables and subsets of sizes ranging from 1 to 11 results in 2047 possible combinations. Since SPSS has a command line interface, a script was written to automate the running of the simulations for all possible combinations. The results are summarized in Table 32.

Table 32: Maximum accuracy using different numbers of variables (N=2052, LDF).

Variables	Number of Variables Used										
	1	2	3	4	5	6	7	8	9	10	11
Rd1				X	X	X	X	X	X	X	X
Rd2		X	X		X	X	X	X	X	X	X
Rd3										X	X
Rd4	X			X	X	X	X	X	X	X	X
Rd5				X	X	X	X	X	X	X	X
Rd6			X				X		X	X	X
Rd7											X
Rd8		X	X	X	X	X	X	X	X	X	X
Rd9								X	X	X	X
Rd10								X	X	X	X
N2						X	X	X	X	X	X
Pc (%)	73.86	92.43	95.77	96.90	97.41	97.85	97.84	97.40	97.02	96.76	96.02

Using only Rd4 resulted in a surprisingly high classification rate (73.86%) over all classes. The greatest accuracy was achieved using six variables (Rd1, Rd2, Rd4, Rd5, Rd8, and N2). This happens to be the same combination of variables that was hand-picked from the previous discussion! Using more than six variables did not improve the accuracy; in fact, using more variables increasingly reduced the overall accuracy. This is a fortunate result as reducing the number of variables from 11 to 6 reduces the number of computations required by LDFs by 55%.

Table 32 shows the results of using different combinations of variables using pseudo quadratic discriminant functions. Again, using only six variables results in the best accuracy (100%); however these six are not the same as determined for LDFs.

When using the pseudo quadratic method, the discriminants that provide the best results are Rd1, Rd2, Rd3, Rd5, Rd6, and Rd7. Actually, when using 6 to 9 variables, many different combinations yielded essentially the same average classification accuracies, within our ability to measure accuracy. The table below shows only one combination when using 6 to 9 variables, with the remaining summarized in Appendix F.

Table 33: Accuracy using different combinations of variables (N=2052, Pseudo QDF).

Variables	Number of Variables Used										
	1	2	3	4	5	6	7	8	9	10	11
Rd1		X			X	X	X	X	X	X	X
Rd2	X		X	X	X	X	X	X	X	X	X
Rd3		X			X	X	X	X	X	X	X
Rd4			X	X					X	X	X
Rd5				X	X	X	X	X	X	X	X
Rd6			X	X	X	X	X	X	X	X	X
Rd7						X	X	X	X	X	X
Rd8							X	X			X
Rd9									X	X	X
Rd10								X	X	X	X
N2										X	X
Pc (%)	76.55	96.17	99.73	99.96	99.99	100.0	100.0	100.0	99.99	99.99	99.62

When using pseudo quadratic discriminant functions, the variable ranking is different than when using LDFs. Simulations were carried out using the data Sewall collected and the resulting ranking is summarized below. The variable ranking when using LDFs can be found in [2].

Table 34: Discriminant variable rankings (*N*=2048, Pseudo QDF).

Rank	Wilks' Lambda	Mahalanobis Distance	F-ratio	Rao's V	Unexplained Variance
1	Rd2	Rd4	Rd4	Rd2	Rd2
2	Rd3	Rd8	Rd1	Rd4	Rd1
3	Rd7	Rd5	Rd5	Rd5	Rd4
4	Rd1	Rd7	Rd8	Rd7	Rd5

Table 34: Discriminant variable rankings (*N*=2048, Pseudo QDF).

Rank	Wilks' Lambda	Mahalanobis Distance	F-ratio	Rao's V	Unexplained Variance
5	Rd4	Rd9	Rd7	Rd1	Rd3
6	Rd5	Rd6	Rd9	Rd6	Rd6
7	Rd6	Rd10	Rd6	Rd3	Rd8
8	Rd8	Rd1	Rd10	Rd9	Rd7
9	N2	N2	N2	Rd8	Rd9
10	Rd9	Rd3	Rd3	N2	N2
11	Rd10	Rd2	Rd2	Rd10	Rd10

Using these optimum variable subsets, simulations were done using QDFs. The accuracies were identical to the pseudo quadratic accuracies when using 6 variables. Additional simulations were done using some of the 7 and 8 variable combinations, again yielding the same results.

For both the linear and pseudo quadratic methods, the results when using the optimum subset of variables does not result in a great improvement over using all variables. Also, different combinations of six or more variables produced comparable accuracies. For example, all variable combinations above 4 resulted in a 99% accuracy when using pseudo quadratic discriminant functions. For LDFs the accuracy varied from 95.77% to 97.85% when using 3 or more variables. To further explore this effect the accuracy range, when using different numbers of variables, was investigated and is summarized in Figures 28 and 29.

For LDFs the upper limit is above 92% when using the best set of 2 or more feature variables. The lower range increases as the number of variables increases. This suggests that, when using relatively few variables, it is especially important to select them properly or the accuracy could be considerably reduced. When using 9 or more feature variables, the lower limit is above 90%, so the particular choice of variables is not as important.

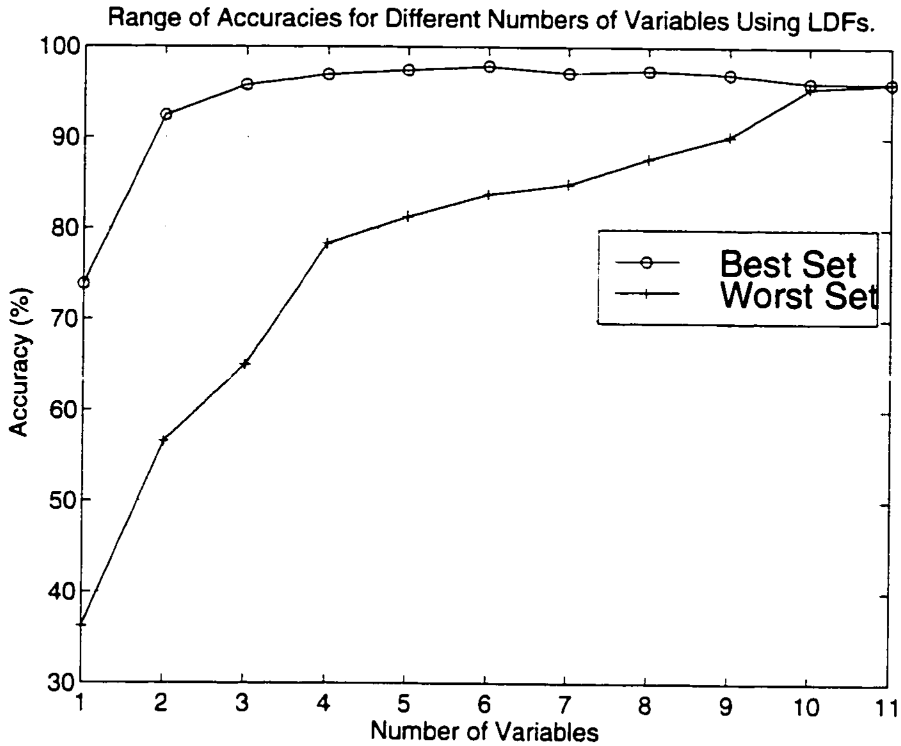


Figure 28: Range of accuracies for different numbers of variables using LDFs.

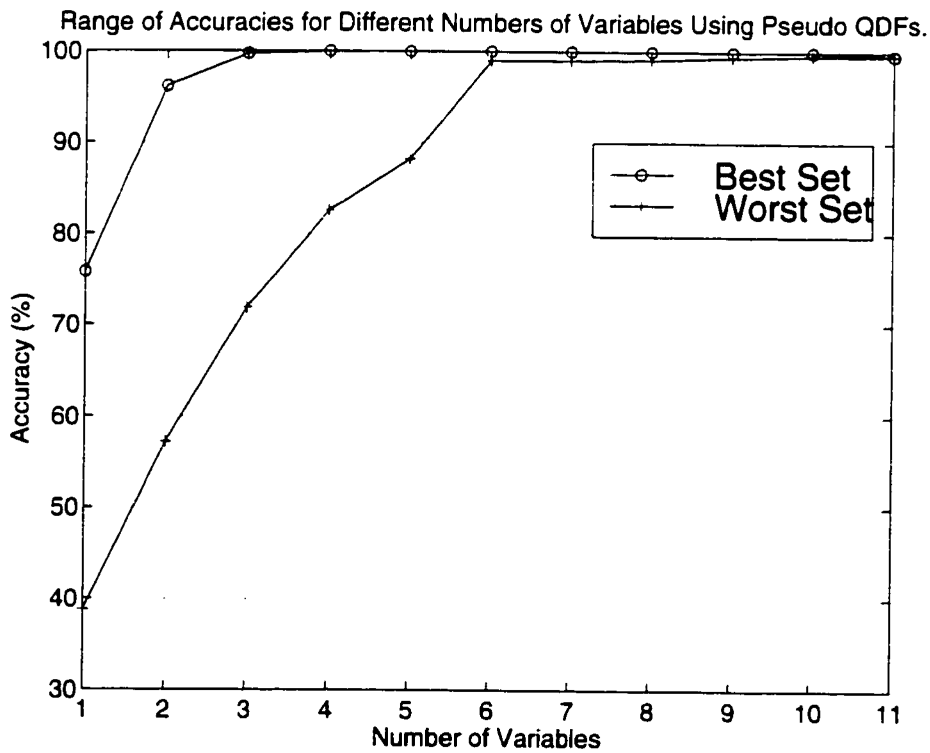


Figure 29: Range of accuracies for different numbers of variables using pseudo QDFs.

The results when using pseudo quadratic discriminant functions are similar to those when using LDFs. The fewer the number of discriminants that are used, the more the accuracy varies; however, any more than 6 variables results in less than a 1% deviation. This suggests that any arbitrary combination of 6 or more variables will results in an accuracy of greater than 99%. By reducing the number of variables from 11 to 6, the number of computations is reduced significantly by 66%. Since there is no added value in using more than 6 variables, why waste scarce real-time resources by including any more variables.

Using the optimal variable selections for linear and pseudo quadratic discriminant functions, simulations were conducted with the data collected at the first field trial. The results are summarized in the tables below for both 9 and 12 classes, and both LDFs and QDFs.

Table 35: Percent classification accuracy using variables Rd1, 2, 4, 5, 8, and N2 (N=2052, LDF, std V.34).

Class	1	2	3	4	5	6	7	8	9
1	100.00	-	-	-	-	-	-	-	-
2	-	87.55	-	12.45	-	-	-	-	-
3	-	-	99.74	0.01	0.01	0.03	0.02	0.03	0.16
4	-	-	-	91.78	8.22	-	-	-	-
5	-	-	-	8.06%	91.94	-	-	-	-
6	-	-	-	-	0.02	99.71	0.27	-	-
7	-	-	-	-	-	0.37	99.62	-	0.01
8	-	-	3.34	0.77	-	-	-	95.89	-
9	-	-	-	-	-	-	-	-	100.00

The accuracy over all classes using the optimum set of variables is 96.2%, compared with 96.02% when using all variables. For QDFs the overall accuracy is 99.71% when using the optimum variable set, compared to 99.70% when using all variables. Again, reducing the number of variables does not significantly change the overall accuracy of the classifier, but the number of computations is considerably reduced when using fewer feature variables.

Table 36: Percent classification accuracy using variables Rd1, 2, 3, 5, 6, and 7 (N=2052, QDF, std V.34).

Class	1	2	3	4	5	6	7	8	9
1	100.00	-	-	-	-	-	-	-	-
2	-	100.00	-	-	-	-	-	-	-
3	-	-	99.90	-	-	-	-	0.10	-
4	-	-	-	98.34	1.66	-	-	-	-
5	-	-	-	0.11	99.89	-	-	-	-
6	-	-	-	-	0.46	99.54	-	-	-
7	-	-	-	-	-	0.22	99.72	0.06	-
8	-	-	-	-	-	-	-	100.00	-
9	-	-	-	-	-	-	-	-	100.00

The next two tables show the results when the number of classes is expanded from 9 classes to 12 classes (counting the 12 DTMF signals as one class). Using LDFs the overall accuracy is reduced to 94.56% (from 96.2%), and the accuracies using QDFs is also reduced from 99.71% to 99.68%. In both cases note that the addition of 14 classes reduced the accuracy by less than 2%.

Table 37: Percent classification accuracy using variables Rd1, 2, 4, 5, 8, and N2 (N=2052, LDF, std V.34).

Class	1	2	3	4	5	6	7	8	9	10	11	12
1	76.97	-	-	-	-	-	-	-	-	-	-	23.03
2	-	86.67	-	13.33	-	-	-	-	-	-	-	-
3	-	-	99.73	-	0.02	0.02	0.03	0.01	0.17	-	0.02	-
4	-	-	-	90.52	9.48	-	-	-	-	-	-	-
5	-	-	-	10.19	89.63	0.02	-	-	-	-	-	0.16
6	-	-	-	-	0.06	99.69	0.25	-	-	-	-	-
7	-	-	-	-	-	0.43	99.56	-	0.01	-	-	-
8	-	-	2.22	0.99	-	-	-	92.11	0.12	-	1.85	2.71
9	-	-	-	-	-	-	-	-	100.00	-	-	-
10	-	-	-	-	-	-	-	-	-	100.00	-	-
11	-	-	-	-	-	-	-	-	-	-	100.00	-

Table 37: Percent classification accuracy using variables Rd1, 2, 4, 5, 8, and N2 (N=2052, LDF, std V.34).

Class	1	2	3	4	5	6	7	8	9	10	11	12
12	-	-	-	-	-	-	-	-	-	-	-	100.00

Table 38: Percent classification accuracy using variables Rd1, 2, 3, 5, 6, and 7 (N=2048, QDF, std V.34).

Class	1	2	3	4	5	6	7	8	9	10	11	12
1	100.00	-	-	-	-	-	-	-	-	-	-	-
2	-	100.00	-	-	-	-	-	-	-	-	-	-
3	-	-	99.90	-	-	-	-	0.10	-	-	-	-
4	-	-	0.02	98.57	1.41	-	-	-	-	-	-	-
5	-	-	-	0.16	99.82	-	-	0.02	-	-	-	-
6	-	-	-	-	0.27	99.71	0.02	-	-	-	-	-
7	-	-	-	-	-	0.33	99.59	0.08	-	-	-	-
8	-	-	0.25	-	-	-	-	99.75	-	-	-	-
9	-	-	-	-	-	-	-	-	100.00	-	-	-
10	-	-	-	-	-	-	-	-	-	100.00	-	-
11	-	-	-	-	-	-	-	1.23	-	-	98.77	-
12	-	-	-	-	-	-	-	-	-	-	-	100.00

6.2.8 Performance Measures

The classification accuracies presented thus far have assumed a significantly high signal power within the given segment window. If a mixture of signals (including silence) is present within a segment window, then signal power can drop significantly. Also, in real signals, transitions from one signal type to another do not necessarily occur conveniently at segment window boundaries. Figure 30 shows an example of the call structure of three separate calls: speech, data and facsimile. The vertical lines that run through each bar indicate the boundaries between adjacent segment windows. Within the speech call, a segment window could include only ringback, ringback and silence, ringback and speech, speech and silence, or finally speech alone. For the data call, a segment window could include only ringback, ringback and silence, ringback and the negotiation phase, the negotiation phase, data and the negotiation phase, data

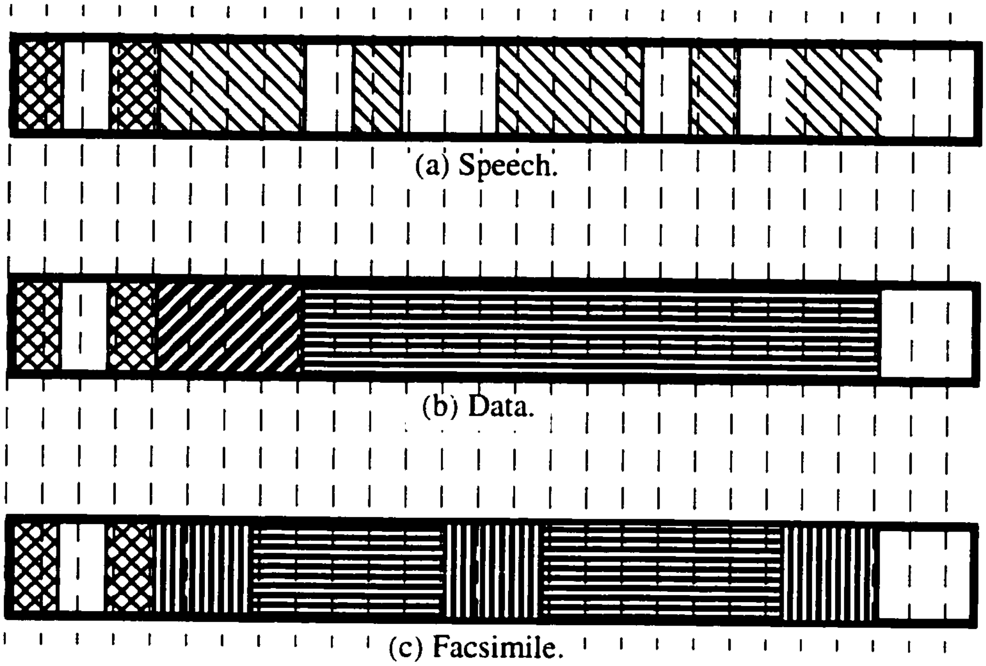


Figure 30: Segment window effects on calls with different structures.

alone, or data with silence. For the facsimile call, a segment window could include only ringback, ringback and silence, ringback and FSK signalling, FSK signalling alone, FSK signalling and data, or data alone. Given that it is inevitable that the classifier will be called upon to classify mixtures of signals, several important questions arise: How will mixtures of different signals affect the accuracy of the classifier? How accurately can the actual signal transitions be tracked? Can incorrect classification decisions be produced? If so, what can be done to control the problem?

In an early attempt to answer some of these questions, a demo file was created and passed to the classifier. This file was created using equal-duration data samples from all signal classes, which were then joined together in round robin format (e.g. class 1, class 2, class 3, ..., class 11, class 12). This sequence was then repeated, each time reducing the duration of each signal class by a factor of two.

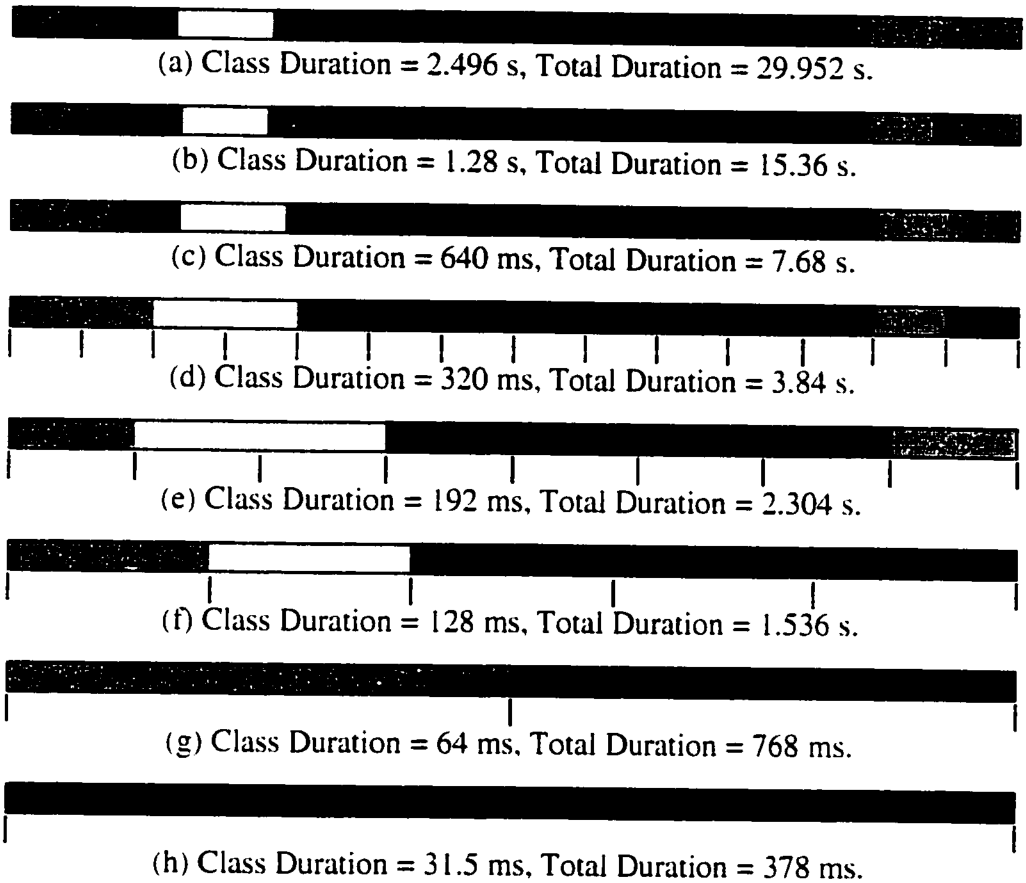
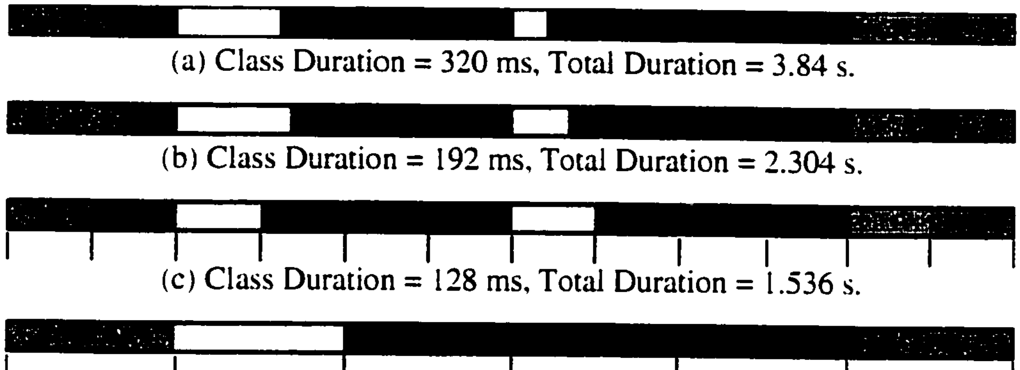


Figure 31: Classification results using different class durations (N=2052).

The results of this simulation, using a 2052 segment size, are shown in Figure 31 (please note that each bar has a different time scale). By the class duration we mean the amount of time that each signal type is present during the particular simulation. By the total duration we mean the total length of the demo file. The different colours represent the different classes (12 in total). The tick marks below the bars represent the classification segment boundaries. Tick marks were not shown for (a), (b), or (c); however, the number of classification vectors for these bars are 116, 59, and 29, respectively. The distinction between each class can be clearly made for class durations as small as 640 *ms*. Interestingly, a mixture of all signal classes gets classified as speech (as shown in Figure 31 (h)). For this simulation the segment size was 256.5 *ms*, or 2052 samples. As a rule of thumb, the classifier's segment length should be no greater than half the smallest class duration. Even if the class duration is greater than 513 *ms*, misclassifications can still occur between signal transitions. Referring to Figure 31a, the class duration is 2.5 *s*, but one misclassification still occurs at the transition from class 6 to class 7. Around this transition, a mixture of both signal types is present within the 2052 segment window. In this instance the signal is classified into class 4; however, in general, the resulting class is unpredictable. If a smaller segment size is used, the signal transitions can be more precisely tracked. This can be seen in Figure 32, which shows the classification results using a segment size of 127.5 *ms* (1020 samples).



(a) Class Duration = 320 ms, Total Duration = 3.84 s.

(b) Class Duration = 192 ms, Total Duration = 2.304 s.

(c) Class Duration = 128 ms, Total Duration = 1.536 s.

(d) Class Duration = 64 ms, Total Duration = 768 ms.

Figure 32: Classification results using different class durations (N=1020).

With a class duration of 320 *ms* and 192 *ms*, the different classes can still be seen. Even at 128 *ms* the only class that has disappeared is FSK signalling (dark blue). Even smaller class durations result in too many mixtures of different signals within one segment window.

Figure 33 shows the classification results using an even smaller segment size of 31.5 *ms* (252 samples). For class durations greater than twice the segment size, or 63 *ms*, each signal class is present. Note that by reducing the segment size, the number of misclassifications has also increased, even when a strong signal is present during the entire segment window.

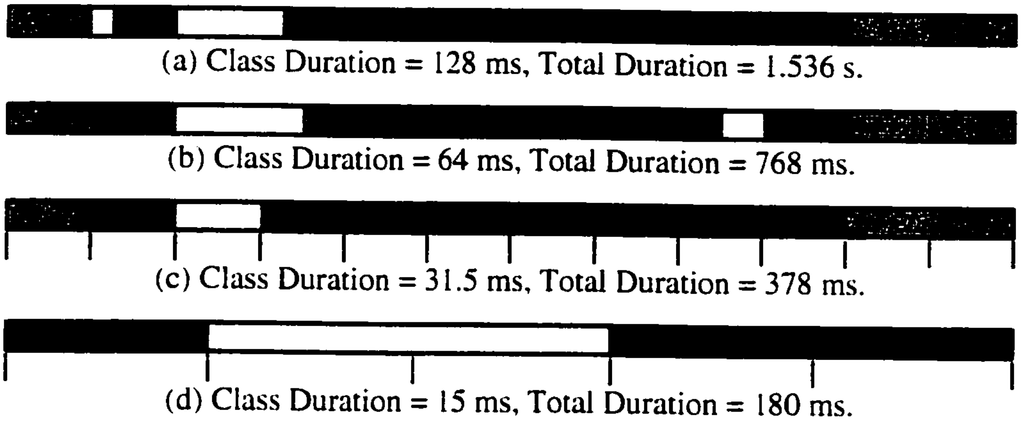


Figure 33: Classification results using different class durations (N=252).

Another limitation when attempting to classify real signals is that the segment boundaries are asynchronous with respect to signal transition times. The "phase" of the segment windows could effect the classification results. Figure 34(a) shows the actual signal class sequence in the simulated signal, while Figure 34(b) shows the corresponding classification results. In Figure 34(a) the high level represents class 3 (V.34), and the low level represents silence. This test signal was generated as follows: the V.34 was turned on for exactly 8160 samples and then turned off for 8160 samples, repeatedly each time increasing the off segment by 102 samples. This caused the signal boundaries to advance with respect to the classifier's segment boundaries by 102 samples each time. This resulted in the segment window not falling completely on the signal pulse at every interval.

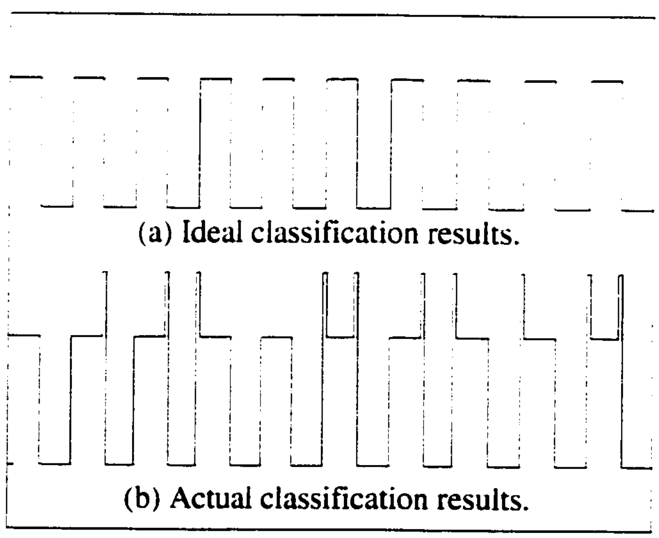


Figure 34: Comparison of ideal and actual classification results.

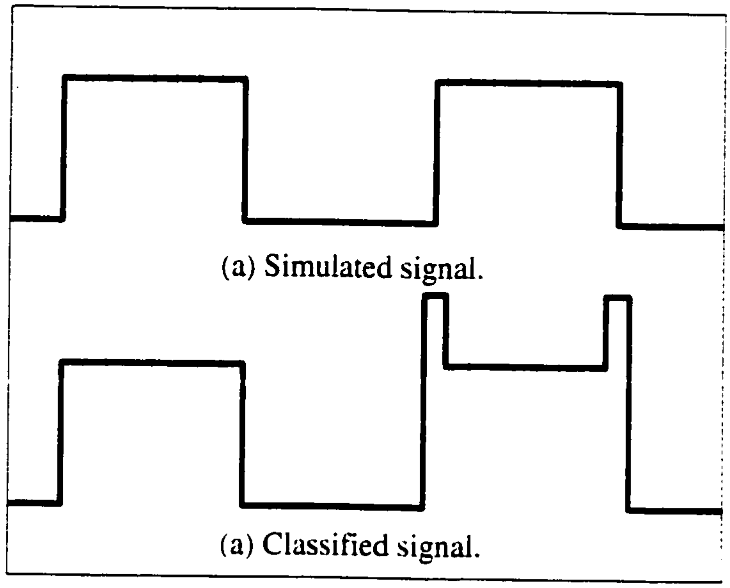


Figure 35: Misclassifications resulting from a mixture of V.34 and silence.

For example, in Figure 35 the vertical dashed lines represent classifier segment boundaries. The first pulse is perfectly aligned on segment window boundaries thus producing correct classifications. The second pulse is not aligned on segment boundaries. This results in the signal being misclassified at both boundaries (shown as the spikes up to a third class type). The classification results are not always incorrect near signal transitions, but they can be unpredictable. One strategy for minimizing the time that the classifier's decisions are erroneous is to use the smallest segment size consistent with reasonably high levels of accuracy for constant signals.

6.3 Results of the Second Field Trial

The second field trial took place at the TELUS Toll Building in downtown Edmonton, and lasted approximately three weeks. For this field trial, the classifier was trained using a mixture of data acquired during the first field trial and (where required) data collected by Sewall. The LDFs were trained to handle all 12 classes, including the three additional signal classes; FSK, ringback, and DTMF tones (really 12 separate classes). The QDFs were only trained to handle the classes with which the LDFs had problems, namely, classes 1, 3, 4, 5, 8, and 12. Training for the QDFs for all classes would have been ideal; however, this was not an option due to real-time limitations in the DSP.

The classifier performed quite well over the three weeks. Attempts were made to monitor T1s that were heavily loaded with various different mixes of signal types. Long distance T1s to Calgary, Vancouver, and Toronto were monitored as well as T1s originating from local neighbourhoods in Edmonton and nearby surrounding areas. Each T1 was monitored for at least 24 hours; some were monitored over an entire weekend. The classification results for the LDFs and QDFs of all 24 channels of the T1 were stored into the classifier's database. All parameters (e.g. power threshold, QDFs and LDFs) for each T1 remained the same except for the segment size. Appendix D contains a complete listing of all T1s monitored along with the appropriate parameters.

6.3.1 Busy Hour and Pie Chart Graphs

Busy hour and pie chart graphs can easily be generated by the classifier from the stored data gathered during the field trials. A few selected graphs are presented in this section (each graph shown was generated using the "Print Screen" key while in an MS-DOS window under Windows 3.11). The data displayed in this section represent the results of the LDFs only since the QDF results apply for only a subset of all the signal classes. Please note that the vertical scale for each busy hour graph is different depending on the load for that particular T1 on that day.

Figures 36 and 37 show the traffic patterns for a T1 selected at random for a Saturday. Classes 3 and 5 (both data modem classes) account for 64% of the traffic on this T1. Class 3 (V.34) is, at present, widely used by dial-up users to connect to ISPs to access the Internet. Class 5 could either be data calls (V.32, V.32*bis*) or fast facsimile (V.17). At present the faster facsimile standard is not commonly used so the traffic that our classifier observed was probably exclusively data modem. Class 4 (common 9600 bps fax) accounted for only 3% of the traffic and was restricted to between 9 in the morning to midnight. Speech (Class 8) only accounted for 24% of the traffic. This number can be misleading as it does not account for the silent intervals commonly scattered within speech calls. To our surprise, a noticeable amount of class 1 (V.22 F) was present, mainly between the hours of 11 in the morning to as late as 10 in the evening.

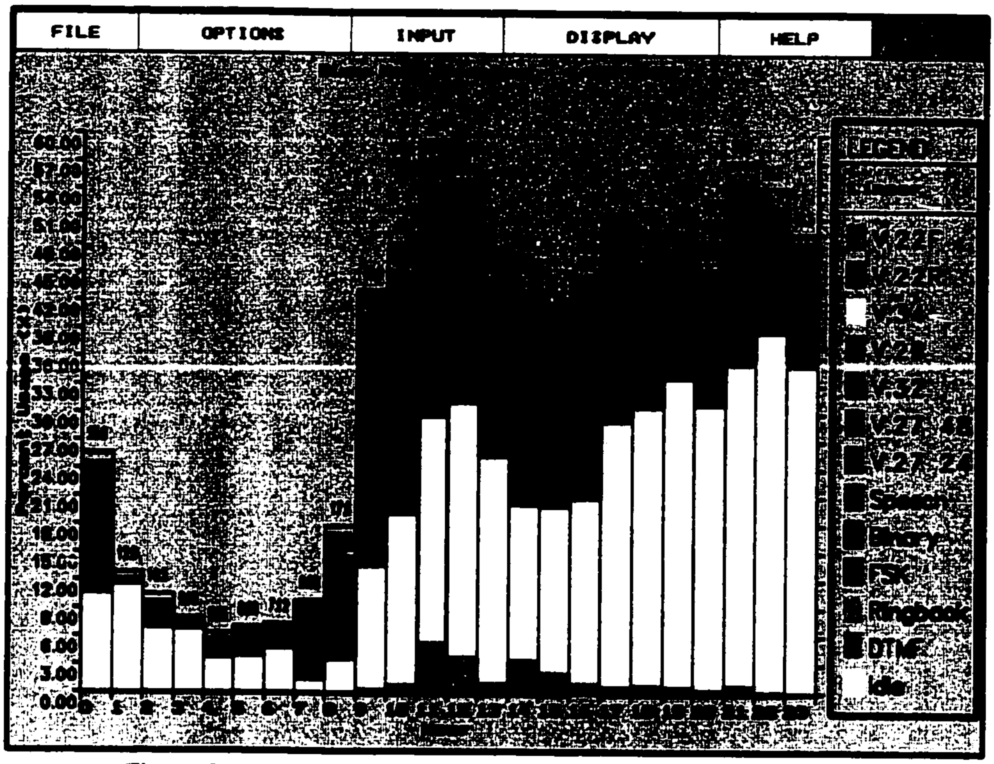


Figure 36: Busy hour graph for a T1 on Saturday April 25, 1998.

This is surprising as the maximum bit rate supported by the older V.22*bis* standard is only 2400 bps. Why would anybody be using this old standard when new standards such as V.34 support bit rates of up to 33600 bps? When these calls were seen on the real-time display during the field trial, they lasted for only a very short time. One possible explanation for these calls is that they are credit/debit card calls made from point-of-sale terminals which continue to use this old standard (possibly to achieve greater data transfer reliability and/or to minimize the delay due to the training sequence). Two classes that appeared more often than we expected were classes 11 (ringback) and 12 (DTMF tones). Ringback of course is present at the start of most calls, but to account for 4% of the total traffic may seem surprising. However, if one assumes that voice calls last on average 3.0 minutes and that ringing lasts on average 6 seconds, the 4% figure looks reasonable. Standard North American ringback is on for 2 seconds and off for 4 seconds, after which the cycle is repeated [30]. In fact, during the actual field trial, misclassifications into ringback were occasionally observed for voice calls. The amount of DTMF tone traffic (1%) present looks a little high. However, DTMF tones

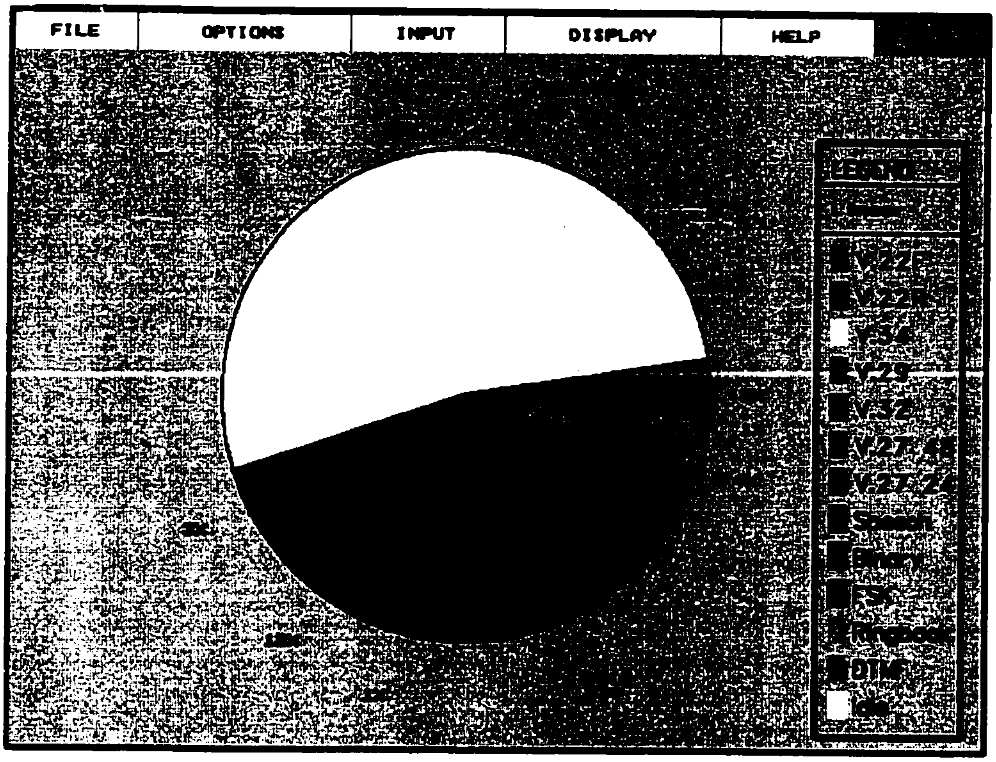


Figure 37: Pie chart graph for a T1 on Saturday April 25, 1998.

will appear for so-called "digital subscriber loops", where dialed digits are sent back to a remote CO. Usually, DTMF tones entered by users to route their calls are intercepted and translated into messages carried by the SS7 signalling network. Tones entered after a call has been set up will, however, still appear on the PSTN. Such tones are used today in several situations: e.g. controlling voice mail and answering machines, telephone banking, and automated call handling systems.

Figure 38 shows the busy hour graph for Mothers Day 1998 (Sunday, March 10). The T1 monitored was a long distance trunk between Edmonton and Calgary. As expected, most of the traffic was speech along with some V.29 facsimile and V.34 modem traffic. The busy hour for this particular day was between 10 and 11 in the evening (most probably son's who just remembered to call their mother's).

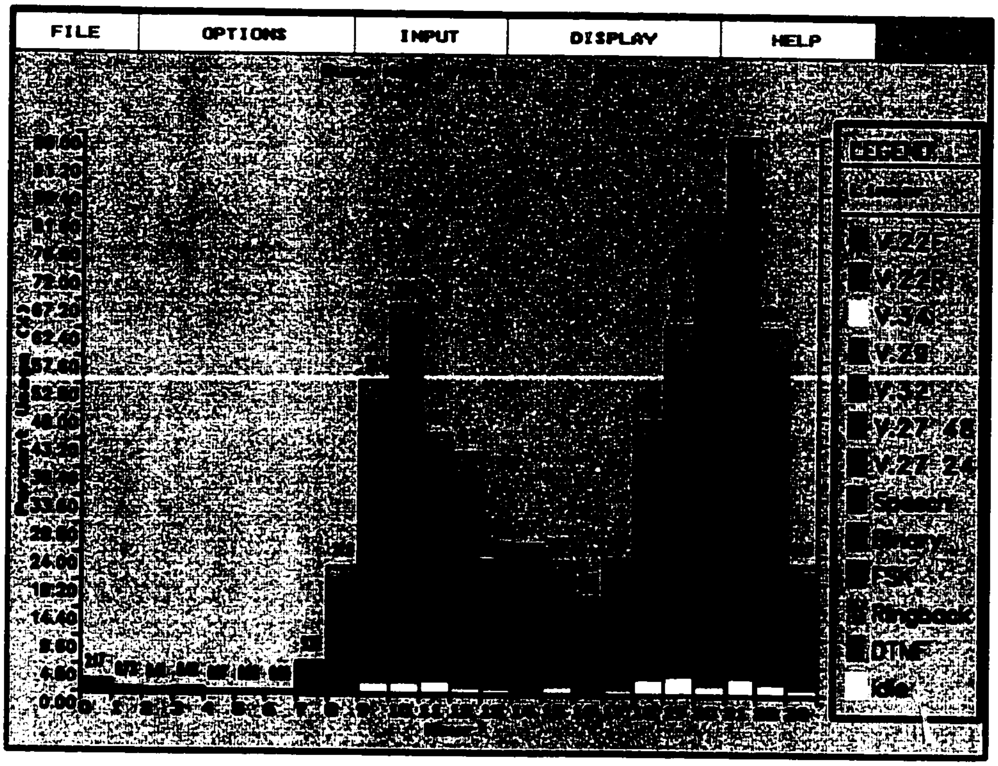


Figure 38: Busy hour graph for a long distance T1 on Mother's Day 1998.

6.4 Results of the Third and Fourth Field Trials

The third and fourth field trials took place at the Bonnie Doon TELUS Mobility base station in Edmonton. This base station carries analogue cellular traffic from the local cell area, and is also the central hub site for all base stations in the TELUS Mobility's northern Alberta network. The third field trial lasted for one week during which the classifier monitored two T1s. One T1 was for Quigley, Alberta and the second T1 was for Worsley, Alberta. Unfortunately, likely due to a power bump, the classifier crashed and data from the Worsley T1 was lost. The fourth field trial lasted for a total of ten days during which the classifier monitored one T1 for Peace River, Alberta. The classifier ran for a total of 10 days uninterrupted, resulting in a 900 MB database file. Some of the results are discussed in the next section.

6.4.1 Busy Hour and Pie Chart Graphs

Figure 39 shows a busy hour graph for a T1 to Quigley, Alberta which is a small town in northern Alberta. Many of the base stations installed in northern Alberta serv-

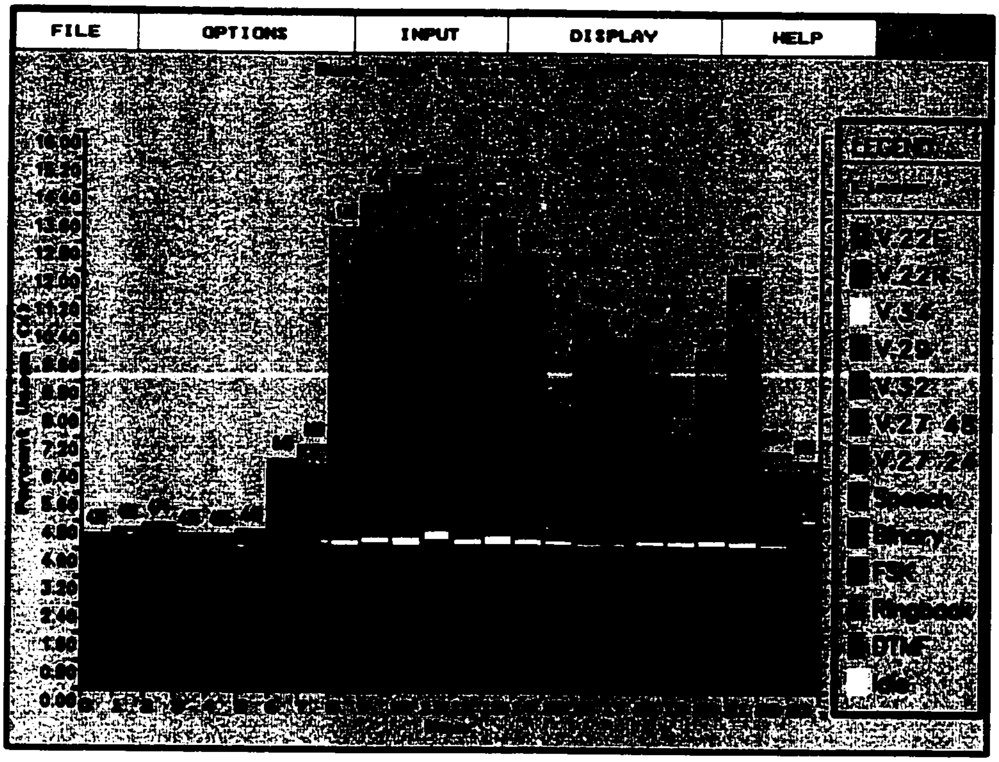


Figure 39: Busy hour graph for a T1 to Quigley, Alberta.

ice demand generated by the oil and gas industries. One interesting feature of this graph is the steady amount of class 9 (Random Binary) traffic that is present throughout the entire day. This was actually due to the in-band control information carried on one DS-0 by this particular T1. In land line connections, the control information is usually carried by a separate SS7 signalling network (packet switched). This line, however, was leased by TELUS Mobility from TELUS, and therefore carried its own signalling channel. Perhaps this T1 carried an ISDN (Integrated Services Digital Network) 23B+D configuration [9]. There is a noticeable amount of V.27*ter* facsimile calls. This traffic may be accurately classified because, in many cases, the connecting modems in a wireless network might be expected to have problems establishing a connection at the faster bit rates. Actually, we observed a call where a modem attempted to connect using the faster V.29 standard. After two unsuccessful attempts, presumably at 9600 at 7200 bps, the bit rate fell to 4800 bps. Experiments conducted at TR*Labs*, with a wireless modem, allowed for faxes to be sent at bit rates of up to 14400 bps. As the speed increased problems, such as missing lines and dropping connections, were increasingly observed. Another bit of interesting information extracted from Figure 39 is the amount of V.22F traffic at 11 in the evening. Upon further investigation, it was determined that this traffic was due to a single call that continued until about 1:30 in the morning on the next day. Perhaps this call was an older data logger sending back a batch of production data for an oil or gas field. Alternatively, the low bit rate could have been due to a transmission for a location near the outer limits of a wireless cell.

Figure 40 shows a busy hour graph for a Peace River T1 monitored during the fourth field trial.

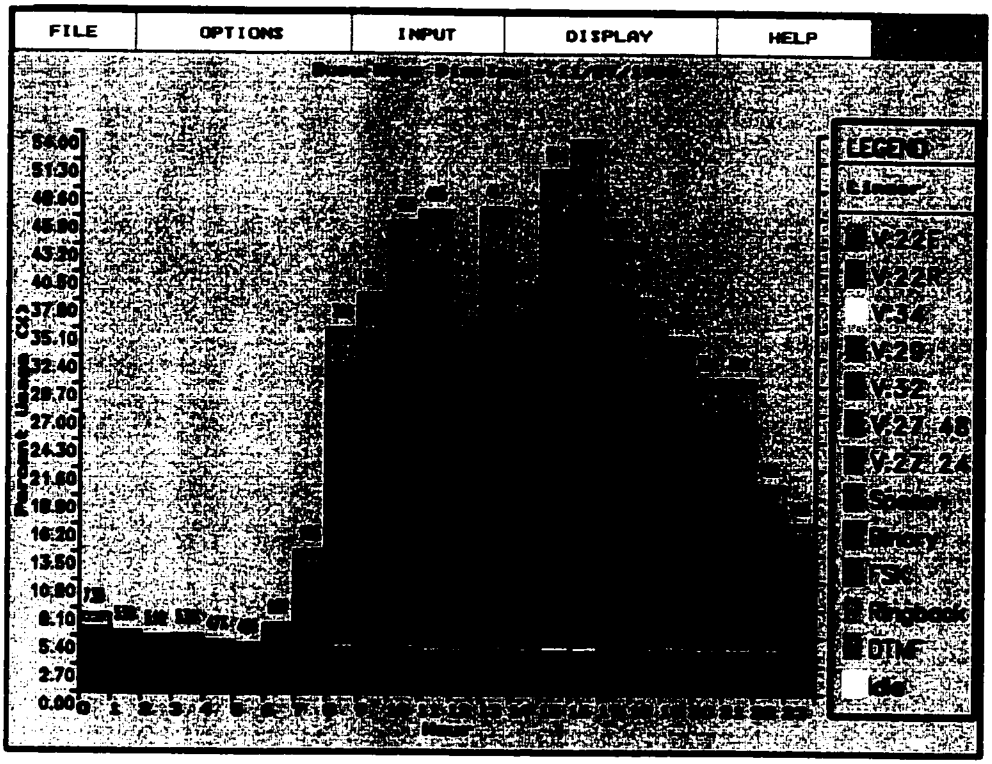


Figure 40: Busy hour graph for a T1 to Peace River, Alberta.

This T1, again, has a control channel in the first DS-0. This caused the steady black stripe at the bottom of the plot corresponding to 1 Erlang of traffic. Aside from that, the majority of traffic is simply speech. This line was monitored for ten straight days, with the other plots looking similar to the one above.

6.5 Classifier Improvements

Although the classifier performed quite well, further improvements were possible after the experience of the field trials. These improvements include the use of filters to help "ride out" possible misclassifications during a call. These misclassifications include silent intervals within voice calls or simply misclassifications due to signal transitions. Also, thus far the results of the QDFs could not be used because the QDFs were only trained on a subset of the total number of classes. If the signal did not belong to one of these expected classes, then the QDF results are meaningless. We therefore investigated ways of combining the LDF decisions with the QDF decisions to get a hybrid method that would have the strengths of both methods.

6.5.1 Improvements From Using Filters

By using simple filtering methods, many of the potential misclassifications can be removed from all types of calls. For example, Figure 41 shows an example of what actual speech and V.29 facsimile calls looked like without filtering. The misclassifica-

(a) Unfiltered Speech.

(b) Unfiltered V.29 Fax.

Figure 41: Voice unfiltered and filtered (1020 segment size).

tions in the speech call are partly because of the silent intervals, and partly because of mixtures of the silence with speech in the same segment. The misclassifications in the facsimile call are mainly confusions between classes 4 and 5, which were known to difficult classes to separate. Note that the dark blue (FSK signalling), is expected in facsimile calls.

The type of filter to use strongly depends on the characteristics of the traffic that the classifier is attempting to classify. Fortunately, many of the calls placed on the PSTN have a predictable call structure. For example, data calls have a negotiation phase, followed by the data using an unchanging modulation method. Facsimile calls also have a negotiation phase, but the data transmission is intermixed with FSK signalling. Finally, voice calls have a mixture of speech with silent intervals. It is also known that once a call has ended, a finite amount of time passes before another call begins on the same DS-0. The start of new calls can, in most cases, be determined by the presence of ringback. It is important that any filter not affect the call boundaries by merging together subsequent calls. Another important concern in filter design is that real transitions in signal class not be moved around in time. Instead of attempting to exploit high-level knowledge of call structure, we decided to evaluate a simple and relatively low-level majority filter technique.

Such a filter simply looks at a user specified window size and, for each channel, counts the number of classification decisions (before filtering) going into each signal class.

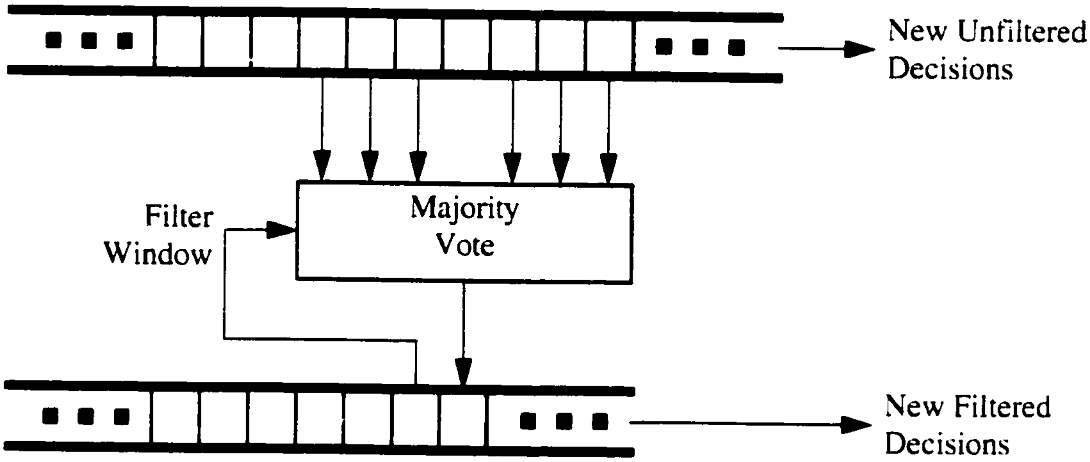
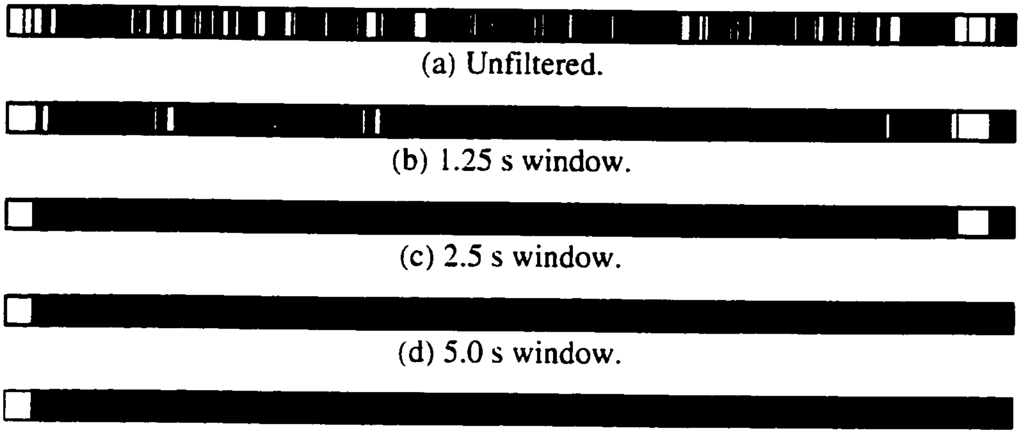


Figure 42: Majority rules filter.

If there is no clear majority class, then the filter backs off and passes the previous filtered decision. One advantage of this filter is that signal boundaries are still maintained (within one sample) between adjacent long stretches of different signals. The filter's simplicity ensures that it requires the minimum of computation.

An example of this filter on an actual voice call is shown in Figure 43.



(a) Unfiltered.

(b) 1.25 s window.

(c) 2.5 s window.

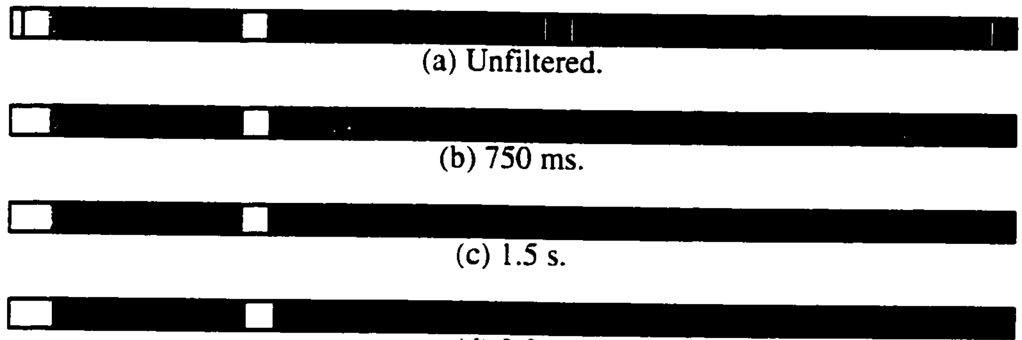
(d) 5.0 s window.

(e) 10.0 s window.

Figure 43: Voice unfiltered and filtered (N=1020).

The unfiltered signal has misclassifications into various classes including, notably, classes 3, 4, and 5. The speech call also contains brief intervals of silence. Figures 43 (b), (c), (d), and (e) show the results of filtering the data using four progressively larger window sizes. By using a 1.25 second filter window the vast majority of the misclassifications are removed. There are still, however, a couple of misclassifications into class 3, and also a few of the silent intervals were not bridged. After increasing the filter window to 2.5 seconds, all of the misclassifications have been cleaned up; however, one large silent interval remains. Using a filter setting of 5.0 seconds removes all of the silent intervals.

In Figure 43 (a) filter setting greater than 5.0 seconds does not appear to result in any improvements. This is usually the case because the silent intervals that we observed were not long (less than 1.25 seconds). In other calls observed at the field trials, silent intervals in the 20 - 30 second range were observed. Such a situation might arise if a caller is placed on hold. Perhaps a larger filter window would help bridge the larger silent intervals. This would, unfortunately, increase the danger of bridging separate calls. During the field trials, a busy T1 clearly had subsequent calls with little intervening silence. The smallest observed time between calls was approximately 5 seconds. Thus, using a filter window of greater than 10 seconds risks bridging adjacent calls on a busy T1.



(a) Unfiltered.

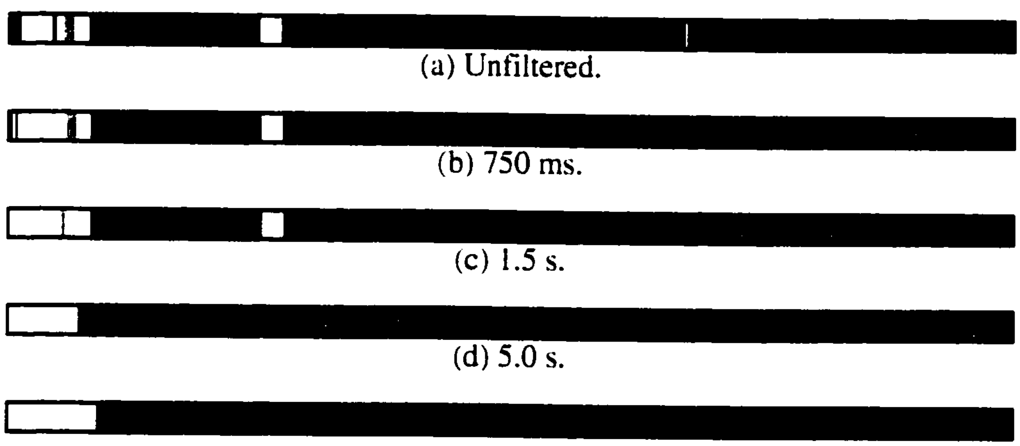
(b) 750 ms.

(c) 1.5 s.

(d) 3.0 s.

Figure 44: V.29 facsimile call, unfiltered and filtered.

Figure 44(a) shows an actual V.29 fax call. The dark blue is the FSK signalling present in all fax calls. The unfiltered bar shows many misclassifications into class 5, which is not unexpected as classes 4 (V.29) and 5 are two easily confused classes. By applying a very short 750 ms filter, the vast majority of the misclassifications are filtered away. All misclassifications are filtered away by a 3.0 second filter.



(a) Unfiltered.

(b) 750 ms.

(c) 1.5 s.

(d) 5.0 s.

(e) 7.5 s.

Figure 45: V.29 facsimile call, unfiltered and filtered.

Figure 45 shows a very clean V.29 fax call. In this particular case filtering does not actually improve the overall accuracy; however, using a filter setting below 1.5 seconds does not appear to do any harm. Using a filter setting of 5.0 seconds reduces the duration of the FSK signalling. Finally, at a filter setting of 7.5 seconds, the FSK signalling is completely filtered away. By using an even larger filter window on facsimile calls, the FSK signalling gets filtered away. It would be undesirable to lose the FSK signalling since, apart from affecting class statistics, it would also corrupt interesting fax page count statistics that might otherwise be collected by the classifier.

From these experiments it appears that speech calls and non-speech calls might best have different filter settings. For speech a larger filter window is desired to filter away as many silent intervals as possible. However, using an overly long filter window on non-speech calls, actual signals are lost. Perhaps an adaptive, multiple-window filter is required here. For example, if the present call has a majority of speech in the filter window, then the filter can be made to change the window size to the speech window filter setting for the next filter output. If the filter determines that the majority is non-speech, then it could be made to change back to the non-speech window filter setting. This two-window setting approach is implemented in the final version of the classifier.

The maximum filter window that can be used without filtering out actual signal transitions depends on the signal that is present for the shortest period of time. FSK signalling and ringback are clearly not present in an actual call for a long period of time compared with, say, facsimile or modem calls. DTMF tones are only actually present for a fraction of a second, possibly only 50 ms for automatic dialers. Manually activated DTMF signals will of course be several times longer. Even if a small 1.5 second filter window is selected, the DTMF tone would have to be present for at least 750 *ms* or else the filter would remove it. Another method would be to disable filtering when DTMF tones can reasonably be expected. The problem with this method is that the classifier would have to be very certain that any DTMF detected were in fact not misclassifications. Unfortunately, class 1 (V.22F), and class 8 (speech) are two classes that have been seen to be sometimes misclassified as DTMF tones.

The problem of recognizing DTMF digits in the presence of other signals was not completely solved in this project. Certainly adaptive filter techniques should be investigated further. As a last resort, one could consider the use of the widely available and inexpensive DTMF digit-collecting integrated circuits.

6.5.1.1 Comparisons of Filtered and Unfiltered Data

Figure 46 shows a pie chart graph for a T1 which terminates in Quigley,

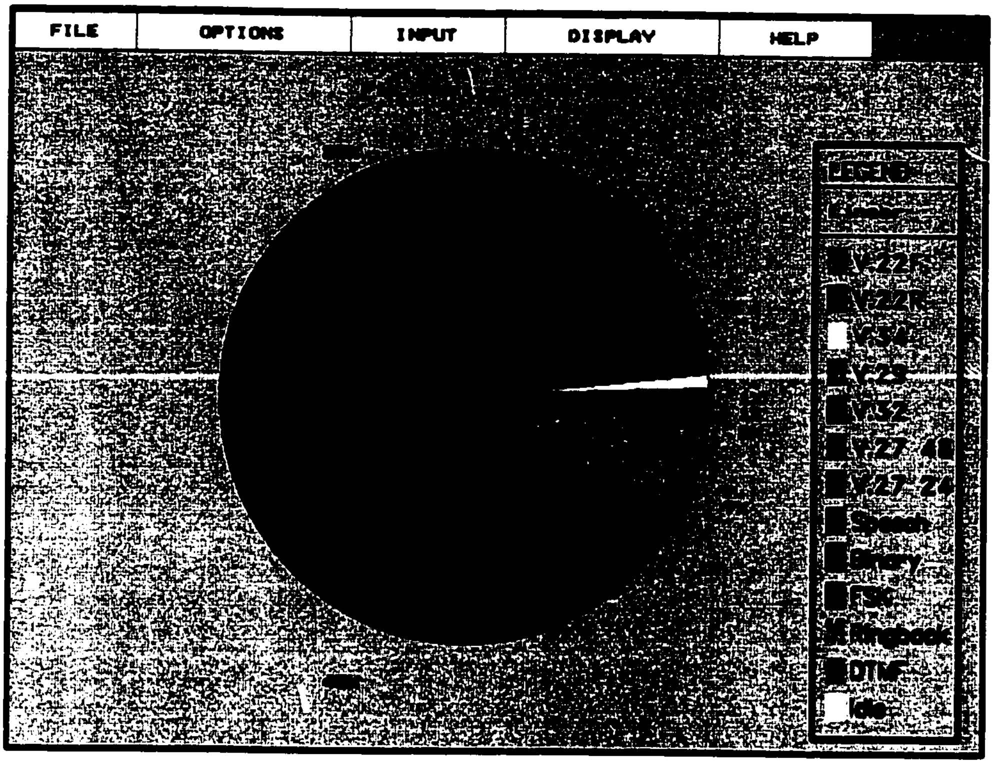


Figure 46: Pie chart for a T1 for Quigley, Alberta before filtering.

Alberta. The results show the breakdown of the un-filtered data. Figure 47 shows the filtered results from the same T1. The filter settings used for this T1 was 5.0 seconds for speech, and 1.0 seconds for non-speech. With filtering the data classes 12 (DTMF tones), 3 (V.34), and 7 (V.27*ter* 24) have disappeared from the pie chart. Also, the amount of ringback has been reduced from 7% to 5%, and the amount of V.29 has been reduced from 9% to 8%. The classes that have increased are speech (25% to 28%), and binary (45% to 49%). The other classes have been essentially unaffected.

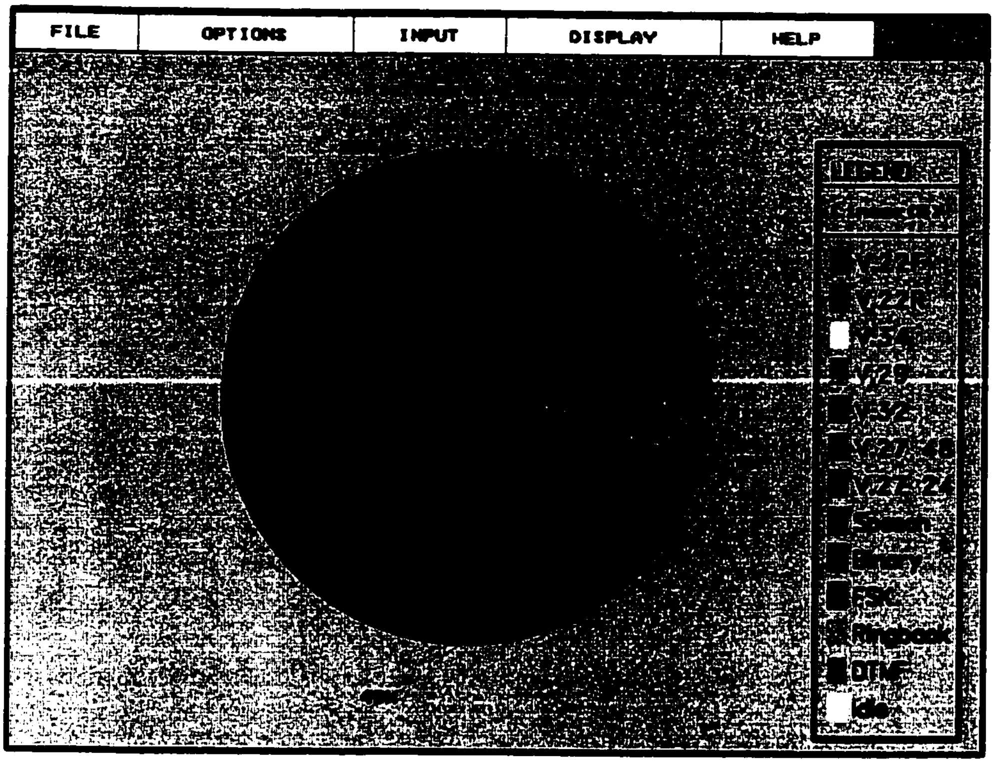


Figure 47: Pie chart for a T1 for Quigley, Alberta after filtering.

6.5.2 Hybrid Discriminant Function

Thus far, all of the busy hour and pie chart graphs have shown the results of the results using LDFs. Over all classes the LDFs are reasonably accurate, but simulations have shown that the QDFs are even more accurate. This is especially true when LDFs attempt to classify a signal that belongs to one of the problem classes (e.g. classes 4 and 5). One solution is to train the QDFs over all classes and then simply to use only those classification results, completely ignoring the LDF results. Unfortunately, the real-time performance limits of our 40 MHz DSP restricted the number of classes the QDFs could be trained on. One compromise would be to only train the QDFs on classes that pose problems to LDFs. Perhaps what is needed is a hybrid method that combines, in some effective way, the decisions of the LDFs and QDFs. But this creates a problem: when should the classifier only use the LDF results, and when should it be allowed to use the QDFs? Since the QDFs are not trained on all signal classes, the results of the LDFs must be considered first because the signal could belong to a class that the QDFs do not recognize. If the signal can be placed by the LDF with high confidence into a set of classes handled by the QDFs, then using the quadratic result is desired. One method to determine the set of class candidates is to look at the actual numerical values generated by the LDFs. For example, currently the classifier evaluates the LDFs and assigns the signal to the class with the largest value (assuming prior probabilities are equal). If instead the classifier looked at the top two or three values, and if these classes were all recognized by the QDF, then the QDF result could be used to make the final classification instead of the linear result (see Figure 48). Of course, this method assumes that the

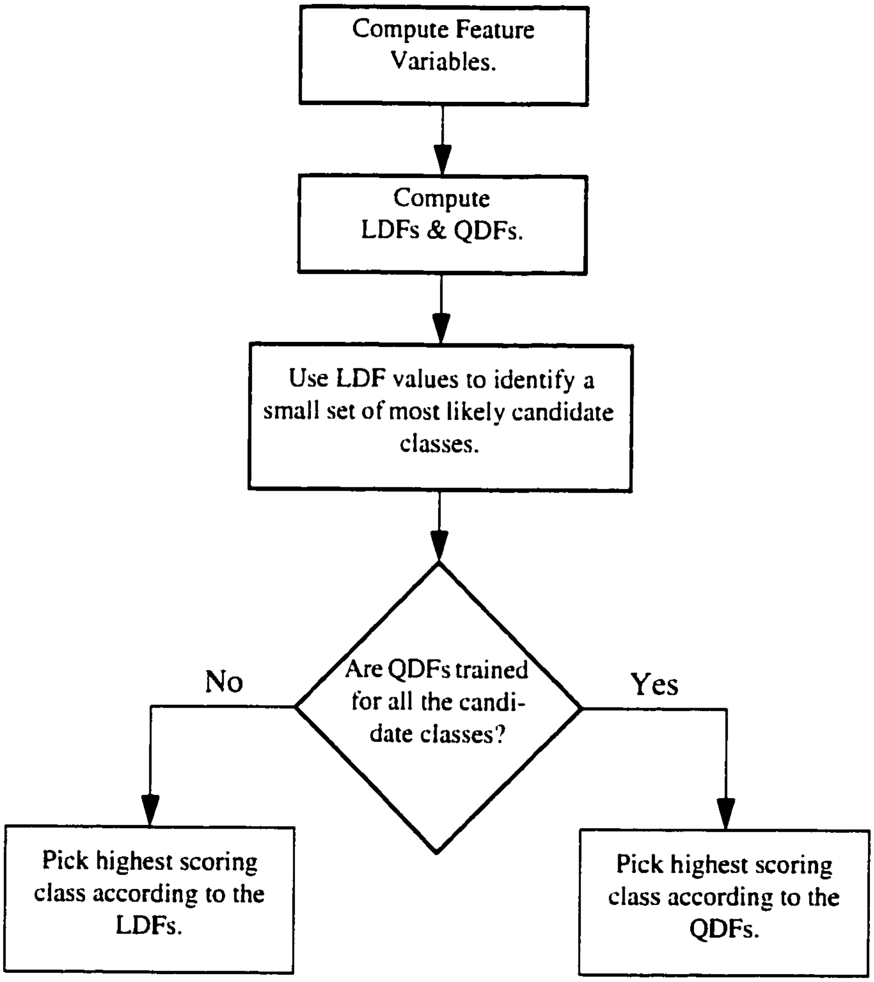


Figure 48: Hybrid decision rule.

LDF will be reliable enough to include the ultimately selected class in a set of candidate classes. To verify this, the top two signal classes of the LDF were recorded and are summarized in Tables 39 and 40 for the problem classes. The ">12" class refers to the signal classes from 13 to 23 inclusive.

Table 39: First choice selection (%) using LDFs (N=2052).

Class	1	2	3	4	5	6	7	8	9	10	11	12	>12
1	89.44	-	-	-	-	-	-	-	-	-	-	10.56	-
2	-	100.00	-	-	-	-	-	-	-	-	-	-	-
3	-	-	99.90	-	0.04	-	0.03	0.01	-	-	0.02	-	-
4	-	-	-	85.63	14.37	-	-	-	-	-	-	-	-
5	-	-	-	11.56	88.40	0.04	-	-	-	-	-	-	-
6	-	-	-	-	-	98.90	1.10	-	-	-	-	-	-
7	-	-	-	-	-	1.20	98.79	-	-	-	-	-	0.01
8	-	0.25	1.97	1.35	-	-	-	91.63	0.49	-	1.72	-	2.59
9	-	-	-	-	-	-	-	-	100.00	-	-	-	-
10	-	-	-	-	-	-	-	-	-	100.00	-	-	-
11	-	-	-	-	-	-	-	-	-	-	100.00	-	-
12	-	-	-	-	-	-	-	-	-	-	-	100.00	-
>12	-	-	-	-	-	-	-	-	-	-	-	-	100.0

Class 1 has a classification accuracy of 89.44%, with 10.56% being classified as class 12 accounting for 100% of class 1. The second choice for class 1 has classes 1 and 12, accounting for 99.1% of the classifications. This suggests that if the actual signal belongs to class 1, then most of the time the top two selections of the LDFS will be 1 and 12.

Table 40: Second choice selection (%) using LDFs (N=2052).

Class	1	2	3	4	5	6	7	8	9	10	11	12	>12
1	10.49	-	-	-	-	-	-	-	-	-	-	88.62	0.89
2	-	-	-	57.86	-	-	-	-	-	-	-	2.15	39.99
3	-	-	-	44.48	55.45	0.02	-	0.02	-	0.02	0.01	-	-
4	-	-	-	14.37	85.63	-	-	-	-	-	-	-	-
5	-	-	-	88.40	11.58	0.02	-	-	-	-	-	-	-
6	-	-	-	1.53	8.35	1.10	89.02	-	-	-	-	-	-
7	-	-	-	0.10	-	98.79	1.10	-	-	-	-	0.01	-
8	-	-	53.07	1.97	0.62	-	-	3.08	0.49	-	33.99	-	6.78
9	-	-	1.03	-	63.64	-	-	-	-	-	-	-	35.33

Table 40: Second choice selection (%) using LDFs (N=2052).

Class	1	2	3	4	5	6	7	8	9	10	11	12	>12
10	-	-	-	-	-	-	100.00	-	-	-	-	-	-
11	-	-	-	-	-	-	-	100.00	-	-	-	-	-
12	100.00	-	-	-	-	-	-	-	-	-	-	-	-

This argument also holds true for easily confused classes 4 and 5. In our experiment classes 4 and 5 account for 100% of the first and second choices. Class 8, however, is misclassified into many different classes with no one clear majority. Also, class 8 (speech) is selected as a second choice only 3.08% of the time. This would suggest that using the hybrid method will not improve the accuracy for speech calls unless the QDFs are trained for virtually all of the classes. Even after doing this, the accuracy for a hybrid method that considers only the top two ranked classes cannot increase by more than 3.08%.

Using the proposed hybrid method with two LDF classes for signals of classes 1, 4, 5, and 12, the accuracy improved significantly as shown in Table 41. The classification accuracy of class 1 jumped nearly 10% to 99.93, and the accuracy for classes 4 and 5 have improved to 98.80% and 99.95%, respectively. The classification accuracy of the remaining classes remains unchanged at high levels. Over all classes, the average accuracy has improved to 98.99% from 96.06%.

Table 41: Percent classification accuracy using the hybrid method (N=2052, Std V.34, Incl. EN).

Class	1	2	3	4	5	6	7	8	9	10	11	12	>12
1	99.93	-	-	-	-	-	-	-	-	-	-	0.07	-
2	-	100.00	-	-	-	-	-	-	-	-	-	-	-
3	-	-	99.90	-	0.04	-	0.03	0.01	-	-	0.02	-	-
4	-	-	-	98.80	1.20	-	-	-	-	-	-	-	-
5	-	-	-	0.02	99.94	0.04	-	-	-	-	-	-	-
6	-	-	-	-	-	98.90	1.10	-	-	-	-	-	-
7	-	-	-	-	-	1.20	98.79	-	-	-	-	-	0.01
8	-	0.25	1.97	1.23	0.12	-	-	91.63	0.49	-	1.72	-	2.59
9	-	-	-	-	-	-	-	-	100.00	-	-	-	-
10	-	-	-	-	-	-	-	-	-	100.00	-	-	-

Table 41: Percent classification accuracy using the hybrid method (N=2052, Std V.34, Incl. EN).

Class	1	2	3	4	5	6	7	8	9	10	11	12	>12
11	-	-	-	-	-	-	-	-	-	-	100.00	-	-
12	-	-	-	-	-	-	-	-	-	-	-	100.00	-
>12	-	-	-	-	-	-	-	-	-	-	-	-	100.00

The above results were generated using all variables to train the QDFs. Using the six optimum variables (as determined from a previous section) reduces the number of computations without reducing accuracy. These results are summarized below in Table 42. Clearly the accuracy has remained the same.

Table 42: Percent classification accuracy using the hybrid method and variables Rd1, 2, 3, 5, 6, and 7 (N=2052, Std V.34, Incl. EN).

Class	1	2	3	4	5	6	7	8	9	10	11	12	>12
1	99.93	-	-	-	-	-	-	-	-	-	-	0.07	-
2	-	100.00	-	-	-	-	-	-	-	-	-	-	-
3	-	-	99.90	0.04	-	-	0.03	0.01	-	-	0.02	-	-
4	-	-	-	98.59	1.41	-	-	-	-	-	-	-	-
5	-	-	-	0.16	99.80	0.04	-	-	-	-	-	-	-
6	-	-	-	-	-	98.90	1.10	-	-	-	-	-	-
7	-	-	-	-	-	1.20	98.80	-	-	-	-	-	-
8	-	0.25	1.97	1.23	0.12	-	-	91.63	0.49	-	1.72	-	2.59
9	-	-	-	-	-	-	-	-	100.00	-	-	-	-
10	-	-	-	-	-	-	-	-	-	100.00	-	-	-
11	-	-	-	-	-	-	-	-	-	-	100.00	-	-
12	-	-	-	-	-	-	-	-	-	-	-	100.00	-
>12	-	-	-	-	-	-	-	-	-	-	-	-	100.00

6.5.3 Speech and Voiceband Data Classifier

The hybrid classifier described in the previous section improved the classification accuracy of classes 1, 4, 5, and 12; however, the accuracy of speech is still relatively low (91.63%). Benvenuto found that by using only two feature variables, namely the autocorrelation at lag 2 and the central second-order moment, speech and voiceband data could be accurately distinguished from each other. Using these same feature variables, an experiment was conducted using only two classes, speech and non-speech. The non-speech class was generated by grouping together all of the non-speech classes (1 to 7 and 9 to 23). The resulting classification accuracies are given in Tables 43 and 44. This two-stage method was actually introduced by Sewall in [2].

Table 43: Percent classification accuracy using only two classes (N=2052, LDF, Std V.34, Incl. EN).

Class	Non-Speech	Speech
Non-Speech	99.88	0.12
Speech	5.42	94.58

Table 44: Percent classification accuracy using only two classes (N=2052, QDF, Std V.34, Incl. EN).

Class	Non-Speech	Speech
Non-Speech	98.51	1.49
Speech	0.25	99.75

Using LDFs the classification accuracy of speech has improved from 91.63% to 94.58%, and by using QDFs the classification accuracy of speech has improved to 99.75%. In both cases, the accuracy for recognizing non-speech remains high at 99.88% for LDFs, and 98.51% for QDFs. At first this seemed like a favourable result, however, the majority of misclassifications of non-speech into speech were from classes 9 (random binary) and 11 (ringback). This reduced the classification accuracy for these two classes substantially. To try and improve the accuracy for these two classes, an experiment was done using 4 classes: non-speech (classes 1 to 7, 10, and 12 to 23), speech, random binary, and ringback. SPSS was then used to determine which feature variables are best at separating these four classes. Using only four variables (Rd1, 3, 5, and N2), the classification accuracy for classes 9 and 11 were increased to 100.0% and 97.53%, respectively as shown in Table 45. This result is very useful as the accuracy for speech has also increased to 99.26% without affecting the classification accuracy of other classes significantly.

Table 45: Percent classification accuracy using only four classes (N=2052, Std V.34, Incl. EN).

Class	Non-Speech (Classes 1-7, 10, & 12-23)	Speech	Random Binary	Ringback
Non-Speech	99.99	0.01	-	-
Speech	0.74	99.26	-	-
Random Binary	-	-	100.0	-
Ringback	-	2.47	-	97.53

This suggests that perhaps a two stage classifier is required. If this classifier returns a non-speech result then the data could be further processed using the previously discussed hybrid classifier. If speech, random binary, or ringback is detected at the first stage, then no further processing is required. The classification accuracies, using this approach is summarized in Table 42.

Table 46: Percent classification accuracy using a two-stage classifier (N=2052, QDF, Std V.34, Incl. EN).

Class	1	2	3	4	5	6	7	8	9	10	11	12	>12
1	99.93	-	-	-	-	-	-	-	-	-	-	0.07	-
2	-	100.00	-	-	-	-	-	-	-	-	-	-	-
3	-	-	99.93	0.04	-	-	0.03	-	-	-	-	-	-
4	-	-	-	98.59	1.41	-	-	-	-	-	-	-	-
5	-	-	-	0.16	99.80	0.04	-	-	-	-	-	-	-
6	-	-	-	-	-	98.96	1.04	-	-	-	-	-	-
7	-	-	-	-	-	0.99	99.0	-	-	-	-	-	0.01
8	-	-	0.74	-	-	-	-	99.26	-	-	-	-	-
9	-	-	-	-	-	-	-	-	100.00	-	-	-	-
10	-	-	-	-	-	-	-	-	-	100.00	-	-	-
11	-	-	-	-	-	-	-	2.47	-	-	97.53	-	-
12	-	-	-	-	-	-	-	-	-	-	-	100.00	-
>12	-	-	-	-	-	-	-	-	-	-	-	-	100.00

Using this method, the second stage of the LDFs would only need to classify between 20 classes (down from 23), and the QDFs would only need to classify 4 classes (down from 6).

Chapter 7

7.0 Recommendations

The intent of this chapter is to provide design recommendations for future voiceband signal classifiers based upon the results of the previous chapters. These will include hardware and software recommendations along with the associated settings to be used by the classification algorithm.

7.1 Computer Hardware, OS, and GUI

The prototype Voiceband Signal Classifier, designed and built for this thesis, was a generic PC with a 486 DX4 100 MHz CPU, a 40 MIPS floating-point DSP card, and a T1 interface card. The PC was used to run the GUI software, and the DSP was used to compute the classification vectors. The DSP was necessary because the host CPU was not fast enough to compute the classification vectors as well as running the GUI software. The OS chosen for the classifier was MS-DOS 6.22. Although this is not a real-time OS, it proved to be satisfactory for this application. The prototype proved to be very stable over all four field trials.

During the project many faster CPUs became available. At the time of writing, Pentium II 450 Mhz CPUs were available in multiple processor motherboards. By using such fast processors, the need for a separate DSP card becomes questionable as the CPU is fast enough to run the GUI as well as compute the classification vectors. This is an advantage as the DSP card is relatively expensive, and it adds complexity to the system.

In fact, a DSP-less Windows NT 4.0 system is already under development by Mr. Xiangqian Xu, a TR*Labs* M.Eng. student, using a Pentium II 266 MHz CPU, a single processor motherboard, and a similar T1 interface card. This system is able to classify, in real-time, all 24 channels of a T1 with many CPU cycles to spare. Xu estimates that the current Windows NT system can handle up to four T1 lines if the real-time display is not active (assuming no other applications are running). By using Windows NT, a GUI was developed which will be much easier to extend than the MS-DOS GUI.

Also, using Windows NT will permit the classifier application to interact with other Windows-based applications, which should make it easier to extend the classifier with new features. One example of this is using standard text editors to view the logfile, or importing the classification vectors directly into standard spreadsheet programs for processing the data and for plotting graphs. After further testing and field trials it is anticipated that the Windows NT system will match the reliability of the MS-DOS prototype and will supersede it as the research development platform.

The two systems described above have only been designed to handle one T1 line. Typically a CO or base station has multiple T1 lines. For example, the Bonnie Doon TELUS Mobility base station terminated over 500 T1 lines, and the TELUS Toll Building had even more. In this situation it would be desirable to monitor more than just one T1 line at a time. One solution would be to scan through multiple T1s using an external multiplexer with the output going to the existing prototype. Another solution would be to use a multiple processor DSP card to perform the classification calculation in parallel for many T1s, perhaps as many as 28 T1s (that is enough for one 44.736 Mbps T3 trunk). A T1 card which switches under software control among up to 8 separate T1 inputs is currently sold by GL Communications Inc. (Multi T1).

7.2 Training Data and Signal Classes

The data used to train and test the classifier included the recordings made at the first field trial, and the recordings made by Sewall in [2]. Signal samples for unusual class 3 (V.34) operating modes were not used in the training of the coefficients. The V.34 standard allows for various baud rates, bit rates, and carrier frequencies. These different modes had quite variable PSDs that caused some difficulty to the classifier. At the first field trial it was observed that when using land line connections, the modems appear to default to a symbol rate of 3529 baud and a carrier frequency of 1959 Hz. The only time that any other modes were observed was if the modem was forced by using the AT command set. This conclusion might hold true for land lines; however, it might not hold true for wireless connections. A data file recorded by VisionSmart (a TR*Labs* small business associate) using a wireless modem fooled the classifier into calling it a V.29 facsimile signal. All that was known about this signal was that it was a data connection (not a facsimile call), and that the bit rate reported from the application program was 14400 bps. Interestingly, although the LDF decisions were V.29, the QDF decisions were V.34. One possible explanation is that the signal was in fact a V.34 signal using an unusual symbol rate and carrier frequency. This is quite possible as the characteristics of wireless voice channels can be quite variable, thus using the V.34 connection would be beneficial since the parameters (carrier frequency, baud rate, and bit rate) can be negotiated to adapt to the channel's limitations. We therefore believe that the non-standard V.34 connection modes cannot be ignored when using the classifier for wireless channels. Additional classes may be required.

The results presented in this thesis were for signals that were separated into 12 classes. Using all 12 classes is recommended if the traffic mix is not already known. If the classifier is to be used in only a specific application, where only certain classes are known to be present, then the number of classes could be reduced. An example of such an application would be a classifier simply distinguishing between speech and modem data; e.g. for lines leading to an ISP. Also, customers may not be interested in DTMF tones. By removing the DTMF tones, 12 fewer classes are required and almost one half of the computations can be saved. Such optimizations, however, are probably not advisable for a research platform.

The LDFs were trained to recognize all 23 classes, while the QDFs were trained to recognize only 6 classes. This was necessary to meet the real-time requirements of the DSP card. The 6 QDF classes included classes 1, 3, 4, 5, 8, and 12. These were chosen because they were observed to cause problems for the LDFs. The QDFs should definitely be trained to at least recognize classes 1, 4, 5, and 12 because using QDFs, the accuracy for these classes can be readily raised to close to 100%. If real-time constraints limit the number of QDF-trained classes to two, then selecting classes 4 and 5 would be best as these classes are more likely to appear then classes 1 and 12. If real-time constraints allow additional classes to be handled by the QDFs, then selecting classes into which speech is frequently misclassified would be desirable. Two such classes are ringback and DTMF tones. With a sufficiently fast processor, the QDFs might be trained to recognize all classes. This would eliminate the need to use LDFs and any multistage hybrid methods.

7.3 Classification Methods

The classification methods evaluated for this thesis included LDFs, QDFs, Pseudo QDFs, and ALNs. A new hybrid method was also developed which combined the strengths of the LDFs and QDFs. The LDFs, QDFs, and the hybrid method were implemented in the prototype classifier. There does not appear to be any advantage to using pseudo QDFs instead of QDFs as, in all cases observed, the QDFs classified signals at least as accurately as the pseudo QDFs. Implementing ALNs in the prototype classifier might improve the classification accuracy as ALNs did perform better over certain classes than QDFs. Perhaps a new hybrid method could be derived from all three of the LDFs, QDFs, and ALNs?

Our hybrid method was developed originally because real-time limitations in the DSP did not allow for the QDFs to be employed in real-time for all signal classes. In a faster system, if enough real-time is available, then using the QDFs over all classes would be preferred since the QDF results are more accurate then the LDF results. This would certainly improve the classification accuracy of speech, which appears to be a priority for many potential users of voiceband signal classifiers.

7.4 Classifier Settings

7.4.1 Segment Size

Using segment sizes below 1020 samples noticeably reduced the overall accuracy of the prototype classifier. Segment sizes of 1020 samples and greater did appear to clean up many misclassifications. The disadvantage of using a very large segment size is that the actual signal transitions may not be accurately tracked. Also, using a larger segment size increases the possibility of having signal mixtures within the segment window. Using a 2052 segment size appears to be a good compromise between high accuracy and precision. This size implies approximately four classification vectors every second, which is fast enough to track the signal transitions in most signal classes. This segment size is still too large to accurately collect DTMF digits at their maximum arrival rate (12.5 digits/sec); however it is more than accurate enough to track signal transitions among the other classes.

The prototype classifier was designed to use the filter coefficients that assumed a 2052 segment size. These coefficients can be used regardless of the segment size actually selected by the user through the GUI because all of the variables are normalized with respect to size. To verify this, a simulation was conducted using 2052 segment size training data for an LDF classifier, and then using the 252, 516, 1020, and 2052 segment sizes in test sets to measure the resulting classification accuracies. The same experiment was repeated for a training segment size of 1020. The results are summarized in Table 47.

Table 47: Classification accuracy using different training sets.

Accuracy (%)	2052 Training Set				1020 Training Set			
	2052	1020	516	252	2052	1020	516	252
Measured Accuracy	96.06	94.35	91.45	86.73	95.38	93.90	91.35	87.31

The measured classification accuracy shows no significant improvement regardless of the segment size used in the training set. We conclude that only one set of coefficients (say for the 2052 segment size) needs to be stored in the classifier, regardless of the segment size selected by the user.

7.4.2 Averaging Filter Settings

We found filters to be effective at improving the accuracy of the classifier on real signals. Without filtering, many misclassifications were observed within speech calls. Fewer misclassifications were observed in data or facsimile calls, but they were still present. If average filtering is to be most effective at improving classifier accuracy, then a different filter window must be used for speech and non-speech. This is necessary to help bridge the sometimes long silent intervals unique to speech calls. For data calls, a large filter window would actually remove important signal structures, such as FSK signalling in facsimile calls and ringback. Filter settings below 3 seconds did not appear to remove these signals and should be safe for non-speech calls. Using a filter of below 1.5 seconds for non-speech is not recommended because of the relatively common problem of misclassifying classes 4 and 5. For speech calls, the upper limit on the size of the filter is determined by the duration between the end of a call and the start of a new call on the same DS-0. From the field trials this time was observed to be approximately 5 seconds, so using a maximum filter window of less than 10 seconds should not bridge the gaps between most speech calls. In many analogue wireless calls, silent intervals did not appear due to the significantly high levels of background noise. This might also be true in some land line connections, but it is difficult to distinguish land line speech calls from wireless speech calls. To be safe, a 10 second filter should be used for all speech calls.

7.4.3 Power Threshold

The power threshold used in the field trials was $P_{th} = 816$ (60 dB lower than the maximum). This value was used by Sewall and was carried forward as a default value into the prototype classifier. The power threshold setting is only applicable to voice calls because, in data calls, there is usually plenty of power in the signal. During the field trials, we adjusted the power threshold in an attempt to reduce the silent intervals. By increasing the threshold, the number of silent intervals in voice calls did in fact increase, and by decreasing the value the number of silent intervals did in fact decrease. The danger in overly reducing the power threshold is that the resulting low signal-to-noise ratio would lead to misclassifications. The danger in overly increasing the power threshold is that silent intervals become longer and, as a result, more difficult to filter out. This was noted at the field trials by simply adjusting the default value of the power threshold, and observing how it effected the classification results of voice calls by using the real-time display. Since no significant improvements were observed, at the field trials, when increasing or decreasing the default value of the power threshold, we recommend that the default value of 816 be used.

7.4.4 Feature Variables

The LDFs and QDFs were trained using all 11 of the variables; however, we showed later that using all 11 variables does not maximize classifier accuracy. Using the optimum sets of variables for both the LDFs and QDFs does not improve the accuracy significantly; however, the number of computations required is reduced. The optimum variables to use for LDFs are Rd1, Rd2, Rd4, Rd5, Rd8, and N2. The optimum variables to use for QDFs are Rd1. Rd2. Rd3. Rd5. Rd6. and Rd7. If the same set of variables must be shared by the LDFs and QDFs, then we recommend that the optimum set for the LDFs be used by both methods. Any six variables used for QDFs yields almost identical classification accuracies (see Figure 29), whereas any six variables used for LDFs results in classification accuracies that vary by up to 12% (see Figure 28).

7.4.5 Database

The prototype classifier saves the classification vectors into a dBase-formatted database. This format is compatible with many different commercial database packages. One important practical issue is the growth in the size of the database file generated while the classifier monitors a T1 line. The database size depends on how many different classification results are stored, the segment size selected, the filter window, the amount of traffic carried by the T1, the specific mix of traffic, and the duration of the classifier run. The largest database file collected was for a T1 to Peace River carrying wireless calls. For this trial the unfiltered and filtered classification vectors for the linear, quadratic, and hybrid discrimination methods were all stored. The segment size used was 2052 samples, and the T1 was monitored for 10 days. The size of the resulting database was almost 900 MB. Other T1s monitored storing only the linear and quadratic results for one day resulted in a database size of 200 MB. We found it useful to store the results of all methods so that all of the information could be retrieved at a later date. If hard disk space is limited, then only the filtered results could be recorded. This would considerably reduce the database size, as entries are only made when a signal on a given DS-0 changes. A typical size for a database resulting from storing only filtered hybrid vectors over one week would be 200 MB. A classifier could conceivably be allowed to run unattended for months at a time given several Gbytes of storage capacity.

7.4.6 Prior Probabilities

In all of the field trials and off-line simulations, the prior probabilities were assumed to be equal over all signal classes. If all classes are not equally probable then these prior probabilities can be changed, which should increase the accuracy of the classifier. One problem in doing this is to determine those prior probabilities. From the experience of the field trials, the type of traffic carried by each T1 varied considerably. One way around this is to change the prior probabilities adaptively to reflect the traffic on the T1 that is being monitored. This too might cause problems as the type of traffic carried often changes depending on the time of day. Since the classifier is already very accurate by assuming that all classes are equally probable, we do not believe that there would be very much to gain by using complex methods to tune the prior probabilities to conform with estimates of the expected signal mix. One possible exception is if a user is certain that some signal classes will never be encountered on a particular T1 trunk. Such signal classes could then be safely assigned a prior probability of 0 to effectively remove them from the classifier's possible decisions, and thereby increase the accuracy of the classifier on the remaining classes.

Chapter 8

8.0 Conclusions

In this thesis we designed, implemented, and experimentally verified an accurate real-time voiceband signal classifier. The classifier was designed and implemented using an inexpensive PC enhanced with T1 and DSP cards. The parts cost of the classifier was approximately $8250 CAD. A GUI was developed which runs on the PC, with classification algorithms implemented in the DSP. The classification vectors generated by the DSP card are displayed graphically on the screen in a real-time display window, and are stored simultaneously into a database to allow off-line database queries. The robustness of the prototype was demonstrated by using it in actual field trials conducted in cooperation with TELUS and TELUS Mobility. Further development work is continuing as part of a collaboration with VisionSmart Inc.

The classifier is trained to recognize a total of 23 classes. These classes include 4 data modem signals, 4 facsimile standards, speech, random binary, FSK signalling, ringback, and a class that contains the twelve common DTMF tones. These classes were selected because each class has a sufficiently unique PSD, and because these signals are known to be present on the PSTN.

Different discrimination methods were evaluated including LDFs, QDFs, ALNs, and Pseudo QDFs. A hybrid method was also evaluated using the LDFs and QDFs in combination. The measured classification accuracy approached 100% for most signal classes using the LDFs. By using a hybrid approach, the accuracy for all classes nearly reached 100%. The accuracies of our best configuration are given in Table 48.

Table 48: Percent classification accuracy using a two-stage classifier (N=2052, Std V.34, Incl. EN).

Class	1	2	3	4	5	6	7	8	9	10	11	12	>12
1	99.93	-	-	-	-	-	-	-	-	-	-	0.07	-
2	-	100.00	-	-	-	-	-	-	-	-	-	-	-
3	-	-	99.93	0.04	-	-	0.03	-	-	-	-	-	-
4	-	-	-	98.59	1.41	-	-	-	-	-	-	-	-
5	-	-	-	0.16	99.80	0.04	-	-	-	-	-	-	-
6	-	-	-	-	-	98.96	1.04	-	-	-	-	-	-
7	-	-	-	-	-	0.99	99.0	-	-	-	-	-	0.01
8	-	-	0.74	-	-	-	-	99.26	-	-	-	-	-
9	-	-	-	-	-	-	-	-	100.00	-	-	-	-
10	-	-	-	-	-	-	-	-	-	100.00	-	-	-
11	-	-	-	-	-	-	-	2.47	-	-	97.53	-	-
12	-	-	-	-	-	-	-	-	-	-	-	100.00	-
>12	-	-	-	-	-	-	-	-	-	-	-	-	100.00

8.1 Summary of Achievements and Results

The specific achievements and results of the research work completed as part of this M.Sc. include:

1. Demonstrating that all 24 channels of a T1 can be accurately classified in real-time using a classifier constructed using only off-the-shelf PC-based hardware. Software was written for the DSP to compute the classification vectors, in real-time, based on algorithms developed by Sewall. Software was also written for the GUI which controlled the classification parameters, displayed the classification results on a real-time display, and stored the classification results into a database for off-line queries such as busy hour and pie chart graphs.

2. The algorithms developed by Sewall were implemented, modified, improved, and experimentally verified using real signal data. Additional hybrid classification techniques were also designed and implemented to further increase the accuracy of the classifier for certain signal classes. Simple averaging filters were also implemented and then shown to eliminate most misclassifications in real signals without effecting the structure of a call.

3. Data was collected for all signal classes (excluding random binary) at the first field trial, which resulted in the addition of three new signal classes: FSK signalling, ring-

back, and DTMF tones. The data was in addition to the roughly 2.5 hours of data previously recorded by Sewall.

4. The stability of the classifier was demonstrated at the field trials while monitoring and storing the classification results of real T1 lines. The classifier ran unattended for 24 hour periods, for several days, and for a period of ten straight days. The T1 lines monitored included local and long distance trunks, as well as both land line and wireless T1s. Data from these field trials were used to generate busy hour and pie chart graphs.

5. The accuracy of the classifier was difficult to determine in the context of live traffic. Off-line experiments suggest accuracies that approach 100% for all signal classes. Practical considerations (e.g. finite segment length, modem and fax training signals, silent intervals in voice calls, etc.) limit the achievable accuracy of the classifier. Also, the classifier can only correctly classify signals that it is trained to recognize. Attempts were made to generate a broad range of signal classes that would cover most signals that the classifier would see on the PSTN. Clearly situations are likely to arise in which the classifier will attempt to classify a signal that it is not trained to recognize (e.g. proprietary protocols, new standards, etc.), and this will degrade the classifiers accuracy.

6. Different discrimination methods were investigated including LDFs, QDFs, pseudo QDFs, and ALNs. Only the LDF and QDF methods were used in the classifier. A third hybrid method was also used which combined the strengths of the LDFs and QDFs.

7. An optimum set of feature variables to be used for LDFs, QDFs, and pseudo QDFs was proposed based on off-line experiments done using both the data Sewall collected and new data collected as part of this M.Sc. thesis. The optimum number of feature variables to use was found experimentally to be six for both LDFs and QDFs. Using only these optimal variables does not increase the accuracy of the classifier significantly, however, it does reduce the computations required from the DSP.

8. A two-stage classifier was proposed, which helped increase the unfiltered classification accuracy of speech from 91% to 99%.

8.2 Classifier Interest

The favourable attention that the classifier has received by TR*Labs* telco sponsors suggests that the classifier is a useful tool. A recent ruling by the Canadian Radio-television and Telecommunications Commission (CRTC) may increase interest in the classifier. The ruling allows ISPs to provide long-distance phone services, but they are required to share their resulting revenues with the telcos. A voiceband signal classifier, such as the one developed in this thesis, would allow telcos to monitor and enforce the ruling.

8.3 Future Research Work

Many follow-up research projects are possible based on the research reported in this M.Sc. thesis.

At the time of writing a, DSP-less Windows NT classifier was under development by another student. This prototype needs to be tested on field trials to ensure its correctness. Further development of classifier technology should probably be conducted with this new platform because of the greater flexibility of the Windows GUI.

A SS7 signalling interface could be added to the classifier. The advantage of an SS7 link would be that call boundaries could be easily determined. If the call boundaries are known, then the problems with silent intervals in voice calls could easily be resolved. The classifier is currently unable to recognize the signals that are used in the negotiation phase in modem and fax calls. This results in the classifier misclassifying these signals into other classes. If the call boundaries were to be known then these misclassifications could be avoided. The addition of an SS7 interface would likely be quite involved due to the complexity of the protocol.

An alternative to the SS7 interface is to use a higher level of filtering that exploits call structure. Such filtering could exploit knowledge of the different possible call types and their structure to better estimate the call boundaries. This method would likely be as accurate as the SS7 signalling method for non-voice calls. For voice calls only an estimate of the end of a call is possible. Such a method would have to be cautious about relying on the presence of ringback to identify the start of calls, since ringback is not always used.

While the classifier has been implemented using a PC, it could possibly be implemented in an Application Specific Integrated Circuit (ASIC) such as a large field-programmable gate-array (FPGA). Such an ASIC might be useful to be able to build a scaled up classifier that could handle higher bit rate trunks, such as T3s.

References

[1] N. Benvenuto, "A Speech/Voiceband Data Discriminator", IEEE Transactions on Communications, Vol. 41, No. 4, April 1993, pp. 539 - 543.

[2] J. S. Sewall, "Signal Classification in Digital Telephone Networks", M.Sc. thesis, Department of Electrical Engineering, University of Alberta, Edmonton, Alberta, Canada, January 3, 1996.

[3] N. Benvenuto, W. R. Daumer, "Classification of Voiceband Data Signals", Proc. IEEE International Conference on Communications, Atlanta, GA, April 1990, Vol. 3, pp. 1010 - 1013.

[4] N. Benvenuto, "Detection of Modem Type and Bit Rate of FSK Voiceband Data Signals", Proc. IEEE International Conference on Communications, Boston, MA, June 1989, Vol. 2, pp 1101 - 1105.

[5] A. V. Oppenheim, R. W. Schafer, "Discrete-Time Signal Processing", Prentice-Hall Inc., 1989.

[6] J. G. Proakis, "Digital Communications", Third Edition, McGraw-Hill Inc., 1995.

[7] M. J. Norusis, SPSS Inc., "SPSS Professional Statistics 6.1", SPSS Inc., 1994.

[8] SPSS Inc., "SPSS Statistical Algorithms", Second Edition, SPSS Inc.

[9] R. L. Freeman, "Telecommunication System Engineering", Second Edition, John Wiley & Sons Inc., 1989.

[10] J. Bellamy, "Digital Telephony", John Wiley & Sons Inc., 1982.

[11] S. Morgan, "The Internet and the Local Telephone Network: Conflicts and Opportunities", IEEE Communications Magazine, Vol. 36 No. 1, January 1998.

[12] W. R. Klecka, "Discriminant Analysis", SAGE Publications Inc., Ninth Printing, 1988.

[13] "Minitab Reference Manual", Minitab Inc., 1995.

[14] E. R. Dougherty, "Probability and Statistics for the Engineering, Computing, and Physical Sciences", Prentice-Hall Inc., 1990.

[15] J. L. Devore, "Probability and Statistics for Engineering and the Sciences". Third Edition, Brooks/Cole Publishing Company, 1991.

[16] Dendronic Decisions Limited, Edmonton, Alberta, "Adaptive Logic Networks Technical Overview", 1997.

[17] R. H. Shumway, "Discriminant Analysis for Timer Series", Handbook of Statistics, Vol. 2, North-Holland Publishing Company, 1982.

[18] P. A. Lachenbruch, "Discriminant Analysis", Hafner Press, 1975.

[19] G. R. McClain, "The Handbook of International Connectivity Standards", Van Nostrand Reinhold, 1992

[20] G. Held, "Digital Networking and T-Carrier Multiplexing", John Wiley & Sons Ltd., 1990.

[21] ITU-T Recommendation Q.315, "PCM line signal sender (transmitter)", Fascicle VI.4, Geneva, 1993.

[22] ITU-T Recommendation Q.316, "PCM line signal receiver", Fascicle VI.4, Geneva, 1993.

[23] CCITT Recommendation V.17, "A 2-wire modem for facsimile applications with rates up to 14 400 bits/s", Geneva, 1991.

[24] ITU-T Recommendation V.22, "1200 bits per second duplex modem standardized for use in the general switched telephone network and on point-to-point 2-wire leased telephone-type circuits", 1993.

[25] ITU-T Recommendation V.22*bis*, "2400 bits per second duplex modem using the frequency division technique standardized for use on the general switched telephone network and on point-to-point 2-wire leased telephone-type circuits", 1993.

[26] ITU-T Recommendation V.27*ter*, "4800/2400 bits per second modem standardized for use in the general switched telephone network", 1993.

[27] ITU-T Recommendation V.34, "A modem operating at data signalling rates of up to 33 600 bits/s for use on the general switched telephone network and on leased point-to-point 2-wire telephone-type circuits", Geneva, October 1996.

[28] 3Com, "3Com V.90 Technology", 3Com Technical Paper (http://www.3com.com), 1998.

[29] P. M. Henderson, "56Kbps Data Transmission Across the PSTN: How does it work?", White Paper, Rockwell Semiconductor Systems, 1997.

[30] W. D. Grover, D. E. Dodds, "EE589: Telecommunication Systems Engineering", Course notes for EE 589, University of Alberta, 1996.

[31] B. F. Cockburn, D. P. Sarda, "Implementation and Evaluation of an Accurate Real-Time Voiceband Signal Classifier", Proceedings 1998 IEEE Canadian Conference on Electrical and Computer Engineering, May 24-28, Waterloo, Ontario, Canada, Volume 1, pages 133-136.

[32] "Canadians Swipe the Stripe One Billion Times in 1997", Canada NewsWire (http://www.newswire.ca), Toronto, Ontario, Canada, January 7, 1998.

[33] "Agreement Reached on 56K Modem Standard", International Telecommunication Union Press Release, Geneva, February 6, 1998.

[34] Unitel, "Unitel Communications Incorporated Network Plan", pages 110-118.

[35] USRobotics, "Sportster Voice For Macintosh Computers, Installation & Troubleshooting", 1996.

[36] Marija J. Norusis, SPSS Inc., "SPSS Professional Statistics 6.1", 1994.

[37] USRobotics Technical Support, "Explanation of ATI6/ATI11 commands", Revision 1.01, July 7, 1995, http://web.aimnet.com/~jnavas/modem/usri6_11.txt.

Appendix A

A.1 Prototype Voiceband Signal Classifier Features

- Real-time display for all 24 channels, with the ability to display a maximum of two algorithms per channel in the form of advancing horizontal bars
- Busy hour and pie chart queries on stored database files
- Load, save, and append of data into a dBASE IV formatted database file
- Select input from either a T1 or a stored data file
- T1 framing on SF or ESF format
- DSP options include setable power threshold, segment size, and variable selection for LDF and QDF
- Important events and errors logged into a text log file

Figure 49 shows a screen shot of the signal classifier GUI. The user can select from one of the five pull-down menus shown. The next few sections describe the options available in each pull-down menu. Menu options preceded by an "*" describe options that have not been implemented.

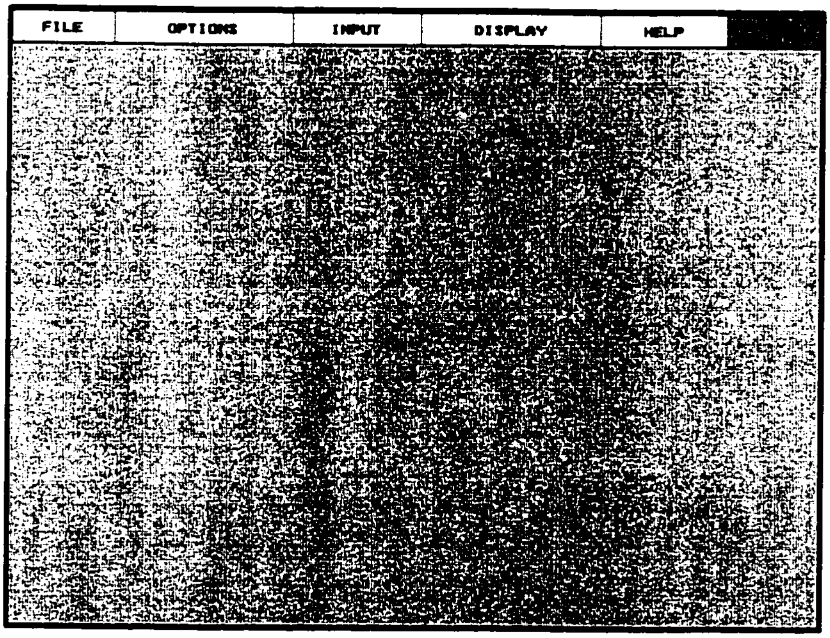


Figure 49: Signal classifier GUI.

Figure 50 shows the options available through the "File" pull-down menu.

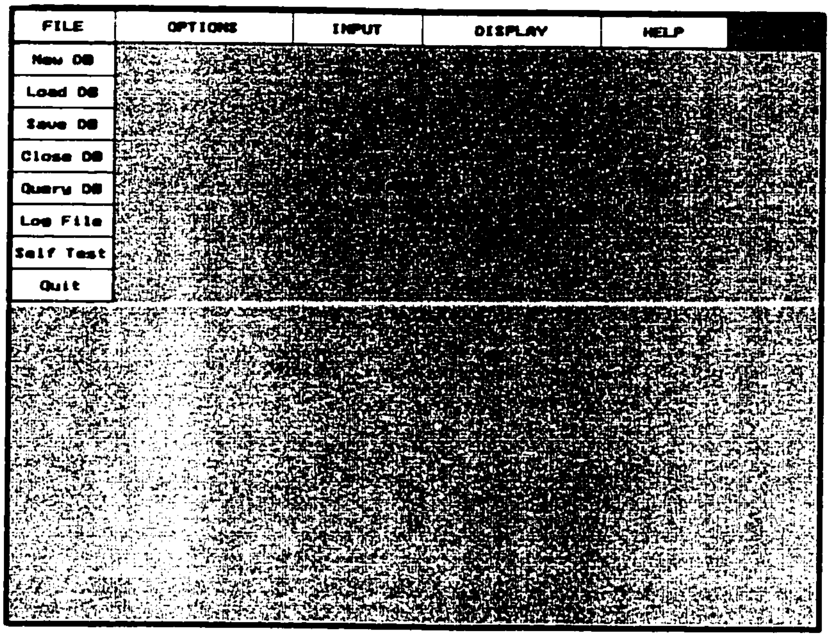


Figure 50: GUI "File" pull-down menu options.

- New DB: Initialize a new database file.
- Load DB: Prepare the classifier to open a previously stored database file.
- Save DB: Save the current database to a file. If a database with the same filename exists, the user will be prompted to overwrite, append, or cancel.
- Close DB: Close an opened database. If the database has not yet been saved the user will be warned.
- *Query DB: Feature to allow the user to query the contents of a database file.
- *Logfile: Feature to allow the user to view the contents of the ASCII text logfile without having to exit the GUI.
- *Self Test: Feature to allow the user to execute a self test. This would simply verify that the T1 and DSP cards are present. It would also test the T1 line.
- Quit: Exit the program. If a database is opened and not yet saved, the user will be warned.

Figure 51 shows the options available through the "Options" pull-down menu.

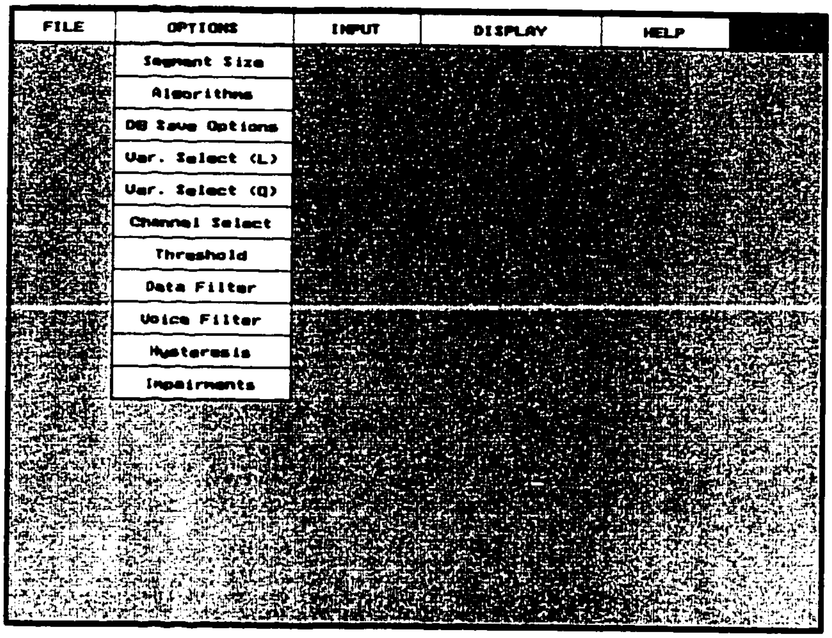


Figure 51: GUI "options" pull-down menu options.

- Segment Size: Allows the user to change to one of many pre-selected segment sizes. A segment size of less than 1024 samples will result in data loss if all 24 channels are active.
- *Algorithms: Allows the user enable or disable the LDF and QDF computation.
- *DB Save Options: Allows the user to select which classifications vectors to store into a database. This is useful if disk space is limited.
- Var. Select (L): Allows the user to select which set of variables are to be used in the LDFs.
- Var. Select (Q): Allows the user to select which set of variables are to be used in the QDFs.
- Channel Select: Allows the user to select which channels are to be active. When using a T1 line all 24 channels should be active.
- Threshold: Allows the user to change the power threshold value.
- Data Filter: Allows the user to set the window size for the data filter.

- Voice Filter: Allows the user to set the window size for the voice filter.
- *Hysteresis: Was suppose to be used to build inertia into the system to ride out silent intervals. This functionality is now covered by the filter settings.
- *Impairments: Feature to allow the user to simulate line impairments.

Figure 52 shows the options available through the "Options" pull-down menu.

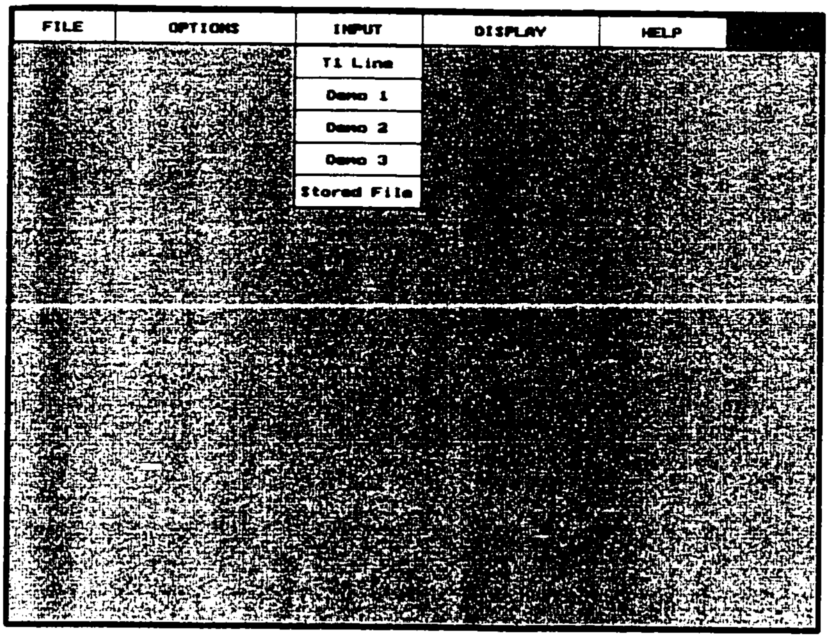


Figure 52: GUI "input" pull-down menu options.

- T1 Line: Selects the input data to be the physically connected T1 line.
- Demo 1: Selects a file named "demo1.mat" as the input file.
- Demo 2: Selects a file named "demo2.mat" as the input file.
- Demo 3: Selects a file named "demo3.mat" as the input file.
- Stored File: Allows the user to specify the name of a file to use as input data. The file must exist in the current directory, and must have a "mat" extension. The filename must also conform to the MS-DOS 8.3 specification (e.g. no more than 8 characters for the name and 3 for the extension).

Figure 53 shows the options available through the "Display" pull-down menu.

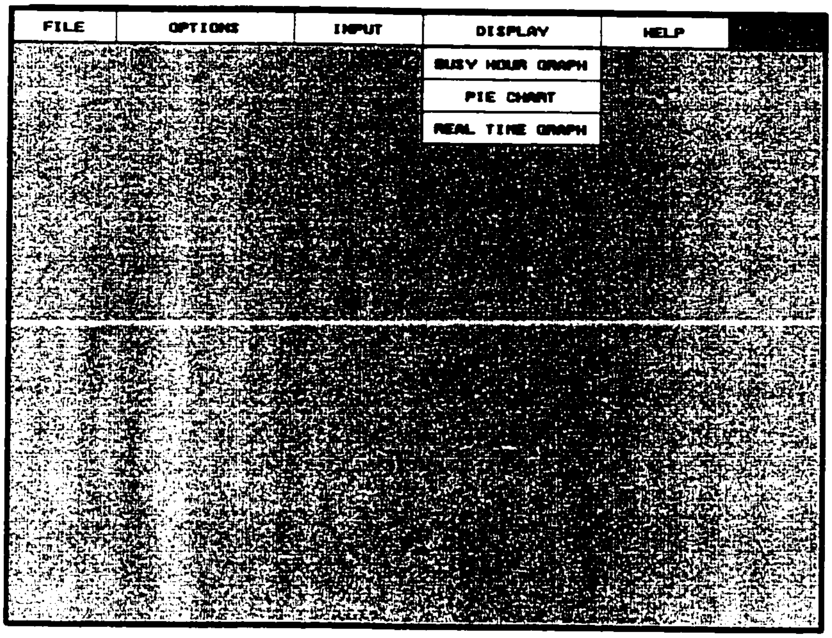


Figure 53: GUI "display" pull-down menu options.

- Busy Hour Display: Allows the user to display a busy hour graph for the opened database. The user will first be prompted for the date. If no records exist for the given date the user will be notified. Otherwise, the busy hour graph for the requested 24-hour period will be displayed with the busy hour outlined in red.

- Pie Chart: Allows the user to display a pie chart for the opened database. Again the user will be prompted for the date, and if no records exist for the given date the user will be notified. Otherwise, the pie chart for the requested 24-hour period displays the percentage of data for each class.

- Real Time Graph Displays a real time graph for the selected input. If T1 is selected as the input then all 24 channels would normally be selected.

Figure 54 shows the options available through the "Help" pull-down menu.

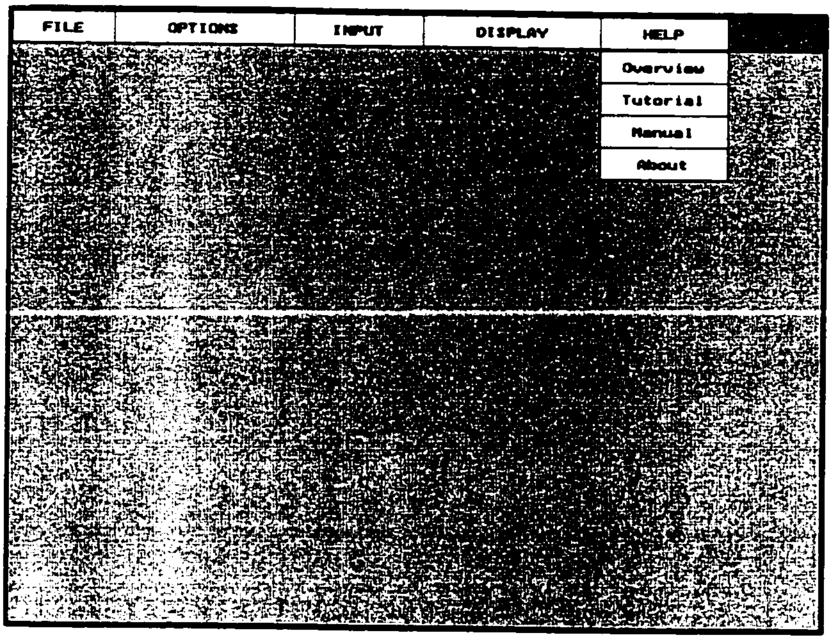


Figure 54: GUI "help" pull-down menu options.

- *Overview: Give the user an overview of the features.
- *Tutorial: Give the user a brief tutorial.
- *Manual: Allow the user to browse the on-line user manual.
- About: Display the current version of the classifier.

Figure 55 shows the typical appearance of a real-time display.

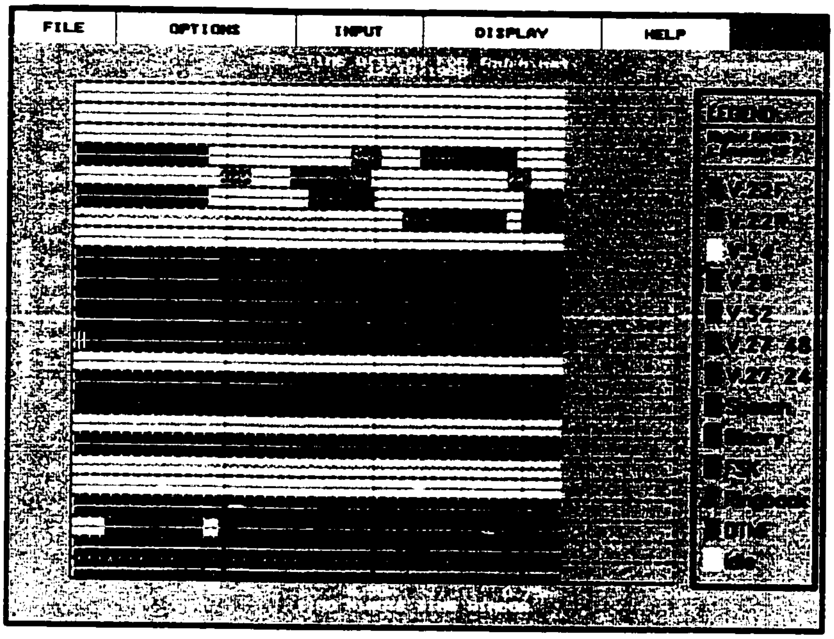


Figure 55: GUI real-time display.

The real-time display shows all 24 channels on the vertical axis, with time advancing to the right on the horizontal axis.

Appendix B

Table 49: Mu-law PCM encoding/decoding table.

Input Amplitude Range	Step Size	Segment Code S	Quantization Code Q	Decimal Code Value	Decoder Amplitude
0-1	1	000	0000	0	0
1-3	2	000	0001	1	2
3-5	2	000	0010	2	4
...	...	...	...	...	...
29-31	2	000	1111	15	30
31-35	4	001	0000	16	33
...	...	...	....	...	...
91-95	4	001	1111	31	93
95-103	8	010	0000	32	99
...	...	...	...	...	...
215-223	8	010	1111	47	219
223-239	16	011	0000	48	231
...	...	...	...	...	...
463-479	16	011	1111	63	471
479-511	32	100	0000	64	495
...	...	...	...	...	...
959-991	32	100	1111	79	975
991-1055	64	101	0000	80	1023
...	...	...	...	...	...
1951-2015	64	101	1111	95	1983
2015-2143	128	110	0000	96	2079
...	...	...	...	...	...
3935-4063	128	110	1111	111	3999
4063-4319	256	111	0000	112	4191
...	...	...	...	...	...
7903-8159	256	111	1111	127	8031

Appendix C

C.1 PC Specifications

The specifications for the PC are shown below:

- AMD 486 DX4/100 MHz CPU
- ISA/PCI generic motherboard
- 32 MB RAM
- 2.1 GB EIDE hard drive
- ATI PCI PC to TV VGA video card (for demonstration purposes)
- 12x EIDE CD-ROM Drive
- 3 1/2" floppy drive
- 14" monitor
- Fujitsu keyboard and MS-mouse
- MS-DOS 6.22 and Windows 3.11 for Workgroups
- Borland C++ V4.5 for Windows
- CodeBase V6.0
- ALN library functions (LIBALN) V1.1
- SPSS V6.1.2 for Windows

C.2 TMS320C30 DSP Specifications

- 40 MHz clock
- Floating point processor
- 50-ns, single-cycle instruction execution time
- 40 MFLOPS
- 20 MIPS
- Two 1K x 32-bit, single-cycle, dual-access, on-chip, RAM blocks
- 64x32-bit instruction cache
- Parallel ALU and multiplier instructions in a single cycle
- Block repeat capability
- Zero-overhead loops with single-cycle branches
- TMS320C3X/4X ANSI C compiler and linker V4.6

C.3 TELUS Maintenance Engineering Lab

C.3.1 Test Equipment

- Newbridge 3600 mainstreet bandwidth manager channel bank
- Newbridge LGS module (90-1228-02/D)
- Newbridge E&M 2/4W module (90-1203-01)

- Newbridge LGE module (90-1229-01)
- HP 4935A transmission test set tone generator
- Tellabs 8103 ringing tone generator
- Apple Powerbook 150
  - System 7.5.5
  - MacComCenter fax/data software (1994)
  - FreePPP 1.0.5
  - USR sportster voice 33.6 fax modem with speakerphone and personal voice mail

C.3.2 Internet Service Providers

Table 50: Internet service provider information.

ISP	Phone Number	Type of Call	Max Bit Rate (bps)
University of Alberta	403-492-3214	Local	28 800
TRLabs Edmonton	403-424-5691	Local	14 400
Corporate Computers Inc. (CCI)	403-450-0705	Local	33 600

C.3.3 Faxback Services

Table 51: Fax services information.

Fax Services	Phone Number	Class of Call	Max Bit Rate (bps)
Hewlett-Packard Faxback	800-333-1917	Long Distance	14 400
Intel Faxback	800-628-2283	Long Distance	14 400
TRLabs Edmonton	403-441-3600	Local	9 600
Author's Home Computer	403-458-3874	Local	14 400

C.4 TELUS Toll Building

- T-Com Digital Communications Test Set
  - Model 440B/T-ACE

Appendix D

D.1 Description of Files Recorded by Sewall

Note that the "?" marks in the "Time" column indicate that the exact length of the call was not recorded in the original table.

Table 52: Description of recorded data files by Sewall [2].

File Name	FAX/ Data/ Voice	Standard	bps	Org.	Dest.	Time (s)	Description
v32bis	d	V.32bis	14,400	local	492-3214	20	SupraFAXmodem; called UofA modem pool; negotiation
v32bis.2	d	V.32bis	14,400	"	"	20	SupraFAXmodem; called UofA modem pool; no neg.
v32	d	V.32	9,600	"	"	20	including negotiation
v32.2	d	V.32	9,600	"	"	20	no negotiation
data1	d	V.22bis	2,400	"	"	25	including negotiation
data2	d	V.22bis	2,400	"	"	25	no negotiation
data3	d	V.32bis	12,000	"	"	25	including negotiation
data4	d	"	9,600	"	"	25	including negotiation
data5	d	"	12,000	"	"	25	no negotiation
data6	d	"	9,600	"	"	25	including negotiation
data7	d	"	9,600	"	"	25	no negotiation
data10	d	"	14,400	"	492-3214	45	Called UofA; incl. negotiation; modem option N8 forces bps
data11	d	"	12,000	"	"	60	"; "; N7
data12	d	"	9,600	"	"	60	"; "; N6
data13	d	"	9,600	"	492-0096	60	"; "; N6
data14	d	"	4,800	"	"	60	"; "; N4
data15	d	V.32bis	2,400	local	492-0096	60	"; "; N3
data16	d	"	2,400	"	492-0024	60	"; "; N3; retrain?
data17	d	"	1,200	"	492-0096	60	"; "; N2
data18	d	"	1,200	"	492-0024	60	"; "; N2
data19	d	"	300	"	492-0096	60	"; "; N1
data20	d	V.34	24,000/ 26,400	"	444-7685	90	called WorldGate; incl. negotiation; speed not forced (N0)
data21	d	"	28,800/ 28,800	"	"	"	"; "; "
data22	d	"	26,400/ 26,400	"	"	"	"; "; N13
data23	d	"	24,000/ 24,000	"	"	"	"; "; N12
data24	d	"	21,600/ 21,600	"	"	"	"; "; N11
data25	d	"	19,200/ 19,200	"	"	"	"; "; N10

Table 52: Description of recorded data files by Sewall [2].

File Name	FAX/ Data/ Voice	Standard	bps	Org.	Dest.	Time (s)	Description
data26	d	"	16,800/ 16,800	"	"	"	"; "; N9
drun1	d	V.34	28,800/ 28,800	TRLabs	U of A 492-3214	30	no setup; &N0
drun2	d	"	"	"	"	"	"
drun3	d	"	"	"	"	"	"
drun4	d	"	"	"	"	"	"
drun5	d	"	"	"	"	"	"
drun6	d	V.34	14,400/ 14,400	"	"	"	"; &N8
drun7	d	V.22bis	2,400	"	492-0024	"	"
drun8	d	"	"	"	"	"	"
drun9	d	"	"	"	"	"	"
drun10	d	"	"	"	"	"	"
drun11	d	"	"	"	"	"	"
drun12	d	V.32bis	14,400	"	TRLabs	"	"
drun13	d	"	"	"	"	"	"
drun14	d	"	"	"	"	"	"
drun15	d	"	"	"	"	"	"
drun16	d	"	"	"	"	"	"
drun17	d	"	12,000	"	"	"	"; &N7
drun18	d	"	"	"	"	"	"
drun19	d	"	9,600	"	"	"	"; &N6
drun20	d	"	"	"	"	"	"
drun21	d	"	"	"	"	"	"
drun22	d	"	"	"	"	"	"
drun23	d	"	"	"	"	"	"
drun24	d	"	7,200	"	"	"	"; &N5
drun25	d	"	"	"	"	"	"
drun26	d	"	4,800	"	"	"	"; &N4
drun27	d	"	"	"	"	"	"
drun28	d	"	"	"	"	"	"
-	-	-	-	-	-	-	-
fax1	f	V.17	14,400	local	441-3600	?	fax to main office; setup included; two pages plus cover
fax2	f	V.17	12,000	"	"	?	"
fax3	f	V.17	9,600	"	"	?	"
fax4	f	V.17	7,200	"	"	?	"
fax5	f	V.27ter	4,800	"	"	?	"
fax6	f	V.27ter	2,400	"	"	?	"; "; no cover page; error after p. 1; part of p. 2
fax9	f	V.29	7,200	local	492-1811	?	fax out; setup included; two pages plus cover
fax10	f	V.27ter	4,800	local	"	?	"

Table 52: Description of recorded data files by Sewall [2].

File Name	FAX/ Data/ Voice	Standard	bps	Org.	Dest.	Time (s)	Description
fax11	f	V.27ter	2,400	local	"	?	" ; errored
fax12	f	V.29	9,600	local	"	?	"
fax13	f	V.29	9,600	local	"	?	" ; errored
fax14	f	V.29	9,600	local	"	?	"
fax15	f	V.29	9,600	local	"	?	"
fax16	f	V.29	9,600	local	"	?	"
fax17	f	V.29	7,200	local	"	?	"
fax21	f	V.29	7,200	local	604-721-0852	?	"
fax22	f	V.27ter	4,800	local	"	?	"
fax23	f	?	2,400	local	"	?	errored fax
fax24	f	?	?	441-3600	local	?	fax received; errored
fax26	f	?	?	"	local	?	fax received; errored
-	-	-	-	-	-	-	-
voice1	v	-	-	498-8397	local	70	Male/Female conversation
voice_1_6_14_17_8	v	-	-	remote	local	?	Pre-recorded message (my voice); two sentences read by others
voice_1_6_14_40_23	v	-	-	"	"	?	"
voice_1_6_14_44_36	v	-	-	"	"	?	Pre-recorded message; person mimicing modem
voice_1_6_14_45_13	v	-	-	"	"	?	"
voice_1_6_14_46_2	v	-	-	"	"	?	Pre-recorded message; person whistling
voice_1_6_14_47_48	v	-	-	"	"	?	Pre-recorded message; person saying nothing
voice_1_6_14_49_27	v	-	-	"	"	?	"
voice_1_6_15_21_15	v	-	-	"	"	?	"
voice_1_6_15_2_18	v	-	-	"	"	?	"
voice_1_6_15_38_52	v	-	-	"	"	?	"
voice_1_6_15_44_14	v	-	-	"	"	?	"
voice_1_6_15_4_27	v	-	-	"	"	?	"
voice_1_6_15_54_39	v	-	-	"	"	?	"
voice_1_6_15_5_35	v	-	-	"	"	?	"
voice_1_6_15_5_4	v	-	-	"	"	?	"

Table 52: Description of recorded data files by Sewall [2].

File Name	FAX/ Data/ Voice	Standard	bps	Org.	Dest.	Time (s)	Description
voice_1_6_ 17_11_41	v	-	-	"	"	?	"
voice_1_6_ 17_54_22	v	-	-	"	"	?	"
voice_1_6_ 18_55_44	v	-	-	"	"	?	"
voice_1_7_ 10_33_11	v	-	-	"	"	?	"
voice_1_7_ 13_17_54	v	-	-	"	"	?	"
voice_1_7_ 14_49_38	v	-	-	"	"	?	"
voice_1_7_ 14_57_46	v	-	-	"	"	?	"
voice_1_7_ 18_6_59	v	-	-	"	"	?	"
voice_1_7_ 20_32_3	v	-	-	"	"	?	"
voice_1_7_ 7_21_34	v	-	-	"	"	?	"
voice_1_7_ 7_57_37	v	-	-	"	"	?	"
voice_1_8_ 14_51_6	v	-	-	"	"	?	"
voice_1_8_ 16_58_8	v	-	-	"	"	?	"
voice_1_8_ 18_25_20	v	-	-	"	"	?	"
voice_1_8_ 8_17_1	v	-	-	"	"	?	"
voice_1_9_ 14_26_10	v	-	-	"	"	?	"
-	-	-	-	-	-	-	-
sim1	d	V.22	1200	-	-	10	simulated call; recorded at alpha point in 4-wire connection
sim2	d	V.22	1200	-	-	30	" ; beta
sim3	d	V.22bis	2400	-	-	10	" ; alpha
sim4	d	V.22bis	2400	-	-	30	" ; beta
sim5	f	V.27ter	4800	-	-	10	" ; alpha
sim6	f	V.27ter	2400	-	-	10	" ; alpha ; fallback mode
sim7	f	V.29	9600	-	-	10	" ; alpha
sim8	f	V.29	7200	-	-	10	" ; alpha ; fallback mode
sim9	d	V.32	9600	-	-	10	" ; alpha
sim10	d	V.32bis	14,400	-	-	10	" ; alpha
sim11	f	V.17	14,400	-	-	10	" ; alpha ; identical simulation to V.32bis ; correct?
sim12	d	V.22	1200	-	-	30	" ; beta

Table 52: Description of recorded data files by Sewall [2].

File Name	FAX/ Data/ Voice	Standard	bps	Org.	Dest.	Time (s)	Description
sim13	d	V.22bis	2400	-	-	30	" ; beta
sim_1_a_1	d	V.22	1200	-	-	25	impairment model 1, alpha monitoring point
sim_1_a_2	d	V.22bis	2400	-	-	25	impairment model 1, alpha monitoring point
sim_1_a_3	f	V.27ter	4800	-	-	25	impairment model 1, alpha monitoring point
sim_1_a_4	f	V.27ter	2400	-	-	25	impairment model 1, alpha monitoring point
sim_1_a_5	f	V.29	9,600	-	-	25	impairment model 1, alpha monitoring point
sim_1_a_6	f	V.29	7,200	-	-	25	impairment model 1, alpha monitoring point
sim_1_a_7	d	V.32	9,600	-	-	25	impairment model 1, alpha monitoring point
sim_1_a_8	d	V.32bis	14,400	-	-	25	impairment model 1, alpha monitoring point
sim_1_a_9	f	V.17	14,400	-	-	25	impairment model 1, alpha monitoring point
sim_1_b_10	d	V.22	1200	-	-	25	impairment model 1, beta monitoring point
sim_1_b_11	d	V.22bis	2400	-	-	25	impairment model 1, beta monitoring point
sim_2_a_1	d	V.22	1200	-	-	25	impairment model 2, alpha monitoring point
sim_2_a_2	d	V.22bis	2400	-	-	25	impairment model 2, alpha monitoring point
sim_2_a_3	f	V.27ter	4800	-	-	25	impairment model 2, alpha monitoring point
sim_2_a_4	f	V.27ter	2400	-	-	25	impairment model 2, alpha monitoring point
sim_2_a_5	f	V.29	9,600	-	-	25	impairment model 2, alpha monitoring point
sim_2_a_6	f	V.29	7,200	-	-	25	impairment model 2, alpha monitoring point
sim_2_a_7	d	V.32	9,600	-	-	25	impairment model 2, alpha monitoring point
sim_2_a_8	d	V.32bis	14,400	-	-	25	impairment model 2, alpha monitoring point
sim_2_a_9	f	V.17	14,400	-	-	25	impairment model 2, alpha monitoring point
sim_2_b_10	d	V.22	1200	-	-	25	impairment model 2, beta monitoring point
sim_2_b_11	d	V.22bis	2400	-	-	25	impairment model 2, beta monitoring point
sim_3_a_1	d	V.22	1200	-	-	25	impairment model 3, alpha monitoring point
sim_3_a_2	d	V.22bis	2400	-	-	25	impairment model 3, alpha monitoring point
sim_3_a_3	f	V.27ter	4800	-	-	25	impairment model 3, alpha monitoring point
sim_3_a_4	f	V.27ter	2400	-	-	25	impairment model 3, alpha monitoring point
sim_3_a_5	f	V.29	9,600	-	-	25	impairment model 3, alpha monitoring point
sim_3_a_6	f	V.29	7,200	-	-	25	impairment model 3, alpha monitoring point
sim_3_a_7	d	V.32	9,600	-	-	25	impairment model 3, alpha monitoring point
sim_3_a_8	d	V.32bis	14,400	-	-	25	impairment model 3, alpha monitoring point
sim_3_a_9	f	V.17	14,400	-	-	25	impairment model 3, alpha monitoring point
sim_3_b_10	d	V.22	1200	-	-	25	impairment model 3, beta monitoring point
sim_3_b_11	d	V.22bis	2400	-	-	25	impairment model 3, beta monitoring point
sim_4_a_1	d	V.22	1200	-	-	25	impairment model 4, alpha monitoring point
sim_4_a_2	d	V.22bis	2400	-	-	25	impairment model 4, alpha monitoring point
sim_4_a_3	f	V.27ter	4800	-	-	25	impairment model 4, alpha monitoring point
sim_4_a_4	f	V.27ter	2400	-	-	25	impairment model 4, alpha monitoring point
sim_4_a_5	f	V.29	9,600	-	-	25	impairment model 4, alpha monitoring point
sim_4_a_6	f	V.29	7,200	-	-	25	impairment model 4, alpha monitoring point
sim_4_a_7	d	V.32	9,600	-	-	25	impairment model 4, alpha monitoring point

Table 52: Description of recorded data files by Sewall [2].

File Name	FAX/ Data/ Voice	Standard	bps	Org.	Dest.	Time (s)	Description
sim_4_a_8	d	V.32bis	14,400	-	-	25	impairment model 4, alpha monitoring point
sim_4_a_9	f	V.17	14,400	-	-	25	impairment model 4, alpha monitoring point
sim_4_b_10	d	V.22	1200	-	-	25	impairment model 4, beta monitoring point
sim_4_b_11	d	V.22bis	2400	-	-	25	impairment model 4, beta monitoring point
sim_5_a_1	d	V.22	1200	-	-	25	impairment model 5, alpha monitoring point
sim_5_a_2	d	V.22bis	2400	-	-	25	impairment model 5, alpha monitoring point
sim_5_a_3	f	V.27ter	4800	-	-	25	impairment model 5, alpha monitoring point
sim_5_a_4	f	V.27ter	2400	-	-	25	impairment model 5, alpha monitoring point
sim_5_a_5	f	V.29	9,600	-	-	25	impairment model 5, alpha monitoring point
sim_5_a_6	f	V.29	7,200	-	-	25	impairment model 5, alpha monitoring point
sim_5_a_7	d	V.32	9,600	-	-	25	impairment model 5, alpha monitoring point
sim_5_a_8	d	V.32bis	14,400	-	-	25	impairment model 5, alpha monitoring point
sim_5_a_9	f	V.17	14,400	-	-	25	impairment model 5, alpha monitoring point
sim_5_b_10	d	V.22	1200	-	-	25	impairment model 5, beta monitoring point
sim_5_b_11	d	V.22bis	2400	-	-	25	impairment model 5, beta monitoring point
-	-	-	-	-	-	-	-
dlib1	-	-	-	-	-	6	information signal from file v32bis; ie. no neg, no retrain
dlib2	-	-	-	-	-	20	information signal from file v32bis.2
dlib3	-	-	-	-	-	8	information signal from file v32
dlib4	-	-	-	-	-	20	information signal from file v32.2
dlib5	-	-	-	-	-	13	information signal from file data1
dlib6	-	-	-	-	-	25	information signal from file data2
dlib7	-	-	-	-	-	9	information signal from file data3
dlib8	-	-	-	-	-	9	information signal from file data4
dlib9	-	-	-	-	-	25	information signal from file data5
dlib10	-	-	-	-	-	15	information signal from file data6
dlib11	-	-	-	-	-	25	information signal from file data7
dlib14	-	-	-	-	-	24	information signal from file data10
dlib15	-	-	-	-	-	41	information signal from file data11
dlib16	-	-	-	-	-	44	information signal from file data12
dlib17	-	-	-	-	-	42	information signal from file data13
dlib18	-	-	-	-	-	38	information signal from file data14
dlib19	-	-	-	-	-	46	information signal from file data15
dlib20	-	-	-	-	-	51	information signal from file data16
dlib21	-	-	-	-	-	41	information signal from file data17
dlib22	-	-	-	-	-	46	information signal from file data18
dlib23	-	-	-	-	-	45	information signal from file data19
dlib24	-	-	-	-	-	62	information signal from file data20
dlib25	-	-	-	-	-	50	information signal from file data21
dlib26	-	-	-	-	-	62	information signal from file data22

Table 52: Description of recorded data files by Sewall [2].

File Name	FAX/ Data/ Voice	Standard	bps	Org.	Dest.	Time (s)	Description
dlib27	-	-	-	-	-	62	information signal from file data23
dlib28	-	-	-	-	-	62	information signal from file data24
dlib29	-	-	-	-	-	62	information signal from file data25
dlib30	-	-	-	-	-	62	information signal from file data26
dlib31	-	-	-	-	-	25	information signal from file fax1
dlib32	-	-	-	-	-	25	information signal from file fax2
dlib33	-	-	-	-	-	38	information signal from file fax3
dlib34	-	-	-	-	-	50	information signal from file fax4
dlib35	-	-	-	-	-	75	information signal from file fax5
dlib36	-	-	-	-	-	75	information signal from file fax6
dlib37	-	-	-	-	-	38	information signal from file fax9
dlib38	-	-	-	-	-	50	information signal from file fax10
dlib39	-	-	-	-	-	25	information signal from file fax11
dlib40	-	-	-	-	-	30	information signal from file fax12
dlib41	-	-	-	-	-	7	information signal from file fax13
dlib42	-	-	-	-	-	30	information signal from file fax14
dlib43	-	-	-	-	-	30	information signal from file fax15
dlib44	-	-	-	-	-	30	information signal from file fax16
dlib45	-	-	-	-	-	38	information signal from file fax17
dlib46	-	-	-	-	-	36	information signal from file fax21
dlib47	-	-	-	-	-	56	information signal from file fax22
dlib48	-	-	-	-	-	27	information signal from file fax23
dlib49	-	-	-	-	-	14	information signal from file fax24
dlib50	-	-	-	-	-	15	information signal from file fax26
rand1	rand.	-	-	-	-	125	random PCM sample stream
rand2	rand.	-	-	-	-	125	random PCM sample stream

D.2 Description of Data Files Recorded by Sarda

Table 53: Description of data files excluding V.34.

File Name	Connect Rate (bps)	Call Origination	Call Destination	Monitor Port	Length (sec)	Signal Type
* tr_b	14400	TELUS Lab	TR*Labs*	Rx	106	V.32
cci_n1a1	300	TELUS Lab	CCI	Tx	139	Bell 103
cci_n1b1	300	TELUS Lab	CCI	Rx	90	Bell 103
cci_n2a1	1200	TELUS Lab	CCI	Tx	106	V.22 F
cci_n2b1	1200	TELUS Lab	CCI	Rx	155	V.22 R

Table 53: Description of data files excluding V.34.

File Name	Connect Rate (bps)	Call Origination	Call Destination	Monitor Port	Length (sec)	Signal Type
tr_2_a	1200	TELUS Lab	TR*Labs*	Tx	57	V.22 F
tr_2_b	1200	TELUS Lab	TR*Labs*	Rx	57	V.22 R
tr_3_a	2400	TELUS Lab	TR*Labs*	Tx	57	V.22*bis* F
tr_3_b	2400	TELUS Lab	TR*Labs*	Rx	65	V.22*bis* R
tr_4_a	4800	TELUS Lab	TR*Labs*	Tx	57	V.32
tr_4_b	4800	TELUS Lab	TR*Labs*	Rx	57	V.32
tr_5_a	7200	TELUS Lab	TR*Labs*	Tx	57	V.32
tr_5_b	7200	TELUS Lab	TR*Labs*	Rx	57	V.32
tr_6_a	9600	TELUS Lab	TR*Labs*	Tx	57	V.32
tr_6_b	9600	TELUS Lab	TR*Labs*	Rx	57	V.32
tr_7_a	12000	TELUS Lab	TR*Labs*	Tx	57	V.32
tr_7_b	12000	TELUS Lab	TR*Labs*	Rx	57	V.32
tr_n1_a1	300	TELUS Lab	TR*Labs*	Tx	123	Bell 103
tr_n1_b1	300	TELUS Lab	TR*Labs*	Rx	139	Bell 103
tr_n3_a1	2400	TELUS Lab	TR*Labs*	Tx	131	V.22*bis* F
tr_n3_b1	2400	TELUS Lab	TR*Labs*	Rx	90	V.22*bis* R
u_n2_a1	1200	TELUS Lab	UofA	Tx	?	V.22 F
u_n2_b1	1200	TELUS Lab	UofA	Rx	?	V.22 R
u_n3_a1	2400	TELUS Lab	UofA	Tx	139	V.22*bis* F
u_n3_b1	2400	TELUS Lab	UofA	Rx	155	V.22*bis* R

All data files listed in Table 54 belong to the V.34 class and originated from the TELUS Lab. A "?" mark in the symbol rate and carrier frequency columns indicate that these values were not specifically recorded; however, they were probably 3429 baud and 1959 Hz, respectively. These values represent the default rates that the connecting modems appear to negotiate on land lines.

Table 54: V.34 data files.

File Name	Connect Speed (bps)	Symbol Rate (sym/s)	Carrier Freq. (Hz)	Call Destination	Monitor Port	Length (sec)
* cci_b	26400	?	?	CCI	Rx	106
cci_10_a	19200	?	?	CCI	Tx	57
cci_10_b	19200	?	?	CCI	Rx	57
cci_11_a	21600	?	?	CCI	Tx	57
cci_11_b	21600	?	?	CCI	Rx	57
cci_12_a	24000	?	?	CCI	Tx	57
cci_12_b	24000	?	?	CCI	Rx	57
cci_13_a	26400	?	?	CCI	Tx	57
cci_13_b	26400	?	?	CCI	Rx	57
cci_4_a	4800	?	?	CCI	Tx	57
cci_4_b	4800	?	?	CCI	Rx	57
cci_5_a	7200	?	?	CCI	Tx	57
cci_5_b	7200	?	?	CCI	Rx	65
cci_6_a	9600	?	?	CCI	Tx	57
cci_6_b	9600	?	?	CCI	Rx	57
cci_7_a	12000	?	?	CCI	Tx	57
cci_7_b	12000	?	?	CCI	Rx	57
cci_8_a	14400	?	?	CCI	Tx	57
cci_8_b	14400	?	?	CCI	Rx	57
cci_9_a	16800	?	?	CCI	Tx	57
cci_9_b	16800	?	?	CCI	Rx	57
cci_a_1	26400	3429	1959	CCI	Tx	65

Table 54: V.34 data files.

File Name	Connect Speed (bps)	Symbol Rate (sym/s)	Carrier Freq. (Hz)	Call Destination	Monitor Port	Length (sec)
cci_a_10	21600	2400	1800	CCI	Tx	180
cci_a_11	4800	2400	1800	CCI	Tx	123
cci_a_12	14400	2400	1800	CCI	Tx	131
cci_a_13	26400	3000	2000	CCI	Tx	123
cci_a_14	4800	3000	2000	CCI	Tx	155
cci_a_15	14400	3000	2000	CCI	Tx	90
cci_a_16	24000	2800	1867	CCI	Tx	196
cci_a_17	4800	2800	1867	CCI	Tx	139
cci_a_18	14400	2800	1867	CCI	Tx	147
cci_a_19	26400	3429	1959	CCI	Tx	123
cci_a_2	26400	3200	1920	CCI	Tx	65
cci_a_20	9600	3200	1920	CCI	Tx	114
cci_a_21	9600	3000	2000	CCI	Tx	123
cci_a_22	9600	2400	1800	CCI	Tx	131
cci_a_3	26400	3000	2000	CCI	Tx	65
cci_a_4	24000	2800	1867	CCI	Tx	147
cci_a_5	21600	2400	1800	CCI	Tx	106
cci_a_6	21600	3429	1959	CCI	Tx	82
cci_a_7	4800	3429	1959	CCI	Tx	114
cci_a_8	7200	3429	1959	CCI	Tx	106
cci_a_9	9600	3429	1959	CCI	Tx	82
cci_b_1	26400	3429	1959	CCI	Rx	65
cci_b_10	21600	2400	1600	CCI	Rx	188
cci_b_11	4800	2400	1600	CCI	Rx	114
cci_b_12	14400	2400	1600	CCI	Rx	131
cci_b_13	24000	3000	1800	CCI	Rx	172
cci_b_14	4800	3000	1800	CCI	Rx	139
cci_b_15	14400	3000	1800	CCI	Rx	204

Table 54: V.34 data files.

File Name	Connect Speed (bps)	Symbol Rate (sym/s)	Carrier Freq. (Hz)	Call Destination	Monitor Port	Length (sec)
cci_b_16	24000	2800	1680	CCI	Rx	131
cci_b_17	4800	2800	1680	CCI	Rx	237
cci_b_18	14400	2800	1680	CCI	Rx	131
cci_b_19	26400	3429	1959	CCI	Rx	123
cci_b_2	26400	3200	1829	CCI	Rx	65
cci_b_20	9600	3200	1829	CCI	Rx	139
cci_b_21	9600	3000	1800	CCI	Rx	82
cci_b_22	9600	2400	1600	CCI	Rx	114
cci_b_3	24000	3000	1800	CCI	Rx	65
cci_b_4	24000	2800	1680	CCI	Rx	98
cci_b_5	21600	2400	1600	CCI	Rx	114
cci_b_6	21600	3429	1959	CCI	Rx	106
cci_b_7	4800	3429	1959	CCI	Rx	114
cci_b_8	7200	3429	1959	CCI	Rx	74
cci_b_9	9600	3429	1959	CCI	Rx	98
d_1_ab	19200	?	?	Author's Home	Tx/Rx	
d_n4_ab	4800	?	?	Author's Home	Tx/Rx	
d_n6_ab	9600	?	?	Author's Home	Tx/Rx	
d_n7_ab	12000	?	?	Author's Home	Tx/Rx	
d_n8_ab	14400	?	?	Author's Home	Tx/Rx	
d_n9_ab	16800	?	?	Author's Home	Tx/Rx	
u_ab	21600	?	?	UofA	Tx/Rx	
u_n10_ab	19200	?	?	UofA	Tx/Rx	
u_n11_ab	21600	?	?	UofA	Tx/Rx	
u_n4_ab	4800	?	?	UofA	Tx/Rx	
u_n5_ab	7200	?	?	UofA	Tx/Rx	
u_n6_ab	9600	?	?	UofA	Tx/Rx	
u_n7_ab	12000	?	?	UofA	Tx/Rx	

Table 54: V.34 data files.

File Name	Connect Speed (bps)	Symbol Rate (sym/s)	Carrier Freq. (Hz)	Call Destination	Monitor Port	Length (sec)
u_n8_ab	14400	?	?	UofA	Tx/Rx	
u_n9_ab	16800	?	?	UofA	Tx/Rx	

Table 55: Description of facsimile files recorded.

File Name	Connect Speed	Call Orig.	Call Dest.	Mon. Port	Length (sec)	No. Pages	Correct Result
deep_1	2400	Author's Home	TELUS Lab	Rx	556	4	V.27 24
deep_2	4800	Author's Home	TELUS Lab	Rx	311	4	V.27 48
deep_3	9600	Author's Home	TELUS Lab	Rx	441	9	V.29
deep_4	14400	Author's Home	TELUS Lab	Rx	319	9	V.32
deep_5	14400	TELUS Lab	Author's Home	Tx	74	2	V.32
deep_6	14400	TELUS Lab	Author's Home	Tx	155	5	V.32
fax_n3aa	ERROR	ERROR	ERROR	-	ERROR	ERROR	ERROR
fax_n3ab	ERROR	ERROR	ERROR	-	ERROR	ERROR	ERROR
fax_n3_a	ERROR	ERROR	ERROR	-	ERROR	ERROR	ERROR
fax_n4_a	4800	TELUS Lab	Author's Home	Tx	188	3	V.27 48
fax_n5_a	7200	TELUS Lab	Author's Home	Tx	139	3	V.29
fax_n6_a	9600	TELUS Lab	Author's Home	Tx	123	3	V.29
fax_n7_a	12000	TELUS Lab	TR*Labs*	Tx	106	3	V.32
fax_n8_a	14400	TELUS Lab	TR*Labs*	Tx	98	3	V.32
hp_1	14400	HP Fax	TELUS Lab	Rx	229	6	V.32
hp_10	14400	HP Fax	TELUS Lab	Rx	131	3	V.32

Table 55: Description of facsimile files recorded.

File Name	Connect Speed	Call Orig.	Call Dest.	Mon. Port	Length (sec)	No. Pages	Correct Result
hp_11	14400	HP Fax	TELUS Lab	Rx	474	10	V.32
hp_12	4800	HP Fax	TELUS Lab	Rx	343	3	V.27 48
hp_2	ERROR	ERROR	ERROR	-	ERROR	ERROR	ERROR
hp_3	9600	HP Fax	TELUS Lab	Rx	302	6	V.29
hp_4	4800	HP Fax	TELUS Lab	Rx	417	3	V.27 48
hp_5	2400	HP Fax	TELUS Lab	Rx	923	2	V.27 24
hp_6	2400	HP Fax	TELUS Lab	Rx	768	1	V.27 24
hp_7	2400	HP Fax	TELUS Lab	Rx	556	2	V.27 24
hp_8	4800	HP Fax	TELUS Lab	Rx	343	3	V.27 48
hp_9	9600	HP Fax	TELUS Lab	Rx	188	3	V.29
hp_n3_a1	2400	HP Fax	TELUS Lab	Tx	466	2	V.27 24
hp_n4_a1	4800	HP Fax	TELUS Lab	Tx	294	2	V.27 48
hp_n6_a1	9600	HP Fax	TELUS Lab	Tx	523	10	V.29
hp_n8_b1	9600	HP Fax	TELUS Lab	Tx	498	6	V.29
intel_1	9600	Intel Fax	TELUS Lab	Rx	760	15	V.29
intel_2	14400	Intel Fax	TELUS Lab	Rx	131	3	V.32
intel_3	4800	Intel Fax	TELUS Lab	Rx	311	3	V.27 48
intel_4	2400	Intel Fax	TELUS Lab	Rx	580	3	V.27 24
intel_5	4800	Intel Fax	TELUS Lab	Rx	204	2	V.27 48
tr_1	2400	TR*Labs*	TELUS Lab	Rx	139	2	V.27 24
tr_2	4800	TR*Labs*	TELUS Lab	Rx	98	2	V.27 48
tr_3	4800	TR*Labs*	TELUS Lab	Rx	188	4	V.27 48
tr_4	4800	TR*Labs*	TELUS Lab	Rx	188	4	V.27 48
tr_5	14400	TR*Labs*	TELUS Lab	Rx	180	10	V.32

Table 56: Description of other files recorded.

File Name	Call Orig.	Call Dest.	Mon. Port	Correct Result
c_mess	TR*Labs*	St. Albert	Tx	DTMF digits: 9, 8, 5, 7, 1, 0, 0, 0, 0, 0
DTMF_1	TR*Labs*	St. Albert	Tx	DTMF digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, *, 0, #
dt_0	TR*Labs*	St. Albert	Tx	DTMF digit: 0
dt_1	TR*Labs*	St. Albert	Tx	DTMF digit: 1
dt_2	TR*Labs*	St. Albert	Tx	DTMF digit: 2
dt_3	TR*Labs*	St. Albert	Tx	DTMF digit: 3
dt_4	TR*Labs*	St. Albert	Tx	DTMF digit: 4
dt_5	TR*Labs*	St. Albert	Tx	DTMF digit: 5
dt_6	TR*Labs*	St. Albert	Tx	DTMF digit: 6
dt_7	TR*Labs*	St. Albert	Tx	DTMF digit: 7
dt_8	TR*Labs*	St. Albert	Tx	DTMF digit: 8
dt_9	TR*Labs*	St. Albert	Tx	DTMF digit: 9
dt_pound	TR*Labs*	St. Albert	Tx	DTMF digit: pound
dt_star	TR*Labs*	St. Albert	Tx	DTMF digit: star
mimic_1	TR*Labs*	St. Albert	Tx	Tape Recording of 14.4 V.32 modem
mimic_2	TR*Labs*	St. Albert	Tx	Tape Recording of 26.4 V.34 modem
ring1	TELUS Lab	463-4099 (Millwoods)	Rx	Ringback
ring2	TELUS Lab	458-3874 (St. Albert)	Rx	Ringback
ring3	TELUS Lab	437-6373 (Riverbend)	Rx	Ringback
ring4	TELUS Lab	452-4567 (West End)	Rx	Ringback
ring5	TELUS Lab	988-6444 (Riverbend)	Rx	Ringback
voice_1	TR*Labs*	TELUS Lab	Rx	Male Speech, Early 30's, Cantonese
voice_2	TR*Labs*	TELUS Lab	Rx	Male Speech, 24, French
voice_3	TR*Labs*	TELUS Lab	Rx	Male Speech, Early 30's, Mandarin

Table 56: Description of other files recorded.

File Name	Call Orig.	Call Dest.	Mon. Port	Correct Result
voice_4	Riverbend (437-6373)	TELUS Lab	Rx	Female, Early 20's, English
voice_5	Riverbend (437-6373)	TELUS Lab	Rx	Female, Late 40's, Fiji Hindi
voice_6	TR*Labs*	TELUS Lab	Rx	Female, Early 30's, English
voice_7	TR*Labs*	TELUS Lab	Rx	Male - Male, Late 20's, English Conversation
voice_8	TR*Labs*	TELUS Lab	Rx	Male - Male, Late 20's, English Conversation

D.3 Description of Database files Recorded by Sarda

The files listed in Table 57 were recorded during the second field trial at the TELUS Toll Building. These files do not contain the actual PCM μ -law encoded samples, instead they contain the classification vectors returned by the DSP.

Table 57: Description of database files recorded at the Telus Toll Building.

Filename (.dbf)	Start Time/Date	Stop Time/Date	Segment Size	Monitor Point	Comments
04_20_98	April 20 16:08	April 21 08:42	1 020	Bay 2954 EDTN 05T, 11th floor Port No. 3A	
day2_1	April 21 10:03	April 21 10:38	1 020	Bay 2954 EDTN 05T, 11th floor Port No. 3A	
04_22_98	April 21 11:27	April 22 11:38	1 020	Bay 2953 EDTN 07T Port No. 12A	
04_23_98	April 22 12:52	April 23 08:47	2 052	Bay 2954 EDTN05T Port No. 3A	

Table 57: Description of database files recorded at the Telus Toll Building.

Filename (.dbf)	Start Time/Date	Stop Time/Date	Segment Size	Monitor Point	Comments
Binary1	April 23	April 23	1 020	Bay 2954 Stratacom, 623-650 Point No. 2B	Digital Switch
04_24_98	April 23 11:43	April 24 13:28	1 020	Bay 2954 EDTN 05T, 11th floor Port No. 4	
04_27_98	April 24 14:33	April 27 8:41	1 020	Bay 2953 EDTN 07T Point 9B	
04_28_98	April 27 9:04	April 28 9:06	4 092	Bay 2953 EDTN 07T Point 9B	
04_29_98	April 28 10:05	April 29 9:56	1 020	Bay 2221 225-252 Point 19B	Long Distance, Toronto
04_30_98	April 29 10:06	April 30 10:14	8 196	Bay 2221 225-252 Point 19B	Long Distance, Toronto
05_01_98	April 30 10:36	May 1 10:39	1 020	Bay 2955 STAL 1ST01, 533-546, LEDC 4ST01, 547-560 Point 16B	Traffic going to Leduc
05_04_98	May 1 11:11	May 4 11:14	2 052	Bay 2955 STAL 1ST01, 533-546, LEDC 4ST01, 547-560 Point 16B	Traffic going to Leduc
05_05_98	May 4 11:35	May 5 11:50	16 380	Bay 2221 281-308 Point 23A	Long distance Vancouver
05_06_98	May 5 11:57	May 6 12:03	2 052	Bay 2221 281-308 Point 23A	Long distance Vancouver

Table 57: Description of database files recorded at the Telus Toll Building.

Filename (.dbf)	Start Time/Date	Stop Time/Date	Segment Size	Monitor Point	Comments
05_07_98	May 6 12:18	May 7 13:06	4 092	Bay 2953 EDTN 071, 501-550 Point 1B	
05_08_98	May 7 13:25	May 8 9:46	1 020	Bay 2954 LEND 8T3, 201-224, LEND 9T3, 225 Point 8B	Local
05_11_98	May 8 11:12	May 11 10:19	8 196	Bay 2952 601 Point No. 17	Long distance Calgary

The files listed in Table 58 were recorded during the third and fourth field trials at the Bonnie Doon TELUS Mobility base station in Edmonton. These files do not contain the actual PCM μ-law encoded samples, instead they contain the classification vectors returned by the DSP.

Table 58: Description of database files recorded at the TELUS Mobility base station.

Filename (.dbf)	Start Time/Date	Stop Time/Date	Segment Size	Monitor Point	Comments
10_21_98	Oct 19 10:22	Oct 21 9:33	2052	Bay DSX 1-6 Rack 2 Point No. 17 (left side)	- T1 to Quigley, Alberta - Speech Filter: 3 sec. - Non-Speech Filter: 1 sec.
11_20_98	Nov. 20 9:15	Nov. 30 13:48	2052	Bay DSX 1-4 Row 3A Point No. 5	- T1 to Peace River, Alberta - Speech Filter: 10 sec. - Non-Speech Filter: 1 sec.

Tables 59 to 62 list the files used from the Multilingual Speech Database obtained from CSLU. All filenames are suffixed with ".story.wav". Each file is approximately 1 minute in length.

Table 59: Japanese data files used from the CSLU multilingual speech database.

Filename	Gender
JA-10	Female
JA-30	Male
JA-58	Female
JA-70	Male
JA-91	Female
JA-104	Male
JA-155	Female
JA-199	Male
JA-220	Female
JA-233	Male

Table 60: English data files used from the CSLU multilingual speech database.

Filename	Gender
EN-16	Female
EN-21	Male
EN-38	Female
EN-44	Male
EN-51	Female
EN-60	Male
EN-80	Female
EN-77	Male
EN-96	Female
EN-101	Male

Table 61: French data files used from the CSLU multilingual speech database.

Filename	Gender
FR-6	Female
FR-19	Male
FR-35	Female
FR-46	Male
FR-50	Female
FR-81	Male
FR-91	Female
FR-104	Male
FR-114	Female
FR-126	Male

Table 62: German data files used from the CSLU multilingual speech database.

Filename	Gender
GE-10	Male
GE-224	Female
GE-242	Male
GE-278	Female
GE-290	Male
GE-314	Female
GE-397	Male
GE-426	Female
GE-462	Male
GE-357	Female

Appendix E

Parameters used for experiments using the ALN method:

- Minweight: -10 000
- Maxweight: 10 000
- Input Epsilon: 0.001
- Ouput Epsilon: 0.2
- Jitter: True
- Learn Rate: 0.3
- Min Rmse: 0.001
- Epochs: 14
- Random Seed: 238

The train file should be named "1_all.txt", and the test file should be named "2_all.txt". Each file should be formatted so that the feature and class variables are all on one row separated by tab characters. The class needs to be the last column in each row. Also, any row that begins with a ";" character is ignored. All parameters are read in as command line arguments. To get the syntax simply type in the name of the executable file and the syntax will be displayed on the screen.

Appendix F

This appendix lists all combinations of variables that give the maximum classification accuracy using pseudo quadratic discriminant functions for 6, 7, 8, and 9 variables. Only these are listed because multiple combinations of variables provide the maximum classification accuracy.

Table 63: Maximum classification accuracy using 6 variables (N=2052, Pseudo QDF).

Variables	Number of Variables Used	
	6	6
Rd1	X	X
Rd2	X	X
Rd3	X	X
Rd4		
Rd5	X	X
Rd6	X	X
Rd7	X	
Rd8		X
Rd9		
Rd10		
N2		
Pc (%)	100.0	100.0

Table 64: Maximum classification accuracy using 7 variables (N=2052, Pseudo QDF).

Variables	Number of Variables Used			
	7	7	7	7
Rd1	X	X	X	X
Rd2	X	X	X	X
Rd3	X	X	X	X
Rd4				
Rd5	X	X	X	X
Rd6	X	X	X	X
Rd7	X	X	X	

Table 64: Maximum classification accuracy using 7 variables (N=2052, Pseudo QDF).

Variables	Number of Variables Used			
	7	7	7	7
Rd8	X			X
Rd9				
Rd10		X		
N2			X	X
Pc (%)	100.0	100.0	100.0	100.0

Table 65: Maximum classification accuracy using 8 variables (N=2052, Pseudo QDF).

Variables	Number of Variables Used			
	8	8	8	8
Rd1	X	X	X	X
Rd2	X	X	X	X
Rd3	X	X	X	X
Rd4				
Rd5	X	X	X	X
Rd6	X	X	X	X
Rd7	X	X	X	
Rd8	X	X		X
Rd9				
Rd10	X		X	X
N2		X	X	X
Pc (%)	100.0	100.0	100.0	100.0

Table 66: Maximum classification accuracy using 9 variables (N=2052, Pseudo QDF).

Variables	Number of Variables Used											
	9	9	9	9	9	9	9	9	9	9	9	9
Rd1	X	X	X	X	X	X	X	X	X	X	X	X
Rd2	X	X	X	X	X	X	X	X	X	X	X	X
Rd3	X	X	X	X	X	X	X	X	X			
Rd4	X	X	X	X	X	X	X			X	X	X
Rd5	X	X	X	X				X	X	X	X	X
Rd6	X				X	X	X	X	X	X	X	X

Table 66: Maximum classification accuracy using 9 variables (N=2052, Pseudo QDF).

Variables	Number of Variables Used											
	9	9	9	9	9	9	9	9	9	9	9	9
Rd7	X	X	X		X	X	X	X	X	X	X	
Rd8		X	X	X	X	X	X	X	X	X	X	X
Rd9	X	X		X	X	X		X		X		X
Rd10	X	X	X	X	X		X	X	X	X	X	X
N2			X	X		X	X		X		X	X
Pc (%)	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99