**FADING CHARACTERISTICS OF THE LAND
MOBILE RADIO CHANNEL AT 820 MHz**

by

T. Anthony Knight, B.A.Sc.

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

Faculty of Engineering
Department of Systems and Computer Engineering
Carleton University

Ottawa, Ontario

May 1985

© copyright

1985, T.A. Knight

1

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Services des thèses canadiennes

Ottawa, Canada
K1A 0N4

CANADIAN THESES

THÈSES CANADIENNES

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

**THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED**

**LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE**

National Library
of CanadaBibliothèque nationale
du CanadaCanadian Theses
on Microfiche ServiceServices des thèses canadiennes
sur microfichesOttawa, Canada
K1A 0N4

PERMISSION TO MICROFILM — AUTORISATION DE MICROFILMER

• Please print or type — Écrire en lettres moullées ou dactylographier

Full Name of Author — Nom complet de l'auteur

THOMAS ANTHONY KNIGHT

Date of Birth — Date de naissance

10 - NOV - 57

Country of Birth — Lieu de naissance

CANADA

Permanent Address — Résidence fixe

2475 ROSEBERY AVE
WEST VANCOUVER, BC
V7V 2Z8

Title of Thesis — Titre de la thèse

"FADING CHARACTERISTICS OF THE LAND
MOBILE RADIO CHANNEL AT 820 MHZ"

University — Université

CARLETON UNIVERSITY

Degree for which thesis was presented — Grade pour lequel cette thèse fut présentée

M. ENG.

Year this degree conferred — Année d'obtention de ce grade

1985

Name of Supervisor — Nom du directeur de thèse

H. M. HAFEZ

Permission is hereby granted to the NATIONAL LIBRARY OF CANADA to microfilm this thesis and to lend or sell copies of the film.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

L'autorisation est, par la présente, accordée à la BIBLIOTHÈQUE NATIONALE DU CANADA de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans l'autorisation écrite de l'auteur.

Date

JULY 4/85

Signature

**FADING CHARACTERISTICS OF THE LAND
MOBILE RADIO CHANNEL AT 820 MHZ**

by

T. Anthony Knight, B.A.Sc.

A thesis submitted to the
Faculty of Graduate Studies and Research
in partial fulfilment of the requirements
for the degree of
Master of Engineering

Faculty of Engineering
Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario
May 1985

© copyright
1985, T.A. Knight

The undersigned hereby recommend to the Faculty of
Graduate Studies and Research, acceptance of the thesis

"FADING CHARACTERISTICS OF THE LAND MOBILE

RADIO CHANNEL AT 820 MHZ"

submitted by T. Anthony Knight, B.Sc. in partial
fulfilment of the requirements for the degree of Master of
Engineering.

Thesis Supervisor

Chairman,
Department of Systems
and Computer Engineering

Department of Systems and Computer Engineering

Faculty of Engineering

Carleton University

July 1985

ABSTRACT

The signal strength at a mobile receiver is a spatial random process, characterized by two principal components: variation in mean level due to shadowing, and fast fading due to multipath propagation. Random fading of the 800 MHz land mobile channel is studied, based on field measurements collected in the city of Ottawa, Canada.

The signal envelope was recorded in areas representing urban, suburban and highway terrain, and then resampled to remove the effects of variable vehicle speed. Statistics of the spatial shadowing and multipath fading processes were then estimated. New experimental evidence concerning the spatial correlation of multipath fading is found, which confirms existing theoretical results. Multipath, or fast, fading is approximately Rician in distribution, with parameters that depend on the type of terrain. Variation in the local mean due to shadowing is approximately lognormally distributed, with a standard deviation of 3 to 4 dB.

ACKNOWLEDGEMENTS

I want to thank my thesis supervisor, Professor H.M. Hafez, for his support and guidance of this research. As well, thanks go to Professor A.U.H. Sheikh for his valuable advice and contributions.

I especially want to thank Diane Dodds for her work in word processing the manuscript from my handwritten copy, her patience in making numerous changes, and finally her discipline, without which, the changes would have continued forever. I wish to thank Janice Miller of the Carleton English Department for her review of the manuscript, and Dr. R. Gordon Mouldey and family for their support.

I acknowledge the financial support received from the Natural Sciences and Engineering Research Council of Canada.

TABLE OF CONTENTS

Abstract	1
Acknowledgements	11
Table of Contents	111
List of Figures	v
List of Acronyms	vii
Mathematical Notation	viii
Chapter 1 INTRODUCTION	1
1.1 Background	1
1.2 Land Mobile Radio Channel Characteristics	3
1.3 Synopsis of Thesis Research	7
1.4 Thesis Organization	10
Chapter 2 ANALOG CHANNEL FADING CHARACTERISTICS	11
2.1 Introduction	11
2.2 Multipath Propagation	15
2.3 Local Mean Estimation	19
2.4 Envelope Probability Density Functions	23
2.5 Envelope Autocovariance Function	28
Chapter 3 DATA COLLECTION AND PREPROCESSING	33
3.1 Introduction	33
3.2 Data Acquisition	36
3.3 Bit Error Rates	40
3.4 Envelope Calibration	44
3.5 Equidistant Resampling	46
Chapter 4 STATISTICAL DATA ANALYSIS	53
4.1 Introduction	53
4.2 Estimation of Local Mean	56
4.3 Mean and Standard Deviation	61
4.4 Sample Probability Density Functions	63
4.5 Autocovariance Function	71
4.6 Power Spectral Density	81
4.7 Prediction of Area Mean	84

Chapter 5	CONCLUSIONS	88
5.1	Discussion of Results	88
5.2	Applications	93
5.3	Suggestions for Further Work	96
REFERENCES		99
APPENDIX A		
	DATA ACQUISITION HARDWARE	A-1
APPENDIX B		
	ENVELOPE CALIBRATION POINTS	B-1
APPENDIX C		
	RESAMPLING LPF FREQUENCY RESPONSES	C-1
APPENDIX D		
	PROGRAM LISTINGS	D-1
APPENDIX E		
	FADING STATISTICS BY AREA	E-1

LIST OF FIGURES

Figure		Page
1.2-1	The mobile radio communication channel	4
2.1-1	On-off channel model	14
2.1-2	Two-state Markov BSC model or "Gilbert model", after [17]	14
2.1-3	Time-varying BSC and probability of error, after Aulin [17]	14
2.1-4	Analog channel fading simulator	16
2.2-1	Multipath propagation	17
2.2-2	Field strength fluctuation with one reflected wave	17
2.3-1	Typical signal envelope with varying local mean	20
2.4-1	Standard deviation of local mean by various authors, after [19]	24
2.4-2	Unit-mean Rayleigh probability density function	26
2.4-3	Unit-mean Rician probability density functions	26
2.5-1	Autocovariance function of magnitude of vertical electric field component	30
2.5-2	Estimates of time-correlation of field strength by Aulin [17]	30
3.1-1	Base station transmitter equipment	34
3.1-2	Mobile receiver equipment	34
3.1-3	Areas of data collection in the Ottawa-Hull region	37
3.1-4	Description of areas covered	38
3.2-1	Data acquisition equipment	39
3.3-1	Typical received envelope, bit errors and speed pulse	42
3.3-2	Average bit-error-rate by area	43
3.4-1	Signal envelope voltage path	43
3.4-2	Piecewise-linear approximation of RF signal level to ADC output	45
3.5-1	Illustration of equi-distant resampling	49
3.5-2	Resampling low-pass filter specifications	49
3.5-3	Resampled envelope data files	51
3.5-4	Typical resampled data sequences (Spatial signal strength)	52
4.1-1	Steps of statistical analysis	54
4.2-1	Frequency response of 99 point $\cos^2$ window	58
4.2-2	Local mean low-pass filter characteristics	58

4.2-3	Typical local mean and fast fading components (urban Ottawa - 99 point filter)	59
4.2-4	Local mean estimates using various length lowpass filters	60
4.4-1	Sample probability density functions (lowpass filter N=99)	65
4.4-2	Examples of typical curve-fitting results: Urban Ottawa fast fading data	68
4.4-3	Curve fitting envelope and fast fading data by area	69
4.4-4	Optimum Rician curve fitting parameters for normalized fast fading component	70
4.5-1	Typical autocovariance estimates for urban envelope and components (downtown Ottawa)	74
4.5-2	Autocovariance functions of local mean for four urban Ottawa data files	76
4.5-3	Zero-padded FFT interpolation. (16:1 time interpolation)	77
4.5-4	Autocovariance of fast fading in urban Ottawa for several lowpass filter lengths	79
4.5-5	Autocovariance function of fast fading by area	80
4.6-1	Power spectral density of urban fading envelope	83
4.6-2	Typical local mean filter cutoff characteristics	83
4.7-1	Predicted path loss and received signal strength	86
A-1	Data acquisition hardware block diagram	A-3

LIST OF ACRONYMS

ACF	autocovariance function
ACSSB	amplitude companded single side-band
AGC	automatic gain control
BER	bit error rate
BPS	bits per second
BSC	binary symmetric channel
DOC	(Canadian Federal) Department of Communications
FFT	fast Fourier transform
FIR	finite impulse response
FM	frequency modulation
IPS	inches per second
LPF	low-pass filter
NBFM	narrow-band frequency modulation
PDF	probability density function
PSD	power spectral density
RAM	random access memory
RC	resistive-capacitance
SINAD	signal to interference, noise and distortion (ratio)

MATHEMATICAL NOTATION

t	continuous time variable
x	continuous space variable
λ	wavelength
N	number of samples
μ	mean value
σ	standard deviation
ξ	spatial offset in correlation/covariance function
h_i	filter coefficients
d_c	spatial cutoff period of lowpass filter
$\hat{y}$	estimate of value/function y
$\bar{y}$	mean value of y
$v(t)$	vehicle speed as a function of time
$r(x)$	spatial signal envelope
$m(x)$	local mean component
$r_o(x)$	fast fading component
$K_r(\xi)$	spatial autocovariance function of $r(x)$
$R_r(\xi)$	spatial correlation function of $r(x)$
$I_0(x)$	modified Bessel function of zeroth order
$J_0(x)$	ordinary Bessel function of zeroth order
$\Gamma(x)$	Gamma function

Chapter 1

INTRODUCTION

1.1 Background

Many new mobile communication systems are now being introduced to serve a growing demand for land mobile voice and data communications. One area of rapid growth is cellular mobile radio telephone systems, which are now operational in some cities in Europe, North America and Japan. In 1984, it was estimated that, worldwide, there were 100,000 users of cellular radio. Projections suggest that this figure will increase to three million users by 1990, with 50% of the users located in North America [1]. Many of these new systems will operate at frequencies in the "900 MHz" band (825-890 MHz). Advances made in RF technology and the allocation of the 900 MHz band for mobile telephony have made the use of higher frequencies practical, relieving the demand for channels in the now congested VHF (150 MHz) and low UHF (450 MHz) bands [2].

Despite many improvements in the design of mobile radio systems, further research is needed to cope with impairments of the analog mobile radio channel. Compared with fixed, point-to-point channels, the mobile radio channel is subject to serious problems of fading, noise and interference. Advances in VLSI technology have made

which compares the performance of the three configurations, with some views on the directions for future digital mobile radio regulations.

The field data was recorded on 7-track analog FM tapes in six areas of the Ottawa-Hull region, representing various urban and suburban terrain types. The seven parallel tracks consist of the demodulated signal, received data and clock, stable reference data clock, signal envelope, 20-second start/stop signal, and the vehicle speed. Four tapes recorded in the 1983 field experiment were selected for our research. They contain envelope data for base-to-mobile transmission at 820 MHz. These tapes were chosen because they yield the most interesting results in terms of future mobile radio developments. Two of the tapes represent urban terrain: downtown Ottawa and Hull. One tape was recorded on the western portion of the Queensway, representing typical open, highway terrain, and one tape was recorded in typical suburban residential terrain (South Ottawa). Each tape contains about six minutes of alternating 20-second intervals of data transmission with 20-second intervals of unmodulated carrier. About 18.8 seconds of each 20-second interval of data transmission was sampled and stored in a data file on the computer. A total of 28 files or five megabytes was stored, representing 8.8 minutes of real-time data.

The equipment used for data acquisition and processing consisted of a desktop microcomputer, the

it possible to design compact mobile radio telephones that incorporate many features found in land-line telephones, but progress has been less spectacular in other aspects such as the type of modulation used. Frequency and phase modulation remain preferred for mobile radio use because they are relatively immune to noise and amplitude disruptions. FM with ± 12 KHz deviation was selected as a standard for cellular mobile radio to yield speech quality similar to that of land-line systems [3]. The chief disadvantage of FM is that, compared to AM and other narrow-band schemes, a larger bandwidth is required. Other modulation schemes that are more spectrum efficient than FM include narrow-band FM (NBFM) and amplitude-companded single side-band (ACSSB). Research continues into the use of these various modulation schemes, but to date, frequency/phase modulation has dominated commercial mobile radio systems.

An understanding of the nature of the mobile radio channel is necessary in the development of efficient mobile radio systems. The radio channel typically includes all disruptions occurring between modulation in the transmitter and demodulation in the receiver. The channel is the one aspect of any communications system over which the system designer has least control. The properties of the channel essentially dictate the design of all other aspects of the system. Theoretical study and experimental results have led to the development of models of the land mobile radio channel, which have then been

used in the design of software and hardware simulators to reproduce field conditions in the laboratory. The results presented in this thesis contribute to further understanding of the nature of the mobile radio channel at 900 MHz, a frequency band where the continued development of new systems is expected.

1.2 Land Mobile Radio Channel Characteristics

The nature of the mobile radio channel has been under study for years. A considerable body of literature describing signal propagation and reception in the mobile radio environment now exists. Figure 1.2-1 shows the principal aspects of any mobile communications system: the radio channel, modulation, and source and channel coding. The disruptions over the radio channel, and the desired system performance, essentially determine the design of modulation and coding schemes used. Disruptions of the radio channel include signal fading, noise, interference, and random FM. Ignition noise from vehicles has been cited as a particular source of bit errors in digital systems. In one experiment, French determined that ignition noise contributed to approximately 20% of the total bit errors in the transmission of 4800 bps at 462 MHz [16]. Ignition noise can be controlled by the use of high quality narrow-band filters in the receiver, and by proper shielding. Interference in mobile radio may originate from other transmitters operating on the same

Figure 1.2-1 The mobile radio communication channel.

frequency (co-channel) or on adjacent channels. The large number of allocated channels and channel reuse in cells make interference problems especially acute in cellular mobile radio systems. Solutions to the problems of interference range from proper equipment and system design, to responsible spectrum management and frequency allocation by regulatory agencies.

Signal fading is probably the most serious impairment of the mobile radio channel. Fading occurs when the mobile receiver moves through a stationary field strength pattern which varies randomly with position. Large variations in signal strength require radio receivers with wide dynamic range and automatic gain control (AGC). Even conventional AGC can only track changes in average signal level, and cannot compensate for all fading. Inevitably, some speech and data will be lost when the signal level drops below the input sensitivity of the receiver. There are many techniques for combatting the effects of signal fading. One method that reduces the effects of fading without an increase in transmitted power or bandwidth is space diversity. This method employs two antennas spaced apart such that the random fading processes at each antenna are sufficiently decorrelated with each other, making the probability of simultaneous fades low. This method does not reduce the information capacity of the channel, but requires extra hardware in the form of antennas and combining circuits. In digital

communications, the entire subject of information coding is applied to combat the error bursts that occur in fades.

The subject of signal fading has been widely studied in the literature. The theoretical basis for signal fading was understood many years ago, and many subsequent experiments have confirmed these early theoretical results. Experimental results have also yielded empirical models for the prediction of mean signal strength and fading statistics. Nevertheless, there are still a number of aspects of signal fading that have not been adequately covered in the open literature. There has been little published work on aspects such as the spatial correlation of the random fading process, and the behaviour of the probability density function of fading with the type of terrain.

The results of this thesis aim to fully characterize the random fading process of the 820 MHz mobile radio channel. In the experiments, both the signal envelope and the vehicular speed were recorded simultaneously in urban, suburban and highway terrains. In subsequent processing, the signal envelope was resampled to remove the effects of fluctuating speed, and then statistics of the spatial fluctuations were extracted. Correlation of the random fading process was studied by computing estimates of the autocovariance function of the envelope. The results provide new, experimental evidence confirming theoretical results about the correlation of the fading envelope.

The principal application of the results in this thesis is in the area of channel modelling and simulation. The results here add to existing empirical knowledge of the channel fading characteristics, which may be used in subsequent channel simulations. Details of the signal processing and statistical analysis of the envelope may find use in future experiments. The autocovariance function may be used in a receiver to predict the envelope fluctuation, and adaptively correct for it.

1.3 Synopsis of Thesis Research

The experimental data used in the research for this thesis was collected in a 1983 study of digital land mobile systems. The work was done at Carleton University under contract to the Federal Department of Communications. The experiment evaluated three different modulation schemes in the 150, 450 and 850 MHz bands in terms of bit error rates, equipment complexity, compatibility with current systems and spectrum usage. The results of the experiment are reported in "Experimental Study of Digital Land Mobile Systems", prepared for the DOC [21]. In this report, the theory of signal propagation and channel characteristics are reviewed, and some analog channel characteristics including probability density function, fade duration, and level crossing rate are derived from the field measurements. The DOC report concludes with a discussion

which compares the performance of the three configurations, with some views on the directions for future digital mobile radio regulations.

The field data was recorded on 7-track analog FM tapes in six areas of the Ottawa-Hull region, representing various urban and suburban terrain types. The seven parallel tracks consist of the demodulated signal, received data and clock, stable reference data clock, signal envelope, 20-second start/stop signal, and the vehicle speed. Four tapes recorded in the 1983 field experiment were selected for our research. They contain envelope data for base-to-mobile transmission at 820 MHz. These tapes were chosen because they yield the most interesting results in terms of future mobile radio developments. Two of the tapes represent urban terrain: downtown Ottawa and Hull. One tape was recorded on the western portion of the Queensway, representing typical open, highway terrain, and one tape was recorded in typical suburban residential terrain (South Ottawa). Each tape contains about six minutes of alternating 20-second intervals of data transmission with 20-second intervals of unmodulated carrier. About 18.8 seconds of each 20-second interval of data transmission was sampled and stored in a data file on the computer. A total of 28 files or five megabytes was stored, representing 8.8 minutes of real-time data.

The equipment used for data acquisition and processing consisted of a desktop microcomputer, the

'Texas Instruments Professional Computer' or TI PC, with a 10-megabyte hard disk, 8087 numeric co-processor, parallel dot matrix printer and video graphics board. A commercial 12-bit analog-to-digital converter, the 'Tecmar Lab Master', was used to sample the envelope signal, and some custom hardware was built to interface other digital control signals to the computer.

Programs to filter the data and extract statistics were developed in Pascal, in a short single-function style. This simplified the task of program development. Several versions of a program were easily maintained. The most complicated program is RESAMP, used to filter and resample the data to equi-distant intervals. Most programs operate on a data file specified at run-time, typically taking a few minutes to produce another file of results. An MS-DOS command file was used to run programs repeatedly on several data files. This has the advantage of permitting examination of intermediate results, but increases the number of data files to be stored. Some programs such as the curve fitting routines run interactively and plot data on the screen using the video graphics options. A utility program was used to obtain a hard-copy of the graphics on the printer.

1.4 Thesis Organization

Chapter 2 discusses the analog channel fading characteristics and the methods of statistical analysis used in this work. The theory and models of multipath propagation, shadowing and terrain effects are summarized. Complete details may be found in the references.

Chapter 3 discusses the details of field data collection and preprocessing to obtain equi-distant real-valued numeric representation of the signal strength at the mobile receiver.

Chapter 4 describes the methods and results of statistical analysis. It begins with the method used to estimate the local mean and fast fading components. Statistics of these estimates are then computed, including sample probability density functions, autocovariance function and power spectral density. These are compared with the theoretical results in Chapter 2.

Chapter 5 contains a summary of salient results and conclusions, with suggestions for applications and further work.

Chapter 2

ANALOG CHANNEL FADING CHARACTERISTICS

2.1 Introduction

Fading over the analog urban mobile radio channel arises directly from variations in signal strength with position. Fluctuation in field strength is characterized by three principal components, all having spatial origins. First of these components is the mean signal strength at the receiver which is determined by free-space path loss, plus additional losses due to the geography of the area. In urban areas, the path of propagation may vary widely, ranging from a few hundred metres line-of-sight to a path tens of kilometers, which is obstructed by buildings and terrain. Empirical results reported by Okumura and Hata [23] may be used to predict mean signal strength. The second component is the variation of actual signal strength about this mean level due to individual structures and topographical features of the terrain. Together, these first two components determine a signal level known as the local mean level. The random variations are frequently referred to as lognormal shadowing, since these fluctuations have a probability density function which is approximately lognormal. The third component is 'fast' or Rayleigh fading which

consists of fluctuations on the order of one-half wavelength, due to multipath propagation.

Theoretical results and statistics derived from experiments on fading over the mobile radio channel are widely reported [4-21]. The fading envelope and its components can be characterized by estimates of the mean, variance, probability density function, autocovariance function and power spectral density. In turn, these statistics are used to predict system performance. Local mean statistics may be used to predict probability of coverage in a given area. The received envelope is easily measured in any receiver following the linear gain of RF and IF stages, but before any detection or demodulation, with any AGC circuits disabled.

Characterization of the fading envelope remains an important aspect of the study of digital mobile channel performance. Models of the random digital error process can be derived from the behaviour of the analog fading process. Hansen followed this method, modelling the channel as on or off, according to the envelope level [14] (see Figure 2.1-1). When the envelope is above the receiver error threshold, the probability of bit error is zero. Below this threshold, the channel is interrupted and the bit error rate is 0.5. Hansen combined Rayleigh fading with lognormal shadowing to derive a model which estimates performance with just three parameters: receiver error threshold, local mean signal strength and standard deviation of shadowing. Subsequent field

measurements by French at 465 MHz agreed well with predictions using this model.

The digital channel may also be modelled directly, having several discrete states. The simplest is a two state Markov model of a binary symmetric channel (BSC), known as Gilbert's model (see Figure 2.1-2). Aulin continued by generalizing this model to many states. In the limit, an infinite number of states leads to a time-varying BSC with cross-over probability given by the field strength [17]. Aulin shows that the Markov dependence on the previous state is lost and the channel characteristics are determined entirely by the statistics of the relative field strength F and the probability of bit error $P(F)$ (see Figure 2.1-3). He estimated $P(F)$ from field data, and then used it to simulate digital errors, with reasonable success.

Other researchers have conducted experiments with fading simulators and found results consistent with actual field tests. Daikoku reported such results in tests of new digital modulation schemes for transmission of high speed data (16 k.bits/sec) over the 920 MHz mobile radio channel. Bit error rates measured in the field agreed with those obtained using a Rayleigh fading simulator [19].

The need to further examine the statistics of random fading is the motivation for our research. Statistics such as the mean, variance and probability density

Figure 2.1-1 On-off channel model.

Figure 2.1-2 Two-state Markov BSC model or "Gilbert model", after [17].

Figure 2.1-3 Time-varying BSC and probability of error, after Aulin [17].

Superimposed on the lognormal shadowing is fast fading due to multipath propagation. The probability density function of Rayleigh fading, predicted by the scattering multipath model, is:

$$f_r(x) = \frac{x}{a^2} \exp\left(-\frac{x^2}{2a^2}\right) \quad (7)$$

where x is linear signal strength in volts or similar units, and a = mean value. Figure 2.4-2 illustrates the Rayleigh PDF with unity mean ($a=1$).

Some investigators have compared the measured envelope to a Rayleigh PDF without removing the fluctuating local mean and obtained reasonable fits [10,13,15,16]. In situations where the field at the receiver is dominated by one plane wave such as a line-of-sight component or a strong specular (reflected) component, the signal envelope has a distribution which is Rician with a PDF of [6]:

$$f_r(x) = \frac{x}{b^2} \exp\left(-\frac{(x^2+a^2)}{2b^2}\right) \cdot I_0\left(\frac{ax}{b^2}\right) \quad (8)$$

where $I_0(x)$ = modified Bessel function of zeroth order, for a line-of-sight component $a \cos 2\pi ft$ plus scattered waves with mean power b^2 (see Figure 2.4-3). This expression includes the Rayleigh PDF as a special case when $a=0$.

Experimental envelope density functions have also been compared to the Nakagami PDF [11,17]:

Figure 2.1-4 Analog channel fading simulator.

Figure 2.2-1 Multipath propagation.

Figure 2.2-2 Field strength fluctuation with one reflected wave.

many interfering waves, the signal fading becomes a random process, but is still predominated by nulls spaced approximately one-half wavelength apart. One property of the multipath model is the description of the Rayleigh probability density function of the fading envelope. The fading due to multipath propagation is frequently known as Rayleigh fading even though the statistics may be different from Rayleigh. In this work, we simply refer to fading due to multipathing as fast fading.

With few modifications, this model has remained widely accepted in the literature. It predicts many first and second order statistics which have been confirmed by experiment. Turin was one of the first to report experimental results on Rayleigh fading combined with lognormal fluctuation in local mean [5]. Work by Rice led to an extension of the model in situations where one plane wave, such as a line-of-sight component, dominates the received signal. In this case, the probability density function has been shown to become Rician. Many investigators have reported the statistics of envelope data and compared the distributions with Rayleigh, Rician, Nakagami and other distribution functions [11,17].

Some researchers have studied other aspects of the channel such as the impulse response, but the fading envelope has remained the principal feature studied. In [12], Hashemi modelled the wide-band response of the channel using an equivalent baseband linear filter having

a complex impulse response. This is useful for describing the response to a particular pulse shape, but requires specialized equipment to measure the complex impulse response of the channel. Direct measurements of the dispersive properties of the channel show little dispersion of excess of 7 μ s [5]. Indirect evidence from experiments that use amplitude fading and ignore dispersion [19], indicate that dispersion due to multipath propagation does not contribute significantly to degradation in performance at data rates up to 16 K.bits/sec.

2.3 Local Mean Estimation

Following Lee and Yeh [8], we represent the received signal strength (or envelope) as a stationary Rayleigh process with a slowly varying mean signal level:

$$r(x) = m(x) r_0(x) \quad (1)$$

where $r_0(x)$ = a stationary Rayleigh stochastic process with unit mean

and $m(x)$ = mean value of $r(x)$.

This is illustrated in Figure 2.3-1. The mean value includes signal strength fluctuations due to shadowing and gross variation with terrain, while $r_0(x)$ represents Rayleigh fading due to multipath propagation. Since we can only measure $r(x)$, the mean value and fast fading components must be estimated. If we assume $m(x)$ is a bandlimited process, then an estimate of the local mean,

Figure 2.3-1 Typical signal envelope with varying local mean.

$\hat{m}(x)$, may be obtained by averaging $r(x)$ over some interval [8]:

$$\hat{m}(x) = \frac{1}{2L} \int_{x-L}^{x+L} r(y) dy \quad (2)$$

The interval $2L$ must be sufficiently long to obtain a statistically significant sample space of $r(y)$ but not so long that the mean $m(x)$ changes significantly. In [8], Lee computed the variance of estimate of local mean for various interval lengths from estimation theory. For an interval $2L=40\lambda$, the standard deviation of the estimate is $0.06 \bar{m}$, that is, the estimate is within 1dB of the actual value 86% of the time. Lee suggests that $2L$ could be further increased, but $m(x)$ would become significantly nonconstant over longer intervals.

We may view a moving estimate of the local mean, $\hat{m}(x)$, as a low-pass filtered version of the signal envelope $r(x)$:

$$\hat{m}(x) = \frac{1}{2L} \int_{x-L}^{x+L} r(y) h(y) dy \quad (3)$$

where $h(y)$ is some weighting function.

In discrete samples, this is simply a finite-impulse-response (FIR) low-pass filter:

$$\hat{m}(x_j) = \sum_{i=-N}^N h_i r(x_{j+i}) \quad (4)$$

With $m(x)$ computed, the fast fading component may be

estimated by the ratio:

$$\hat{r}_0(x) = \frac{r(x)}{\hat{m}(x)} \quad (5)$$

Any low-pass filter with stop-band attenuation sufficient to reject all high-frequency multipath fading is acceptable. Choosing the averaging interval, $2L$, mainly involves judging where the transition from multipath fading to shadowing occurs. Multipath or fast fading is basically wavelength dependent, while fluctuations due to shadowing depend on the dimensions of structures and variations in terrain. Most investigators have estimated the local mean over an interval on the order of a hundred wavelengths, or tens of metres at VHF and UHF. Reudink used 10 to 20 metres [9], while French averaged over 42 metres (65λ) [10]. The most rapid fluctuations due to multipath fading have a period of $\lambda/2$ or 0.176 metre at 820 MHz. With many scattered waves incident on the receiver, there will be additional fluctuations in field strength over longer periods, up to 10λ or 4 metres. Averaging intervals in the range of 5 to 10 metres were used for most of our statistical analysis.

The statistics of $\hat{m}(x)$ and $\hat{r}_0(x)$ are both dependent on the degree of averaging. As the filter length is increased, the spatial cutoff frequency decreases, and the estimate $\hat{m}(x)$ becomes smoother. At the same time, the fast fading estimate $\hat{r}_0(x)$ becomes increasingly like the original envelope. Conversely, a shorter filter results

in more fluctuation in $\hat{m}(x)$, and the estimate of $r_0(x)$ remains closer to unity. With less long-term variation in $\hat{r}_0(x)$, we predict that estimates of its statistics will be increasingly consistent.

2.4 Envelope Probability Density Functions

The mean value $m(x)$ includes signal strength fluctuations due to shadowing caused by both man-made structures and natural features in the terrain. In addition, signal strength varies with distance to the transmitter and the overall terrain type. If the distance to the transmitter and the type of terrain remain reasonably constant, then experiments show that the mean value variation has an approximately lognormal probability density function (PDF) given in [5] as:

$$f_m(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp \left(- \frac{(x-\bar{m})^2}{2\sigma^2} \right) \quad (6)$$

The mean value $\bar{m}$ represents the average local mean and may be predicted if transmitter power and path losses are known. The standard deviation is a function of frequency and terrain as shown in Figure 2.4-1, reproduced from Figure 21 of Reudink [9]. Measurements by French [10] and the results of our research have been added to the figure. At 800 MHz, typical values fall within the range 5 to 8 dB, with more variation in suburban and hilly terrain.

Figure 2.4-1 Standard deviation of local mean by various authors, after [9].

Superimposed on the lognormal shadowing is fast fading due to multipath propagation. The probability density function of Rayleigh fading, predicted by the scattering multipath model, is:

$$f_r(x) = \frac{x}{a^2} \exp\left(\frac{-x^2}{2a^2}\right) \quad (7)$$

where x is linear signal strength in volts or similar units, and a = mean value. Figure 2.4-2 illustrates the Rayleigh PDF with unity mean ($a=1$).

Some investigators have compared the measured envelope to a Rayleigh PDF without removing the fluctuating local mean and obtained reasonable fits [10,13,15,16]. In situations where the field at the receiver is dominated by one plane-wave such as a line-of-sight component or a strong specular (reflected) component, the signal envelope has a distribution which is Rician with a PDF of [6]:

$$f_r(x) = \frac{x}{b^2} \exp\left(-\frac{(x^2+a^2)}{2b^2}\right) \cdot I_0\left(\frac{ax}{b^2}\right) \quad (8)$$

where $I_0(x)$ = modified Bessel function of zeroth order, for a line-of-sight component $a \cos 2\pi ft$ plus scattered waves with mean power b^2 (see Figure 2.4-3). This expression includes the Rayleigh PDF as a special case when $a=0$.

Experimental envelope density functions have also been compared to the Nakagami PDF [11,17]:

Figure 2.4-2 Unit-mean Rayleigh probability density function.

Figure 2.4-3 Unit-mean Rician probability density functions.

$$f_r(x) = \frac{2}{\Gamma(a)} \cdot \left(\frac{a}{b^2}\right)^a \cdot x^{2a-1} \cdot \exp\left(\frac{-ax^2}{b^2}\right) \quad (9)$$

where $a \geq 1/2$ and $\Gamma(a)$ = Gamma function. The shape of this PDF is similar to the Rician PDF.

Some researchers have emphasized the importance of matching experimental data histograms to the PDFs above. Using the straight-line method of plotting data distributions against theoretical cumulative distribution functions. Aulin [17] concluded that his data fit the Nakagami distribution best, while Suzuki [11] found that both the Rician and Nakagami PDFs fit his data best. Suzuki noted that these two PDFs fit better than Rayleigh chiefly because they both have two degrees of freedom, while the Rayleigh PDF only has one. Other researchers discount the small differences in curve fitting. Momo [15] notes the unimportance of these differences because, according to him, system performance is determined principally by the low-level "tails" of these distributions, which are all similar. In our work, curve fitting forms an essential part of characterizing the shadowing and fast fading components of the fading process. The finer differences in the residual fast fading PDFs become more apparent with the removal of the nonstationary local mean. For example, it is interesting to compare the line-of-sight to scattered energy ratio of the Rician PDF with the type of terrain.

2.5 Envelope Autocovariance Function

The correlative properties of the signal envelope may be directly related to the spatial correlation of the magnitude of the electric field at the receiving antenna. For example, in a typical system using vertical polarization, the envelope of the received signal at the mobile receiver is proportional to the magnitude of the vertical electric field; $|E_z|$. Either the autocorrelation or the autocovariance function of the electric field may be considered. When considering magnitude only, it is most convenient to first remove the positive mean value of the field strength, then correlate the data to obtain an estimate of the autocovariance function (ACF). Assuming a stationary random process $r(x)$ with a positive mean value $\bar{r}$, the autocovariance is defined as:

$$\begin{aligned} K_r(\xi) &= E[(r(x) - \bar{r})(r(x+\xi) - \bar{r})] \\ &= E[r(x)r(x+\xi)] - \bar{r}^2 \\ &= R_r(\xi) - \bar{r}^2 \end{aligned} \tag{11}$$

Clarke derived the correlation and covariance functions for the three field components E_z , H_x and H_y based on the scattered wave multipath model. In experiments, these field components are not measured directly, but the envelope or squared magnitude are observed. The normalized ACF of the squared magnitude of the vertical field component, $|E_z|^2$, is given by equation (16) of [4]:

$$K_{|Ez|}(\xi) = J_0^2\left(\frac{2\pi\xi}{\lambda}\right) \quad (12)$$

where

$J_0(x)$ = zeroth order Bessel function of the first kind.

This is plotted in Figure 2.5-1.

Zander derived similar theoretical expressions based on a stochastic multipath model with a large number of reflectors distributed in a plane according to a planar Poisson process. The model yields a channel transfer function in terms of frequency and displacement $H(f, r)$ with a single parameter D , the approximate distance between reflectors. The autocovariance of the envelope with frequency and space is then derived, independent of D . Field trials were conducted using a fixed receiver and mobile transmitter. The received signal envelope was measured at 256 frequencies at each of 70 points in a rectangular grid 2.0 x 1.35 m. The correlation with frequency was confirmed, but there was insufficient spatial data to study the correlation with displacement [18].

The only other experimental results concerning the correlation of the field strength process appear in the work of Aulin [17]. He estimated the time correlation function from measured envelope data. Two of his estimates, presented in his paper, are reproduced in Figure 2.5-2. Each estimate was computed over 51100 points (42.5 seconds) of field strength data, chosen where

Figure 2.5-1 Autocovariance function of magnitude of vertical electric field component.

(a) high fade rate (high vehicle speed)

(b) low fade rate (low vehicle speed)

Note: No units specified by Aulin.

Figure 2.5-2 Estimates of time-correlation of field strength by Aulin [17].

the fading was judged to be stationary. No mention is made of the criteria used for his selection. The vehicle speed was not recorded; so the spatial correlation could not be examined, but his results obviously show that the time correlation of the envelope decreases with increasing vehicle speed. In his conclusions, Aulin himself, cites the need for further study of the spatial correlative properties of the random fading process.

In this thesis, the issue of spatial correlation of the fading process is addressed directly. With vehicle speed information recorded with the envelope data, the effects of fluctuating speed were removed and the spatial correlation examined directly. The statistics of the fading envelope are assumed independent of vehicle orientation and direction travelled. The spatial envelope data is treated as essentially one-dimensional although the vehicle actually makes turns following the route in each area. This is reasonable since the basic multipath model assumes no directionality in the interfering plane waves. In one reported field test, however, orientation of the streets with respect to the transmitter had an observed effect on the local mean [20].

An estimate of the autocovariance function of our data is computed using a discrete form of (11):

$$\hat{K}_r(m) = \frac{1}{N} \sum_{n=0}^{N-m-1} (r(n) - \bar{r})(r(n+m) - \bar{r}) \quad (13)$$

where $r(n)$ = discrete equi-distance envelope samples

N = number of samples

and

$$\bar{r} = \frac{1}{N} \sum_{n=0}^{N-1} r(n)$$

In order to compute the estimate $\hat{K}_r(m)$, $r(n)$ is assumed to be wide-sense stationary over the interval N .

In our work, the autocovariance estimates were made over several intervals of data recorded in each area. Statistical variation in the estimates is always expected, even if $r(n)$ is stationary. Our goal was to examine the results visually and look for similarities. Visual consistency is useful as a first step in statistical analysis, but further work is needed to objectively assess qualities such as stationarity.

Chapter 3

DATA COLLECTION AND PREPROCESSING

3.1 Introduction

The signal envelope data analyzed in this thesis was collected in a previous experimental study of digital land mobile systems, conducted in Ottawa in the spring of 1983. The base station equipment used in the experiment is illustrated in Figure 3.1-1. The transmitter was a commercial (General Electric) FM unit rated at 30 watts output power with a 100% duty cycle. The 820 MHz base station antenna was a 2.5 m mast of vertically stacked quarter-wave coaxial elements with a gain of 8 db. The antenna, located on the top of the Carleton University Mackenzie Engineering Building, about 20 m above ground level, was connected to the transmitter with 15 m of 50 Ω RG-213/U cable having a total loss of approximately 2 dB. The area around the transmitter site consists mostly of buildings 2-5 storeys in height, with the exception of the 24-storey Arts Tower to the west, and a 14-storey student residence to the north.

Figure 3.1-2 illustrates the data collection equipment installed in the mobile van. The FM receiver was a commercial (General Electric) transceiver having a noise bandwidth of 11 KHz and a sensitivity of 0.35 μ V for

Figure 3.1-1 Base station transmitter equipment.

Figure 3.1-2 Mobile receiver equipment.

12 dB SINAD. A quarter-wave whip antenna was mounted on the roof of the van, about 3 meters above road level. The received signal envelope was detected at the output of the IF stage in the receiver before the limiter/discriminator by a linear detector having a dynamic range of 30 dB. The signal then passed through a logarithmic amplifier and was recorded on one track of a 7-track tape recorder. The recorded voltage was calibrated using known RF input levels to the receiver. The vehicle speed was monitored by an electromagnetic pick-up which produced a pulse for each revolution of a wheel. This speed pulse recorded on another track of the tape.

The transmitted signal consisted of a 20 second interval of modulated carrier using a repeating 2047-bit pseudo-noise (PN) sequence at 4800 bits/second, alternating with a 20 second interval of unmodulated carrier. A commercial radio modem (Mobile Data International) was used to generate the digitally modulated baseband signal for transmission. A start/stop signal indicated the beginning and end of a 20 second data transmission interval. The data clock and timing signals were generated simultaneously at the transmitter and receiver using synchronized highly stable frequency standards. The demodulated data and clock from the receiver were recorded. As well, the reference clock and start/stop signals derived from the frequency standard were recorded.

Field data at 820 MHz was collected in four regions of the greater Ottawa area representing typical urban, suburban residential and highway terrains. Figure 3.1-3 is a map showing the areas covered and Figure 3.1-4 summarizes some salient features of each area. Except for the Queensway, a looping route was chosen in each area. The aim was to obtain statistical variations in signal strength which are due to fading and shadowing while maintaining a relatively constant terrain type and distance to transmitter. In the case of the Queensway, the mobile travelled in one direction along the highway approaching the transmitter, but the terrain remained relatively constant.

3.2 Data Acquisition

This author's investigation began by sampling the experimental data recorded on tape and storing it on the 10 Mb Winchester disk of the 'TI PC' computer. Figure 3.2-1 illustrates the data-acquisition equipment. The tapes were played back in real-time (30 IPS) and the recorded reference clock provided timing for sampling the 4800 bits/second data. The signal envelope voltage was sampled at the same rate using a commercial 12-bit analog-to-digital converter, having a linear range of -1.0 to +1.0 volt. The recorded voltage varied over 0 to -1.0 volt, using half of the available dynamic range. A spectrum analyzer showed that the spectral power of the envelope above 100 Hz was more than 40 dB below that of

Figure 3.1-3 Areas of data collection in the Ottawa-Hull region.

Area	Terrain type	Average distance to Tx (Km)	Description of Area
downtown Ottawa	urban	4	<ul style="list-style-type: none"> typical small-medium size city 10-25 storey buildings
downtown Hull	urban	5	<ul style="list-style-type: none"> small city core 5-15 storey buildings
Queensway	rural, open	10-12	<ul style="list-style-type: none"> 4-lane divided highway some low-rise industrial and residential development line-of-sight path obstructed by rolling terrain
residential Ottawa (south)	suburban	4	<ul style="list-style-type: none"> single-family and semi-detached townhouse development scattered, light vegetation

Figure 3.1-4. Description of areas covered.

Figure 3.2-1 Data acquisition equipment.

lower frequencies. Since this bandwidth is much less than the sampling rate of 4800 Hz, a simple RC filter was used to prevent any aliasing. More precise digital filters were used later on in the resampling process. Some hardware was built to provide an electrical interface from the digital control signals to the TI PC system bus. Details of the hardware built are contained in Appendix A.

Data acquisition is synchronized to the 20-second intervals of data transmission. Sampling begins when a 'start' pulse is detected on the start/stop channel of the recorder, indicating the start of one such interval. The 12-bit envelope value, received data bit and speed pulse are packed into a 16-bit word in RAM. 90000 samples or 18.75 seconds of the interval is acquired, which is limited by total available RAM. The data for one interval is stored in a file on disk during the 20-second unmodulated interval. About six minutes of recordings were made for each area yielding 8 to 9 data files. The program 'MRDA', listed in Appendix D, was used to acquire the data on the TI computer. A total of 28 files were created from the tapes for the four areas.

3.3 Bit Error Rates

The raw data was first processed to determine the occurrence of bit errors and to compute the bit error rate (BER). While this is not essential to the study of the fading signal envelope, it provides some insight into the correlation between error bursts and fades in signal

strength. Errors are found by comparing the received data with the known 2047-bit transmitted PN sequence.

Before this comparison may be made, the received data must be synchronized with the known sequence. This is done by correlating the two until a minimum BER is found. Correlating the first 250 bits yielded satisfactory results. In most files there are few error bursts, and a unique synchronization offset is easily found. In a few files, apparently a bit in the sequence is skipped, and synchronization is lost. This is apparent when the BER suddenly increases to about 0.5, or simply random chance. These files were not included in the computation of average BER. The program 'BERCOM1' (Appendix D) performs the synchronization and computation of bit error rate.

Figure 3.3-1 shows some typical sequences of the logarithmic envelope signal, occurrence of errors and speed pulse. While it is beyond the scope of this research to investigate the random error process and its correlation with signal strength, the coincidence of error bursts with fades is easily visible. In the urban Ottawa and Hull data, where the signal is normally strong, error bursts are observed only in significant fades, below -100 dBm of received signal strength. In the highway and residential data, where the average signal is lower, about -103 to -107 dBm (see Appendix E), errors are more frequent throughout the data. Figure 3.3-2 summarizes the BER

(a) urban Ottawa, $v = 25$ Km/h

(b) highway (Queensway), $v = 60$ Km/h

Figure 3.3-1 Typical received envelope, bit errors and speed pulse.

Area	Number of files averaged *	Bit error rate	
		Average	Std. deviation
urban Ottawa	9	$2.3 \cdot 10^{-3}$	$1.7 \cdot 10^{-3}$
urban Hull	9	$7.2 \cdot 10^{-3}$	$2.2 \cdot 10^{-3}$
highway (Queensway)	6	0.12	0.076
residential Ottawa	3	0.069	0.041

* Each file = 90000 bits

Figure 3.3-2 Average bit-error-rate by area.

Figure 3.4-1 Signal envelope voltage path.

results for each area. There is a fair degree of variation of BER between files for each area. Due to the bursty nature of the errors and non-stationarity of the channel, digital performance of the channel cannot be completely characterized by average bit-error rate alone. A more extensive study of the error bursts in this data is reported in [22].

3.4 Envelope Calibration

In processing the signal envelope, the first step that was taken was the conversion of the stored 12-bit integer values (Q) into real-valued numbers representing the RMS RF level at the input to the receiver (v_{RF}). Figure 3.4-1 illustrates the signal envelope voltage path from the RF level to the quantized values. A set of calibration points relating the RF input level to the logarithmic amplifier output (v_A) were recorded during the original experiment. This calibration data is listed in Appendix B. As part of our work, a test was then conducted to determine the DC transfer function from the tape recorder input (v_A) to the ADC output (Q). From these results, a piecewise linear approximation of the overall transfer function of Figure 3.4-1 was then derived. This relationship between v_{RF} and Q is plotted in Figure 3.4-2. Finally, the points of this linear approximation were used in a program (QRFGEN, Appendix D) to compute the corresponding RF level for all possible values of Q , and store them in a file. The resampling

Figure 3.4-2 Piecewise-linear approximation of RF signal level to ADC output.

program, described in the next section, converted the stored envelope data to real values using the look-up table generated by QRFGEN.

3.5 Equidistant Resampling

In order to study the spatial fluctuation of signal strength and make meaningful comparisons over different terrain types, the envelope data is resampled to equidistant intervals, independent of vehicle speed. Figure 3.5-1 illustrates the process of equi-distant resampling. The number of envelope samples between two speed pulses varies inversely with vehicle speed. The distance travelled in this interval, equivalent to one full revolution of the wheel, is always 2.83 m. When the speed of the vehicle is constant, equidistant samples are uniformly spaced. If the vehicle accelerates, then the equidistant samples become more closely spaced in time. Thus, conversion from a time coordinate to a spatial coordinate involves non-uniform subsampling of the envelope data.

With reference to Figure 3.5-1, it is possible to simplify the resampling process. While instantaneous vehicle speed is not available, fluctuations in speed are not rapid, and are sampled adequately by the rate of speed pulses. The instantaneous speed is approximated by linear segments between pulses. The speed over the first interval may be expressed as:

$$v(t) = v_0 + \frac{t}{t_1} \cdot (v_1 - v_0) \quad (14)$$

The distance is easily obtained:

$$d(t) = v_0 t + \frac{t^2}{2t_1} \cdot (v_1 - v_0) \quad (15)$$

If there are N samples between the first two speed pulses in the data:

$$t_1 = \frac{N}{f_s} \quad (16)$$

where: f_s = sample rate = 4800 Hz

Letting j = equidistant sample number

d_j = distance to j^{th} sample

M = number of equidistant samples in interval

r_w = wheel radius (0.45 m)

$$\text{Then } d_j = \Delta d \cdot j = \frac{2\pi r_w j}{M} \quad (17)$$

To obtain the corresponding time of the d_j sample, solve for t_j :

$$d_j = v_0 \cdot t_j + \frac{t_j^2}{2t_1} \cdot (v_1 - v_0) \quad (18)$$

Since the data is highly oversampled, it is sufficient to take the nearest envelope sample in the data to time t_j for output as the d_j equidistant sample.

In choosing the equi-distant sample spacing, a trade-off was made between retaining sufficient spatial information and keeping storage and processing within

practical limits. Assuming that adjacent fades are spaced by $\lambda/2$, the maximum sample spacing must be less than $\lambda/4$, or 8.8 cm at 820 MHz, to ensure adequate sampling. Fading, however, is not sinusoidal and has some energy at higher frequencies. As well, some margin must be allowed for variations in speed, and to permit the design of practical anti-aliasing filters. For resampling the envelope data, the number of samples between two speed pulses was chosen to be $M=100$. This corresponds to $\Delta d = 2.83$ cm, or about $\lambda/12$.

By oversampling the spatial fading frequency by about three, a fixed bandwidth anti-aliasing low-pass filter can be designed for each terrain type. Three filters were designed to the specifications given in Figure 3.5-2, which are based on the maximum speed reached in each area. The McClellan-Parks equi-ripple FIR filter design program was used to design the three filters. Further details and frequency response plots appear in Appendix C.

The program RESAMP (Appendix D) was used to process the integer envelope data. Values representing the approximate logarithm of the signal strength filtered and resampled, and then converted to a real-valued RF signal levels. Of the original 28 files of data acquired, 19 files containing a total of 120K equi-distant samples were obtained. Only files where the mobile remains in motion were processed, because memory requirements become unbounded when the vehicle stops. This occurs because the number of time samples between speed pulses increases as

Figure 3.5-1 Illustration of equi-distant resampling.

Terrain type	Maximum speed [Km/h]	Fading rate [Hz]	Subsampled rate [Hz]	Filter design*	
				Cutoff freq. [Hz]	SB Atten.** [dB]
urban	20	32	197	100	39
residential	50	79	490	250	39
highway	80	126	785	400	39

* sampling rate = 4800 Hz

** stop-band attenuation

Figure 3.5-2 Resampling low-pass filter specifications.

the vehicle slows and the program requires two speed-pulse intervals of time data for resampling.

Figure 3.5-3 summarizes the files processed and resampled. The length of the file varies with the distance travelled in each 18.8 second interval of data acquisition.

Figure 3.5-4 shows examples of received field strength data in each area after equidistant resampling. The RF signal strength is plotted in dBm calculated from the data assuming 50Ω impedance:

$$v[\text{dBm}] = 10 \log \left(\frac{v_{\text{RF}}^2}{(50\Omega)(1.0 \cdot 10^{-9} \text{W})} \right) \quad (19)$$

In this figure, with the data now a function of distance, some features in the behaviour of field strength by area become apparent. The regular ripple in field strength, due to multipath fading, appears greatest in the urban and residential data, while less on the highway. This is reasonable, since more multipath scattering is expected in the developed areas, compared to the uncluttered highway terrain.

Area	File number	Number of equi-distant samples	Distance travelled [metres]	Range of speed [m/s]
urban Ottawa	1	3700	105.	3.9-6.8
	4	3600	102.	4.3-6.0
	5	2400	68.	1.6-5.4
	8	2100	59.	2.0-6.0
urban Hull	2	2400	68.	2.7-4.7
	4	3500	99.	5.2-5.8
	5	2000	57.	2.0-5.2
	7	2300	65.	1.9-5.6
	8	1700	48.	1.5-3.9
	9	3700	105.	5.4-6.5
highway (Queensway)	1	5300	150.	8.1-9.4
	2	11700	331.	15.5-19.3
	3	13500	382.	20. -21.
	4	13800	390.	21.
	5	13300	376.	19.6-21.2
	6	14500	410.	22.
residential (south Ottawa)	1	7100	201.	9.9-10.4
	2	6800	192.	10.3-11.0
	3	6700	189.	10. -11.2

Figure 3.5-3 Resampled envelope data files.

Figure 3.5-4 Typical resampled data sequences.
(Spatial signal strength)

Chapter 4

STATISTICAL DATA ANALYSIS

4.1 Introduction

In this chapter, we consider first and second order statistics of the spatial fluctuations in the random field strength process. Figure 4.1-1 illustrates the steps taken to compute the local mean, and estimate various statistics. As discussed in Section 2.2, the envelope is first low-pass filtered to obtain an estimate of the fluctuating local mean due to shadowing. The remaining fast fading ratio is then computed. Several estimates of the local mean were made using low-pass filters with different spatial cut-off frequencies.

Programs were written to compute individual statistics of the signal envelope, local mean and remaining fast fading ratio. These include sample mean and standard deviation, probability density function and autocovariance function. First-order statistics of the fading envelope and its components have been widely reported, and our results compare favourably. For other statistics, such as autocovariance and power spectral density, there is less published material for comparison.

In our statistical analysis, the assumption of stationarity of the random processes is accepted with

Figure 4.1-1 Steps of statistical analysis.

qualifications. This assumption allows us to compute useful short-term statistics. However, throughout this work, the actual degree of stationarity was subsequently judged by examining the consistency of these statistics over longer intervals. A trade-off must be made in the choice of sample size used for the estimates. The estimation error decreases as the size of the sample space increases. However, over larger intervals, the statistical invariance becomes more significant. We estimated each statistic using the data from one data file, typically 2000-15000 samples. In most cases, the estimates for files in one area were consistent, so the same statistic was subsequently estimated over all data from one area. Judging the degree of consistency and stationarity remains a qualitative problem. In the case of the autocovariance of the local mean, the results of each file were plotted and found to vary widely. Other statistics such as the probability density functions of local mean and fast fading, and the autocovariance of the fast fading were found to be consistent between files. One sample probability density function was computed using all the data collected in an area. In the case of the autocovariance function of the fast fading, we simply averaged the individual estimates made for each file, then to perform an autocorrelation over all files in an area.

4.2 Estimation of Local Mean

From equation (4), the estimate of the local mean is computed using a FIR filter:

$$\hat{m}(x_j) = \sum_{i=-M}^M h_i r(x_{i+j})$$

having an impulse response length of $N = 2M+1$.

The envelope is then normalized with respect to the local mean to obtain an estimate of the remaining fast fading component:

$$\hat{r}_o(x_j) = \frac{r(x_j)}{\hat{m}(x_j)}$$

The filter coefficients h_i are computed from an N -point cosine-squared function:

$$h_i = \cos^2 \left(\frac{\pi i}{2M} \right), \quad i=-M, \dots, M \quad (20)$$

and normalized such that $\sum h_i = 1$. The frequency response of this filter is shown in Figure 4.2-1. The choice of filter is not critical; however, the cos-squared filter has excellent roll-off characteristics and is easily computed. The normalized cutoff frequency varies inversely with the filter length, and was actually measured for $N=35$, 99 and 201 from the frequency responses. For other values, the cutoff frequency can be approximated by $\bar{f}_c \approx 0.73/N$. The spatial cutoff frequency was then calculated knowing the sampling rate of $1/2.83$ cm $= 35.3 \text{ m}^{-1}$.

Figure 4.2-2 summarizes the characteristics of the filters used. The distance cutoff d_c represents, approximately, the interval averaged. Variations in local mean with a spatial period of d_c are attenuated 3dB relative to a steady signal. More rapid variations than this are highly attenuated. The response is down 6dB at $0.7d_c$ and more than 30dB down for spatial fluctuations less than $0.3d_c$.

As discussed in Chapter 2, fluctuations over intervals 5 to 10 metres or more constitute significant shadowing, and filter lengths were chosen with this in mind. The lengths $N=99$ and 201 best meet the shadowing criteria, but the other filters were included to examine the effects of filter length on the statistics.

A program was written (ENFILT, Appendix D) to compute the local mean and fast fading ratio for each data file. Figure 4.2-3 is an example of the results obtained for downtown Ottawa envelope data using a 99-point filter. The estimates $\hat{m}(x)$ and $\hat{r}_0(x)$ are plotted on logarithmic dBm and dB scales, respectively. Removal of the varying local mean is clearly evident in (c), as the fast fading fluctuates about 0dB.

Figure 4.2-4 shows the local mean estimate using three different filters. Obviously, the longer filter yields a smoother estimate of the local mean. Therefore, more fluctuation is found in the remaining fast fading $\hat{r}_0(x)$. Conversely, the shorter filters leave more short-term fluctuations in the estimate of the local mean. In

Figure 4.2-1 Frequency response of 99 point $\cos^2$ window.

Number of points, N	Normalized Cutoff freq. f_c	Spatial cutoff frequency f_{cd} [m^{-1}]	Spatial cutoff distance d_c (m)
15	0.049**	1.73	0.58
35	0.0212*	0.75	1.33
51	0.0144*	0.51	1.96
99	0.00733*	0.26	3.84
201	0.0036**	0.127	7.87
399	0.00183**	0.065	15.4

* Measured.

** Extrapolated by approximation.

Figure 4.2-2 Local mean low-pass filter characteristics.

Figure 4.2-3 Typical local mean and fast fading components.
Urban Ottawa - 99 point filter

Figure 4.2-4 Local mean estimates using various length lowpass filters.

this case, there is less variation in $\hat{F}_0(x)$ due to shadowing, making it more stationary about 0dB. The effects of filter length on stationarity and statistics are examined further in following sections.

4.3 Mean and Standard Deviation

Estimates of the mean and standard deviation were computed for the envelope $r(x)$, local mean $\hat{\mu}(x)$ and fast fading $\hat{F}_0(x)$. Each statistic was calculated from all the data in one area. For N points, the mean is estimated as (summation over all x):

$$\hat{\mu}_x = \frac{1}{N} \sum x \quad (21)$$

and the sample standard deviation:

$$\hat{\sigma}_x = \sqrt{\frac{1}{N-1} \left[\sum x^2 - \frac{1}{N} (\sum x)^2 \right]} \quad (22)$$

The estimates were computed using various local mean filter lengths and tabulated in Appendix E. The area mean, $\hat{\mu}_r$, for each area is related to gross terrain effects and is examined later. The estimates of statistics for the local mean, and fading, show the effect of varying filter lengths. The decreasing standard deviation of the local mean, $\hat{\sigma}_\mu$ with increasing filter length, obviously reflects the reduced variability in the smoother estimates obtained with longer filters. However, decreases in $\hat{\sigma}_\mu$ may also be attributed to the fixed sample size over which this statistic was computed. These statistics were computed over all data files for an area.

Statistics by individual file showed less variability. For example, $\hat{\sigma}_m^2$ was computed for each of the four urban Ottawa files using a 99-point filter. Values of 1.9, 2.5, 3.0 and 3.3 dB were obtained. By comparison, the standard deviation computed over all four files (11800 points) was 3.4 dB. If the random local mean process was stationary, then a larger sample size would yield a better estimate of these statistics. However, we found the local mean is not stationary, so a larger sample size will not make any improvement.

The mean and standard deviation were computed over differing sample sizes which makes comparison of areas less precise, but some features are worth noting. The standard deviation of the local mean $\hat{\sigma}_m^2$ in the urban and residential areas is about 3 to 4 dB, except in those cases where the very shortest filter was used. The same statistic ranges from 4 to 5 dB on the Queensway. The higher variability of the local mean on the Queensway may be attributed to the decreasing path length as the mobile receiver travelled east on the highway towards the transmitter. The path length decreased from 12 to 10 km. The corresponding free-space path loss decreases by 1.6 dB. If excess losses using Okumura's method are included, this difference increases to 2.9 dB. However, it should also be noted that there are about 72000 samples of highway data, while there are fewer than 20000 samples in each of the other areas. The standard deviation of the

local mean for individual highway files, over 5000 to 15000 samples, average about the same as the standard deviations obtained for the other areas.

Based on these results, we find no significant differences in standard deviation of the lognormal shadowing by area. The average value for $\hat{\sigma}_n$ for all areas is about 3.0 to 3.5 dB for filters $N=99$ to $N=399$. These values are low in comparison to other reported results. French obtained 5 dB from measurements at 462 MHz in urban London [10]. Generally accepted values for small cities are about 6 dB, and 8 to 12 dB for large cities.

4.4 Sample Probability Density Functions

The distributions of the random envelope $r(x)$ and its components $\hat{m}(x)$ and $\hat{r}_0(x)$ were examined by computing sample probability density functions, or data histograms. The experimental PDFs were accumulated on a logarithmic scale divided into 0.5 dB intervals. Each sample PDF was accumulated over all files for an area, ranging from 11800 samples in urban Ottawa, to 72100 samples of highway data. Most of the results were obtained using a 99-point lowpass filter. Other filters were tried for comparison, but changes in the sample PDFs essentially reflect the statistics in Appendix E. As the filter length is increased, the standard deviation of the local mean decreases, while the fast fading ratio approaches the distribution of the original envelope. Conversely,

shorter filters cause the local mean distribution to approach that of the original envelope, while variance of the fast fading decreases. Even for the shortest filter ($N=15$, averaging interval $\approx 1.5\lambda$), there is still a remarkably good spread of fast fading data. For urban Ottawa, the fast fading standard deviation drops from 6.5 dB ($N=400$) to 5.0 dB ($N=15$).

Figure 4.4-1 shows the sample PDFs by area, obtained with a 99-point filter. The PDFs of the local mean are somewhat rough and ragged, but bear resemblance to a lognormal distribution. Both the envelope $r(x)$ and fast fading $\hat{r}_0(x)$ in the urban areas, (a) and (b), have a shape similar to a Rayleigh PDF (see Figure 4.4-2(b)). However, the fast fading from the highway and residential data have distributions which are narrow and peaked, almost lognormal looking. Neither have significant "tails" into low signal levels, as seen in the urban and residential fast fading distributions.

The sample PDFs in Figure 4.4-1 were fitted with the theoretical PDFs introduced in Chapter 2. We used a sum-of-squares criterion as a measure of goodness-of-fit, similar to Suzuki [11]:

$$e = \sum_{\substack{i=-70\text{dBm} \\ i=-130\text{dBm} \\ (0.5 \text{ dB steps})}} (\hat{f}(x_i) - f(x_i))^2 \quad (23)$$

where

$\hat{f}(x_i)$ = fraction of total samples in interval i ,

(a) urban Ottawa - all data files

(b) urban Hull - all data files

Figure 4.4-1 Sample probability density functions (page 1 of 2)
(lowpass filter $N=99$)

(c) Queensway (highway) data

(d) residential Ottawa data

Figure 4.4-1 Sample probability density functions (page 2 of 2)

$f(x_1)$ = theoretical PDF evaluated at center of interval i .

An interactive program, was used to plot the PDFs and allows adjustment of parameters to seek an optimum fit based on minimizing the error above. Figure 4.4-2 illustrates the curve-fitting of urban Ottawa fast fading data to a Rician and Rayleigh PDF. The PDF curves are all scaled to have equal areas. The best-fit Rayleigh PDF is poor, while a good fit to the Rician PDF is obtained.

Figure 4.4-3 summarizes the curve fitting for all areas. We found the envelope $r(x)$ and fast fading $\hat{r}_0(x)$ fit best to the Rician and Nakagami distributions. The differences between the two are quite small. By comparison, the Rayleigh PDF fit poorly. As noted by Suzuki in similar work [11], a better fit to these Rician and Nakagami PDFs can be expected since both of these PDFs have two parameters, and both include the Rayleigh PDF as a special case. When fitting the fast fading ratio to a unit mean Rayleigh PDF, there are no free parameters, while there is one remaining free parameter in the Rician and Nakagami PDFs.

In the case of the unit mean Rician PDF, this remaining free parameter is the ratio of scattered energy to the line-of-sight component. With reference to equation (8), Figure 4.4-4 summarizes the optimum Rician parameters by area, based on our curve fitting criteria. The results show that scattered energy is highest in urban areas, compared to the uncluttered highway terrain. The

(a) Best-fit Rician PDF - good fit.

(b) Best-fit Rayleigh PDF - poor fit.

Figure 4.4-2 Examples of typical curve-fitting results:
Urban Ottawa fast fading data.

Area	Component	Rayleigh	Rician	Nakagami	Lognormal
urban Ottawa	$r(x)$	8.6E-4 good	8.6E-4 good	4.2E-4 good	6.2E-4 good
	$\hat{r}_0(x)$	4.5E-3 fair	1.9E-4 good	3.0E-4 good	1.5E-3 fair
urban Hull	$r(x)$	1.7E-3 fair	7.2E-4 good	6.0E-4 good	3.0E-4 good
	$\hat{r}_0(x)$	5.8E-3 poor	3.3E-4 good	2.1E-4 good	8.2E-4 good
highway (Queensway)	$r(x)$	2.2E-3 fair	1.1E-3 fair	8.8E-4 good	1.4E-3 fair
	$\hat{r}_0(x)$	3.5E-2 poor	2.2E-3 fair	3.1E-3 fair	3.2E-3 fair
residential Ottawa	$r(x)$	3.6E-3 fair	6.6E-4 good	6.1E-4 good	1.0E-3 fair
	$\hat{r}_0(x)$	2.3E-2 poor	6.3E-4 good	8.0E-4 good	1.5E-3 fair

Quality of fit:	
Good:	$e < 1E-3$
Fair:	$1E-3 < e < 5E-3$
Poor:	$e > 5E-3$

Boundaries of the quality of fit are subjective. See Figure 4.4-2 for examples of good and poor fits.

Figure 4.4-3 Curve fitting envelope and fast fading data by area.

Area	Rician parameters		Mean value μ
	a (direct)	b (scattered)	
urban Ottawa	1.11	0.487	1.20
urban Hull	1.06	0.438	1.14
highway	1.03	0.208	1.05
residential	1.04	0.270	1.07

Figure 4.4-4 Optimum Rician curve fitting parameters for normalized fast fading component.

fraction of scattered energy present in the residential area falls between these two extremes. These results are basically consistent with the nature of the terrain in each area.

The error in fitting the urban fast fading ratio to a Rician PDF is $1/2$ to $1/4$ of that for the original envelope. This indicates a substantial improvement in fit is obtained by removing the local mean. However, it was observed earlier that the fast fading component from the highway and residential data has a sample PDF which is very peaked, and looks neither Rayleigh or Rician. There is little improvement, or an actual deterioration, in the quality of fit for the same data from the residential and highway areas. We are not certain whether this lack of improved fit is due to the relative absence of multipath fading, or to the low signal level in these areas. A strong line-of-sight component should yield a Rician PDF, but this was not the case. We suspect that the low signal level may have forced the logarithmic detector to operate near the bottom of its dynamic range, thereby compressing any fading below that level, and distorting the resulting distributions.

4.5 Autocovariance Function,

This section, forming the core of our research, examines the spatial correlative properties of the random field strength process. Of principal interest is the autocovariance of the fast fading component, due to

multipath propagation. As outlined in Chapter Two, the autocovariance of multipath fading is evident over an interval on the order of $1-2\lambda$, described by equation (12). The autocovariance of the fast fading component is estimated from the equi-distant envelope data, then interpolated for finer comparison. The autocovariance of the original envelope and local mean are also examined over about 300 samples, or about 8.5 metres. From equation (13), the autocovariance function estimator is:

$$\hat{K}_r(m) = \frac{1}{N} \sum_{n=0}^{N-m-1} [r(n) - \bar{r}][r(n+m) - \bar{r}]$$

where $r(n)$ = envelope samples,

and $\bar{r}$ = mean value over N samples.

The program COVAR (Appendix D) was used to compute autocovariance estimates for the envelope and its components. The program correlates up to 8000 samples, limited by maximum array sizes, which is a limitation in correlating the highway data only. The other estimates were limited by the individual file sizes (see Figure 3.5-3), typically 2000 to 8000 samples, or 50-225 metres. A comparison of autocovariance estimates from one file showed that the larger sample size produced more pronounced peaks and troughs. However, we judged that the variation in file sizes in each area did not warrant the extra effort required to compute the estimates over a fixed interval. Instead, the entire file was used for the estimate. The estimated functions were then interpolated.

and plotted for comparison.

Figure 4.5-1 illustrates some typical autocovariance estimates for some urban Ottawa data. The sample spacing in the autocovariance estimates is the same as the original data, namely 2.83 cm. The local mean was estimated with a 35-point filter, which is equivalent to averaging about 1 metre. The correlation of $\hat{m}(x)$ over the first metre due to averaging is clearly evident. Adjacent peaks in the autocovariance of the fast fading are spaced about 0.19 m apart, or just about $\lambda/2$ (0.183 m), reflecting the periodicity of fast fading in the data. From these results, it is apparent that the autocovariance of the local mean is a smoothed version of the autocovariance of the original envelope. Further work would be needed to verify that the autocovariance function of the overall envelope is actually the sum of the autocovariances of the two components. As an alternate method of normalization, the difference $\hat{y}(x) = r(x) - \hat{m}(x)$ was also computed. The function $\hat{y}(x)$ also represents the fast fading; however, $\hat{y}(x)$ is bipolar with approximately zero mean. Except for a scale factor, the autocovariance of $\hat{y}(x)$ is very similar to that of $\hat{r}_0(x)$. Other statistics of $\hat{y}(x)$ were not examined, but this result suggests that either the ratio estimate $\hat{r}_0(x)$ or the difference estimate $\hat{y}(x)$ are both valid methods of representing the multipath fading component of the signal strength process.

Figure 4.5-1 Typical autocovariance estimates for urban envelope and components. (downtown Ottawa)

There is a large degree of variation in the autocovariance of the local mean from file to file in one area, reflecting the non-stationary nature of the shadowing process. Figure 4.5-2 shows the autocovariance of $\hat{m}(x)$ for four urban Ottawa files. The local mean shows substantial correlation over the first 2 metres, or approximately the filter length (99 points = 2.8 m), as expected, however, the shapes vary widely. Longer filters ($N=201, 399$) increase the spatial correlation proportionally. Longer filters increase the consistency in local mean autocovariance, but they tend to obscure the finer fluctuations between 1 and 5 metres, visible in Figure 4.5-2. The 99-point filter apparently offers the best compromise between extracting these fluctuations in the local mean and the fluctuations which are attributable to multipath fading.

In order to examine further the short-term correlative behaviour of the fast fading component $\hat{f}_0(x)$, the autocovariance functions were interpolated 16:1 using the zero-padded FFT method. The method is illustrated in Figure 4.5-3. The first 64 lag points are reproduced symmetrically in a 128 point array. The FFT is computed, then expanded to a 2048 array with zeros in the middle. The inverse FFT is computed and the resulting interpolated data extracted. The spatial sampling of the new data is $2.83/16 = 0.177$ cm. This technique yields good results for bandlimited signals.

Figure 4.5-2 Autocovariance functions of local mean for four urban Ottawa data files.

Figure 4.5-3 Zero-padded FFT interpolation.
(16:1 time interpolation)

Figure 4.5-4 illustrates some interpolated autocovariance functions from four urban Ottawa files, where two different low-pass filters to estimate the local mean. Figure 4.5-4(a) and (b) were derived with 99 and 201-point filters, respectively. The curves all have a similar shape, peaking at about 0, 0.5λ , λ and so on, as expected. The results for the shorter filters have less offset and tend to oscillate more symmetrically about zero. This symmetry about zero is especially apparent in the uninterpolated data of Figure 4.5-1(c), obtained with a 35-point filter. Very short filters leave essentially only $\lambda/2$ variations in the fast fading, with a corresponding autocovariance that is periodic in $\lambda/2$. Conversely, a very long filter would remove very little local mean fluctuation, and so the autocovariance of the fast fading component would resemble that of the original envelope. Similar results were obtained for the fast fading component of other areas, although the autocovariance from several Queensway files (#1 and #3) had poor shape.

The autocovariance functions of the fast fading component from each area were first averaged, then plotted for comparison in Figure 4.5-5. As well, the theoretical autocovariance of the vertical electric field envelope from equation (12) is included. The shape of our averaged functions is similar to the theoretical result. It should be noted that the first two sidelobes of $J_0^2(\xi)$ peak

(a) 99 point lowpass filter

(b) 201 point lowpass filter

Figure 4.5-4 Autocovariance of fast fading in urban Ottawa for several lowpass filter lengths.

Figure 4.5-5 Autocovariance function of fast fading by area.
(99 point filter)

greater than $\lambda/2$ and λ . They actually occur at 0.61 λ and 1.11 λ . The peaks in our data occur closer together than the theoretical. A possible inaccurate measurement of the wheel radius would account for the difference in the spatial scale. Another notable feature in Figure 4.5-5 is that the experimental autocovariance functions make excursions below zero, while $J_0^2(\xi)$ is a positive-valued function. As noted above, fast fading estimates obtained with longer local mean filters have an autocovariance function which approaches that of the original envelope. As the autocovariance approaches that of the original envelope, it becomes increasingly positive-valued, making less excursions below zero.

4.6 Power Spectral Density

The power spectral density (PSD) may be obtained by computing the Fourier transform of the autocorrelation function. The FFT of the autocovariance functions of the previous section yield essentially the same results, since the two differ only a non-zero DC level. Otherwise, the spectral characteristics remain the same. While an extensive investigation of the power spectral density was not undertaken in this work, the example presented here provides some insight into the spectral characteristics of the spatial fluctuations in signal strength.

Figure 4.6-1 is an example of the PSD of the urban fading envelope. The spatial sampling frequency is $f_s = 1/0.0283 = 35.3 \text{ m}^{-1}$. A prominent peak is visible

about $0.15 f_s = 5.3 \text{ m}^{-1}$. The corresponding period is 18.8 cm or 0.51λ , which agrees well with the position of the first sidelobe in the autocovariance function of Figure 4.5-5. Above this point, the PSD drops by 20 dB, indicating that there is little energy at higher frequencies. A significant minimum, approximately $0.075 f_s$, or 1λ in period, suggests that a possible choice for the cutoff frequency of the local mean filter might be made at this minimum. The choice of cutoff frequency could be based on the criteria of splitting the power spectrum into two equal-energy parts. This choice of cutoff is much less than the 3 to 15 metre intervals used in our work, or 50-100 λ used by other researchers. As shown in the example of Figure 4.6-2, the choice of these low cutoff frequencies, in terms of power spectrum is essentially arbitrary.

Although this examination of the envelope PSD is only cursory, it points out the spectral inseparability of fast fading from shadowing. The power spectrum suggests that a division may be made around spatial fluctuations of 1λ , but this choice is much smaller than fluctuations typically considered shadowing. We conclude, therefore, that the envelope power spectrum does not indicate any natural choice of a longer averaging interval.

Figure 4.6-1 Power spectral density of urban fading envelope.

Figure 4.6-2 Typical local mean filter cutoff characteristics.

4.7 Prediction of Area Mean

The empirical results published by Okumura may be used to predict the average signal strength. Okumura describes total path loss as the sum of free-space loss and additional losses due to the terrain. These additional losses depend on several factors including type of terrain (urban, suburban, rural, etc.) and antenna height of the base and mobile. Okumura's results are based on measurements in Tokyo, but have been compared with results obtained in other areas. Since radio propagation is influenced by many geographic and atmospheric conditions, these predictions are naturally only approximate. Okumura's results have been conveniently summarized in equation form by Hata [23], which are used here. We begin the calculations with free space path loss, which may be conveniently expressed as:

$$L_{fs} = 32.44 + 20\log_{10}d + 20\log_{10}f \text{ [dB]} \quad (24)$$

where

d = distance in Km,

and f = frequency in MHz.

In our experiment, transmitter power output $P_t = 30.W$
= 44.8 dBm. The effective radiated power,

$$\begin{aligned} P_{erp} &= P_t + \text{antenna gain} - \text{line losses} \\ &= 44.8 + 8 - 2 \\ &= 50.8 \text{ dBm} \end{aligned}$$

Area	Free space path loss L_{fs} [dB]	Additional terrain loss L_t [dB]	Power at receiver input		Difference: predicted - measured [dB]
			Predicted P_r [dBm]	Measured P_r [dBm]	
urban Ottawa (d = 4 Km)	102.8	43.2	-94.2	-89.8	-4.4
urban Hull (d = 5 Km)	104.7	44.8	-97.2	-94.9	-2.8
highway: d = 10 Km d = 12 Km	110.7	49.7	-108.6	-105.9	-2.7
	112.3	51.0	-111.5	-106.4	-5.1
residential (d = 5 Km)	104.7	44.8	-97.7	-103.8	+6.1

Figure 4.7-1 Predicted path loss and received signal strength.

Net gain at the receiver is:

G_r = mobile antenna gain - line losses

≈ 2 (quarter-wave vertical) - 1 (estimated losses)

≈ 1 dB

The predicted received power is then

$$P_r = P_{erp} - L_{fs} + G_r - L_t \quad (25)$$

where L_t = additional losses due to terrain.

Removing the free-space loss component from equation (11) of [23], an empirical expression for additional losses over quasi-smooth terrain in an urban area:

$$L_t = 37.11 + 6.16 \log_{10} f - 13.82 \log_{10} h_b - a(h_m) \\ + (24.9 - 6.55 \log_{10} h_b) \log_{10} d \quad (26)$$

where h_b = base station antenna height in metres,

and $a(h_m)$ = correction for mobile antenna height over 1.5m.

For a medium-small city,

$$a(h_m) = (1.1 \log_{10} f - 0.7) h_m - (1.56 \log_{10} f - 0.8)$$

Substituting $h_m = 3$ m, $f = 820$ MHz yields $a(h_m) = 3.8$ dB, and so (26) becomes:

$$L_t = 33.31 + 16.38 \log_{10} d \text{ [dB]} \quad (27)$$

$$\text{and } P_r = 51.8 - L_{fs} - L_t \text{ [dBm]} \quad (28)$$

Using equations (24), (27) and (28) for the quasi-smooth urban model, estimates of the received signal strength were found to agree reasonably well with measured values (see Figure 4.7-1). The estimates are 3 to 4 dB below actual values, except for the residential area where

Area	Free space path loss L_{fs} [dB]	Additional terrain loss L_t [dB]	Power at receiver input		Difference: predicted - measured [dB]
			Predicted P_r [dBm]	Measured P_r [dBm]	
urban Ottawa (d = 4 Km)	102.8	43.2	-94.2	-89.8	-4.4
urban Hull (d = 5 Km)	104.7	44.8	-97.2	-94.9	-2.8
highway: d = 10 Km d = 12 Km	110.7	49.7	-108.6	-105.9	-2.7
	112.3	51.0	-111.5	-106.4	-5.1
residential (d = 5 Km)	104.7	44.8	-97.7	-103.8	+6.1

Figure 4.7-1 Predicted path loss and received signal strength.

the estimate is 6 dB too high. Corrections for suburban and open terrain by Hata significantly worsened the estimates. For suburban terrain, there is a correction factor which, for 820 MHz, worsens the path loss estimate by 9 dB. A similar correction of 28 dB for open terrain, if included, would predict a signal strength on the highway about 25 dB too high. The results using the basic quasi-smooth urban model matches our measured results best, while these other correction factors appear to worsen the estimates. In our experiment, any increase in signal strength in less urbanized areas might have been attenuated by other features of the terrain. For example, buildings of the university are located in a depression next to the Rideau River, and adjacent terrain obscures a line-of-sight path to both the Western Queensway and residential areas. This is one such specific feature that might account for additional path losses. The basic quasi-smooth urban model is obviously valuable for making rough estimates of received signal strength, but the differences between these estimates and actual values only serve to highlight the need to conduct actual field measurements of received signal strength in desired coverage areas.

Chapter 5

CONCLUSIONS

5.1 Discussion of Results

The following is a summary of our research. Statistics of envelope fading over the 820 MHz base-to-mobile radio channel were derived from data collected in an earlier field experiment conducted in the greater Ottawa-Hull region. The effects of changing vehicle speed were removed, and the random spatial field strength process was examined directly. Estimates of the mean, standard deviation, sample probability density function and autocovariance function were computed. Some of our statistical results confirm those reported in previous experiments, while other results, particularly those concerning the correlative properties of the field strength process, provide new experimental evidence supporting existing theoretical results.

Much of the analysis is based on a model which combines multipath fading and shadowing effects. The model essentially describes the overall fading envelope as the product of fast (or Rayleigh) fading due to multipath propagation with a slowly varying local mean. We have assumed these two effects are separable in spatial frequency. The most rapid variations in signal strength

have a spatial period of about $\lambda/2$ or 18 cm, while slower variations with a period of 5-10 metres or more, constitutes shadowing. An estimate of the shadowing component was obtained by low-pass filtering the spatial envelope data. The definition of shadowing tends to be inexact, making the choice of cut-off frequency somewhat arbitrary. No natural division between shadowing and multipath fading was indicated in the power spectral density at these spatial frequencies, although a much smaller period of about 1 metre was suggested by a minimum in the example PSD computed. Several filters with spatial cut-off periods ranging from 1 to 15 metres were used. The fast fading and shadowing components have statistics which depend on the filter length, but each component retains distinctive features such as probability density and correlation functions. This evidence supports our assumption that the two components are uncorrelated, but this was not verified with cross-correlation estimates. Even the absence of correlation between shadowing and multipath fading does not imply stochastic independence. The use of independent Rayleigh (or Rician) and lognormal random processes for fading and shadowing in a channel model is only approximate. However, previous studies have shown that results obtained with such a channel model agree well with field measurements.

Our research has shown that the shadowing component has a probability density function which is approximately lognormal, in agreement with previous studies. We found

COMPONENT SIDE

DIGITAL FILTER INTERFACE

ASSEMBLY DRAWING

DWN' T KNIGHT 9-JULY-84

fast fading was derived by normalizing the original envelope to the local mean. Sufficient data from each area permitted detailed scrutiny of the sample probability density function. In comparison to the theoretical density functions, the data from the urban Ottawa, residential, and highway areas fitted best to a Rician PDF. The urban Hull data fitted best to the Nakagami PDF, although the Rician density function fit nearly as well. Still, considering the randomness of the sample distributions, the differences between the Rician and Nakagami PDFs are not significant. In terms of our goodness-of-fit criteria, the urban Ottawa and Hull data have density functions which fit closest to Rician. The residential and highway data did not fit as well, a situation we suspect, might have been due to nonlinear effects in the logarithmic amplifier at low signal levels. The Rician distributions all have a mean value near unity, with the remaining free parameter, the ratio of line-of-sight energy to scattered energy, being different for each area. The fraction of scattered energy was highest in the urban areas where more multipath scattering is to be expected, and lowest in open areas such as highway terrain. The results here are representative of those that might be expected in the urban core and suburban surroundings of a small-medium city. This empirical ratio is a terrain-dependent parameter which may be adjusted when modelling multipath fading using a Rician random process.

Stationarity in the spatial correlation of the fast fading component was indicated by consistent estimates of the autocovariance function within one area. As well, the average autocovariance function from each of the four areas had similar shapes and compared favourably with the autocovariance predicted by the multipath fading model. This consistency could be further examined by computing a confidence interval to bound statistical variation in the autocovariance estimates.

Invariance with terrain is supported by the basic multipath model, which attributes fading to an indeterminate number of scattered, interfering waves. The results here indicate that the spatial correlation remains unchanged whether a Rayleigh or Rician random process is assumed. As with shadowing, the autocovariance function may be used to correlate samples of an uncorrelated Rician random process. The merits of using such a stochastic model in the simulation of multipath fading must be considered in light of alternative approaches such as replay of actual data or deterministic calculations.

U We found predictions of mean signal strength using Okumura's quasi-smooth urban terrain model compared well with estimates of mean signal strength from the experimental data. Typical predictions were 3 to 4 dB low of measured values, except in the residential area. The basic urban terrain model yields the best predictions,

while the correction factors for suburban and open terrain significantly worsened the estimates. There are no firm guidelines in Okumura's method for the choice of these correction factors which may radically alter the estimate.

5.2 Applications

The area to which this work is most applicable is that of fading/simulation. Frequently, the performance of both equipment and systems may be tested in the lab solely on the basis of amplitude fading. Analog fading simulators may be designed for operation at either baseband or radio frequencies. Baseband fading simulators can be used to test modems or similar equipment while excluding the effects of the transmitter and receiver. A RF fading simulator reproduces conditions very similar to those encountered in the field, and is useful for lab testing a complete mobile radio system. In both cases, the simulation may range from one involving simple Rayleigh fading to one that combines fading, shadowing and terrain effects, such as shown in Figure 2.1-4. Using the autocovariance functions of the fast fading and local mean components, samples from Rician and lognormal random processes may be correlated to produce a stationary simulation of multipath fading and shadowing.

As indicated by previous researchers, knowledge of the analog fading characteristics is useful in modelling the digital mobile radio channel. A number of models exist which can be used to reproduce the variable, bursty

nature of errors. Simulations of this kind may be used to predict system performance, and test various coding schemes and protocols.

The level of complexity and sophistication remains a major issue in the choice of a fading channel model. Useful results have been reported using simple Rayleigh fading simulation alone. The question remains whether or not increased complexity of the analog fading model is justified in view of other disruptions in the mobile radio channel. Uncertainty in the gross features of amplitude fading such as the prediction of the area mean may obliterate any gains made through these refinements. The finer aspects of amplitude fading must also be weighed against the relative importance of random FM, dispersion, ignition noise, interference and other factors. An associated issue involves making tradeoffs between simulation and reproduction. A very effective simulation may be obtained by simply reproducing stored data. The mass storage in desktop computers such as the one used in our work and the large capacity of IC memories now make it practical to store and recall actual fading data recorded in the field. With spatial data stored, any fixed or variable vehicle speed may be used in resampling this to time-dependent data.

The autocovariance function describing the short-term correlative behaviour of the signal envelope has application in space diversity, and potential use in

linear prediction of the envelope. Ordinarily, constant-amplitude modulation schemes are chosen for use in mobile radio because they are relatively immune to fading. However, digital amplitude modulation schemes such as QAM might be feasible if the signal envelope is predictable. With knowledge of past envelope samples, the autocovariance function may be used in the linear prediction of the current envelope level. This technique offers the potential for demodulating QAM or similar modulation schemes, increasing the information transmitted per symbol. Unfortunately, this novel approach faces some serious challenges. The autocovariance function is a spatially invariant function, while the receiver inherently requires it in temporal form to predict the received envelope. The conversion from space to time requires information about the speed of the vehicle. Obtaining this information in a mobile receiver would be tedious and virtually impossible when the mobile transmits to a base station receiver. Also, fading over the mobile channel is simply too severe to recover reliable estimates of the signal envelope in any AM scheme. When the signal level drops below the receiver noise threshold, envelope information simply cannot be recovered. Previous experience indicates it is better to be content with the unavoidable digital error bursts in a mobile system, and use simple, robust modulation and decoding schemes which minimize the propagation of errors.

Space diversity offers an effective means to combat fading. The field strength process is sufficiently decorrelated at antennas spaced greater than about $\lambda/4$, making the probability of simultaneous fades very low. Ideal spacing occurs where the autocovariance function becomes zero. In the theoretical case, the zeros of the Bessel function occur at 0.38λ , 0.88λ , and so on. Our results indicate that the autocovariance function of the original envelope does not pass through zero, but has minima near these values. The first minimum occurs about 0.30λ where the autocovariance is about 30% of the peak value.

5.3 Suggestions for Further Work

Our research has concentrated on the fading of the 820 MHz base-to-mobile channel, but this certainly does not represent an exhaustive analysis of the experimental data collected in 1983. Our methods could be applied to data collected at 150 and 450 MHz as well. Analysis of this other data would reveal any frequency-dependent behaviour of the fading statistics. Also, data collected for transmission from mobile to base could be analyzed to reveal any nonreciprocal properties of the channel. There is little published work on these reciprocal properties, but it is believed that the fading statistics may depend on the direction of transmission..

The data acquired on the computer can be further analyzed to determine statistical relationships between channel fading and digital error bursts. The correlation of error bursts with fades is apparent visually, and further analysis could focus on the probability of bit error as a function of signal strength. Following the discussion in Chapter 2, such results would lead to a digital channel model which might be developed and tested. The channel performance obviously depends on the specific modems and radio equipment used, but would provide experimental evidence to support such a model. These results may be further corroborated by measurement in the lab of the bit-error-rate versus RF signal level into the receiver. Any differences between the results would then reflect differences between the effects of amplitude fading in the lab and actual field measurements. The relative importance of other factors such as noise and interference can then be judged on this basis.

The results of this research can be used in the further development of analog channel fading simulators. Work in this direction is now underway in the mobile radio laboratory here at Carleton University. As noted previously, there remains the issue of the requisite complexity of a stochastic fading simulator. Simple Rayleigh fading and lognormal shadowing may be adequate in many situations. In other cases, it may be more appropriate to reproduce fading from actual stored data, avoiding the need to make stochastic assumptions about

fading, shadowing and terrain effects. The use of fading data recorded over specific terrain does not necessarily compromise the results of any study of various coding and modulation schemes obtained using such a simulation.

Experimental measurements of the field strength process and channel characteristics, obviously, will remain an important area of research. The variable nature of mobile radio propagation and its dependence on specific geography makes field studies an essential part of mobile radio system engineering. Further experimental work may include closer examination of the complex analog channel response. This would require specialized receiving equipment capable of quadrature demodulation, and some method of extracting the time-varying channel impulse response. Such work would lead to further understanding of the disruptions over the mobile radio channel. Increasingly sophisticated models and simulations might be developed for predicting performance, but the final evaluation will still remain in actual system field tests.

REFERENCES

- [1] P. Thorne, "Data Transmission on the Move", Canadian Electronics Engineering, May 1984.
- [2] R. Bowers, A.M. Lee and C. Hershey, Eds., Communications for a Mobile Society, Beverley Hills: Sage, 1978, Chapter 2.
- [3] International Telecommunications Union, Recommendations and Reports of the CCIR 1982, Vol.VIII: Mobile Services, Geneva, 1982.
- [4] R.H. Clarke, "A Statistical Theory of Mobile-Radio Reception", Bell Syst. Tech. J., Vol.47, July-August, 1968.
- [5] G.L. Turin, et al., "A Statistical Model of Urban Multipath Propagation", IEEE Trans. on Veh. Tech., Vol.VT-21, No.1, February 1972.
- [6] M.J. Gans, "A Power-Spectral Theory of Urban Multipath Propagation", IEEE Trans. on Veh. Tech., Vol.VT-21, No.1, February 1972.
- [7] W.C. Jakes, Ed., Microwave Mobile Communication, New York: Wiley, 1974.
- [8] W.C.Y. Lee and Y.S. Yeh, "On the Estimation of the Second-Order Statistics of Log Normal Fading in Mobile Radio Environment", IEEE Trans. on Communications, Vol.COM-22, June 1974.
- [9] D.O. Reudink, "Properties of Mobile Radio Propagation Above 400 MHz", IEEE Trans. on Veh. Tech., Vol.VT-23, No.4, November 1974.
- [10] R.C. French, "Radio Propagation in London at 462 MHz", The Radio and Electronic Engineer, Vol.46, No.7, July 1976.
- [11] H. Suzuki, "A Statistical Model for Urban Radio Propagation", IEEE Trans. on Communications, Vol.COM-25, No.7, July 1977.
- [12] H. Hashemi, "Simulation of the Urban Radio Propagation Channel", Proc. 1977 Nat. Telecomm. Conf., Paper 38-1.

- [13] R. Jensen, "900-MHz Mobile Radio Propagation in the Copenhagen Area", IEEE Trans. on Veh. Tech., Vol.VT-26, No.4, November 1977.
- [14] F. Hansen and F.I. Meno, "Mobile Fading-Rayleigh and Lognormal Superimposed", IEEE Trans. on Veh. Tech., Vol.VT-26, No.4, November 1977.
- [15] F.I. Meno, "Mobile Radio Fading in Scandinavian Terrain", IEEE Trans. on Veh. Tech., Vol.VT-26, No.4, November 1977.
- [16] R.C. French, "Error Rate Predictions and Measurements in the Mobile Radio Data Channel", IEEE Trans. on Veh. Tech., Vol.VT-27, No.3, August 1978.
- [17] T. Aulin, "Characteristics of a Digital Mobile Radio Channel", IEEE Trans. on Veh. Tech., Vol.VT-30, No.2, May 1981.
- [18] J. Zander, "A Stochastic Model of the Urban UHF Radio Channel", IEEE Trans. on Veh. Tech., Vol.VT-30, No.4, November 1981.
- [19] K. Daikoku, et al., "High-Speed Digital Transmission Experiments in 920 MHz Urban and Suburban Mobile Radio Channels", IEEE Trans. on Veh. Tech., Vol.VT-31, No.2, May 1982.
- [20] B. Compson and S.C. Bhattacharya, "Mobile Radio Communication in 800 MHz Band in Metropolitan Toronto", IEEE 1982 Veh. Tech. Conference.
- [21] H.M. Hafez, et al., "Experimental-Study of Digital Land Mobile Systems", Technical Report, Department of Communications, Canada, September 1983.
- [22] H.M. Hafez, A.U.H. Sheikh and B. McPhail, "Experimental Evaluation of Packet Error Rates for the 850 MHz Mobile Channel", Paper accepted for publication in a forthcoming issue of Proceedings of IEE in 1985.
- [23] M. Hata, "Empirical Formula for Propagation Loss in Land Mobile Radio Services", IEEE Trans. on Veh. Tech., Vol.VT-29, No.3, August 1980.

APPENDIX A

DATA ACQUISITION HARDWARE

Description

Some custom hardware was built to interface external analog and digital signals from the Racal 7-track tape recorder to the system bus of the TI 'Professional Computer'. A commercial FIR digital filter (ADF-16) is also interfaced to the system bus. Collectively this is called the digital signal processing unit, or DSPU. Figure A-1 is a block diagram of the unit. The expansion bus interface (EBI) is a circuit board which fits into the expansion bus inside the TI PC. It provides bidirectional buffering of system bus signals. The EBI is connected to an external chassis which contains the ADF-16 digital filter, digital filter interface (DFI) and power supply. The DFI performs address decoding and data buffering of signals for the ADF-16 and external inputs. External digital inputs at RS-232 levels are buffered with line receivers. The analog speed pulse and start/stop signals from the recorder are amplified and shaped by op-amp circuits, then detected with line receivers.

The DSPU is memory mapped onto the system bus in the absolute address space B0000H to B03FFH. The locations are mapped as follows: (offset from segment B0000H)

<u>Offset</u>	<u>Read/ Write</u>	<u>Description</u>
0000H	R/W	ADF-16 coefficient h(0) least sig. byte
0001H	R/W	ADF-16 coefficient h(1) most sig. byte
0002H	R/W	ADF-16 coefficient h(1) LSB
0003H	R/W	ADF-16 coefficient h(1) MSB
.	.	.
01FEH	R/W	ADF-16 coefficient h(255) LSB
01FFH	R/W	ADF-16 coefficient h(255) MSB
0200H	R	ADF-16 output Y: LSB
0200H	W	ADF-16 input X: LSB
0201H	R	ADF-16 output Y: CSB (centre byte)
0201H	W	ADF-16 input X: MSB
(ADF-16 filter cycle starts after writing to 0201H)		
0202H	W	Control register (not used)
0202H	R	ADF-16 output Y: MSB
0203H	R	Status register

The status register bits are assigned as follows:

D7 (MSB)								D0 (LSB)	
1	1	RCVD DATA	RCVD CLOCK	SPEED PULSE	START	STOP	ADF DONE

Figure A-1 Data acquisition hardware block diagram

ASSEMBLY DRAWING

DWN. T KNIGHT 9-JULY-84

DIGITAL DIGITAL PA VFD AND UNIT

EXPANSION BUS INTERFACE (TO TI PC) PART 1 OF 1

	C	B	A
1	GND3	GND2	GND1
2	VCC3	VCC2	VCC1
3	Y22	Y23	STATUS
4	Y19	Y20	Y21
5	Y16	Y17	Y18
6	Y15	RXD	RXC
7	Y12	Y13	Y14
8	Y9	Y10	Y11
9	Y6	Y7	Y8
10	Y3	Y4	Y5
11	Y0	Y1	Y2
12	H6	H7	
13	H3	H4	H5
14	H0	H1	H2
15	SELECT ST	SG	
16	ADFCCLK	-12V	+12V
17	SC		
18	H14	H15	
19	H11	H12	H13
20	H8	H9	H10
21	X6	X7	HWR
22	X3	X4	X5
23	X0	X1	X2
24	D5	D6	D7
25	D2	D3	D4
26	MRD	D0	D1
27	BA9	ENA	NHF
28	BA6	BA7	BA8
29	BA3	BA4	BA5
30	BA0	BA1	BA2
31			
32	GND6	GND5	GND4
	C	B	A

EUROCARD PIN-OUT

TABLE OF I.C.'S			
DESIGNATOR*	PART NO	+5V	GND
M7	74LS10	14	7
M15	74LS27	14	7
H29, K29	74LS32	14	7
M23	74LS74A	14	7
D30, F30	74LS138	16	8
H21, F10, F21, D10, D21, B10, B21	74LS244	20	10
H10, K10, K21	74LS374	20	10
B29	MC1489	14	7

POWER SUPPLY CONNECTIONS

UNUSED GATES

DIGITAL FILTER INTERFACE

PART 1 OF 6

RS-232 LINE RECEIVERS

TYPICAL SPEED PULSE

START PULSE

DIGITAL SIGNAL PROCESSING UNIT

DIGITAL FILTER INTERFACE

- LINE RECEIVERS

PART 6 OF 6

DES. BY DWN. T.A. KNIGHT 16-AUG-84

SYSTEMS ENGINEERING
CARLETON UNIVERSITY

OUTPUT BUFFERS

DIGITAL SIGNAL
PROCESSING UNIT

DIGITAL FILTER
INTERFACE

PART 4 of 6

DESIGNED BY: T. A. KNIGHT
10-11-84
SYSTEMS ENGINEERING
CARLETON UNIVERSITY

SPEED PULSE CONDITIONING

START/STOP PULSE SEPARATOR

DIP HEADERS

A-10

DIGITAL SIGNAL PROCESSING
UNIT

DIGITAL FILTER INTERFACE

— ANALOG INPUT CONDITIONING

PART 5 OF 6

DES: DW: T.A. KNIGHT

16-AUG-84

SYSTEMS ENGINEERING
CARLETON UNIVERSITY

RS-232 LINE RECEIVERS

TYPICAL SPEED PULSE

START PULSE

A-11

DIGITAL SIGNAL PROCESSING UNIT

DIGITAL FILTER INTERFACE

- LINE RECEIVERS

PART 6 OF 6

DES. & DWN. T.A. KNIGHT 18-AUG-74

SYSTEMS ENGINEERING
CARLETON UNIVERSITY

```

***** CSIMOD *****
MODULE NAME
CSIMOD

DESCRIPTION:
Control/status register interface.
This module contains procedures to manipulate the control
and status registers in the DSPU. The status register
holds values of external digital inputs from the 7-track
recorder. The control register may be written to from
software, allowing an external device to observe CPU
activity. This is useful for testing time-critical S/W.

PROCEDURES PROVIDED:
CSI_WAIT_START

PROCEDURES REQUIRED:
None.

PREREQUISITES:
The DSPU must be connected to the TI PC.

NOTES, ORGANIZATION AND RESTRICTIONS:
1. The DSPU is memory mapped into the TI PC at absolute
address B0000H. See module ADMOD for more details on
the DSPU. This module only uses two locations, the
control and status registers

Copyright 1984 by Tony Knight,
Dept. of Systems and Computer Engineering,
Carleton University, Ottawa, Ontario, K1S 5B6

NAME CSIMOD
DSPU Control and Status Interface Procedures

--- Linker information ---
PUBLIC CSI_WAIT_START

--- Symbolic constants ---
START_MASK EQU 00000010B

--- Declare memory-mapped DSPU segment ---
CSIMOD_ESEG SEGMENT AT 0B0000H
ORG 202H
CTRLREG DB 1 DUP (?) ; Control register - write only
STATUS DB 1 DUP (?) ; Status register - read only
CSIMOD_ESEG ENDS

--- Code segment ---
CSIMOD_CODE SEGMENT 'CODE'
PAGE
PROCEDURE
CSI_WAIT_START
DESCRIPTION:
Wait for 'START' pulse
USE
INPUT
OUTPUT

```

```

NOTES:
1. The 'START' pulse marking the beginning of a 20-second
interval of data and the 'STOP' pulse marking the end
of same are recorded as a positive and negative impulse
lasting about 600 microseconds. The DSPU hardware
separates the positive and negative pulses into digital
bits. Since the impulse 'undershoots' or rebounds in
opposite polarity, a START pulse will always be
followed closely by an apparent STOP pulse, and vice
versa. Therefore, to prevent false triggering the
STOP pulse is also checked. If one occurs, the
software ignores any START pulses for about 10 mill-
seconds to let things settle down.

VERSION HISTORY:
VER DATE AUTHOR COMMENT
0.0 DD-MM-YY T. Knight Original

--- Procedure begins ---
CSI_WAIT_START PROC FAR
ASSUME CS:CSIMOD_CODE,DS:CSIMOD_ESEG

--- Entry prolog ---
PUSH CX
DS DS
MOV AX,CSIMOD_ESEG
MOV DS,AX

--- Read status until '1' is detected ---
WS_L1: MOV AL,STATUS
AND AL,00000110B ;Check start and stop bits
JZ WS_L1 ;Wait until some activity
AND AL,00000100B ;Check if stop occurred.
JZ WS_START ;If not, must be a start.
MOV CX,4000D ;Wait 10 milliseconds
LOOP WS_WAIT
JMP WS_L1

--- Check 'START' is sustained for about 100 microseconds.
--- Pulse lasts typically 600 microseconds. ---
WS_START: MOV CX,15D
MOV AL,STATUS
AND AL,00000110B
CNP AL,00000010B
JNE WS_L1
LOOP WS_L2

--- Arrive here when a bonafide START pulse received. ---
POP DS
POP CX
RET
CSI_WAIT_START ENDP
CSIMOD_CODE ENDS
END

```


Logarithmic amplifier output [volts]	RF input level	
	[μ V]	[dBm]
0.010	0.10	-127
0.012	0.13	-125
0.017	0.18	-122
0.023	0.24	-120
0.037	0.30	-117.5
0.039	0.35	-117
0.046	0.36	-116
0.058	0.45	-114
0.074	0.59	-112
0.098	0.70	-110
0.128	0.90	-108
0.166	1.15	-106
0.200	1.42	-104
0.234	1.80	-102
0.264	2.35	-100
0.290	2.9	-98
0.324	3.6	-96
0.359	4.51	-94
0.383	5.9	-92
0.411	7.2	-90
0.432	9.0	-88
0.445	11.7	-86
0.450	14.2	-84
0.464	18.	-82
0.474	23.	-80
0.476	29.	-78
0.480	36.	-76
0.483	45.	-74
0.482?	58.	-72
0.495	63.	-70
0.497	115.	-66

Notes:

1. Logarithmic supply voltage = 65.0V
2. Radio supply voltage = 13.8 V.
3. No load on the detector (tape recorder not connected.)
4. GE mobile radio.
5. ? = questionable data point

APPENDIX C

RESAMPLING LPF FREQUENCY RESPONSES

APPENDIX D

PROGRAM LISTINGS

Source Filename	Page
MRDA.ASM	D-2
ADCMOD.ASM	D-6
CSIMOD.ASM	D-8
DOSFNS.ASM	D-9
UTLMOD.ASM	D-13
GCIMOD.ASM	D-16
BERCOM1.PAS	D-21
QRFGEN.PAS	D-25
RESAMP.PAS	D-26
ENFILT.PAS	D-30
RICIAN.PAS	D-32
DSPMOD.PAS	D-35
COVAR.PAS	D-38
INTERP.PAS	D-39

```

***** MRDA *****
PROGRAM NAME
MRDA

DESCRIPTION:
Mobile Radio Data Acquisition
This program is used to sample and store digital mobile radio
data recorded on 7-track tapes. The START signal is monitored,
to determine the beginning of a 20-second data transmission
interval. Once a START is detected, the program samples
the received data, speed pulse and analog envelope signals
synchronous with the recorded REFERENCE CLOCK. At real-time
playback (30 IPS) this corresponds to 4800 baud.

PROCEDURES PROVIDED:
None. This is a program level procedure

PROCEDURES REQUIRED:
DOS_SELECT_DISK
DOS_CREATE_FILE
DOS_CLOSE_FILE
DOS_SEQ_WRITE
DOS_SET_DTA
DOS_INIT_RETURN
DOS_EXIT_TO_DOS
DOS_GET_CHAR
DOS_DISP_STRING
AOC_INIT_GETSAM
AOC_GET_SAMPLE
CSI_WAIT_START

PREREQUISITES:
None.

NOTES, ORGANIZATION AND RESTRICTIONS
1. None

Copyright 1984 by Tony Knight,
Dept of Systems and Computer Engineering,
Carleton University, Ottawa, Ontario, K1S 5B6

*****
NAME MRDA
Mobile Radio Data Acquisition

TITLE
Mobile Radio Data Acquisition

--- Structure Definitions ---
FCBSTRUC
DRIVE DB 5
FILENAME DB '
EXT DB 'DDA'
BLOCK DB
RECORDSIZE DW
FILESIZE DW 2 DUP (?)
DATE DW
TIME DW

Default drive E
Filename, left justified
File extension
Current block
Record size
File size in bytes
Date of last update
Time of last update

```

```

RESERVED DB 8 DUP (?) ;Reserved for MS-DOS
RECORD DB ? ;Current record
RANDRECD DW 2 DUP (?) ;Random record
FCBSTRUC ENDS

--- Data Segments ---
MRDA_DATA SEGMENT 'DATA'
FCBSTRUC <
MSG1 DB 100,130,'Output filename (1-8 chars) ? $'
MSG2 DB 100,130,'Extension is DDA, DOB etc $'
MSG3 DB 100,130,'Start tape and'
MSG4 DB 'hit any key to continue $'
MSG5 DB 100,130,'Hit any key when tape finished $'
MSG6 DB 100,130,'START detected.$'
MSG7 DB 100,130,'Data saved to file:'
MSG8 DB ' $'
MRDA_DATA ENDS

ARRAY_SIZE EQU 225000 ;No of samples per array
DSEG_SIZE EQU 2*ARRAY_SIZE ;No of BYTES per array
MRDA_DSEG1 SEGMENT 'DATA'
ARRAY_1 DB DSEG_SIZE DUP (?)
MRDA_DSEG1 ENDS

MRDA_DSEG2 SEGMENT 'DATA'
ARRAY_2 DB DSEG_SIZE DUP (?)
MRDA_DSEG2 ENDS

MRDA_DSEG3 SEGMENT 'DATA'
ARRAY_3 DB DSEG_SIZE DUP (?)
MRDA_DSEG3 ENDS

MRDA_DSEG4 SEGMENT 'DATA'
ARRAY_4 DB DSEG_SIZE DUP (?)
MRDA_DSEG4 ENDS

--- Declare memory-mapped DSFU control and status registers
CSI_SEG SEGMENT AT 0B000H
ORG 202H
CTRLREG DB 1 DUP (?) ;Control register - write only
STATUS DB 1 DUP (?) ;Status register - read only
CSI_SEG ENDS

--- Symbolic constants associated with CSI_SEG registers ---
START_MASK EQU 00000010B
CLOCK_MASK EQU 00010000B
BUSY_SET EQU 1
BUSY_CLR EQU 0

--- Stack segment. SS:SP are initialized by loader. ---
STACK SEGMENT 'STACK'
DB 100H DUP (?)
STACK ENDS

--- External procedures ---
EXTRN DOS_SELECT_DISK:FAR
EXTRN DOS_CREATE_FILE:FAR
EXTRN DOS_CLOSE_FILE:FAR
EXTRN DOS_SEQ_WRITE:FAR

```



```

AND AL,CLOCK_MASK      , to go low.
JNZ L6A
;--- Set busy flag in output (control) register ---
MOV AL,BUSY_SET
MOV CTRLREG,AL
;--- Read in digital inputs from status register. ---
MOV DH,STATUS
SHL DH,1
SHL DH,1
AND DH,11110000B
CALL ADC_GET_SAMPLE
AND AH,00001111B
OR AH,DH
MOV ES:[BX],AX
INC BX
INC BX
MOV AL,BUSY_CLR
MOV CTRLREG,AL
LOOP L6

;--- Fourth segment of data acquisition --
MOV AX,MRDA_DSEC4
MOV ES,AX
MOV BX,OFFSET ARRAY_4
MOV CX,ARRAY_SIZE
;--- Wait for high-to-low clock transition --
MOV AL,STATUS
AND AL,CLOCK_MASK
JZ L7
L7:
MOV AL,STATUS
AND AL,CLOCK_MASK
JNZ L7A
;--- Set busy flag in output (control) register --
MOV AL,BUSY_SET
MOV CTRLREG,AL
;--- Read in digital inputs from status register --
MOV DH,STATUS
SHL DH,1
SHL DH,1
AND DH,11110000B
CALL ADC_GET_SAMPLE
AND AH,00001111B
OR AH,DH
MOV ES:[BX],AX
INC BX
INC BX
MOV AL,BUSY_CLR
MOV CTRLREG,AL
LOOP L7
STI

;--- Save data in disk file --
MOV AX,MRDA_DATA
MOV DS,AX
ASSUME DS:MRDA_DATA
MOV DX,OFFSET ECB1
CALL DOS_CREATE_FILE

;--- Set file control block to initial values --
MOV ECB1_BLOCK,0

```

```

MOV ECB1_RECNO,0
MOV ECB1_RECNO_SIZE,SIZE ARRAY_1

;--- Set disk transfer address and write the segment to disk --
PUSH DS
MOV AX,MRDA_DSEC1
MOV DS,AX
MOV DX,OFFSET ARRAY_1
CALL DOS_SET_DTA
POP DS
MOV DX,OFFSET ECB1
CALL DOS_SEQ_WRITE

;--- Set disk transfer address and write the segment to disk --
PUSH DS
MOV AX,MRDA_DSEC2
MOV DS,AX
MOV DX,OFFSET ARRAY_2
CALL DOS_SET_DTA
POP DS
MOV DX,OFFSET ECB1
CALL DOS_SEQ_WRITE

;--- Set disk transfer address and write the segment to disk --
PUSH DS
MOV AX,MRDA_DSEC3
MOV DS,AX
MOV DX,OFFSET ARRAY_3
CALL DOS_SET_DTA
POP DS
MOV DX,OFFSET ECB1
CALL DOS_SEQ_WRITE

;--- Set disk transfer address and write the segment to disk --
PUSH DS
MOV AX,MRDA_DSEC4
MOV DS,AX
MOV DX,OFFSET ARRAY_4
CALL DOS_SET_DTA
POP DS
MOV DX,OFFSET ECB1
CALL DOS_SEQ_WRITE

;--- Close file and print filenames --
CALL DOS_CLOSE_FILE
MOV AX,MRDA_DATA
MOV ES,AX
ASSUME ES:MRDA_DATA
MOV SI,OFFSET FILENAME
MOV DI,OFFSET MSG7
MOV CX,8

REP MOVSB
MOV SI,OFFSET EXT
MOV DI,OFFSET MSG8
MOV CX,3

REP MOVSB
MOV DX,OFFSET MSG6
CALL DOS_DISP_STRING
INC FCBI_EXT+2
;Increment file ext name

```



```

;-- End of main program loop.. Check if key pressed. --
MOV     DL,0FEH
MOV     AH,6
INT     21H ;DOS function call
JNZ     LEXIT ;Z=0 if char typed
JMP     L3

;-- Program exit --
LEXIT:  CALL DOS_EXIT_TO_DOS
        ENDP
MRDA

;-- End of code segment --
MRDA_CODE ENDS

;-- End program (program entry at MRDA) --
        END MRDA

```

```

***** ADCMOD *****
MODULE NAME
ADCMOD

DESCRIPTION:
Interface procedures to Tecmar LabMaster unit containing
analog-to-digital and digital-to-analog converters, and timers
These procedures are written for specific applications that
use the analog-to-digital converter for speech and signal
acquisition.

PROCEDURES PROVIDED
ADC_INIT_GETSAM
ADC_GET_SAMPLE

PROCEDURES REQUIRED:
None

PREREQUISITES:
The Tecmar LabMaster must be installed.

NOTES, ORGANIZATION AND RESTRICTIONS:
1. There are numerous jumpers and options selectable on the
LabMaster unit. The unit in Prof. Falconer's TI PC is the
12-bit 5712 module. Refer to LabMaster Installation Manual
and User's Guide for further details.

Some of the more important jumpers that affect this software
include.

Motherboard
SW1, SW2, SW3. Select BASE_ADR
J5: I/O mapped jumpers installed
J2: Jumper pin 3-4: 'OUT5' drives External Start Conv.

Daughterboard
SWID: Select last channel in autoIncrement mode
Set to ??

Copyright 1984 by Tony Knight
Dept of Systems and Computer Engineering
Carleton University, Ottawa, Ontario, K1S 5B6

*****
NAME ADCMOD
Analog-to-digital Conversion Procedures

Linker information
PUBLIC ADC_INIT_GETSAM
PUBLIC ADC_GET_SAMPLE

SYMBOLIC CONSTANTS
I/O addresses to LabMaster
BASE_ADR EQU 0C0H
DAC0LSB EQU BASE_ADR
DAC0MSB EQU BASE_ADR+1
DAC1LSB EQU BASE_ADR+2
DAC1MSB EQU BASE_ADR+3
ADC_CTRL EQU BASE_ADR+4
ADC_STATUS EQU BASE_ADR+4
ADC_CHAN EQU BASE_ADR+5
ADC_LSB EQU BASE_ADR+5

Base I/O address.
DAC channel 0 low byte
DAC chan 0 high byte (nibble)
DAC channel 1 low byte
DAC chan 1 high byte (nibble)
ADC control (W)
ADC status (R)
ADC channel number (W)
ADC low byte (R)

```

```

ADC_MSB EQU BASE_ADR+6 ;ADC most significant 4 bits (R)

--- Immediate operands ---
CMD_BYTE1 EQU 80H

STARTCONV EQU 84H
CHANNEL0 EQU 0
TSIDONE EQU 80H

--- Code segment ---
ADCMOD_CODE SEGMENT 'CODE'

PAGE
***** ADC_INIT_GETSAM *****
PROCEDURE:
ADC_INIT_GETSAM

DESCRIPTION:
Initialize analog-to-digital converter for sampling
channel zero under software control.

USE:
Assembler:
CALL ADC_INIT_GETSAM

Pascal declaration.
PROCEDURE ADC_INIT_GETSAM; EXTERN;

INPUT:
None.

OUTPUT:
None.

NOTES:
None.

VERSION HISTORY:
VER DATE AUTHOR COMMENT
0 0 22-8-84 T. Knight Original

--- Procedure begins ---
ADC_INIT_GETSAM PROC FAR
ASSUME CS:ADCMOD_CODE

--- Disable autoIncrementing mode, external start and interrupts
MOV AL, CMD_BYTE1
OUT ADC_CTRL, AL
IN AL, ADC_MSB ;Clear done flag

--- Select channel zero ---
MOV AL, CHANNEL0
OUT ADC_CHAN, AL

--- Return ---
RET

ADC_INIT_GETSAM ENDP

PAGE
***** ADC_GET_SAMPLE *****
PROCEDURE:
ADC_GET_SAMPLE

```

```

* DESCRIPTION:
* Perform a single analog-to-digital conversion of channel
* zero and return the result in AX
*
* USE:
* Assembler:
* CALL ADC_GET_SAMPLE
*
* Pascal declaration:
* FUNCTION ADC_GET_SAMPLE: INTEGER; EXTERN,
*
* INPUT:
* None.
*
* OUTPUT:
* AX = 12-bit conversion result, returned as 16-bit integer.
*
* NOTES:
* 1. Conversion takes approximately 30 microseconds.
*
* VERSION HISTORY:
* VER DATE AUTHOR COMMENT
* 0.0 22-8-84 T. Knight Original
*
* --- Procedure begins ---
ADC_GET_SAMPLE PROC FAR
ASSUME CS:ADCMOD_CODE
*
* --- Start conversion ---
OUT ADC_MSB,AL ;AL = anything
*
* --- Wait for done bit, and then read result ---
IN AL,ADC_STATUS
AND AL,TSTDONE
JZ CSLOOP
IN AL,ADC_MSB ;Get high byte
OUT DAC0MSB,AL ;Send to DAC
MOV AH,AL
IN AL,ADC_LSB ;Get low byte
OUT DAC0LSB,AL ;Send to DAC
*
* --- Return ---
RET
ADC_GET_SAMPLE ENDP
*
* --- End of code segment ---
ADCMOD_CODE ENDS
*
* --- End module ---
END

```

```

***** CSIMOD *****
MODULE NAME:
CSIMOD

DESCRIPTION:
Control/status register interface.
This module contains procedures to manipulate the control
and status registers in the DSPU. The status register
holds values of external digital inputs from the 7-track
recorder. The control register may be written to from
software, allowing an external device to observe CPU
activity. This is useful for testing time-critical S/W.

PROCEDURES PROVIDED:
CSI_WAIT_START

PROCEDURES REQUIRED:
None.

PREREQUISITES:
The DSPU must be connected to the TI PC

NOTES, ORGANIZATION AND RESTRICTIONS
1. The DSPU is memory mapped into the TI PC at absolute
address B0000H. See module ADEMOD for more details on
the DSPU. This module only uses two locations, the
control and status registers

Copyright 1984 by Tony Knight,
Dept. of Systems and Computer Engineering,
Carleton University, Ottawa, Ontario, K1S 5S6

NAME CSIMOD
DSPU Control and Status Interface Procedures

--- Linker information ---
PUBLIC CSI_WAIT_START

--- Symbolic constants ---
START_MASK EQU 00000010B

--- Declare memory-mapped DSPU segment ---
CSIMOD_ESEG SEGMENT AT 0B0000H
ORG 202H
CTRLREG DB 1 DUP {?} Control register - write only
STATUS DB 1 DUP {?} Status register - read only
CSIMOD_ESEG ENDS

--- Code segment ---
CSIMOD_CODE SEGMENT 'CODE'
PAGE
PROCEDURE CSI_WAIT_START
DESCRIPTION:
Wait for 'START' pulse
USE
INPUT
OUTPUT

```

```

NOTES:
1. The 'START' pulse marking the beginning of a 20-second
interval of data and the 'STOP' pulse marking the end
of same are recorded as a positive and negative impulse
lasting about 600 microseconds. The DSPU hardware
separates the positive and negative pulses into digital
bits. Since the impulse 'undershoots' or rebounds in
opposite polarity, a START pulse will always be
followed closely by an apparent STOP pulse, and vice
versa. Therefore, to prevent false triggering, the
STOP pulse is also checked. If one occurs, the
software ignores any START pulses for about 10 milli-
seconds to let things settle down.

VERSION HISTORY:
VER DATE AUTHOR COMMENT
0.0 DD-MM-YY T. Knight Original

--- Procedure begins ---
CSI_WAIT_START PROC FAR
ASSUME CS:CSIMOD_CODE,DS:CSIMOD_ESEG

--- Entry prolog ---
PUSH CX
DS DS
MOV AX,CSIMOD_ESEG
DS,AX

--- Read status until '1' is detected ---
WS_L1:
MOV AL,STATUS
AND AL,00000110B
JZ WS_L1
AND AL,00000100B
JZ WS_START
MOV CX,4000H
LOOP WS_WAIT
JMP WS_L1

--- Check 'START' is sustained for about 100 microseconds.
--- Pulse lasts typically 600 microseconds ---
WS_L2:
MOV CX,15D
MOV AL,STATUS
AND AL,00000110B
CMP AL,000000010B
JNE WS_L1
LOOP WS_L2

--- Arrive here when a bonafide START pulse received. ---
POP DS
POP CX
RET
CSI_WAIT_START ENDP
CSIMOD_CODE ENDS
END

```



```

MOV     AH,16H      create file request
INT     DOS_FCN
RET
--- END OF PROCEDURE ---
DOS_CREATE_FILE ENDP
PAGE
PROCEDURE
DOS_OPEN_FILE
***** DOS_OPEN_FILE *****
DESCRIPTION:
Open file specified by FCB, pointed to by DS:DX. The diskette
directory is searched for the named file. If found, AL returns
FFH and the FCB is updated as follows. If the drive code was 0
(Default drive), it is changed to the actual drive used (A=1,
B=2, etc.). This allows changing the default drive without
interfering with subsequent operations on this file. The high
byte of the current block field is set to zero. The record size,
time and date are updated with values obtained from the directory
USE:
Callbale from assembler only.
INPUT:
DS:DX = address of unopened FCB
OUTPUT:
register AL = 00 (success - file opened)
FF (failure - file not found)
1. All registers except AL preserved
2. Stack is not used
VERSION HISTORY
VER DATE AUTHOR COMMENT
0 0 DD-MM-YY TAK Original
*****
--- PROCEDURE BEGINS --- FAR
DOS_OPEN_FILE PROC ASSUME CS:DOSSENS_CODE
MOV     AH,0FH      Open file request
INT     DOS_FCN
RET
--- END OF PROCEDURE ---
DOS_OPEN_FILE ENDP
PAGE
PROCEDURE
DOS_CLOSE_FILE
***** DOS_CLOSE_FILE *****
DESCRIPTION:
Close file specified by FCB. DS:DX points to an open FCB. The
diskette directory is searched, and if the file is found, its
position is compared with that kept in the FCB. The directory
is updated with the information in the FCB
USE:
Callbale from assembler only
INPUT:
DS:DX = address of open FCB
OUTPUT:

```

```

register AL = 00 (success - file closed)
FF (failure - file not found)
1. All registers except AL preserved.
2. Stack is not used.
VERSION HISTORY:
VER DATE AUTHOR COMMENT
0 0 DD-MM-YY TAK Original
*****
--- PROCEDURE BEGINS --- FAR
DOS_CLOSE_FILE PROC ASSUME CS:DOSSENS_CODE
MOV     AH,10H      Close file request
INT     DOS_FCN
RET
--- END OF PROCEDURE ---
DOS_CLOSE_FILE ENDP
PAGE
PROCEDURE
DOS_SEQ_READ
***** DOS_SEQ_READ *****
DESCRIPTION:
Sequential read. DS:DX points to an opened FCB. The record
addressed by the current block (FCB bytes 0C-0D) and current
record (FCB byte 1FH) is loaded at the diskette transfer address
(DTA), then the record address is incremented
USE:
Callbale from assembler only.
INPUT:
DS:DX = address of opened FCB
OUTPUT:
register AL = 00H - Successful read
01H - No data in record.
02H - Not enough room in DTA segment for one record
transfer was aborted
03H - Partial record read and filled out with zeros.
NOTES:
1. All registers except AL preserved
2. Stack is not used.
VERSION HISTORY
VER DATE AUTHOR COMMENT
0 0 DD-MM-YY TAK Original
*****
--- PROCEDURE BEGINS --- FAR
DOS_SEQ_READ PROC ASSUME CS:DOSSENS_CODE
MOV     AH,14H      Sequential read request
INT     DOS_FCN
RET
--- END OF PROCEDURE ---
DOS_SEQ_READ ENDP
PAGE

```

```

***** DOS_SEQ_WRITE *****
PROCEDURE
DOS_SEQ_WRITE

DESCRIPTION:
Sequential write DS:DX points to an opened FCB. The record
addressed by the current block and record fields is written from
the diskette transfer address to disk. If a record is less than
a sector size, it is buffered until a sector of data is
accumulated, and then written. The record address is then
incremented.

USE:
Callabale from assembler only.

INPUT:
DS:DX = address of unopened FCB

OUTPUT:
register AL = 00 - Successful write
01H - Diskette full.
02H - Insufficient room in diskette transfer
segment for one record. Transfer aborted.

1. All registers except AL preserved

VERSION HISTORY:
VER DATE AUTHOR COMMENT
0 0 DO-MM-YY TAK Original

--- PROCEDURE BEGINS ---
DOS_SEQ_WRITE PROC FAR
ASSUME CS DOSSENS_CODE
MOV AH,15H
INT DOS_FCN
RET
END OF PROCEDURE
DOS_SEQ_WRITE ENDP
PAGE
***** DOS_SEQ_DTA *****
PROCEDURE
DOS_SEQ_DTA
DESCRIPTION
Set diskette transfer address to value in DS DX
This is used in sequential and random read/write operations
USE:
Callabale from assembler only
INPUT
DS:DX = Diskette transfer address (DTA)
OUTPUT
None
NOTES
1 All registers preserved
2 MS-DOS will not allow data transfers to wrap around within
a segment or to overflow into the next segment.
VERSION HISTORY:
VER DATE AUTHOR COMMENT

```

```

***** 0 0 DO-MM-YY TAK Original *****
--- PROCEDURE BEGINS ---
DOS_SET_DTA PROC FAR
ASSUME CS DOSSENS_CODE
MOV AH,1AH
INT DOS_FCN
RET
END OF PROCEDURE
DOS_SET_DTA ENDP
PAGE
***** DOS_INIT_RETURN *****
PROCEDURE
DOS_INIT_RETURN
DESCRIPTION:
Save return address of program segment header for use in
DOS_EXIT_TO_DOS. On entry to a program DS and ES contain
the segment of the program segment header.
USE:
Callabale from assembler only
INPUT:
None.
OUTPUT:
None.
NOTES:
VERSION HISTORY:
VER DATE AUTHOR COMMENT
0 0 DO-MM-YY TAK Original
--- PROCEDURE BEGINS ---
DOS_INIT_RETURN PROC FAR
ASSUME CS DOSSENS_CODE
MOV PCN_SEG_HDR_CS,ES
RET
END OF PROCEDURE
DOS_INIT_RETURN ENDP
PAGE
***** DOS_EXIT_TO_DOS *****
PROCEDURE
DOS_EXIT_TO_DOS
DESCRIPTION:
Terminate program and transfer control to MS-DOS All files
should be closed before doing this
Return to DOS is performed by jumping to INT 20 instruction
in the program segment header
USE
Callabale from assembler only
INPUT
None

```

DOSENS_CODE ENDS
 --- END MODULE ---
 END

```

*****
** OUTPUT
** None.
**
** NOTES
** 1. DOS_INIT_RETURN must have been called on entry to program
**    before this procedure is called
** 2. This procedure does not return
**
** VERSION HISTORY:
** VER  DATE  AUTHOR  COMMENT
** 0.0  00-00-YY  TAY  Original
*****
*** PROCEDURE BEGINS --- FAR
DOS_EXIT_TO_DOS_PROC ASSUME CS:DOSSENS_CODE
MOV  AX, PGM_SEG_HDR_CS  Get program seg header addr
PUSH AX
SUB  AX, AX               Push offset = 0000H
PUSH AX
RET                       Return to program seg header

*** END OF PROCEDURE ---
DOS_EXIT_TO_DOS ENDP

*** PROCEDURE BEGINS --- FAR
DOS_GET_CHAR PROC CS:DOSSENS_CODE
ASSUME CS:DOSSENS_CODE
MOV  AH, 01H
INT  DOS_FCN
RET

*** END OF PROCEDURE ---
DOS_GET_CHAR ENDP

*** PROCEDURE BEGINS --- FAR
DOS_DISP_CHAR PROC CS:DOSSENS_CODE
ASSUME CS:DOSSENS_CODE
MOV  AH, 02H
INT  DOS_FCN
RET

*** END OF PROCEDURE ---
DOS_DISP_CHAR ENDP

*** PROCEDURE BEGINS --- FAR
DOS_DISP_STRING PROC CS:DOSSENS_CODE
ASSUME CS:DOSSENS_CODE
MOV  AH, 09H
INT  DOS_FCN
RET

*** END OF PROCEDURE ---
DOS_DISP_STRING ENDP

*** PROCEDURE BEGINS --- FAR
DOS_PRINT_CHAR PROC CS:DOSSENS_CODE
ASSUME CS:DOSSENS_CODE
MOV  AH, 5
INT  DOS_FCN
RET

*** END OF PROCEDURE ---
DOS_PRINT_CHAR ENDP

*** END OF SEGMENT ---

```



```

***** UTLMOD *****
MODULE NAME
UTLMOD

DESCRIPTION
  Utilities Module
  This module contains procedures callable from Pascal
  or assembler for performing bit manipulation of data
  that cannot be done conveniently in Pascal.

PROCEDURES PROVIDED
  UTL_UNPACK_DATA
  UTL_PACK_DATA
  UTL_UNPACK_ENVELOPE

PROCEDURES REQUIRED
  None

PREREQUISITES
  None

NOTES, ORGANIZATION AND RESTRICTIONS
  None

*****
Copyright 1984 by Tony Knight
Dept of Systems and Computer Engineering,
Carleton University, Ottawa, Ontario, K1S 5B6

*****
TITLE      NAME      UTLMOD
Utility Procedures

-- Linker information --
PUBLIC UTL_UNPACK_DATA
PUBLIC UTL_PACK_DATA
PUBLIC UTL_UNPACK_ENVELOPE

-- Symbolic constants --

-- Code segment --
UTLMOD_CODE SEGMENT 'CODE'
***** UTL_UNPACK_DATA *****
PROCEDURE:
  UTL_UNPACK_DATA

DESCRIPTION
  Unpacks one 16-bit data element into a 16 bit integer
  envelope, plus four binary data bits, each stored as bytes

USE
  Pascal declaration:
  PROCEDURE UTL_UNPACK_DATA(
    packed_data: INTEGER;
    VAR rx_data: BYTE;
    VAR speed_pulse: BYTE;
    VAR D12: BYTE;
    VAR D14: BYTE;)
    (* IN *)
    (* OUT *)
    (* OUT *)
    (* OUT *)
    (* OUT *)
    (* OUT *)

INPUT
  packed_data, 16-bit data element packed as follows:
  (MSB)
  D15 = received data
  D14 = 1 RAM DATA Reference clock (always 0
  since data sampled on high-to-low

```

```

transition of clock)
2 Error bit after correlation with
reference sequence.
D13 = speed pulse bit
D12 = 1 RAW DATA START pulse - may be '1'
near beginning of a 20-sec
interval)
2 Error bit valid - indicates bit-error
processor successfully synchronized
to reference and computed valid error
occurrences.

D11..D0 (LSB) = 12-bit signed integer envelope
data

OUTPUT:
envelope = 12-bit ADC sample sign-extended to 16-bits
rx_data = received digital data, 0..1 subrange of type
byte.
speed_pulse = 1, speed pulse is occurring, else 0.
Subrange 0..1 of BYTE. This is not synchronized
to anything, and lasts up to 40 milliseconds or
about 80 data samples. There may several pulses
adjacent in this interval.
D12 = Error bit valid
D14 = Error bit, '1' indicates data is in error
*****
-- Declare structure of stack on entry --
UNPACKD_FRAME STRUCT
UNPACKD_BP DW ?
UNPACKD_RET DW ?
UNPACKD_D14 DW ?
UNPACKD_D12 DW ?
UNPACKD_PULS DW ?
UNPACKD_RXD DW ?
UNPACKD_ENVEL DW ?
UNPACKD_DAT DW ?
UNPACKD_FRAME ENDS

-- Procedure begins --
UTL_UNPACK_DATA PROC FAR
ASSUME CS:UTLMOD_CODE
-- Uses Pascal data segment --

Entry prolog:
PUSH BP
MOV BP, SP

Get packed data:
MOV AX, [BP]
MOV DX, AX
UNPACKD_DAT

Compute envelope:
AND AX, 0FFFFH
CMP AX, 07FEFH
JLE SHORT UNP_L1
OR AX, 0F000H
MOV BX, [BP]
MOV [BX], AX
UNPACKD_ENVEL

Compute received data:
ROL DX, 1
MOV AL, DL
AND AL, 1
Rotate data into low bit
Clear higher bits

```



```

* packed_data: 16-bit data element packed as above
*
* OUTPUT
* envelope = 12-bit ADC sample sign-extended to 16-bits
*
*****
--- Declare structure of stack on entry ---
UNPKENVL_FRAME STRUC
UNPKENVL_BP DW ? Previous base pointer
UNPKENVL_RET DD ? Return address
UNPKENVL_ENVEL DW ? Pointer to envelope data
UNPKENVL_DAT DW ? Packed data value
UNPKENVL_FRAME ENDS

--- Procedure begins ---
UTL_UNPACK_ENVELOPE PROC FAR
ASSUME CS,UTLMOD_CODE
--- Uses Pascal data segment ---

--- Entry prolog ---
PUSH BP
MOV BP,SP

--- Get packed data ---
MOV AX,[BP].UNPKENVL_DAT
MOV DX,AX

--- Compute envelope ---
AND AX,0FEFH Mask top nibble
CMP AX,07EEH Figure out sign
JLE SHORT UNPKENV_L1
OR AX,0F000H
UNPKENV_L1: MOV BX,[BP].UNPKENVL_ENVEL
MOV [BX],AX

--- Clean up and return ---
POP BP
RET 4D Discard params on stack
UTL_UNPACK_ENVELOPE ENDP

--- End code segment ---
UTLMOD_CODE ENDS
--- END MODULE ---
END

```

```

***** OCIMOD *****
** MODULE NAME:
** OCIMOD
**
** DESCRIPTION:
** Graphics Controller Interface Module
** This module contains several procedures callable from assembler
** or Pascal for interacting with the graphics video board in the
** TI PC
**
** PROCEDURES PROVIDED
** OCL_INIT_BW
** OCL_SET_PLANE
** OCL_CLEAR_PLANE
** OCL_BORDER
** OCL_XLINE
** OCL_YLINE
**
** PROCEDURES REQUIRED
** None
**
** PREREQUISITES
** Graphics video controller board must be installed.
**
** NOTES, ORGANIZATION AND RESTRICTIONS:
** 1 This module chiefly supports only black and white graphics
** Colour graphics requires more sophisticated use of the
** attribute latches than is provided here
**
** Copyright 1984 by Tony Knight,
** Dept of Systems and Computer Engineering,
** Carleton University, Ottawa, Ontario, K1S 5B6
**
** NAME OCIMOD
** GRAPHICS PROCEDURES
**
** Linker information
** PUBLIC OCL_INIT_BW
** PUBLIC OCL_SET_PLANE
** PUBLIC OCL_CLEAR_PLANE
** PUBLIC OCL_BORDER
** PUBLIC OCL_XLINE
** PUBLIC OCL_YLINE
**
** Symbolic constants
** MONOCHROME_C EQU 0FEH
** PLANE_SIZE_C EQU 13800H
**
** Palette latch value,
** Screen size in words.
**
** The graphics video controller board has three bit-mapped or
** attribute bit memories designated A, B and C each mapping 27600
** bytes of RAM to 720 horizontal by 300 vertical pixels. Three
** 8-bit colour palette latches map the attribute bits for a given
** pixel to a colour or gray level. The attribute bits are displayed
** with 'C' having the highest priority, and 'A' the lowest. See the
** Technical Reference Manual, Section 2.10, for further details
**
** Declare graphics display RAM segments

```

```

** PLANE_A SEGMENT AT 0C000H
** DW PLANE_SIZE_C DUP(?)
** ENDS
**
** PLANE_B SEGMENT AT 0C800H
** DW PLANE_SIZE_C DUP(?)
** ENDS
**
** PLANE_C SEGMENT AT 0D000H
** DW PLANE_SIZE_C DUP(?)
** ENDS
**
** PALETTE_LATCHES SEGMENT AT 0DE00H
** ORG 10H
** DB ?
** ORG 20H
** DB ?
** ORG 30H
** DB ?
** PALETTE_LATCHES ENDS
**
** -- CODE SEGMENT --
** OCIMOD_CODE SEGMENT 'CODE'
**
** -- Some constant data --
**
** -- Module static variables in code segment --
** ACTIVE_PLANE DW ?
** PAGE
**
** ***** OCL_INIT_BW *****
** PROCEDURE
** OCL_INIT_BW
**
** DESCRIPTION:
** Initialize graphics for monochrome (black and white) display.
** Is a '1' bit on any plane will be displayed at full intensity.
** Set active plane to 'A'
**
** USE
** Assembler
** CALL OCL_INIT_BW
**
** Pascal declaration
** PROCEDURE OCL_INIT_BW; EXTERN
**
** INPUT None
**
** OUTPUT None
**
** VERSION HISTORY
** VER DATE AUTHOR COMMENT
** 0 0 05-07-84 T Knight Original
**
** -- Procedure begins --
** OCL_INIT_BW PROC CS; OCIMOD_CODE
** ASSUME CS; OCIMOD_CODE
**
** -- Init palette latches for monochrome display --
** RS
** PUSH AX
** MOV ES, AX
** ASSUME ES; PALETTE_LATCHES
** MOV AL, MONOCHROME_C

```

```

MOV RED_LATCH,AL
MOV GREEN_LATCH,AL
MOV BLUE_LATCH,AL

--- Init active plane
MOV AX,PLANE_A
MOV ES,AX
ASSUME ES:PLANE_A
MOV ACTIVE_PLANE,AX
;Get segment addr.
;Make plane addressable
;Initialize active plane

--- Return ---
POP ES
RET
ENDP
PAGE
***** OCI_SET_PLANE *****
PROCEDURE OCI_SET_PLANE
DESCRIPTION:
  Set active plane to A, B or C
USE:
  Assembler.
  PUSH PLANE_VAL
  CALL OCI_INIT_BW
  Pascal declaration.
  PROCEDURE OCI_SET_PLANE (PLANE_VAL, INTEGER); EXTERN.
INPUT:
  PLANE_VAL = 0 - Plane A
             1 - Plane B
             2 - Plane C
OUTPUT: None
VERSION HISTORY:
  VER DATE AUTHOR COMMENT
  0 0 05-07-84 T Knight Original
*****
--- Declare structure of stack on entry ---
SETPL_FRAME STRUC
SETPL_BP DW
SETPL_RET DD
SETPL_VAL DW
SETPL_FRAME ENDS
--- Procedure begins ---
OCI_SET_PLANE PROC FAR
ASSUME CS:OCIMOD_CODE
--- Entry prolog ---
PUSH BP
MOV BP,SP
--- Get parameter ---
MOV AX,[BP] SETPL_VAL
--- Case on value and set active plane value ---
CMP AX,0
JNE SET_L1
MOV AX,PLANE_A
MOV ACTIVE_PLANE,AX

```

```

SET_L1 SET_EX
JMP SET_L1
CMP AX,1
JNE SET_L2
MOV AX,PLANE_B
MOV ACTIVE_PLANE,AX
SET_EX
CMP AX,2
JNE SET_L3
MOV AX,PLANE_C
MOV ACTIVE_PLANE,AX
--- Clean up and return ---
POP BP
SET_EX
RET 4
;Discard 4 bytes of stack

--- End procedure ---
OCI_SET_PLANE ENDP
PAGE
***** OCI_CLEAR_PLANE *****
PROCEDURE OCI_CLEAR_PLANE
DESCRIPTION:
  Clear currently active plane
USE:
  Assembler.
  CALL OCI_CLEAR_PLANE
  Pascal declaration
  PROCEDURE OCI_CLEAR_PLANE, EXTERN.
INPUT:
  None
OUTPUT:
  None
VERSION HISTORY:
  VER DATE AUTHOR COMMENT
  0 0 05-07-84 T Knight Original
*****
--- Procedure begins ---
OCI_CLEAR_PLANE PROC FAR
ASSUME CS:OCIMOD_CODE
--- Get active plane and make it addressable ---
PUSH ES
MOV AX,ACTIVE_PLANE
MOV ES,AX
--- Clear plane ---
MOV AX,0
MOV BX,AX
MOV CX,PLANE_SIZE_C
MOV ES:[BX],AX
ADD BX,2
LOOP CLR_L1
CLR_L1
--- Return ---
POP ES
RET
OCI_CLEAR_PLANE ENDP
PAGE

```

```

***** OCI_BORDER *****
PROCEDURE
  OCI_BORDER
DESCRIPTION
  Draw border around edge of active plane
USE:
  Assembler
  CALL OCI_BORDER
Pascal declaration:
  PROCEDURE OCI_BORDER, EXTERN.
INPUT:
  None
OUTPUT:
  None
VERSION HISTORY
  VER DATE AUTHOR COMMENT
  0 0 05-07-84 T Knight Original
--- Procedure begins ---
OCI_BORDER PROC CS OCIMOD_CODE
ASSUME
  PUSH ES
  MOV AX, ACTIVE_PLANE
  MOV ES, AX
--- Draw top border ---
  MOV AX, 0FFFFH
  MOV BX, 0
  MOV CX, 45D
  MOV ES, [BX], AX
  ADD BX, 2
  LOOP BDR_L1
--- Bottom border ---
  MOV BX, 27508D
  MOV CX, 45D
  MOV ES, [BX], AX
  ADD BX, 2
  LOOP BDR_L2
--- Left border ---
  MOV AX, 8000H
  MOV BX, 92D
  MOV CX, 298D
  MOV ES, [BX], AX
  ADD BX, 92D
  LOOP BDR_L3
--- Right border ---
  MOV AX, 1
  MOV BX, 180D
  MOV CX, 298D
  MOV ES, [BX], AX
  ADD BX, 92D
  LOOP BDR_L4
  
```

```

--- Return ---
  POP ES
  RET
PAGE
***** OCI_XLINE *****
PROCEDURE
  OCI_XLINE
DESCRIPTION:
  Draw a horizontal line
USE:
  Assembler
  PUSH X1
  PUSH X2
  PUSH Y
  CALL OCI_XLINE
Pascal declaration
  PROCEDURE OCI_XLINE(X1, X2, Y INTEGER); EXTERN.
INPUT:
  X1 = Horizontal coordinate of endpoint (0 <= X1 <= 719)
  X2 = Horizontal coordinate of endpoint (0 <= X2 <= 719)
  Y = Vertical position of line (0 <= Y <= 299)
  X, Y coordinates have the usual first quadrant orientation
  with (0,0) in the lower left corner. Beware No range
  checking or clipping is performed.
OUTPUT: None
VERSION HISTORY:
  VER DATE AUTHOR COMMENT
  0 0 05-07-84 T Knight Original
--- Declare structure of stack on entry ---
XLINE_FRAME STRUC
XLINE_BP DW ? ;Previous stack marker
XLINE_RET DD ? ;Return address
XLINE_Y DW ? ;Y (horizontal) value
XLINE_X2 DW ? ;X-axis endpoint
XLINE_X1 DW ? ;Other endpoint
XLINE_FRAME ENDS
--- Some constant data ---
BEG_LOAD
11111111111111111111B
01111111111111111111B
00111111111111111111B
00011111111111111111B
00001111111111111111B
00000111111111111111B
00000011111111111111B
00000001111111111111B
00000000111111111111B
00000000011111111111B
00000000001111111111B
00000000000111111111B
00000000000011111111B
00000000000001111111B
00000000000000111111B
00000000000000011111B
00000000000000001111B
00000000000000000111B
00000000000000000011B
00000000000000000001B
  
```

```

END_LOAD      DW 10000000000000000000B
               DW 11000000000000000000B
               DW 11100000000000000000B
               DW 11110000000000000000B
               DW 11111000000000000000B
               DW 11111100000000000000B
               DW 11111110000000000000B
               DW 11111111000000000000B
               DW 11111111100000000000B
               DW 11111111110000000000B
               DW 11111111111000000000B
               DW 11111111111100000000B
               DW 11111111111110000000B
               DW 11111111111111000000B
               DW 11111111111111100000B
               DW 11111111111111110000B
               DW 11111111111111111000B
               DW 11111111111111111100B
               DW 11111111111111111110B
               DW 11111111111111111111B

--- Static variables ---
Y_OFFSET      DW 0
BEG_WORD      DW ?
END_WORD      DW ?

--- Register use ---
;AX = working, BX = offset into graphics RAM.
;CX = counter, DX = not used.
;DI = beginning offset in bytes.
;SI = ending
;BP = base pointer to parameters on stack

--- Procedure begins ---
OCI_XLINE     PROC FAR
               ASSUME CS:OCIMOD_CODE

--- Entry prolog ---
               PUSH BP
               MOV BP, SP
               ;Save base pointer
               ;Use BP to access params
               PUSH ES
               MOV AX, ACTIVE_PLANE
               ;Address current graphic plane
               MOV ES, AX

--- Compute Y offset = (299-Y)*92
               MOV AX, 2990
               SUB AX, [BP+].XLINE_Y
               MOV BX, 92D
               MUL BX
               MOV Y_OFFSET, AX
               ;Result in AX

--- Compute beginning and end word in graphics RAM, and remainders
               MOV AX, [BP+].XLINE_X1
               MOV BX, [BP+].XLINE_X2
               CMP BX, AX
               JGE XLINE_L1
               XCHG AX, BX
               MOV CX, AX
               MOV AX, 1
               SHR AX, 1
               SHR AX, 1
               SHR AX, 1
               MOV BEC_WORD, AX
               MOV AX, 1
               SHL AX, 1
               SHL AX, 1
               SHL AX, 1
               SHL AX, 1
               SUB CX, AX
               SHL CX, 1
               MOV DI, CX

XLINE_L1:     ;Save beginning (left) coord
               ;Compute beginning word
               ;Compute remainder
               ;CX = remainder in pixels
               ;Convert to offset in bytes
               ;Save it

```

```

MOV CX, BX
SHR BX, 1
SHR BX, 1
SHR BX, 1
SHR BX, 1
MOV END_WORD, BX
SHL BX, 1
SHL BX, 1
SHL BX, 1
SHL BX, 1
SHL CX, BX
SUB CX, 1
MOV SI, CX

--- Compare begin/end word to determine fill requirement. If
--- BEC_WORD = END_WORD then only one word written to graphic RAM ---
MOV AX, BEC_WORD
CMP AX, END_WORD
JF SHORT XLINE_L5 ;Only one word to write.

--- Else, two or more words to compute and display ---
MOV BX, BEC_WORD
SHL BX, 1
ADD BX, Y_OFFSET
MOV AX, ES:[BX]
OR AX, BEC_LOAD[DI]
MOV ES:[BX], AX
;Get current screen contents
;Turn on new bits
;and save it

--- Turn on all bits in the intervening words (if any) ---
MOV CX, END_WORD
SUB CX, BEC_WORD
DEC CX
JZ SHORT XLINE_L3 ;Skip fill, if beg/end adjacent
MOV AX, 0FFFFH
INC BX
INC BX
MOV ES, [BX], AX
XLINE_L2:     LOOP XLINE_L2
INC BX
INC BX

--- Do last (rightmost) word ---
MOV AX, ES:[BX]
OR AX, END_LOAD[SI]
MOV ES:[BX], AX
JMP SHORT XLINE_EX ;Exit

--- Handle special case of writing single word to graphics RAM ---
XLINE_L5:     MOV BX, BEC_WORD
               SHL BX, 1
               ADD BX, Y_OFFSET
               MOV AX, BEC_LOAD[DI]
               AND AX, END_LOAD[SI]
               OR AX, ES:[BX]
               MOV ES:[BX], AX
               ;Compute bits to turn on
               ;OR it with current screen
               ;and save it.

--- Epilog ---
XLINE_EX:     POP ES
               POP BP
               RET 6
               ;Discard 6 bytes of stack

--- End of procedure ---
OCI_XLINE     ENDP
               PAGE
               ***** OCI_XLINE *****
               ; PROCEDURE

```

```

* OCL_YLINE
* DESCRIPTION:
*   Draw a vertical line
*
* USE:
*   Assembler
*   PUSH Y1
*   PUSH Y2
*   PUSH X
*   CALL OCL_YLINE
*
* Pascal declaration:
*   PROCEDURE OCL_YLINE (Y1, Y2, X: INTEGER); EXTERNAL
*
* INPUT:
*   Y1 = Vertical coordinate of endpoint (0 <= Y1 <= 299)
*   Y2 = Vertical coordinate of endpoint (0 <= Y2 <= 299)
*   X = Horizontal position of line (0 <= X <= 719)
*
* X, Y coordinates have the usual first quadrant orientation
* with (0,0) in the lower left corner. Beware. No range
* checking or clipping is performed
*
* OUTPUT: None
*
* VERSION HISTORY:
*   VER DATE AUTHOR COMMENT
*   0 0 05-07-84 T Knight Original
*
* --- Declare structure of stack on entry ---
YLINE_FRAME STRUCT
YLINE_BP DW ? 'Saved base pointer'
YLINE_RET DW ? 'Saved return address'
YLINE_X DW ? 'X-coordinate of line'
YLINE_Y2 DW ? 'Y-axis endpoint'
YLINE_Y1 DW ? 'Other endpoint'
YLINE_FRAME ENDS

--- Constant data ---
Y_LOAD DW 1000000000000000000B 'Load values'
DW 0100000000000000000B
DW 0010000000000000000B
DW 0001000000000000000B
DW 0000100000000000000B
DW 0000010000000000000B
DW 0000001000000000000B
DW 0000000100000000000B
DW 0000000010000000000B
DW 0000000001000000000B
DW 0000000000100000000B
DW 0000000000010000000B
DW 0000000000001000000B
DW 0000000000000100000B
DW 0000000000000010000B
DW 0000000000000001000B
DW 0000000000000000100B
DW 0000000000000000010B
DW 0000000000000000001B

--- Procedure begins ---
OCL_YLINE PROC FAR
ASSUME CS:OCLMOD_CODE

--- Entry prolog ---
PUSH BP 'Save base pointer'

--- Restore registers and return ---
POP ES
POP BP
RET 6 'End of procedure'
OCL_YLINE ENDP
PAGE
--- END OF SEGMENT ---
OCLMOD_CODE ENDS
--- END MODULE ---
END

```

```

BP,SP
MOV ES
AX,ACTIVE_PLANE 'Access current plane'
MOV ES,AX

--- Get X coordinate and compute load value ---
MOV AX,[BP] YLINE_X
MOV BX,AX
SHR AX,1
SHR AX,1
SHR AX,1
SHR AX,1
SHL AX,1
MOV SI,AX 'X offset in bytes'
SHL AX,1
SHL AX,1
SHL AX,1
SUB BX,AX
SHL BX,1
MOV DI,Y_LOAD[BX]
; BX = remainder in pixels
; Convert rem. to byte offset
; Look up load value and save.

--- Get Y coordinates, calculate starting offset and count ---
MOV AX,[BP] YLINE_Y1
MOV BX,[BP] YLINE_Y2
CMP BX,AX
JGE YLINE_L1 'Make AX <= BX'
XCHG AX,BX
MOV CX,BX 'Compute no. of words to fill'
SUB CX,AX
INC CX 'Compute starting address'
MOV AX,299D
SUB AX,BX
MOV BX,92D
MUL BX 'Result in DX AX'
ADD AX,SI 'Adjust for X offset'
MOV BX,AX

--- Turn on the bits to form a line ---
YLINE_L2 OR ES,[BX],DI
ADD BX,92D 'Offset to next horiz line'
LOOP YLINE_L2

--- Restore registers and return ---
POP ES
POP BP
RET 6 'Discard 6 bytes of stack'

--- End of procedure ---
OCL_YLINE ENDP
PAGE
--- END OF SEGMENT ---
OCLMOD_CODE ENDS
--- END MODULE ---
END

```



```

PROGRAM BERCOM1 (INPUT, OUTPUT, IN_FILE, OUT_FILE, REF_FILE,
  DEBUG, BER_TOLERANCE)

```

```

*****
DESCRIPTION:

```

```

  Process one file of 90K samples in 5625 sample blocks.
  Synchronization is performed once at the beginning and is
  checked after a block is processed. If abnormally high
  bit-error rates are detected, the block is marked invalid

```

```

  This program is suitable for downtown Ottawa/Hull tapes
  where the data is clean and loss of synchronization is
  highly unlikely. For noisier tapes, a more sophisticated
  method of synchronization is desirable

```

```

USE:

```

```

  To run: BERCOM1 <input_data_file> <output_data_file>
           <reference_file> <debug_flag> <BER tolerance>

```

```

  Eg BERCOM1 MRDA01.01A MRDA01 01B PNSEQU DAT TRUE 0.25

```

```

INPUT:

```

```

  IN_FILE: Input packed data file from MRDA

```

```

  REF_FILE: 2047-byte file of 0.1 values of PN
             sequence

```

```

  DEBUG: TRUE causes extra debug info to be printed.
         Otherwise, use FALSE

```

```

  BER_TOLERANCE: Bit-error-rate tolerance. Used to
                  determine maximum permissible errors
                  before loss-of-sync is assumed. Also
                  used as limit on errors when
                  resynchronizing sequence to data

```

```

OUTPUT:

```

```

  OUT_FILE: Output file with error pattern and valid
            error bits computed. File is packed
            with the same format as MRDA. See
            UTILMOD for details on packing

```

```

PROCEDURES REQUIRED:
  UTL_UNPACK_DATA
  UTL_PACK_DATA

```

```

PREREQUISITES

```

```

  None

```

```

NOTES, ORGANIZATION AND RESTRICTIONS.

```

```

  None

```

```

VERSION HISTORY

```

VER	DATE	AUTHOR	COMMENT
0 0	02-10-84	T Knight	Original
0 1	05-10-84	TK	Improved loss-of-sync detection technique

```

  COPYRIGHT 1984 by Tony Knight,
  Dept. of Systems and Computer Engineering,
  Carleton University Ottawa, Ontario, K1S 5B6

```

```

CONST

```

```

BLOCK_LEN = 5625, (* 90000/16 *)
REF_SEQ_LEN = 2047, (* Length of PN sequence *)

```

```

TYPE

```

```

  DATA_BLOCK = ARRAY [0..BLOCK_LEN-1] OF INTEGER,
  REF_SEQ_TYPE = ARRAY [0..REF_SEQ_LEN-1] OF SINT,
  SINT_BLOCK = ARRAY [0..BLOCK_LEN-1] OF SINT,

```

```

VAR

```

```

  PACKED_BLOCK: DATA_BLOCK;

```

```

  RX_DATA: RX_DATA_BLOCK;

```

```

  ERROR_PATTERN: SINT_BLOCK;

```

```

  REF_SEQ_SEQUENCE: REF_SEQ_TYPE;

```

```

  IN_FILE: IN_FILE;

```

```

  OUT_FILE: OUT_FILE;

```

```

  REF_FILE: FILE OF DATA_BLOCK,

```

```

  SYNC_OFFSET: FILE OF REF_SEQ_TYPE,

```

```

  ENVELOPE: ENVELOPE;

```

```

  BLOCK_NUM: BLOCK_NUM;

```

```

  VALID_BLK_COUNT: VALID_BLK_COUNT;

```

```

  VALID_BLK_EC: VALID_BLK_EC;

```

```

  BLOCK_EC: BLOCK_EC;

```

```

  I: INTEGER;

```

```

  DEBUG: INTEGER;

```

```

  SYNCHRONIZED: BOOLEAN;

```

```

  RX_DATA_BIT: RX_DATA_BIT;

```

```

  SPEED_PULSE: SPEED_PULSE;

```

```

  ERROR_BIT: ERROR_BIT;

```

```

  DUMMY_SINT: DUMMY_SINT;

```

```

  BER_TOLERANCE: BER_TOLERANCE;

```

```

  TOTAL_BER: REAL;

```

```

EXTERNAL PROCEDURES

```

```

$INCLUDE 'UTILMOD.PAS'

```

```

PROCEDURE READ_REFERENCE_SEQUENCE;

```

```

DESCRIPTION:

```

```

  Read in reference PN sequence

```

```

INPUT:

```

```

  Global variables

```

```

  REF_FILE = File containing reference sequence

```

```

OUTPUT:

```

```

  Global variables

```

```

  REF_SEQUENCE = reference PN sequence

```

```

NOTES, ORGANIZATION AND RESTRICTIONS.

```

```

  None

```

```

BEGIN

```

```

  RESET(REF_FILE);

```

```

  READ(REF_FILE, REF_SEQUENCE);

```

```

  CLOSE(REF_FILE);

```

```

END.

```

```

PROCEDURE COMPUTE_SYNC_OFFSET;

```

```

*****
** DESCRIPTION:
** Correlate beginning of data block with reference sequence
** shift and repeat to determine synchronization offset. A
** warning is printed if synchronization not possible.
**
** INPUT:
** Global variables.
** RX_DATA = block of received digital data
** REF_SEQUENCE = reference digital PN sequence
**
** OUTPUT:
** Global variables.
** SYNC_OFFSET = Offset into REF_SEQUENCE to line-up data
** and reference.
** SYNCHRONIZED = Boolean flag set TRUE if a unique
** sync offset found.
**
** Console:
** Warning printed if synchronization not possible
**
** NOTES, ORGANIZATION AND RESTRICTIONS:
** None.
*****
CONST
  CORR_LEN = 250;

VAR
  SYNC_FLAG    BOOLEAN;
  OFFSET       INTEGER;
  EC_TOLERANCE INTEGER;
  I, J         INTEGER;

```

```

BEGIN
  EC_TOLERANCE := ROUND((BER_TOLERANCE * FLOAT(CORR_LEN)));
  WRITELN('Resynchronizing data and sequence. ');
  IF DEBUG THEN
    WRITELN(' Error count tolerance = ', EC_TOLERANCE,
    SYNC_OFFSET := 0;
    SYNC_FLAG := FALSE;
  FOR OFFSET := 0 TO REF_SEQ_LEN-1 DO
    BEGIN
      ERROR_COUNT := 0;
      FOR I := 0 TO CORR_LEN-1 DO
        BEGIN
          J := (OFFSET+I) MOD REF_SEQ_LEN;
          IF RX_DATA[I] <> REF_SEQUENCE[J] THEN
            ERROR_COUNT = ERROR_COUNT+1;
        END;
      END;
      IF ERROR_COUNT < EC_TOLERANCE THEN
        BEGIN
          WRITELN(' Synchronization failed - not unique. ');
          SYNCHRONIZED = FALSE;
        END;
      ELSE
        BEGIN
          SYNC_OFFSET = OFFSET;
          SYNC_FLAG = TRUE;
          SYNCHRONIZED = TRUE;
          IF DEBUG THEN

```

```

BEGIN
  WRITELN(' Synchronization achieved. ');
  WRITELN(' Sync offset = ', SYNC_OFFSET);
  WRITELN(' Error count = ', ERROR_COUNT);
END;
END;
(* ENDIF *)
END;
(* FOR *)
IF NOT(SYNCHRONIZED) AND NOT(SYNC_FLAG) THEN
  WRITELN(' Unable to synchronize. ');
END;
(* PROCEDURE *)

PROCEDURE COMPUTE_ERROR_PATTERN;
*****
** DESCRIPTION:
** Using current sync_offset, correlate received data with
** reference sequence to determine bits in error.
**
** INPUT:
** Global variables.
** RX_DATA = Received data
** REF_SEQUENCE = Reference PN sequence
** SYNC_OFFSET = Synchronization offset
**
** OUTPUT:
** Global variables.
** ERROR_PATTERN = Computed error pattern (1=error)
**
** NOTES, ORGANIZATION AND RESTRICTIONS:
** None.
*****
VAR
  I, J: INTEGER;
  FOR I := 0 TO BLOCK_LEN-1 DO
    BEGIN
      J := (I-SYNC_OFFSET) MOD REF_SEQ_LEN;
      IF RX_DATA[I] <> REF_SEQUENCE[J] THEN
        ERROR_PATTERN[I] := 1;
      ELSE
        ERROR_PATTERN[I] := 0;
      END;
    END;
  END;
  (* FOR *)
END;
(* PROCEDURE *)

FUNCTION CHECK_BIT_ERROR_RATE: BOOLEAN;
*****
** DESCRIPTION:
** Compute bit-error rates for current error pattern and
** determine if loss-of-synchronization with the reference
** has occurred. Errors are counted and a value of FALSE is
** returned if two adjacent blocks contain excessive errors.
** One isolated block of excessive errors does not result in
** loss of sync
**
** INPUT:
** Global variables.
** ERROR_PATTERN = Array of error occurrences
**
** OUTPUT:
** Function return value.
*****

```



```

UTL_PACK_DATA(ENVELOPE_RX_DATA,BIT_SPEED,PULSE,
ERROR_BIT_VALID,ERROR_PATTERN[I],
PACKED_DATA[I])
END; (*FOR*)
WRITE(OUT_FILE,PACKED_DATA);
UPDATE_SYNC_OFFSET;
BLOCK_NUM = BLOCK_NUM+1;
END; (*WHILE*)
TOTAL_BER := FLOAT(VALID_BLK_EC)/
(FLOAT(VALID_BLK_COUNT)*FLOAT(BLOCK_LEN));
WRITELN(VALID_BLK_COUNT, ' out of ',BLOCK_NUM,
' blocks have valid error bits. ');
WRITELN('Bit-error-rate for file = ',TOTAL_BER);
END; (*BERCOM1*)

```

```

PROGRAM QRCEN (INPUT, OUTPUT, QRE_OUT, DEBUG)
*****
* PROGRAM
* QRCEN
*
* DESCRIPTION
* Compute and store look-up table relating quantized envelope values
* to receiver RF signal level using linear interpolation
*
* USE
* QRCEN <output_filename> <debug_flag>
*
* INPUT
* <output_filename> = Name of file for QRE table
* <debug_flag> = TRUE - Writes Q and RE values to display
*               FALSE - No output produced
*
* OUTPUT
* 2049-point real-valued lookup table of RF signal strength (
* in RMS volts) written to output file
*
* PROCEDURES REQUIRED
* None
*
* PREREQUISITES
* None
*
* NOTES, ORGANIZATION AND RESTRICTIONS
* None
*
* VERSION HISTORY:
* VER DATE AUTHOR COMMENT
* 0 0 84-10-18 T.Knight Original
*
* Copyright 1984 by Tony Knight,
* Dept of Systems and Computer Engineering,
* Carleton University, Ottawa, Ontario, K1S 5B6
*
*****
TYPE QRE_BLOCK = ARRAY [0..2048] OF REAL;
VAR
  QRE_OUT      FILE OF QRE_BLOCK;
  QRE_TABLE    QRE_BLOCK;
  QUT          ARRAY [0..24] OF INTEGER; (* Linear interp LUT *)
  I, J         INTEGER;
  FRAC         REAL;
  RE           REAL;
  DEBUG        BOOLEAN;
BEGIN (*PROGRAM*)
  (* Initialize the linear interpolation table Note
  * the QUT values are absolute values. The actual values
  * are negative. This is a non-uniform piecewise linear
  * approximation that was derived manually from a graph
  * Refer to envelope conversion notes for more details.
  *)
  QUT[0] := 0;
  QUT[1] := 50;
  QUT[2] := 70;
  QUT[3] := 90;
  QUT[4] := 120;
  QUT[5] := 150;
  QUT[6] := 200;
  QUT[7] := 300;
  QUT[8] := 400;
  QUT[9] := 500;
  QUT[10] := 700;
  QUT[11] := 900;
  QUT[12] := 1100;
  QUT[13] := 1300;
  QUT[14] := 1500;
  QUT[15] := 1600;
  QUT[16] := 1700;
  QUT[17] := 1800;
  QUT[18] := 1850;
  QUT[19] := 1900;
  QUT[20] := 1950;
  QUT[21] := 1980;
  QUT[22] := 2000;
  QUT[23] := 2020;
  QUT[24] := 2050;
  (* These RF values are in dBmV *)
  RELUT[0] := -130.0;
  RELUT[1] := -130.0;
  RELUT[2] := -124.5;
  RELUT[3] := -122.0;
  RELUT[4] := -119.5;
  RELUT[5] := -117.5;
  RELUT[6] := -115.5;
  RELUT[7] := -112.5;
  RELUT[8] := -110.0;
  RELUT[9] := -108.5;
  RELUT[10] := -106.0;
  RELUT[11] := -103.0;
  RELUT[12] := -100.0;
  RELUT[13] := -97.0;
  RELUT[14] := -94.0;
  RELUT[15] := -92.0;
  RELUT[16] := -90.0;
  RELUT[17] := -87.5;
  RELUT[18] := -86.0;
  RELUT[19] := -84.0;
  RELUT[20] := -81.5;
  RELUT[21] := -79.5;
  RELUT[22] := -77.5;
  RELUT[23] := -76.0;
  RELUT[24] := -72.0;
  J := 0; (* Index into linear interpolation table *)
  FOR I := 0 TO 2048 DO
    BEGIN
      IF I >= QUT[J+1] THEN
        J := J+1;
      FRAC := (FLOAT(I - QUT[J]) / (FLOAT(QUT[J+1] - QUT[J])));
      RE := RELUT[J] + FRAC * (RELUT[J+1] - RELUT[J]);
      QRE_TABLE[I] := 0.223607 * EXP(RE / 8.685890);
      IF DEBUG THEN
        WRITE('Q = ', I, ' RE = ', RE, ' dBmV');
      END; (*FOR*)
      REWRITE(QRE_OUT);
      WRITE(QRE_OUT, QRE_TABLE);
    END; (*PROGRAM*)
  END

```

PROGRAM RESAMP (INPUT, OUTPUT, IN_FILE, OUT_FILE, QRF_FILE, LPF_FILE, DEBUG).

```
CONST
  WORKING_BUF_LEN = 16380,
  LPF_LEN = 45,
  LPF_TAIL_LEN = 22,
  RESAMPLE_TABLE_LEN = 100,
  MAX_DEADTIME = 400, (* Deadtme aft pulse found in samples *)

TYPE
  LPF_BLOCK = ARRAY [0..LPF_LEN-1] OF REAL,
  INPUT_STATUS_T = (SUCCESS, BUFFER_FULL, EOF_ENCOUNTERED);
```

```
VAR
  IN_FILE      FILE OF INTEGER,
  OUT_FILE     FILE OF REAL,
  QRF_FILE     FILE OF REAL,
  LPF_FILE     FILE OF LPF_BLOCK,
  DONE         BOOLEAN,
  DATA_FOLDED,
  DEBUG,
  E0_VALID,
  E1_VALID,
  E2_VALID,
  QRF_TABLE    LPF_BLOCK,
  LPF_COEF     LPF_BLOCK,
  WORKING_BUFFER ARRAY [0..WORKING_BUF_LEN-1] OF INTEGER,
  READ_SP_STATUS, INPUT_STATUS T,
  RESAMPLE_TABLE ARRAY [0..RESAMPLE_TABLE_LEN-1] OF INTEGER,
  INTERVAL_COUNT,
  SP_DEADTIME,
  I_PULSE,
  I_BOO,
  I_MES,
  I0_11_12,
  OUTPUT_SAMPLE_CNTR,
  I_RESULT,
  I_LPF,
  I_DATA,
  E0_COUNT,
  E1_COUNT,
  E2_COUNT,
  V0_V1_SUM    INTEGER,
  REAL;
```

```
(* EXTERNAL DECLARATIONS *)
(* INCLUDE 'UTILMOD.PAS' *)

FUNCTION WRAP_AROUND (X INTEGER) INTEGER;
BEGIN
  WHILE X < 0 DO
    X = X + WORKING_BUF_LEN;
  WRAP_AROUND = X MOD WORKING_BUF_LEN;
END; (*PROCEDURE*)

PROCEDURE READ_QRF_TABLE;
(* INPUT/OUTPUT *)
(* Global vars *)
(* QRF_FILE (IN) *)
(* QRF_TABLE (OUT) *)
VAR
  I INTEGER;
```

```
RESET(QRF_FILE);
FOR I := 0 TO 2048 DO
  BEGIN
    READ(QRF_FILE, X);
    QRF_TABLE[I] := X;
  END;
CLOSE(QRF_FILE);
END; (*PROCEDURE*)

PROCEDURE READ_LPF_COEF;
(* INPUT/OUTPUT *)
(* Global vars *)
(* LPF_FILE (IN) *)
(* LPF_COEF (OUT) *)
VAR
  I BOO,
  IN_FILE (IN),
  I_MES (IN),
  SP_DEADTIME (IN),
  READ_SP_STATUS (IN),
  WORKING_BUFFER (IN),
  I_PULSE (IN),
  E1_COUNT (IN),
  E1_VALID (IN),
  E2_COUNT (IN),
  E2_VALID (IN);
```

```
RESET(LPF_FILE);
READ(LPF_FILE, LPF_COEF);
CLOSE(LPF_FILE);
END; (*PROCEDURE*)

PROCEDURE READ_SP_INTERVAL;
(* INPUT/OUTPUT *)
(* Global vars *)
(* I_BOO (IN) *)
(* IN_FILE (IN) *)
(* I_MES (IN) *)
(* SP_DEADTIME (IN) *)
(* READ_SP_STATUS (IN) *)
(* WORKING_BUFFER (IN) *)
(* I_PULSE (IN) *)
(* E1_COUNT (IN) *)
(* E1_VALID (IN) *)
(* E2_COUNT (IN) *)
(* E2_VALID (IN) *)
VAR
  DONE BOOLEAN,
  PREVIOUS_SP,
  CURRENT_SP,
  RX_DATA,
  EB_VALID,
  ERROR_BIT,
  SINT,
  PACKED_INT,
  ENVELOPE,
  TAIL_COUNT INTEGER;
```

```
FUNCTION CHECK_BUFFER_FULL BOOLEAN;
BEGIN
  IF I_MES = WRAP_AROUND(I_BOO-1) THEN
    CHECK_BUFFER_FULL = TRUE;
  ELSE
    CHECK_BUFFER_FULL = FALSE;
  END; (*FUNCTION*)

BEGIN (*PROCEDURE*)
  READ_SP_STATUS := SUCCESS;
  DONE := FALSE;
  PREVIOUS_SP = 0;
  WHILE NOT(DONE) DO
```

X REAL,

```
BEGIN
  RESET(QRF_FILE);
  FOR I := 0 TO 2048 DO
    BEGIN
      READ(QRF_FILE, X);
      QRF_TABLE[I] := X;
    END;
  CLOSE(QRF_FILE);
END; (*PROCEDURE*)

PROCEDURE READ_LPF_COEF;
(* INPUT/OUTPUT *)
(* Global vars *)
(* LPF_FILE (IN) *)
(* LPF_COEF (OUT) *)
VAR
  I BOO,
  IN_FILE (IN),
  I_MES (IN),
  SP_DEADTIME (IN),
  READ_SP_STATUS (IN),
  WORKING_BUFFER (IN),
  I_PULSE (IN),
  E1_COUNT (IN),
  E1_VALID (IN),
  E2_COUNT (IN),
  E2_VALID (IN);
```

```
RESET(LPF_FILE);
READ(LPF_FILE, LPF_COEF);
CLOSE(LPF_FILE);
END; (*PROCEDURE*)

PROCEDURE READ_SP_INTERVAL;
(* INPUT/OUTPUT *)
(* Global vars *)
(* I_BOO (IN) *)
(* IN_FILE (IN) *)
(* I_MES (IN) *)
(* SP_DEADTIME (IN) *)
(* READ_SP_STATUS (IN) *)
(* WORKING_BUFFER (IN) *)
(* I_PULSE (IN) *)
(* E1_COUNT (IN) *)
(* E1_VALID (IN) *)
(* E2_COUNT (IN) *)
(* E2_VALID (IN) *)
VAR
  DONE BOOLEAN,
  PREVIOUS_SP,
  CURRENT_SP,
  RX_DATA,
  EB_VALID,
  ERROR_BIT,
  SINT,
  PACKED_INT,
  ENVELOPE,
  TAIL_COUNT INTEGER;
```

```
FUNCTION CHECK_BUFFER_FULL BOOLEAN;
BEGIN
  IF I_MES = WRAP_AROUND(I_BOO-1) THEN
    CHECK_BUFFER_FULL = TRUE;
  ELSE
    CHECK_BUFFER_FULL = FALSE;
  END; (*FUNCTION*)

BEGIN (*PROCEDURE*)
  READ_SP_STATUS := SUCCESS;
  DONE := FALSE;
  PREVIOUS_SP = 0;
  WHILE NOT(DONE) DO
```

```
RESET(QRF_FILE);
FOR I := 0 TO 2048 DO
  BEGIN
    READ(QRF_FILE, X);
    QRF_TABLE[I] := X;
  END;
CLOSE(QRF_FILE);
END; (*PROCEDURE*)

PROCEDURE READ_LPF_COEF;
(* INPUT/OUTPUT *)
(* Global vars *)
(* LPF_FILE (IN) *)
(* LPF_COEF (OUT) *)
VAR
  I BOO,
  IN_FILE (IN),
  I_MES (IN),
  SP_DEADTIME (IN),
  READ_SP_STATUS (IN),
  WORKING_BUFFER (IN),
  I_PULSE (IN),
  E1_COUNT (IN),
  E1_VALID (IN),
  E2_COUNT (IN),
  E2_VALID (IN);
```

```
RESET(LPF_FILE);
READ(LPF_FILE, LPF_COEF);
CLOSE(LPF_FILE);
END; (*PROCEDURE*)

PROCEDURE READ_SP_INTERVAL;
(* INPUT/OUTPUT *)
(* Global vars *)
(* I_BOO (IN) *)
(* IN_FILE (IN) *)
(* I_MES (IN) *)
(* SP_DEADTIME (IN) *)
(* READ_SP_STATUS (IN) *)
(* WORKING_BUFFER (IN) *)
(* I_PULSE (IN) *)
(* E1_COUNT (IN) *)
(* E1_VALID (IN) *)
(* E2_COUNT (IN) *)
(* E2_VALID (IN) *)
VAR
  DONE BOOLEAN,
  PREVIOUS_SP,
  CURRENT_SP,
  RX_DATA,
  EB_VALID,
  ERROR_BIT,
  SINT,
  PACKED_INT,
  ENVELOPE,
  TAIL_COUNT INTEGER;
```

```
FUNCTION CHECK_BUFFER_FULL BOOLEAN;
BEGIN
  IF I_MES = WRAP_AROUND(I_BOO-1) THEN
    CHECK_BUFFER_FULL = TRUE;
  ELSE
    CHECK_BUFFER_FULL = FALSE;
  END; (*FUNCTION*)

BEGIN (*PROCEDURE*)
  READ_SP_STATUS := SUCCESS;
  DONE := FALSE;
  PREVIOUS_SP = 0;
  WHILE NOT(DONE) DO
```

```

BEGIN
  IF EOF(IN_FILE) THEN
    BEGIN
      READ_SP_STATUS = EOF_ENCOUNTERED,
      DONE := TRUE
    END
  ELSE IF CHECK_BUFFER_FULL THEN
    BEGIN
      READ_SP_STATUS = BUFFER_FULL,
      DONE := TRUE
    END
  ELSE
    BEGIN
      (* Read in a sample, unpack and store *)
      READ(IN_FILE,PACKED_INT),
      UTL_UNPACK_DATA(PACKED_INT,ENVELOPE,RX_DATA),
      CURRENT_SP,EB_VALID,ERROR_BIT),
      WORKING_BUFFER(I_NES) = ENVELOPE,
      I_NES = WRAP_AROUND(I_NES+1),
      (* Handle speed pulse *)
      IF SP_DEADTIME>0 THEN
        SP_DEADTIME = SP_DEADTIME-1
      ELSE
        BEGIN
          IF (PREVIOUS_SP=0) AND (CURRENT_SP=1) THEN
            BEGIN
              (* Speed pulse found *)
              PULSE = WRAP_AROUND(I_NES-1),
              SP_DEADTIME = MAX_DEADTIME,
              DONE = TRUE
            END
          END, (*IF*)
          PREVIOUS_SP = CURRENT_SP,
          (* Handle error count *)
          E1_COUNT := E1_COUNT*ERROR_BIT,
          IF EB_VALID=0 THEN
            E1_VALID = FALSE,
            END, (*IF*)
            END, (*WHILE*)
          IF READ_SP_STATUS=SUCCESS THEN
            (* Speed pulse found, now read in more data to *)
            (* allow for LPF computation *)
            BEGIN
              E2_COUNT = 0,
              E2_VALID := TRUE,
              TAIL_COUNT = 0,
              DONE = FALSE,
              WHILE NOT(DONE) DO
                IF EOF(IN_FILE) OR CHECK_BUFFER_FULL THEN
                  DONE = TRUE
                ELSE
                  BEGIN
                    (* Get a sample, unpack and save *)
                    READ(IN_FILE,PACKED_INT),
                    UTL_UNPACK_DATA(PACKED_INT,ENVELOPE,RX_DATA),
                    CURRENT_SP,EB_VALID,ERROR_BIT),
                    WORKING_BUFFER(I_NES) = ENVELOPE,
                    I_NES = WRAP_AROUND(I_NES+1),
                    (* Update speed pulse dead time *)
                    IF SP_DEADTIME>0 THEN
                      SP_DEADTIME = SP_DEADTIME-1,

```

```

          (* Handle error count *)
          E2_COUNT = E2_COUNT*ERROR_BIT,
          IF EB_VALID=0 THEN
            E2_VALID := FALSE,
          (* Check if done *)
          TAIL_COUNT := TAIL_COUNT+1,
          IF TAIL_COUNT=LPF_TAIL_LEN THEN
            DONE := TRUE,
            END, (*IF*)
            END, (*ENDWHILE*)
            END, (*PROCEDURE*)
          PROCEDURE COMPUTE_RESAMPLE_TABLE,
          (* INPUT/OUTPUT: *)
          (* GLOBAL VARS: *)
          I0,I1,I2 (IN)
          INTERVAL_COUNT (IN)
          DEBUG (IN)
          RESAMPLE_TABLE (OUT)
          V0,V1 (OUT)
          CONST
            D_WHEEL = 2.83; (* Distance travelled in one wheel rev (m) *)
            F_SAMPLE = 4800.0; (* Working array sample rate in Hz *)
            V_MAX = 33.3; (* Max speed for bounds checking = 120 km/h *)
          VAR
            NO,N1,V_AVG,DELTA,V_C,D,
            PREV_X,CURR_X,REAL_I,REAL_J,
            REAL_RST_LEN,DX1,DX2 REAL,
            I,J: INTEGER,
          BEGIN (*PROCEDURE*)
            IF (I0<0) OR (I0>=WORKING_BUF_LEN) THEN
              WRITELN('WARNING: I0 OUT OF BOUNDS '),
            IF (I1<0) OR (I1>=WORKING_BUF_LEN) THEN
              WRITELN('WARNING: I1 OUT OF BOUNDS '),
            IF (I2<0) OR (I2>=WORKING_BUF_LEN) THEN
              WRITELN('WARNING: I2 OUT OF BOUNDS '),
          (* Compute interval speeds *)
          I := I1-I0
          IF I>=0 THEN
            NO = FLOAT(I)
          ELSE
            NO := FLOAT(I+WORKING_BUF_LEN),
            I = I2-I1,
            IF I>=0 THEN
              N1 = FLOAT(I)
            ELSE
              N1 := FLOAT(I+WORKING_BUF_LEN),
              V0 := D_WHEEL*F_SAMPLE/NO,
              V1 := D_WHEEL*F_SAMPLE/N1,
              DELTA_V := V1-V0,
              V_AVG := 0.5*(V0+V1),
          (* Do some checks *)
          IF V0 <= 0 THEN
            WRITELN('WARNING: V0 NEGATIVE OR ZERO '),
          IF V1 <= 0 THEN
            WRITELN('WARNING: V1 NEGATIVE OR ZERO '),
          IF (V0>V_MAX) OR (V1>V_MAX) THEN
            WRITELN('WARNING: EXCESSIVE SPEED '),

```

```

E1_VALID = E2_VALID.
READ_SP_INTERVAL.
I2 = I_PULSE.
IF READ_SP_STATUS<>SUCCESS THEN
  WRITELN('FAILED TO FIND THIRD SPEED PULSE').
WRITELN('Interval v0 [m/s] v1 [m/s] Error count')
(* Prepare for main program loop *)
DONE = FALSE.
INTERVAL_COUNT := 0.
WHILE NOT (DONE) DO
  BEGIN
    COMPUTE_RESAMPLE_TABLE;
    IF E0_VALID THEN
      WRITELN('INTERVAL COUNT:8.V0.V1.E0_COUNT:8')
    ELSE
      WRITELN('INTERVAL COUNT:8.V0.V1. Invalid').
    IF I_BOD<I_NES THEN
      DATA_FOLDED := FALSE
    ELSE
      DATA_FOLDED = TRUE.
    FOR OUTPUT_SAMPLE_CNTR = 0 TO RESAMPLE_TABLE_LEN-1 DO
      BEGIN
        (* Compute one LPF sum-of-products *)
        SUM := 0.0.
        FOR I_LPF := 0 TO LPF_LEN-1 DO
          BEGIN
            I_DATA = WRAP_AROUND(I_LPF-LPF_TAIL_LEN+
              RESAMPLE_TABLE[OUTPUT_SAMPLE_CNTR]).
            IF DATA_FOLDED THEN
              BEGIN
                IF (I_DATA<I_NES) OR (I_DATA>=I_BOD) THEN
                  SUM := SUM+LPF_COEF(I_LPF)*FLOAT(
                    WORKING_BUFFER(I_DATA))
              END
            ELSE
              IF (I_BOD=I_DATA) AND (I_DATA<I_NES) THEN
                SUM := SUM+LPF_COEF(I_LPF)*FLOAT(
                  WORKING_BUFFER(I_DATA))
              END, (*FOR*)
            (* Note that input data is negative,
              * hence SUM is negated to get positive result. *)
            I_RESULT := ROUND(-SUM)
            IF I_RESULT<0 THEN
              I_RESULT := 0
            ELSE
              IF I_RESULT>2048 THEN
                I_RESULT := 2048.
              WRITE(OUT_FILE,QRF_TABLE(I_RESULT)).
            END, (*FOR*)
          (* Get the next speed pulse interval *)
          I0 := I1.
          I1_BOD = WRAP_AROUND(I0-LPF_TAIL_LEN).
          I1 := I2.
          E0_COUNT = E1_COUNT.
          E0_VALID = E1_VALID.
          E1_COUNT := E2_COUNT.
          E1_VALID := E2_VALID.
          READ_SP_INTERVAL.
          CASE READ_SP_STATUS OF
            SUCCESS
              I2 = I_PULSE.
            BUFFER_FULL
              BEGIN

```

```

(* Quadratic coefficients *)
REAL_RST_LEN := FLOAT(RESAMPLE_TABLE_LEN).
C = REAL_RST_LEN*V0/(N0*V_AVG).
D = REAL_RST_LEN*DELTA_V/(2.0*V_AVG*N0*NO).
RESAMPLE_TABLE[0] := 0.
CURR_X := 0.0. (* Offset into working array *)
I := 0.
(* Compute resampling table *)
FOR J := 1 TO RESAMPLE_TABLE_LEN-1 DO
  BEGIN
    REPEAT
      REAL_J := FLOAT(J).
      PREV_X = CURR_X.
      I := I+1.
      REAL_I := FLOAT(I).
      CURR_X := C*REAL_I + D*REAL_I*REAL_I
    UNTIL CURR_X>REAL_J.
    DX1 := REAL_J-REAL_I.
    DX2 := CURR_X-REAL_X.
    IF DX2<DX1 THEN
      RESAMPLE_TABLE[J] = (I-I0) MOD WORKING_BUF_LEN
    ELSE
      RESAMPLE_TABLE[J] = (I-I0-1) MOD WORKING_BUF_LEN.
    END, (*FOR*)
  END, (*PROCEDURE*)
  BEGIN (*PROGRAM*)
    (* Open files *)
    READ_QRF_TABLE.
    READ_LPF_COEF.
    RESET(IN_FILE).
    REWRITE(OUT_FILE).
    (* Initialize some global variables *)
    I_BOD := 0.
    I_NES := 0.
    (* Pointer to beginning of data *)
    (* Pointer to next empty space *)
    (* Counts out the period following a
      * speed pulse when further pulses
      * are ignored
    SP_DEADTIME = 0.
    (* Find first speed pulse in data *)
    E1_COUNT = 0.
    E1_VALID = TRUE.
    READ_SP_INTERVAL.
    I0 = I_PULSE.
    IF I0>=LPF_TAIL_LEN THEN
      I1_BOD = I0-LPF_TAIL_LEN
    ELSE
      I1_BOD = 0.
    IF READ_SP_STATUS<>SUCCESS THEN
      WRITELN('FAILED TO FIND FIRST SPEED PULSE').
    (* Find second speed pulse *)
    E1_COUNT = E2_COUNT.
    E1_VALID = E2_VALID.
    READ_SP_INTERVAL.
    I1 = I_PULSE.
    IF READ_SP_STATUS<>SUCCESS THEN
      WRITELN('FAILED TO FIND SECOND SPEED PULSE').
    (* Find third speed pulse *)
    E0_COUNT = E1_COUNT.
    E0_VALID = E1_VALID.
    E1_COUNT = E2_COUNT

```



```
DOVE := TRUE.  
WRITELN('CANNOT CONTINUE - BUFFER FULL')  
END.  
EOF ENCOUNTERED  
DOVE := TRUE,  
END. (*CASE*)  
INTERVAL_COUNT := INTERVAL_COUNT+1,  
END; (*WHILE*)  
END. (*PROGRAM*)
```

```

PROGRAM ENFILT(INPUT, OUTPUT, IN_FILE, OUT_FILE, RES_FILE, LPF_FILE)
*****
* PROGRAM
* ENFILT
* DESCRIPTION:
* Read in real envelope data and apply FIR smoothing filter
* to obtain a moving average value. Save the average value
* in OUT_FILE, and compute the ratio of actual/average and
* save it in RES_FILE
* INPUT
* IN_FILE = File of real envelope data
* LPF_FILE = Formatted file of LPF coefficients
* OUTPUT
* OUT_FILE = Filtered data or moving average of envelope
* RES_FILE = Residual or ratio of actual to average
* Various statistics computed and output to OUTPUT
*****
CONST
  LPF_MAX_LEN = 401, (* Maximum LP filter size *)
VAR
  IN_FILE,
  OUT_FILE,
  RES_FILE,
  LPF_FILE,
  DATA_ARRAY,
  LPF_COEF,
  LPF_LEN,
  LPF_TAIL_LEN,
  I, J, K, C,
  SUM_RES,
  NPTS,
  DB,
  DBM,
  ACC_AVG,
  ACC_AVG_SQR,
  ACC_AVG_DBM,
  ACC_AVG_DBM_SQR,
  ACC_RAT,
  ACC_RAT_SQR,
  ACC_RAT_DB,
  ACC_RAT_DB_SQR,
  AVG_MEAN,
  AVG_STD_DEV,
  AVG_DBM_MEAN,
  AVG_DBM_STD_DEV,
  RAT_MEAN,
  RAT_STD_DEV,
  RAT_DB_MEAN,
  RAT_DB_STD_DEV
  FUNCTION VOLTS_TO_DBM(X REAL) REAL,
  (* Convert voltage into dB relative to 1 mW assuming 50 ohm impedance *)
  BEGIN
    VOLTS_TO_DBM = 8.685890*LN(X) + 13.01030
  END
  FUNCTION TWENTY LOG10(X REAL) REAL,
  BEGIN

```

```

    TWENTY_LOG10 = 8.685890*LN(X)
  END
  FUNCTION CALC_STD_DEV(SUM_X, SUM_X_SQRD, N: REAL) REAL,
  (* Compute sample standard deviation given, sum of samples,
  * sum of squared values and number of points
  *)
  BEGIN
    CALC_STD_DEV = SQRT((SUM_X_SQRD - SUM_X**2/N)/(N-1.0))
  END
  BEGIN
    WRITELN('Envelope moving average program ')
    (* Read in LPF coefficients *)
    RESET(LPF_FILE);
    READ(LPF_FILE, LPF_LEN);
    WRITELN('ENFILT: Lowpass filter length =', LPF_LEN);
    IF (LPF_LEN > LPF_MAX_LEN) THEN
      WRITELN('WARNING -- Filter too large');
      LPF_TAIL_LEN = (LPF_LEN-1) DIV 2;
      IF (2*LPF_TAIL_LEN <> LPF_LEN-1) THEN
        WRITELN('WARNING -- Filter length not odd');
        FOR I := 0 TO LPF_TAIL_LEN-1 DO
          BEGIN
            J = LPF_LEN-I-1;
            READ(LPF_FILE, LPF_COEF[I]);
            LPF_COEF[J] := LPF_COEF[I];
          END
        END
        READ(LPF_FILE, LPF_COEF[LPF_TAIL_LEN]);
      (* Initialize working data buffer *)
      FOR I := 0 TO LPF_TAIL_LEN-1 DO
        DATA_ARRAY[I] := 0.0;
      RESET(IN_FILE);
      FOR I = LPF_TAIL_LEN TO LPF_LEN-1 DO
        READ(IN_FILE, DATA_ARRAY[I]);
      (* Compute first output *)
      REMWRITE(OUT_FILE);
      REMWRITE(RES_FILE);
      SUM := 0.0;
      FOR I = 0 TO LPF_LEN-1 DO
        SUM := SUM + LPF_COEF[I]*DATA_ARRAY[I];
        WRITE(OUT_FILE, SUM);
        ACC_AVG := SUM;
        ACC_AVG_SQR := SQR(SUM);
        DBM := VOLTS_TO_DBM(SUM);
        ACC_AVG_DBM := DBM;
        ACC_AVG_DBM_SQR := SQR(DBM);
        RES := DATA_ARRAY[LPF_TAIL_LEN]/SUM;
        WRITE(RES_FILE, RES);
        ACC_RAT := RES;
        ACC_RAT_SQR := SQR(RES);
        DB = TWENTY_LOG10(RES);
        ACC_RAT_DB := DB;
        ACC_RAT_DB_SQR := SQR(DB);
        NPTS = 1.0; (* Number of points input *)
      (* Main filter loop *)
      K := 0; (* K = offset and next sample index *)
      WHILE NOT (EOF(IN_FILE)) DO
        BEGIN
          READ(IN_FILE, DATA_ARRAY[K]);
          K := (K+1) MOD LPF_LEN;
          SUM = 0.0;
          FOR I = 0 TO LPF_LEN-1 DO

```

```

WRITELN(RAT_DB_MEAN, ' dB'),
WRITE(' Sample standard dev',
WRITELN(RAT_DB_STD_DEV, ' dB')
END (*PROGRAM*)
= ' RAT_STD_DEV, ' unitless, ');

```

```

BEGIN
  J := (I-K) MOD LPE_LEN;
  SUM := SUM+LPE_COEF[I]*DATA_ARRAY[J]
END; (*FOR*)
WRITE(OUT_FILE, SUM)
ACC_AVG := ACC_AVG+SUM;
ACC_AVG_SQR := ACC_AVG_SQR+SUM*(SUM);
DBM := VOLTS_TO_DBM(SUM);
ACC_AVG_DBM := ACC_AVG_DBM+DBM;
ACC_AVG_DBM_SQR := ACC_AVG_DBM_SQR+DBM*(DBM);
RES := DATA_ARRAY[(K-LPE_TAIL_LEN) MOD LPE_LEN]/SUM;
WRITE(RES_FILE, RES);
ACC_RAT := ACC_RAT+RES;
ACC_RAT_SQR := ACC_RAT_SQR+RES*(RES);
DB := TWENTY_LOG10(RES);
ACC_RAT_DB := ACC_RAT_DB+DB;
ACC_RAT_DB_SQR := ACC_RAT_DB_SQR+DB*(DB);
NPTS := NPTS+1.0
END; (*WHILE*)

(* Finish up processing *)
FOR C := 1 TO LPE_TAIL_LEN DO
  BEGIN
    DATA_ARRAY[K] := 0.0;
    SUM := 0.0;
    FOR I := 0 TO LPE_LEN-1 DO
      BEGIN
        J := (I+K) MOD LPE_LEN;
        SUM := SUM+LPE_COEF[I]*DATA_ARRAY[J]
      END; (*FOR*)
      WRITE(OUT_FILE, SUM)
      ACC_AVG := ACC_AVG+SUM;
      ACC_AVG_SQR := ACC_AVG_SQR+SUM*(SUM);
      DBM := VOLTS_TO_DBM(SUM);
      ACC_AVG_DBM := ACC_AVG_DBM+DBM;
      ACC_AVG_DBM_SQR := ACC_AVG_DBM_SQR+DBM*(DBM);
      RES := DATA_ARRAY[(K-LPE_TAIL_LEN) MOD LPE_LEN]/SUM;
      WRITE(RES_FILE, RES);
      ACC_RAT := ACC_RAT+RES;
      ACC_RAT_SQR := ACC_RAT_SQR+RES*(RES);
      DB := TWENTY_LOG10(RES);
      ACC_RAT_DB := ACC_RAT_DB+DB;
      ACC_RAT_DB_SQR := ACC_RAT_DB_SQR+DB*(DB);
      NPTS := NPTS+1.0
    END; (*FOR*)

    (* Compute some statistics *)
    AVG_MEAN := ACC_AVG/NPTS;
    AVG_STD_DEV := CALC_STD_DEV(ACC_AVG, ACC_AVG_SQR, NPTS);
    AVG_DBM_MEAN := ACC_AVG_DBM/NPTS;
    AVG_DBM_STD_DEV := CALC_STD_DEV(ACC_AVG_DBM, ACC_AVG_DBM_SQR, NPTS);
    RAT_MEAN := ACC_RAT/NPTS;
    RAT_STD_DEV := CALC_STD_DEV(ACC_RAT, ACC_RAT_SQR, NPTS);
    RAT_DB_MEAN := ACC_RAT_DB/NPTS;
    RAT_DB_STD_DEV := CALC_STD_DEV(ACC_RAT_DB, ACC_RAT_DB_SQR, NPTS);

    (* and print results *)
    WRITELN(NPTS, 'samples filtered');
    WRITELN('Moving mean level');
    WRITE(' Average = ', AVG_MEAN, ' volts. ');
    WRITELN(AVG_DBM_MEAN, ' dbm');
    WRITE(' Sample standard dev = ', AVG_STD_DEV, ' volts. ');
    WRITELN(AVG_DBM_STD_DEV, ' db');
    WRITELN('Residual fast fading ratio ');
    WRITE(' Average = ', RAT_MEAN, ' unitless. ');

```

```

PROGRAM RICIAN(INPUT,OUTPUT,ENV_PDF,FILE)
*****
PROGRAM
  RICIAN

DESCRIPTION
  Computes Rician probability density functions and
  cumulative distribution functions for a file of data.
  Provision is made for the user to adjust parameters and
  view the plotted results. A sum-of-errors-squared is
  computed and displayed

  This version computes values at actual RF voltages. This
  seems to fail due to numeric error when the ratio of
  direct path (a) over scattered energy (b) exceeds about 20.
  This problem is partially resolved by using double
  precision real arithmetic

USE
  Run the program, specifying the filename of data,
  for example: RICIAN MDI01.0LM

INPUT
  Rician PDF parameters entered from the keyboard

  ENV_PDF_FILE = formatted file of envelope data density
  Line 1: Number of values below -130 dBm
  Line 2: Number of values above -70 dBm
  Line 3: Number of values in range -130 0 to -70.0 dBm
  Line 4: Maximum count
  Line 5: First value count in -130 0 to -129.5 dBm
  etc.
  Line 124: Count -70 5 to -70 0 dBm
  (All values are real numbers)

OUTPUT
  Output text written to screen and data plotted on display

PROCEDURES REQUIRED
  GCI_INIT_BW
  GCI_CLEAR_PLANE
  GCI_BORDER
  GCI_XLINE
  GCI_YLINE
  CRT_CLEAR_SCREEN
  DSP_RICE_DENSITY
  DOS_GET_CHAR

PREREQUISITES
  The graphics video option must be installed

NOTES
  1 The procedures called in DSPMOD must be compiled using
  double precision real arithmetic

VERSION HISTORY
  VER DATE AUTHOR COMMENT
  1 0 1-Feb-85 T Knight Mod to use REAL8

Copyright 1984, 85 by Tony Knight,
Dept of Systems and Computer Engineering
Carleton University, Ottawa Ontario, K1S 5B6

```

```

(*$REAL 8*)
(* Program global variables *)
VAR
  ENV_PDF_FILE TEXT,
  I INTEGER,
  UNDER_RANGE_C,
  OVER_RANGE_C,
  IN_RANGE_C,
  MAX_C,
  ACCUM_C,
  PDF_TO_PLOT_K,
  FUDGE_FACTOR,
  RICE_A,
  RICE_B,
  PDF_MODE,
  ENVEL_PDF,
  ENVEL_CDF,
  RICIAN_PDF,
  RICIAN_CDF: ARRAY (0..119) OF REAL,
  (*$INCLUDE:'COTMOD.PAS'*)
  (*$INCLUDE:'CRIMOD.PAS'*)
  (*$INCLUDE:'DSPMOD.PSI'*)
  (*$INCLUDE:'DOSENS.PAS'*)
PROCEDURE RIC_PLOT_PDF,
VAR
  I,X,Y: INTEGER,
  ERR,
  S_ERR,
  XR,
  REAL,
BEGIN
  (* Plot distributions *)
  CRT_CLEAR_SCREEN;
  GCI_CLEAR_PLANE;
  FOR I := 0 TO 3 DO
    BEGIN
      X = 200*I+50,
      GCI_XLINE(40,290,X)
    END;
  (*FOR*)
  GCI_XLINE(40,660,50),
  GCI_XLINE(40,660,290),
  FOR I := 1 TO 5 DO
    BEGIN
      Y = 40*I+50,
      GCI_XLINE(40,50,Y),
      GCI_XLINE(650,660,Y)
    END;
  (*FOR*)
  (* Plot theoretical Rician density *)
  S_ERR = 0.0;
  FOR I = 0 TO 119 DO
    BEGIN
      X = ROUND(ENVEL_PDF[I]*PDF_TO_PLOT_K),
      GCI_XLINE(50,X+50,2*I+50),
      X = ROUND(RICIAN_PDF[I]*PDF_TO_PLOT_K),
      IF X>400 THEN X=400,
      IF X<0 THEN X=0,
      GCI_XLINE(250,X+250,2*I+50),
      ERR = RICIAN_PDF[I]-ENVEL_PDF[I],
      S_ERR = S_ERR+SQR(ERR),
      X = ROUND(ERR*PDF_TO_PLOT_K),
      IF X<-100 THEN X=-100,
      IF X>100 THEN X=100;
    END;
  END;

```



```
WHILE TRUE DO
  BEGIN
    RIC_ADJUST_PARAMS,
    RIC_CALC_RICIAN_PDF,
    IF PDF_MODE THEN
      RIC_PLOT_PDF
    ELSE
      RIC_PLOT_CDF,
    END: (*WHILE*)
  END (*PROGRAM*)
```

```

MODULE DSPMOD;
***** DSPMOD *****
** MODULE NAME **
** DSPMOD **
** DESCRIPTION **
** DSP procedures **
** PROCEDURES PROVIDED **
** DSP_FORWARD_FFT **
** DSP_INVERSE_FFT **
** DSP_AUTOCORRELATE **
** DSP_RAYLEIGH_DENSITY **
** DSP_BESSEL_FUNCTION **
** DSP_RICE_DENSITY **
** PROCEDURES REQUIRED **
** None **
** PREREQUISITES **
** None **
** NOTES, ORGANIZATION AND RESTRICTIONS **
** None **
*****
** Copyright 1984 by Tony Knight, **
** Dept of Systems and Computer Engineering, **
** Carleton University, Ottawa, Ontario, K1S 5B6 **
*****

```

```

CONST
PI = 3.141592654;

```

```

TYPE
REAL_DUMMY_ARRAY = ARRAY [1..1] OF REAL;
REAL_DUM_ARRAY_2 = ARRAY [0..1] OF REAL;

```

```

FUNCTION DSP_2_PWR_I (I: INTEGER) INTEGER;

```

```

VAR
J, K: INTEGER;

```

```

BEGIN
K := 1;
FOR J = 1 TO I DO
K := 2 * K;
DSP_2_PWR_I := K;
END; (**FUNCTION**)

```

```

PROCEDURE DSP_FORWARD_FFT (VAR XR: REAL_DUMMY_ARRAY;
VAR XI: REAL_DUMMY_ARRAY;
M: INTEGER);

```

```

VAR
N, J, NV2, NM1, L, LE, LE1, I, IP, K: INTEGER;
UR, UI, WR, WI, TR, TI, XRI, XII, XRIIP, XIIP, PHI: REAL;

```

```

BEGIN
N := DSP_2_PWR_I (M);
WRITELN('M=', M, 'N=', N);
FOR L = 1 TO M DO
BEGIN
LE = DSP_2_PWR_I (M+1-L);
LE1 = LE DIV 2;

```

```

UR := 1.0;
UI := 0.0;
PHI := PI/EOAT (LE1);
WR := COS (PHI);
WI := -SIN (PHI);
FOR J = 1 TO LE1 DO
BEGIN
I := J;
WHILE I <= N DO
BEGIN
IP := I * LE1;
XRI := XR [I];
XII := XI [I];
XRIIP := XR [IP];
XIIP := XI [IP];
XRI [I] := XRI * XRIIP;
XI [I] := XII * XIIP;
XRI [IP] := (XRI * XRIIP) * UR;
XI [IP] := (XRI * XRIIP) * UI;
I := I * LE;
END; (**WHILE**)
TR := (WR * UR - WI * UI);
TI := (WR * UI + WI * UR);
UR := TR;
UI := TI;
END; (**FOR J**)
NM2 := N DIV 2;
NM1 := N - 1;
J := 1;
FOR I = 1 TO NM1 DO
BEGIN
IF I < J THEN
BEGIN
TR := XR [J];
TI := XI [J];
XRI [J] := XRI [I];
XI [J] := XI [I];
XRI [I] := TR;
XI [I] := TI;
END;
K := NV2;
WHILE K < J DO
BEGIN
J := J - K;
K := K DIV 2;
END;
J := J + K;
END; (**FOR I**)
END; (**PROCEDURE**)

```

```

PROCEDURE DSP_INVERSE_FFT (VAR XR: REAL_DUMMY_ARRAY;
VAR XI: REAL_DUMMY_ARRAY;
M: INTEGER);

```

```

VAR
N, J, NV2, NM1, L, LE, LE1, I, IP, K: INTEGER;
UR, UI, WR, WI, TR, TI, XRI, XII, XRIIP, XIIP, PHI: REAL;

```

```

BEGIN
N := DSP_2_PWR_I (M);
WRITELN('M=', M, 'N=', N);
FOR L = 1 TO M DO
BEGIN

```

```

LE := DSP_2_FWR_I (M+1-L),
LE1 := LE DIV 2,
UR := 1.0,
UI := 0.0,
PHI := PI/FLOAT(LE1),
WR := COS(PHI),
WI := SIN(PHI),
FOR J := 1 TO LE1 DO
  BEGIN
    I := J,
    WHILE I <= N DO
      BEGIN
        IP := I-LE1,
        XRI := XR[I],
        XII := XI[I],
        XRIIP := XR[IP],
        XIIP := XI[IP],
        XRII := XRI-XRIIP,
        XIIP := XII-XIIP,
        XRIIP := ((XRI-XRIIP)*UR),
        XIIP := ((XII-XIIP)*UI),
        XI[IP] := ((XRIIP-XRIIP)*UR),
        XII[IP] := ((XIIIP-XIIP)*UI),
        I := I+LE
      END, (*WHILE*)
      TR := (WR*UR-WI*UI),
      TI := (WR*UI-WI*UR),
      UR := TR,
      UI := TI,
      END, (*FOR J*)
    END, (*FOR L*)
    NV2 := N DIV 2,
    NM1 := N-1,
    J := 1,
    FOR I := 1 TO NM1 DO
      BEGIN
        IF I < J THEN
          BEGIN
            TR := XR[J],
            TI := XI[J],
            XR[J] := XR[I],
            XI[J] := XI[I],
            XR[I] := TR,
            XI[I] := TI,
            END,
            K = NV2,
            WHILE K < J DO
              BEGIN
                J := J-K,
                K := K DIV 2,
                END,
                J = J-K,
                END, (*FOR I*)
              END, (*FOR L*)
              (* Perform scaling for inverse FFT *)
              TI := FLOAT(N),
              FOR I := 1 TO N DO
                BEGIN
                  XR[I] := XR[I]/TI,
                  XI[I] := XI[I]/TI,
                  END,
                  END, (*PROCEDURE*)
                END,
                REAL DIM ARRAY Z,
                REAL DIM ARRAY Z,
                VAR OUTPUT DATA
                N, M INTEGER)

```

```

*****
** INPUT:
** INPUT_DATA = Real array of data to correlate
** N = number of points in input data
** M = number of output points to compute
*****
VAR
  MI, NI, INTEGER,
  SUM, REAL,
  BEGIN
    FOR MI := 0 TO M-1 DO
      BEGIN
        SUM := 0.0,
        FOR NI := 0 TO N-MI-1 DO
          SUM := SUM+INPUT_DATA[NI]*INPUT_DATA[NI-MI],
          SUM := SUM/FLOAT(8),
          OUTPUT_DATA[MI] := SUM,
          END,
          END, (*PROCEDURE*)
        FUNCTION DSP_RAYLEIGH_DENSITY(ALPHA, X, REAL) REAL
        *****
        ** DESCRIPTION:
        ** Calculate a point from a Rayleigh probability density
        *****
        VAR
          ASQRD, REAL,
          BEGIN
            ASQRD := SQR(ALPHA),
            DSP_RAYLEIGH_DENSITY := (X/ASQRD)*EXP(-X*X/(2.0*ASQRD))
            END,
            (*REAL:8*)
            FUNCTION DSP_BESSEL_FUNCTION(X, REAL) REAL
            *****
            ** DESCRIPTION:
            ** Compute zero order modified Bessel function value.
            *****
            VAR
              S, D, DS, REAL,
              BEGIN
                S := 1.0,
                D := 0.0,
                DS := 1.0,
                WHILE (DS > 2.0E-12*S) DO
                  BEGIN
                    D := D+2.0,
                    DS := DS*SQR(X/D),
                    S := S+DS,
                    END,
                    DSP_BESSEL_FUNCTION = S,
                    END,
                    FUNCTION DSP_RICE_DENSITY(A, B, X, REAL) REAL
                    *****
                    ** DESCRIPTION:
                    ** Compute Rician density function for X.
                    *****
                    VAR
                      BSQR, XBSQR, T, REAL,
                      BEGIN
                        BSQR = B*B,
                        XBSQR = X/XBSQR,
                        T = DSP_BESSEL_FUNCTION(A*XBSQR),

```


T := T*EXP(-(SOR(X)*SOR(A))/(2.0*BSQR));
DSP_RICE_DENSITY := T*XBBSQR
END;
END. (*MODULE*)

```

PROGRAM COVAR (INPUT OUTPUT IN FILE OUT FILE MAX IN PTS NUM_LAG_PTS)
*****
* PROGRAM
* COVAR
* Compute sample covariance function for real data
* DESCRIPTION
* INPUT
* IN_FILE = File of real values (any length)
* MAX_IN_PTS = Limit on number of input points. If less
* than size of IN_FILE, only MAX_IN_PTS points will be
* processed.
* If N larger than size of IN_FILE, then the entire
* file will be processed. Note there is an
* internal limit to MAX_IN_PTS
* NUM_LAG_PTS = Number of lag points to compute (up to a
* limit of MAX_LAG_PTS)
* OUTPUT:
* OUT_FILE = Textfile of autocovariance values
* Results are displayed on current output device
*****
CONST
  IN_ARRAY_LEN = 8000, (* Max num of input points *)
  COV_ARRAY_LEN = 5000, (* Number of covariance pts *)
TYPE
  INPUT_ARRAY_TYPE = ARRAY [1..IN_ARRAY_LEN] OF REAL,
  COVAR_ARRAY_TYPE = ARRAY [1..COV_ARRAY_LEN] OF REAL,
  IN_FILE = FILE OF REAL,
  OUT_FILE = TEXT,
  MAX_IN_PTS = INTEGER,
  NUM_LAG_PTS = INTEGER,
  DATA_ARRAY = INPUT_ARRAY_TYPE,
  COVAR_ARRAY = COVAR_ARRAY_TYPE,
  NUM_IN_PTS = INTEGER,
  MEAN = REAL,
  X_SUM = REAL,
  DUMMY_CHAR = CHAR,
PROCEDURE DSP_AUTOCORRELATE (VAR INPUT_DATA INPUT_ARRAY_TYPE,
  VAR OUTPUT_DATA COVAR_ARRAY_TYPE,
  N, M INTEGER); EXTERN
BEGIN
  RESET(IN_FILE);
  I = 0;
  SUM = 0.0;
  WHILE (NOT(EOF(IN_FILE))) AND (I < MAX_IN_PTS) AND
    (I < IN_ARRAY_LEN) DO
    BEGIN
      I = I + 1;
      READ(IN_FILE, X);
      SUM = SUM + X;
      DATA_ARRAY[I] = X;
    END; (* WHILE *)
  NUM_IN_PTS = I;
  MEAN = SUM / FLOAT(NUM_IN_PTS);
  WRITEIN('Number of input points =', NUM_IN_PTS)
  WRITEIN('Mean envelope level =', MEAN)

```

```

(* Remove mean value from data *)
WRITEIN('Adjusting data to zero mean. ');
FOR I = 1 TO NUM_IN_PTS DO
  DATA_ARRAY[I] = DATA_ARRAY[I] - MEAN;
(* Compute autocovariance function using autocorrel procedure *)
IF NUM_LAG_PTS > COV_ARRAY_LEN THEN
  BEGIN
    WRITEIN('WARNING - Too many lag points requested ');
    WRITEIN('Hit return to continue, or CTRL-C to abort ');
    READLN(DUMMY_CHAR);
  END;
WRITEIN('Computing', NUM_LAG_PTS, ' autocovariance lag points ');
DSP_AUTOCORRELATE(DATA_ARRAY, COVAR_ARRAY, NUM_IN_PTS, NUM_LAG_PTS);
(* Print lag points to a file *)
REWRITE(OUT_FILE);
WRITEIN(OUT_FILE, NUM_LAG_PTS, ' Number of lag points computed ');
X := COVAR_ARRAY[1];
WRITEIN(OUT_FILE, X, ' Maximum value (zero lag pt) ');
X := -X/3.0;
WRITEIN(OUT_FILE, X, ' Estimated min set -1/3 max ');
FOR I = 1 TO NUM_LAG_PTS DO
  WRITEIN(OUT_FILE, COVAR_ARRAY[I]);
END (* PROGRAM *)

```

```

PROGRAM INTERP (INPUT, OUTPUT, INFILE, OUTFILE);
*****
PROGRAM:
INTERP
*****
DESCRIPTION:
Interpolate real covariance data using zero-padded FFT.
*****
USE:
This is used for interpolating an estimate of envelope
covariance function to obtain finer resolution of distance
for analysis and display
*****
INPUT:
INFILE = Formatted text file of covariance data.
Line 1 = Number of points in file (INTEGER)
Line 2 = Zero lag point of covariance
Line 3 = First
etc
*****
OUTPUT:
OUTFILE = Formatted text file of interpolated data
Same format as above
*****
PROCEDURES REQUIRED:
DSP_FORWARD_FFT
DSP_INVERSE_FFT
*****
PREREQUISITES:
None
*****
NOTES, ORGANIZATION AND RESTRICTIONS:
None
*****
VERSION HISTORY:
VER DATE AUTHOR COMMENT
0 0 10-Dec-84 T. Knight Original
1.0 14-Dec-84 TK Output real part of FFT
*****
Copyright 1984 by Tony Knight,
Dept. of Systems and Computer Engineering,
Carleton University, Ottawa Ontario, K1S 5B6
*****
CONST:
N_INP_FFT = 128,
N_OUT_FFT = 2048,
ORD_INP_FFT = 7,
ORD_OUT_FFT = 11,
*****
TYPE
FFT_ARRAY = ARRAY [0..N_OUT_FFT-1] OF REAL,
*****
VAR
INFILE, OUTFILE, TEXT,
N_INFILE, I, N1, N2,
N3, N4,
REAL_DAT,
IMAG_DAT,
GAIN,
X_MIN
*****

```

```

X_MAX, REAL,
X,
PROCEDURE DSP_FORWARD_FFT (VAR XR, FFT_ARRAY, VAR XI, FFT_ARRAY,
M, INTEGER); EXTERN,
PROCEDURE DSP_INVERSE_FFT (VAR XR, FFT_ARRAY, VAR XI, FFT_ARRAY,
M, INTEGER); EXTERN,
*****
BEGIN
(* Set up time domain sequence The input data is one
side of a symmetric function This is stored in the
FFT array as an even function This is a necessary
condition so the resulting interpolated time data
is real, and has no imaginary part *)
RESET(INFILE);
READLN(INFILE, N_INFILE);
READLN(INFILE, X_MAX);
READLN(INFILE, X_MIN);
N1 := N_INFILE-1;
N2 := N_INP_FFT DIV 2;
N3 := N2-1;
IF N1 > N3 THEN
N1 := N3;
READLN(INFILE, REAL_DAT[0]);
FOR I := 1 TO N1 DO
BEGIN
READLN(INFILE, X);
REAL_DAT[I] := X;
REAL_DAT[N_INP_FFT-I] := X;
END;
IF N1 < N3 THEN
FOR I := N1 TO N3 DO
BEGIN
READLN(INFILE, X);
REAL_DAT[I] := X;
REAL_DAT[N_INP_FFT-I] := X;
END;
REAL_DAT[N2] := 0.0;
(* Forward FFT *)
DSP_FORWARD_FFT(REAL_DAT, IMAG_DAT, ORD_INP_FFT);
(* Zero pad *)
N4 := N_OUT_FFT-N2;
FOR I := 0 TO N3 DO
BEGIN
REAL_DAT[I+N4] := REAL_DAT[I-N2];
IMAG_DAT[I+N4] := IMAG_DAT[I-N2];
END;
FOR I := N2 TO N4-1 DO
BEGIN
REAL_DAT[I] := 0.0;
IMAG_DAT[I] := 0.0;
END;
(* Inverse FFT *)
DSP_INVERSE_FFT(REAL_DAT, IMAG_DAT, ORD_OUT_FFT);
(* Write results to output file *)
REWRITE(OUTFILE);
WRITELN(OUTFILE, 300);
WRITELN(OUTFILE, X_MAX);
WRITELN(OUTFILE, X_MIN);
GAIN := FLOAT(N_OUT_FFT DIV N_INP_FFT);
FOR I := 0 TO 300 DO
BEGIN

```

X:=CAIN*REAL_DAT[1].
WRITELN(OUTFILE,X)
END.
END. (*PROGRAM*)

END

| 8.0 | 4.8 | 6

FIN

ACCUMULATED STATISTICS				
AREA: Ottawa (urban)				
FILES: 1, 4, 5, 8				
NUMBER OF SAMPLES: 11800				
TOTAL DISTANCE: 334.m				
Envelope: $\hat{\mu}_r = -89.8$ dBm, $\hat{\sigma}_r = 7.1$ dB				
Filter Length N	Local mean		Fast fading	
	$\hat{\mu}_m$ [dBm]	$\hat{\sigma}_m$ [dB]	$\hat{\mu}_f$ [dB]	$\hat{\sigma}_f$ [dB]
15	-88.7	4.6	-1.1	5.0
35	-88.5	4.0	-1.3	5.5
51	-88.4	3.7	-1.4	5.7
99	-88.3	3.4	-1.5	6.0
201	-88.2	3.1	-1.6	6.3
399	-88.1	2.9	-1.6	6.5

ACCUMULATED STATISTICS				
AREA: Hull (urban)				
FILES: 2, 4, 5, 7, 8, 9				
NUMBER OF SAMPLES: 15600				
TOTAL DISTANCE: 442.m				
Envelope: $\hat{\mu}_r = -94.9$ dBm, $\hat{\sigma}_r = 6.6$ dB				
Filter Length N	Local mean		Fast fading	
	$\hat{\mu}_m$ [dBm]	$\hat{\sigma}_m$ [dB]	$\hat{\mu}_f$ [dB]	$\hat{\sigma}_f$ [dB]
15	-	-	-	-
35	-	-	-	-
51	-93.8	3.7	-1.2	5.2
99	-93.7	3.4	-1.3	5.5
201	-93.6	3.1	-1.3	5.7
399	-93.6	3.0	-1.3	5.9

ACCUMULATED STATISTICS				
AREA: Queensway (highway)				
FILES: 1, 2, 3, 4, 5, 6				
NUMBER OF SAMPLES: 72100				
TOTAL DISTANCE: 2.0 Km				
Envelope: $\hat{\mu}_r = -107.6$ dBm, $\hat{\sigma}_r = 6.6$ dB				
Filter length N	Local mean		Fast fading	
	$\hat{\mu}_m$ [dBm]	$\hat{\sigma}_m$ [dB]	$\hat{\mu}_{f_0}$ [dB]	$\hat{\sigma}_{f_0}$ [dB]
35	-107.0	5.1	-0.55	3.4
51	-107.0	5.0	-0.6	3.6
99	-106.9	4.7	-0.7	3.9
201	-106.8	4.4	-0.8	4.2
399	-106.7	4.2	-0.9	4.5

ACCUMULATED STATISTICS				
AREA: residential Ottawa				
FILES: 1, 2, 3				
NUMBER OF SAMPLES: 20100				
TOTAL DISTANCE: 570.m				
Envelope: $\hat{\mu}_r = -103.8$ dBm, $\hat{\sigma}_r = 5.9$ dB				
Filter length N	Local mean		Fast fading	
	$\hat{\mu}_m$ [dBm]	$\hat{\sigma}_m$ [dB]	$\hat{\mu}_{f_0}$ [dB]	$\hat{\sigma}_{f_0}$ [dB]
35	-103.2	4.0	-0.64	3.8
51	-103.1	3.9	-0.68	4.0
99	-103.0	3.6	-0.75	4.2
201	-103.0	3.4	-0.82	4.4
399	-102.9	3.2	-0.86	4.6

END

| 8.0 | 4.8 6

FIN