1994

A Study Of Meteor Burst Channel Characteristics

Kerry Jon Ellis

A Study of Meteor Burst Channel Characteristics

by

Kerry J. Ellis

Department of Electrical Engineering

Faculty of Engineering Science

Submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

Faculty of Graduate Studies
The University of Western Ontario
London, Ontario
August 1994

ISBN 0-315-93222-8

Canada

ABSTRACT

To be accepted as a viable competitor for long distance communication, it remains to be demonstrated that meteor burst systems possess the potential to support reasonable throughput at agreeable costs. Using measurements from an existing meteor burst link between London and Ottawa, estimates of the path loss, flight-time, waiting time and burst duration were obtained. These results demonstrated that by employing modest equipment, communication could be maintained with an average waiting time of 3.5 minutes and an average transmission time or burst duration 0.5 seconds. By using these parameters and assuming a transmission rate of 4800 bps, the average throughput for this link was estimated to be 11.4 bps. While this throughput may be adequate for applications having low data volumes such as remote monitoring and vehicle tracking, improvements are achievable through increases in transmitter power, antenna gains and data transmission rate. To date, however, little research has been undertaken to assess the effect of potential bandwidth limitations (if any) which may restrict the signaling rate that can be supported by the link. An investigation of these impairments was carried out using an instrument that was specifically designed to probe the multipath nature of the meteor burst channel These measurements revealed that in the majority of instances only a single path exists between the transmitter and receiver and consequently, serious impairments arising from multipath interference appear to be relatively infrequent. In some rare instances, however, as many as four such paths have been observed. Delay spread measurements were found to confirm the presence of multipath interference for which the statistics reveal that in 90% of all observations the delay spread is below 100 ns. Based on these statistics and observations to date, it is likely that data rates ranging from several hundred kbps to perhaps 1 Mbps can be supported by the meteor burst channel.

ACKNOWLEDGMENTS

I would like to express my gratitude to my wife Dianne Wilker and my supervisor Dr Alan Webster for their unwavering support and encouragement throughout the duration of this work. I would also like to thank Dr. Sherman Chow, Dr. Nur Serinken and Mr. Tom Ohno of the Communications Research Centre for their many good ideas and assistance in constructing and maintaining the meteor burst equipment.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

NOMENCLATURE

AWGN	Additive white Gaussian noise
B	Bandwidth
BER	Bit error rate
bps	Bits per second
BPSK	Binary phase shift keying
B_N	Equivalent noise bandwidth
BW	Chirp signal bandwidth
c	Speed of light in a vacuum
d	Separation distance between transmitter and receiver
D	Diffusion coefficient
D	Average delay time
dB	Decibels
dBm	Decibels relative to a power level of 1 mW
DFT	Discrete Fourier Transform
e	Euler's number
E_g	Received signal energy
f	Frequency
F	Receiver noise figure
f_d	Frequency offset between the chirp signals at the transmitter and receiver
FFT	Fast Fourier Transform
f_s	Sampling frequency
FSK	Frequency shift keying
GPS	Global positioning satellite
G_T	Transmitting antenna gain

G_R	Receiving antenna gain
h(t)	Complex impulse response
H(ω)	Frequency domain transfer function
k	Boltzmann's constant
k	Constant of proportionality
L	Path loss
L_R	Transmission line losses at the receiver
L_T	Transmission line losses at the transmitter
M	Average meteor arrival rate
N	Number of samples in a time series
N_o	Noise power spectral density
OOK	On-off keying
P_{IN}	Receiver input power
P_L	Path length
P_N	Noise power
P_{NB}	Noise power level in a business area
P_{NR}	Noise power level in a rural area
P_{Out}	Receiver output power
P_R	Received signal power
P_{RN}	Peak Normalized Power
P_T	Transmitter power
$p_i(t)$	Unipolar rectangular switching waveform
$P(\tau_k)$	Power level at the k^{th} delay sampling time
q	Electron line density
QPSK	Quadrature phase shift keying
R	Transmission rate
r_e	Classical electron radius

R_{Max}	Maximum transmission rate
r_o	Initial radius of the ionized trail
R_R	Distance from the receiver to the point of reflection at the meteor trail
R_T	Distance from the point of reflection at the meteor trail to the transmitter
S	Average throughput
S	Delay spread
S_E	Estimated delay spread due to the channel
S_M	Delay spread measured by the analysis software
S_S	Delay spread of the simulated signal
s(t)	Transmitted signal
SNR	Signal-to-noise ratio
t	Time variable
T_A	Antenna temperature
T_B	Burst duration
T_d	Delay time between the starting times of the chirps at the transmitter and receiver
T_{eq}	Equivalent noise temperature
$T_n(z)$	Chebyshev polynomials of the first kind
T_o	Ambient temperature (290 K)
T_p	Propagation time or time of flight
$\overline{T}_R$	Average receiver delay time
T_W	Waiting time
w[n]	Discrete time window function
z_o	Meteor trail altitude
α	Ratio of the main to peak sidelobe level in dB for the Dolph-Chebyshev family of windows

α_n	Set of complex weighting coefficients from which the Dolph-Chebyshev window coefficients are obtained
β	Angle between the axis of the meteor trail and the plane of propagation
β_k	Path gain for the k^{th} path between transmitter and receiver
γ	Angle between the incident electric field vector and the line of sight to the receiver
λ	Wavelength
μ	Chirp rate
τ	Pulse width
τ_k	k^{th} delay sampling time
τ_k	Delay time of the k^{th} path
τ_o	Time constant for overdense trails
τ_u	Time constant for underdense trails
ϕ	Half-scatter angle
ϕ_k	Phase shift for the k^{th} path due to the scattering process
ω	Angular frequency

Chapter 1

Meteor Burst Communication

1.1 Introduction

During a typical day, billions of meteoric particles enter the Earth's atmosphere and burn up (ablate), leaving in their wake ionized trails that can be used to "reflect" radio signals. The altitudes of these trails range from 80 to 120 kilometers and can permit communication over distances ranging from 200 to 2000 km. Telecommunication systems capitalizing on these serendipitous events are referred to as either meteor scatter or meteor burst systems. Meteoric particles or meteoroids are pieces of rock, dust and other cosmic debris from comets and asteroids that are encountered in varying concentrations around the Earth's orbit. Meteoroids that follow predictable orbits around the sun are referred to as shower meteors and are encountered at well established times of the year which accounts for seasonal variations in the meteor flux. Meteors associated with these showers are referred to by the name of their radiant which is simply the apparent location in the sky from which the meteor was observed to originate. Sporadic meteors on the other hand, follow seemingly random orbits around the sun but contribute a larger number of particles (roughly 10^{12} meteors per day) to the meteoroid flux than shower meteors.

Meteoroids impinging the Earth have masses that range from tenths of micrograms to tens of kilograms with populations that are inversely related to their masses [1]. As a result, the majority of trails used for meteor burst communication are created by relatively small meteoroids having masses ranging from 10^{-7} to 10 grams. Meteors of these sizes give rise to ionized trails whose electron line density, or number of electrons in a cross-

sectional slice of unit thickness, range from 10^{18} electrons per meter for a 10 gram particle to 10^{10} electrons per meter for a much smaller particle having a mass of only 10^{-7} grams The relationships between mass, radius, electron line density and population are outlined

Table 1.1 Order-of-Magnitude Estimates for Sporadic Meteors (after Sugar [1])

Mass (grams)	Radius	Number per Day	Electron Line Density (e/m)
10^{4}	8.0 cm	10	-
10^{3}	4.0 cm	10^{2}	-
10^{2}	2.0 cm	10^{3}	-
10	0.8 cm	10^{4}	10^{18}
1	0.4 cm	10^{5}	10^{17}
10^{-1}	0.2 cm	10^{6}	10^{16}
10^{-2}	0.08 cm	10^{7}	10^{15}
10^{-3}	0.04 cm	10^{8}	10^{14}
10^{-4}	0.02 cm	10^{9}	10^{13}
10^{-5}	80 μm	10^{10}	10^{12}
10^{-6}	40 μm	10^{11}	10^{11}
10^{-7}	20 μm	10^{12}	10^{10}
10^{-8}	8 μm	?	?

in Table 1.1. Ionized meteor trails are classified according to their electron line densities. Trails having electron line densities greater than 10^{14} electrons per meter are classified as overdense while those having electron line densities below this level are classified as underdense. It should be noted that since the electron densities of meteor trails and their

abundance are related to their mass, trails of the underdense variety are encountered most frequently. Since the electron line density for a given trail may not be readily known, trail classification is usually made on the basis of echo characteristics. Figures 1.1 and 1.2 show typical underdense and overdense echoes recorded on a 500 km forward scatter link by Webster and Jones [2]. Underdense echoes are characterized by their relatively short rise times (on the order of hundreds of microseconds) after which the received power reaches a maximum and immediately begins to decay exponentially. Durations on the order of one second are typical for echoes of this type. Echoes from overdense trails on the other hand, exhibit much slower rise times (on the order of hundreds of milliseconds) after which, a peak received power level is achieved and maintained for a period of time before the onset of decay. In some instances, fading can be observed in the echo as shown in Figure 1.2 (bottom). For trails of these types, durations on the order of several seconds to tens of seconds are typical.

Figures 1.3 and 1.4 illustrate both the backscatter and forward scatter geometries used in meteor studies employing radio techniques. In the backscatter configuration, both transmitter and receiver are located at the same geographical location, as in a conventional monostatic radar installation. Signals emitted from the transmitter are scattered back to the receiver whenever a trail of suitable ionization is illuminated by the transmitting antenna. While this geometry is useful for measuring meteor trail characteristics, it is not particularly useful for communication purposes and has been included here for completeness. The remainder of this thesis will focus exclusively on evaluating the performance of communication systems utilizing the forward scatter geometry. Communication systems relying on the forward scatter mode can exchange information only when trails of suitable ionization and orientation are present. This requires that the angle of incidence equal the angle of reflection (a condition for specular reflection) and is equivalent to the geometric requirement that the trail be tangent to a family of confocal

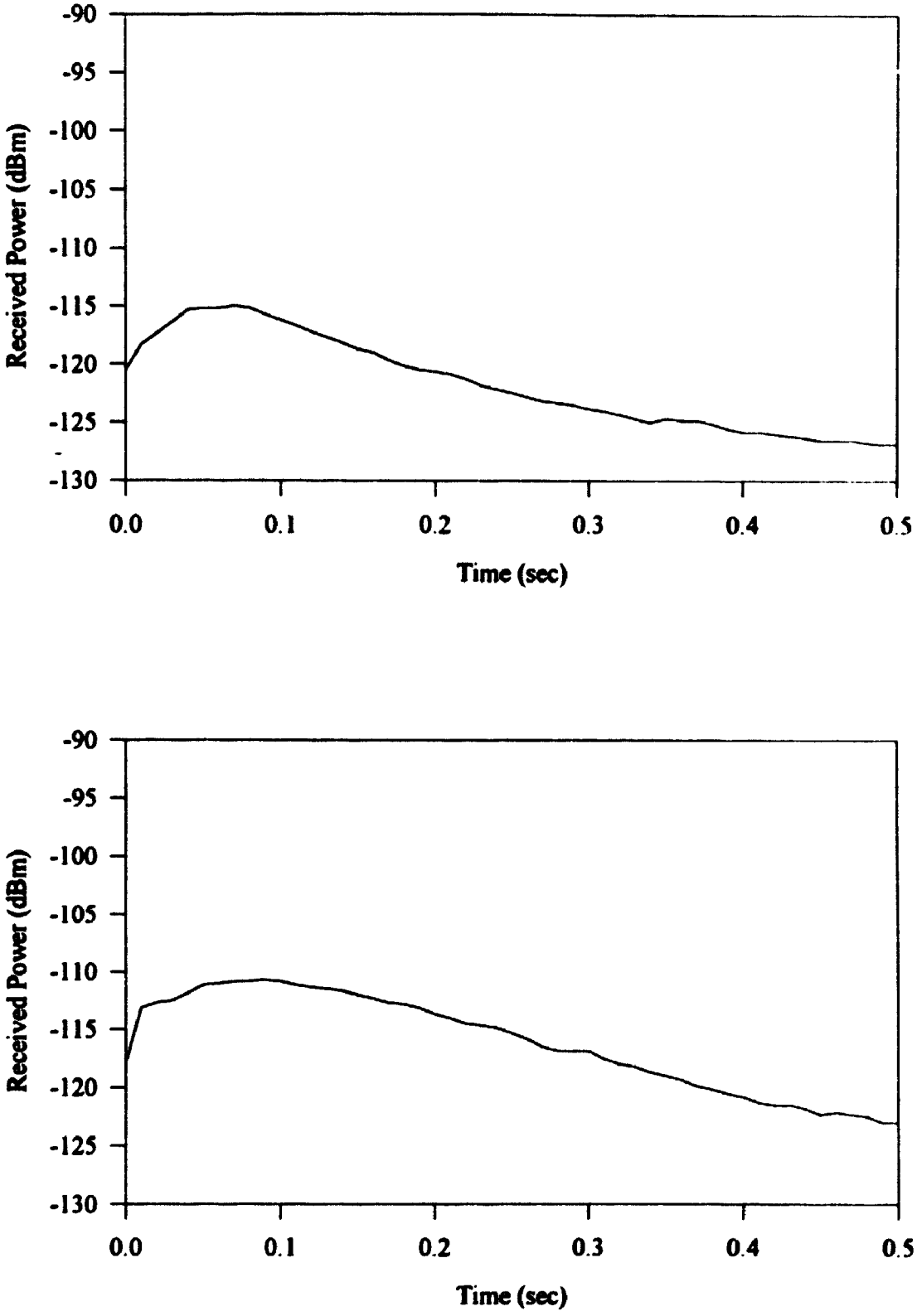


Figure 1.1 Typical Underdense Echoes (after Webster and Jones [2])

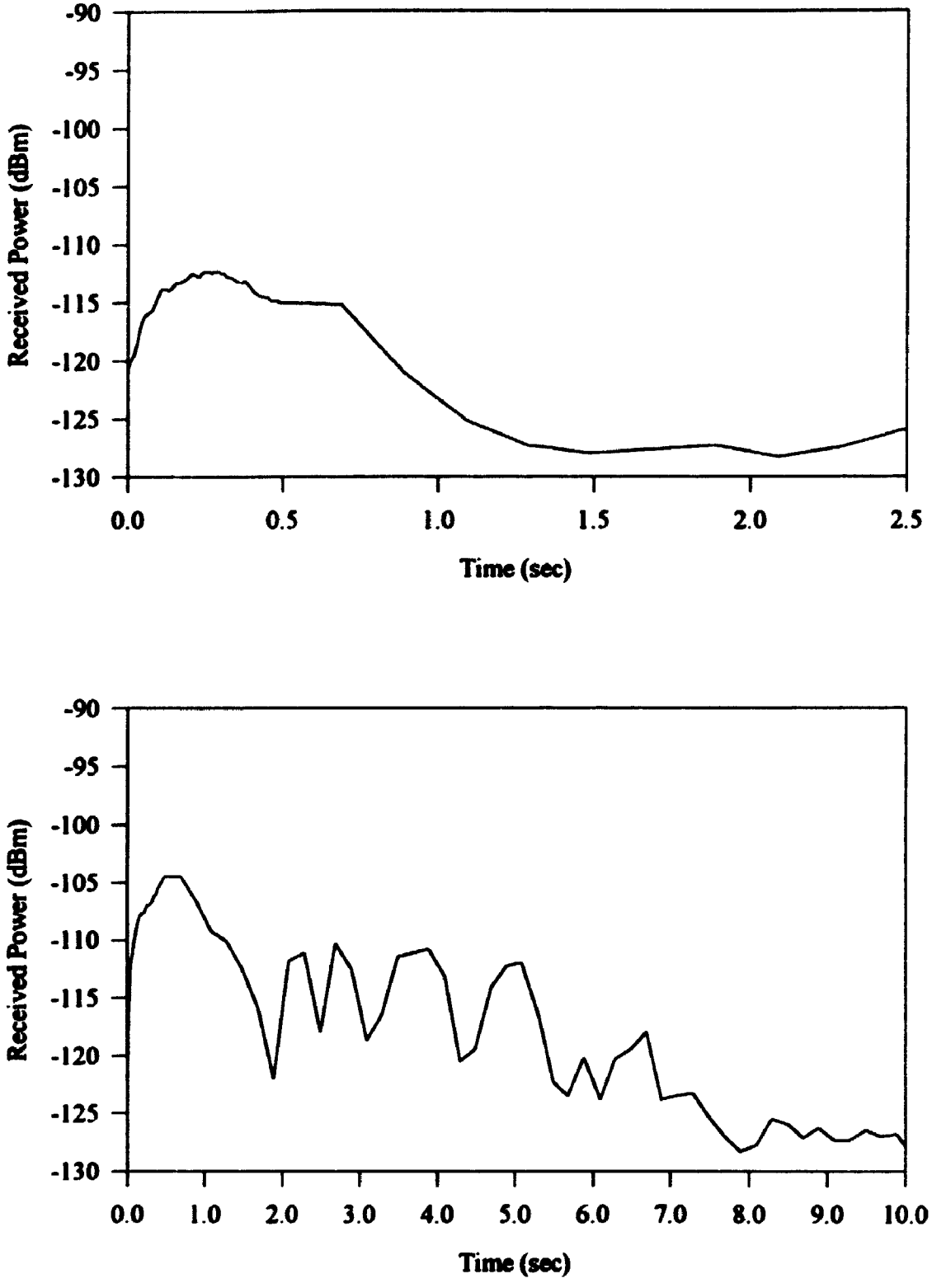


Figure 1.2 Typical Overdense Echoes (after Webster and Jones [2])

prolate ellipsoids whose foci are located at the transmitter and receiver locations [1,3,4]. As long as the conditions for specular reflection are satisfied, any suitably ionized meteor trail can be used for communication, including those occurring in the regions of space behind the transmitter and receiver. Thus in order to take full advantage of all trails having suitable orientation, antennas having omnidirectional radiation patterns should be used. In practice however, directional antennas such as the Yagi-Uda are often used to offset the transmission losses inherent to this mode of propagation.

The geometric conditions required for specular reflection provide meteor burst communication systems with an intrinsic privacy mechanism that prevents other stations from intercepting messages or jamming transmissions, unless they are positioned within the same meteor footprint. These locations however, are a function of the trail orientation and are different for each meteor [5]. Consequently, this requires that jamming or intercepting stations be located geographically close to their intended target to ensure their effectiveness. In military applications this feature offers a two-fold benefit, in that a low probability of intercept can be achieved while at the same time it confines the operation of clandestine stations to "nearby" locals. While this feature is highly desirable in military applications, it is also beneficial in civilian applications since it can permit various meteor burst communication services to share the same frequency spectrum with minimal interference. An additional feature of this mode is that it is neither vulnerable to attack (nuclear or conventional) nor dependent on consumables (propellant for station keeping), unlike satellite systems and their associated ground control stations.

Depending on the nature of the application, the use of meteor burst systems for long distance communication may be advantageous. While satellite communication systems have large bandwidths, are reliable and can carry both analog and digital signals in real time, they have the disadvantage that they are expensive to build and maintain. In

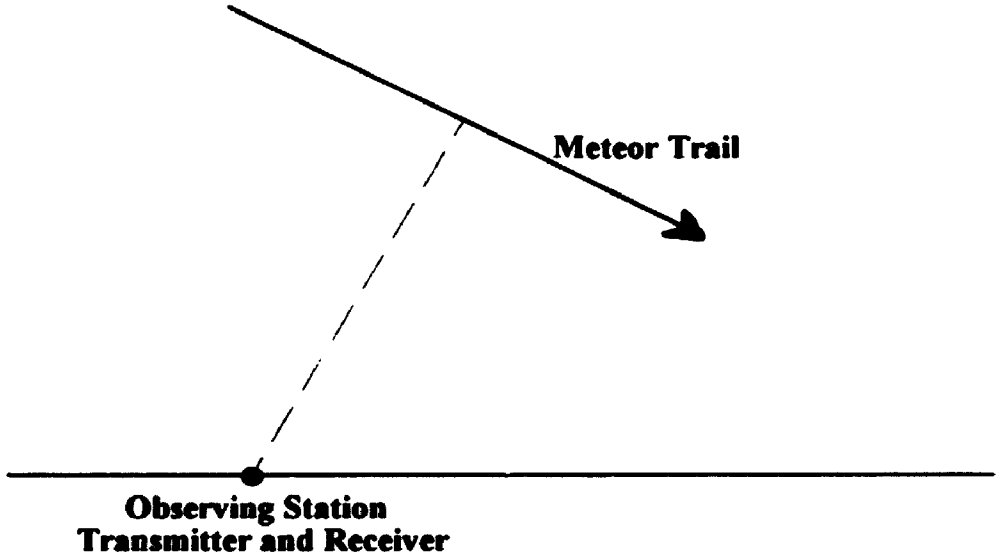


Figure 1.3 Backscatter Geometry

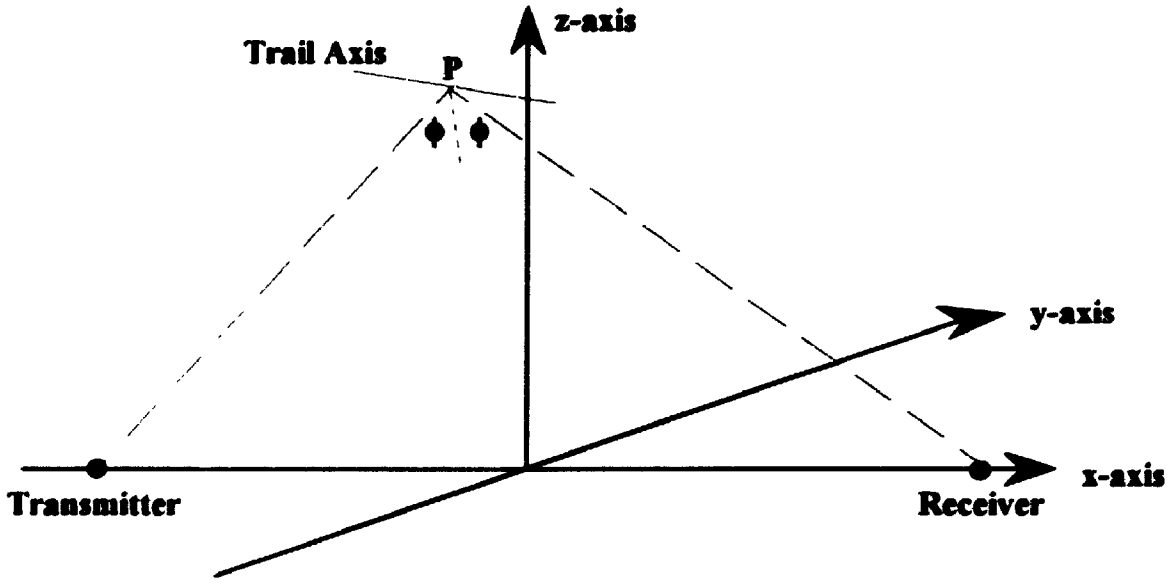


Figure 1.4 Forward Scatter Geometry

applications such as remote monitoring and vehicle tracking where meteor burst technology is currently employed (Schanker [4]), the traffic volume is relatively low and delays on the order of several minutes can be tolerated. For these particular applications the services offered by satellite systems may be underutilized and consequently not very cost effective. It should be noted that since the existence of an ionized meteor trail is typically on the order of one second, and the interarrival time of suitable meteoroids is on the order of one minute, meteor burst systems are best suited to those applications where information exchange need not occur in real-time. As a consequence, meteor burst transmissions are almost exclusively digital, although there appears to be some interest in using this mode for voice transmission [6].

While various data transfer techniques exist, the ensuing discussion summarizes the procedures followed by one such scheme as outlined in Schanker [4]. A meteor burst station wishing to transfer information transmits a "probe" signal either continuously, in the case of a full-duplex system, or intermittently, in the case of a half-duplex system. When the probe signal is detected by the intended receiving station, it assumes that the ionized trail is suitable for transferring information and sends the appropriate signal to the originating station. Upon reception of this signal, the originating station begins to transmit its data. In full-duplex operation, the originating station monitors its receiving channel for acknowledgments from which it determines which portions of the message need to be retransmitted. For stations operating in the half-duplex mode, the originating station must suspend transmission periodically and wait for acknowledgment signals from the intended receiving station. In both modes of operation this procedure is called automatic repeat request (ARQ) and continues until all the information has been successfully transferred to the intended receiving station. It should be noted that due to the intermittent nature of the meteor burst channel, the communication equipment must be computer controlled in order to fully utilize the channel when trails of suitable ionization are present.

1.2 Literature Review

Since the first report of improved broadcast reception associated with meteor showers in 1931 (Pickard [7]), much research in meteor burst communication has been undertaken. This research has encompassed many areas of study, ranging from astronomical observations of meteors and their properties (i.e. mass and radiant distribution, size, etc.) to the analysis of communication protocols for meteor burst communication networks. Due to the voluminous nature of this work a complete review of this material is impractical and for this reason, this section will focus on literature directly related to telecommunication engineering issues.

An important issue in communication system engineering is that of assessing the transmission losses and resulting signal-to-noise ratio for a specific link configuration given the parameters of the transmitting and receiving stations such as transmitter power, antenna gains, operating frequency and separation distance. Fortunately, the loss mechanism associated with forward scatter meteor communication has been researched in detail by Eshleman [8], Manning [9] and others. The significant contribution of this work is a series of equations that can be used to estimate the transmission losses for a specified link geometry. A good discussion of this work and its application in communication system engineering can be found in Sugar [1] and McKinley [3]. More recently, discussions of this work have appeared in Weitzen [10], Spezio [11], and Schanker [4].

A successful analysis of a meteor burst network requires the ability to realistically predict the numbers, arrival rates and durations of meteor trails at any given location on the planet. Information of this type is invaluable since it permits the system designer to estimate such things as the waiting time between transmissions and the amount of information that can be transferred over a link on both a per/burst and daily basis. Additionally, knowledge of the spatial distribution of meteors can be used to determine

how the antenna system may be optimized to take advantage of those areas of the sky having the greatest meteor activity. In some applications, such as large networks and military systems, this knowledge can be used to estimate coverage areas in order to assess potential interference problems (in the case of multi-user systems) or the military equivalent - the probability of intercept. Research detailing models that predict meteor activity, arrival rates, and hot-spot locations can be found in Weitzen [5,12], Ince [13], Oetting [14] and Brown [15].

Recent renewed interest in meteor burst communication has sparked research into the design of communication networks utilizing this mode of propagation. Areas of current activity include the analysis and comparison of protocols (for maximizing throughput), investigations of coding strategies (for reducing the bit error rate) and studies of adaptive rate modulation schemes [16-21]. While protocol evaluation, modulation and coding strategies are important issues in network design, these issues cannot be fully addressed until the channel characteristics have been thoroughly investigated. In the majority of the analyses to date, the channel model assumes that the only impairment resulting in bit errors is additive white gaussian noise (AWGN). Although adequate in some instances, this model does not account for other potential impairments such as those arising from multipath interference. To date however, little work has been undertaken to assess the multipath nature of the meteor burst channel, with the notable exception of that undertaken by Akram [22], Weitzen [23,24] and Eriksson [25]. Since one of the important keys to developing improved meteor communication systems hinges on knowledge of the channel impairments, the measurement and characterization of these impairments will be the focus of this thesis.

1.3 Thesis Objectives

The information presented in the remaining chapters of this thesis focuses exclusively on satisfying the two fundamental objectives of this research. The first objective is to demonstrate the practicality of meteor burst communication systems through the combined use of theoretical arguments and experimental results. Specifically, this assessment will be based on the analysis and measurement of the transmission losses, waiting times, burst durations as well as rough estimates of system performance assuming that the only channel impairment is additive white gaussian noise. The second objective of this work is to measure and characterize the multipath nature of the meteor burst channel. This investigation will focus expressly on the measurement of the channel impulse response with the underlying goal of developing a suitable model for the meteor burst communication channel. The organization and presentation of this material is outlined in the following section.

1.4 Thesis Outline

The arguments and analyses presented in Chapters 2 and 3 serve to establish the practicality of meteor burst communication systems. A complete analysis of a digital meteor burst link requires among other things, the analysis of transmission losses as well as estimates of the signal-to-noise ratio, bit error rate, waiting time, transmission time and throughput. Those topics pertaining to the received signal level quality (transmission losses, SNR and BER) are the focus of Chapter 2, while the remaining topics relating to the statistics of meteor arrivals and the longevity of their ionized trails are the focus of Chapter 3. In both chapters, measurements from an experimental link between London and Ottawa are used to support the validity of analytical results, or in those cases where analytical solutions were not readily available, to estimate values for vital parameters. It should be noted that the results of the bit error rate and throughput analyses presented at the end of these chapters are only rough estimates of these quantities and that these results

are based on the assumption that additive white gaussian noise is the only channel impairment. As a consequence of this assumption, the data rates assumed in the throughput analysis may not be feasible due to the potential bandwidth limitations of the channel.

As part of the investigation into channel impairments, the design and testing of an impulse response measurement system is outlined in Chapters 4 through 6. This system relies on the transmission of a wideband linear F.M. (chirp) signal and its subsequent demodulation to obtain measurements of the channel impulse response, path-loss, flight-times and path-lengths. Details of the system including the designs of the transmitter and receiver are presented in Chapter 4. Although a portion of the receiver function was implemented in software, a discussion of the issues related to signal processing techniques is presented separately in Chapter 5. Complete details regarding the testing and calibration of the measurement system follow in Chapter 6.

Following extensive testing procedures, the impulse response measurement system was deployed at the same London and Ottawa locations from which the experimental data of Chapters 2 and 3 was collected. The advantage of this approach is that any residual measurement anomalies of the "new" system can be easily identified. These "new" measurements, their analysis and interpretation are the subject of Chapter 7. Specifically, the data will be analyzed to obtain information about the path gain, flight-times and delay spread as well as information regarding the channel impulse and frequency response functions. Furthermore, these measurements will be used to characterize the channel impairments and estimate the bandwidth limitations of the meteor burst channel. A summary and discussion of all important results follows in Chapter 8.

Chapter 2

Link Loss Evaluation

2.1 Introduction

The material presented in this chapter is directed at assessing those factors which affect the quality of the received signal on a forward scatter meteor burst link. This assessment will begin with an examination of the equations governing the link loss mechanisms for both the underdense and overdense trail conditions. As part of this analysis, realistic parameter values will be selected and used to estimate the losses for a 500 km link. Following this theoretical evaluation, the validity of these predictions will be confirmed using measurements from an experimental 500 km link between London and Ottawa. The loss estimates obtained from this analysis will then be used to assess the signal-to-noise ratio and error rate performance for a link spanning this distance. It should be noted that while the theoretical results are applicable to links of other distances, the application of these results to the 500 km London-Ottawa link was undertaken to demonstrate the practicality of medium-range meteor burst communication.

2.2 Theoretical Considerations

The equations governing the received power level on a forward scatter meteor burst link are given by Sugar [1] and Weitzen [10] as:

$$P_R(t) = \frac{P_T G_T G_R \lambda^3 q^2 r_e^2 \sin^2 \gamma}{16\pi^2 R_T R_R (R_T + R_R)(1 - \cos^2 \beta \sin^2 \phi)} \exp\left(-\frac{8\pi^2 r_o^2}{\lambda^2 \sec^2 \phi}\right) \exp\left(-\frac{32\pi^2 Dt}{\lambda^2 \sec^2 \phi}\right) \quad (2.1)$$

for the underdense condition and:

$$P_R(t) = \frac{P_T G_T G_R \lambda^2 \sin^2 \gamma}{32\pi^2 R_T R_R (R_T + R_R)(1 - \cos^2 \beta \sin^2 \phi)} \left[\frac{(4Dt + r_o^2)}{\sec^2 \phi} \ln\left(\frac{r_e q \lambda^2 \sec^2 \phi}{\pi^2 (4Dt + r_o^2)} \right) \right]^2 \qquad (2.2)$$

for the overdense condition. In both of these equations, t is the time variable, P_R is the received power, P_T is the transmitted power, G_T and G_R are the gains of the transmitting and receiving antennas respectively, λ is the free space wavelength, q is the electron line density in electrons per meter, r_e is the classical radius of an electron, r_o is the initial trail radius, γ is the angle between the incident electric field vector and the line of sight to the receiver, β is the angle between the plane of propagation and the trail axis, ϕ is the half-scatter angle, D is the diffusion coefficient (m^2/sec) and R_T and R_R are the lengths of vectors from the transmitter and receiver to the point of tangency on the trail (see Figure 2.1). While equation 2.1 is valid for electron line densities less than 10^{14} electrons per meter, in accordance with the definition for the underdense trail, equation 2.2 is valid for overdense trails having electron line densities ranging from 10^{14} to 10^{16} electrons per meter. For overdense trails with electron line densities in excess of 10^{16} (e/m) the received power level is governed by the equation (Weitzen [10]):

$$P_R(t) = \frac{P_T G_T G_R \lambda^2 \sin^2 \gamma}{32\pi^2 R_T R_R (R_T + R_R)(1 - \cos^2 \beta \sin^2 \phi)} \left[\frac{4Dt}{\sec^2 \phi} \ln\left(\frac{r_e q \lambda^2 \sec^2 \phi}{(1 + 0.16t^2)^{1/2} \pi^2 4Dt} \right) \right]^{1/2} \qquad (2.3)$$

where all parameters are as previously defined. Since equation 2.3 applies to trails of very long duration (relatively rare events) the analysis presented in the remainder of this chapter will be restricted to the underdense and overdense trails described by equations 2.1 and 2.2. It should be noted that because the variability of the angles β and γ are generally unknown, both of these angles are customarily taken to be 90 degrees (Schanker [4], Hines and Forsyth [37]). As a result of these assumptions, the equations of interest can be evaluated for a specific set of link parameters.

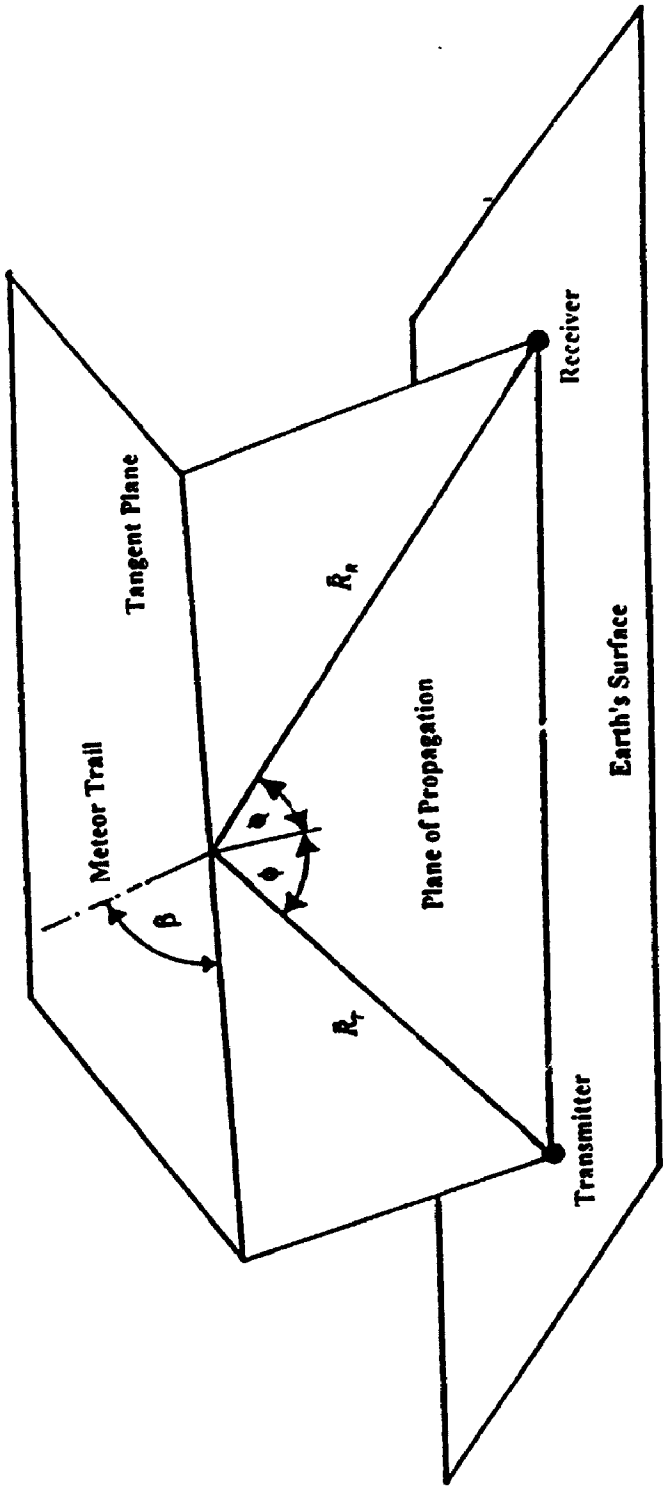


Figure 2.1 Scattering Geometry (After McKinley [3])

In each of the equations describing the underdense and overdense conditions, the received signal level is a multi-variate function, of which many of the parameters cannot be altered by the telecommunication engineer. Of these parameters only the antenna gains, transmitter power and operating frequency can be varied to achieve a viable design for a given link geometry. While modifications to the transmitter power and antenna gains can be achieved with relative ease, selection of an operating frequency or range of frequencies is more involved because of practical limitations on antenna size, unwanted propagation modes (such as ionospheric scattering) and spectrum allocation issues. Currently, the frequency band considered for meteor burst systems ranges from a lower limit of 30 MHz to an upper limit of approximately 100 MHz. Furthermore, the received signal strength is related to the frequency of operation as suggested by equations 2.1 and 2.2. Solving these equations for their peak powers gives:

$$P_R(t) = \frac{P_T G_T G_R \lambda^3 q^2 r_e^2}{16\pi^2 R_T R_R (R_T + R_R)} \exp\left(-\frac{8\pi^2 r_o^2}{\lambda^2 \sec^2 \phi}\right) \tag{2.4}$$

for the underdense condition and:

$$P_R(t) = \frac{P_T G_T G_R \lambda^3}{32\pi^3 R_T R_R (R_T + R_R)} \left[\frac{r_e q}{e}\right]^{\frac{1}{2}} \tag{2.5}$$

for the overdense condition, where the symbol e appearing in the denominator of the radical in equation 2.5 is Euler's number. From the preceding equations it is clear that the peak received power level and the operating frequency share the relationship

$$P_R \propto \frac{1}{f^3} \quad . \tag{2.6}$$

To further investigate this relationship, equations 2.4 and 2.5 were normalized by their peak values at an operating frequency of 30 MHz to facilitate an equitable comparison with systems operating at other frequencies. The equations resulting from this normalization are:

$$P_{RN} = \left(\frac{f_N}{f}\right)^3 \exp\left(-\frac{8\pi\sigma_0^2}{c^2 \sec^2\phi}\left(f^2 - f_N^2\right)\right) \tag{2.7}$$

and

$$P_{RN} = \left(\frac{f}{f_N}\right)^3 \tag{2.8}$$

for the underdense and overdense conditions respectively. In the preceding equations f_N denotes the normalization frequency (30 MHz), P_{RN} denotes the normalized power and all other variables are as previously defined. In order to assess the impact of the power-frequency relationship on a practical meteor burst system, the simplified link geometry and parameters of Figure 2.2 were assumed. A graph showing the normalized power as a function of the operating frequency for the underdense condition is shown in Figure 2.3. It should be noted that the initial radius, r_o, was estimated with the aid of the empirical equation (McKinley[3]):

$$\log_{10}(r_0) = 0.075H - 7.2 \tag{2.9}$$

and results in a value of 1.995 m for r_o - assuming a nominal altitude of 100 km. For the overdense condition, the evaluation of the normalized power equation is straightforward and the resulting relative performance curve is given by the broken line in Figure 2.3. In both cases, the degradation relative to the reference system (operating at 30 MHz) is apparent. At an operating frequency of 40 MHz the penalty, in terms of increased signal loss, is approximately 3 dB and increases to roughly 22 dB at a frequency of 100 MHz. Similar results apply for trails of the overdense type. Due to the severity of the losses incurred at higher operating frequencies, it is desirable to select frequencies closer to the low end of the band (30 MHz) whenever possible. It should be noted that operating frequencies below 30 MHz are seldom used in order to avoid ionospheric propagation modes whose channel characteristics are markedly different from those encountered in meteor scatter communications.

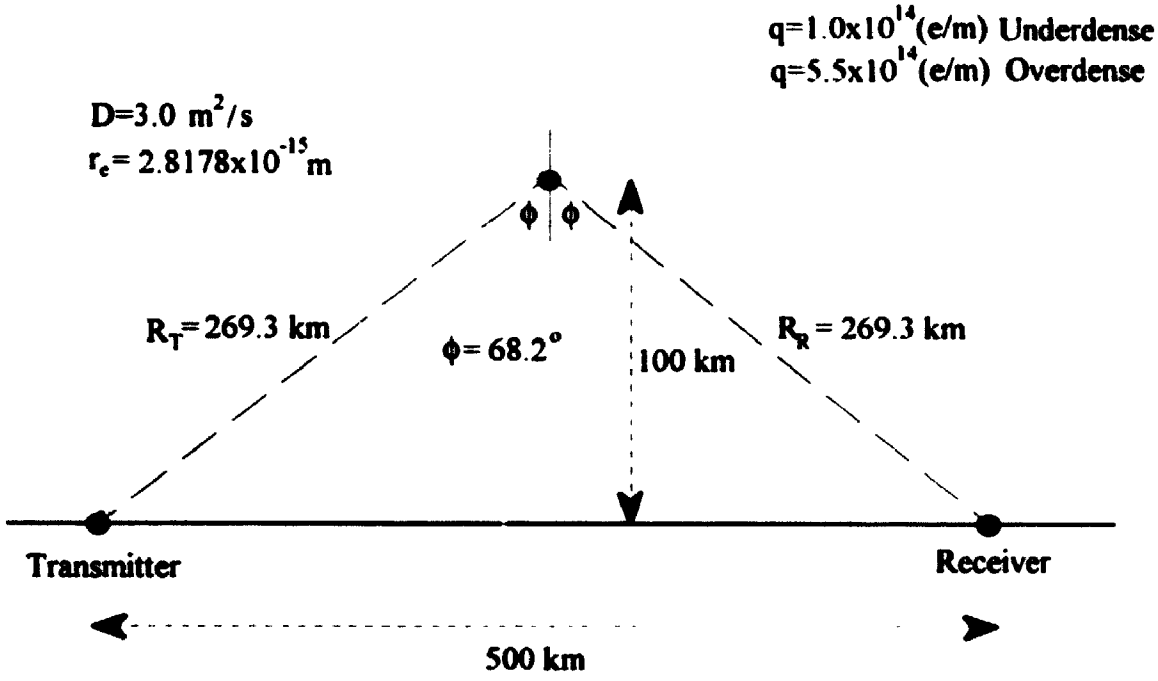


Figure 2.2 Simplified Forward Scatter Link Geometry

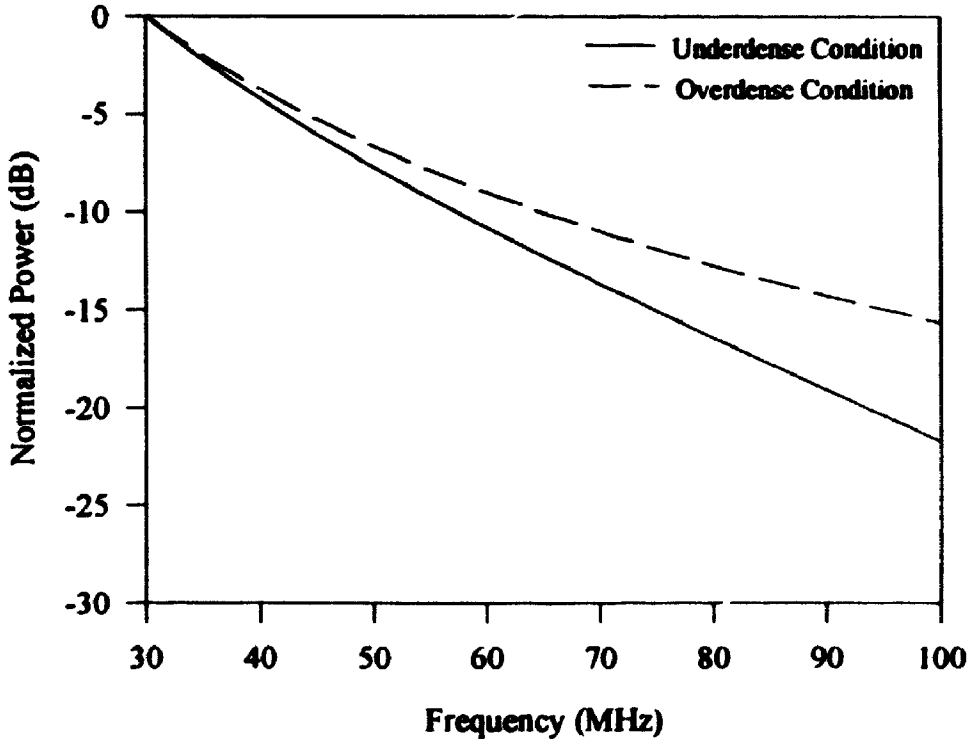


Figure 2.3 Power-Frequency Relationship (Underdense and Overdense Conditions)

Expressions for the propagation loss intrinsic to a specific link geometry and trail characteristics (electron line density, altitude initial radius etc.) can be derived through simple algebraic manipulation of equations 2.1 and 2.2. The equations resulting from such a manipulation are:

$$L = -10\log_{10}\left(\frac{\lambda^3 q^2 r_e^2}{16\pi^2 R_T R_R (R_T + R_R)}\exp\left(-\frac{8\pi^2 r_0^2}{\lambda^2 \sec^2\phi}\right)\right) \qquad (2.10)$$

and

$$L = -10\log_{10}\left(\frac{\lambda^3}{32\pi^3 R_T R_R (R_T + R_R)}\left[\frac{r_e q}{e}\right]^{1/2}\right) \qquad (2.11)$$

and permit the propagation loss, L (in dB), to be estimated for the underdense and overdense conditions respectively. Assuming the simplified link geometry and parameters illustrated in Figure 2.2, equations 2.10 and 2.11 can be evaluated to assess the link losses for trails of varying electron line densities. The results of these calculations are presented in Table 2.1. Because of the relationship between meteoroid mass, ionization and abundance (see Table 1.1), losses greater than 175 dB, corresponding to underdense ionization levels, are more likely to be encountered than the lower losses associated with trails of the overdense type. As a consequence, greater transmitter power and high-gain directional antennas are often employed in practical meteor burst systems to compensate for excessive losses.

2.3 Loss Measurements

In the analysis of the previous section many assumptions were required in order to estimate the propagation losses for a given link geometry. While tractable, the disadvantage of this approach is that the statistics of the path loss cannot easily be determined, since the distributions governing variables such as trail location, altitude, electron line density and scattering angles are unknown. Unfortunately under

Table 2.1 Relationship Between Link Losses and Electron Line Density

q (electrons/meter)	Loss (dB)
10^{16}	165
10^{15}	170
10^{14}	175
10^{13}	195
10^{12}	215
10^{11}	234
10^{10}	255

circumstances such as these, computer simulation techniques are inappropriate since they suffer from the same lack of knowledge that precluded the application of standard analytical approaches. As a result, it is necessary to rely on empirical techniques, from which the relevant statistics can be derived. Fortunately, due to Webster and Jones [2], a sufficient amount of propagation data has been collected from which path loss measurements and their statistics can be obtained.

The measurement system of Webster and Jones [2] (see Figure 2 4) consists of a 100 Watt, 48.7 MHz continuous wave (CW) transmitter located in Ottawa (45° 20' N, 75° 50' W) and a phase-swept interferometer located at Elginfield (43° 11' N, 81° 18' W) near London. When meteor trails of suitable ionization and orientation are present, the received signal activates the recording system and the signal variations from the reference antenna (R) are digitized and written to disk. The echo direction or equivalently its angle-of-arrival relative to the two orthogonal receiving arrays is determined by measuring the signal phases at the individual array elements relative to the phase on the common reference

antenna (R) For the antenna spacings shown in the diagram, the AOA of the echoes can be determined unambiguously from the measured signal phases which are also digitized and written to disk. While use will be made of the AOA measurements in subsequent chapters, the remainder of this section however, will deal exclusively with extracting path loss measurements from the echo recordings.

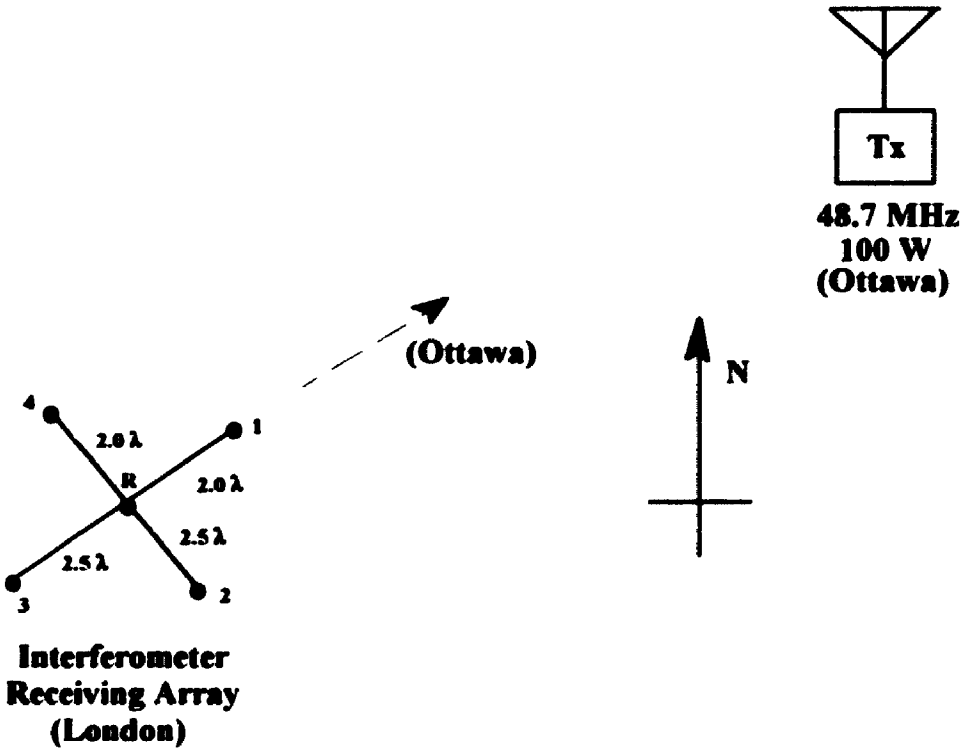


Figure 2.4 The Interferometer System Layout (after Webster and Jones [2])

Assuming that the transmitter power P_T, antenna gains G_T and G_R and the received power, P_R , are known, the path loss, L (in dB), can be estimated using the equation:

$$L_{dB} = P_T + G_T + G_R - P_R \qquad (2.12)$$

where; the quantities P_R and P_T are in dBm and G_T and G_R are in dB. As in the analysis of the previous section, the received power, P_R, is the peak power recorded during the burst and can be estimated from the interferometer echo recordings. Given that the transmitter power is known to be 100 W (50 dBm) and assuming antenna gains of 6 dB for the two-element Yagi antennas at the transmitter and receiver, loss calculations can proceed

according to equation 2.12. Applying this analysis to the data collected from March through December 1991 (≈ 89,000 observations), the minimum, maximum and average propagation losses were found to be 151 dB, 185 dB and 181 dB respectively and reflect

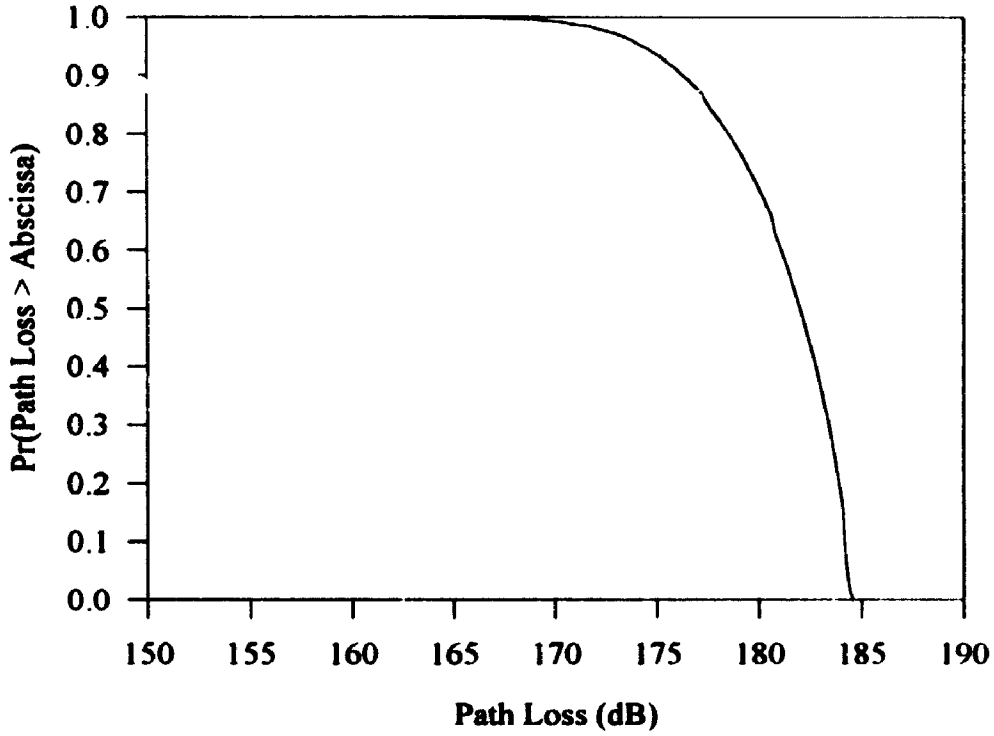


Figure 2.5 Path Loss Statistics for the London-Ottawa Link

contributions from both underdense and overdense trails. The statistics associated with these measurements are illustrated by the cumulative distribution function (CDF) shown in Figure 2.5. An examination of this function reveals that in 90% of the observations the propagation losses range between 170 and 185 dB. It should be noted that loss measurements greater than 185 dB are notably absent from the propagation data due to sensitivity limitations of the receiver. Furthermore it should be noted that the measured losses are consistent with the theoretical predictions of the previous section.

2.4 Signal-to-Noise Ratio

Communication system performance is usually assessed in terms of the bit error rate (BER) - a quantity that is intimately related to the signal-to-noise ratio (SNR) of the link. In this section, the signal-to-noise ratio attainable on a practical meteor burst link will be assessed on the basis of analytical results derived from the use of the propagation equations introduced in section 2.1. The objective of this approach is to evaluate variations in the SNR during typical underdense and overdense bursts. This knowledge will then be used to estimate an upper bound for the error-rate performance of a 500 km forward scatter link.

The signal-to-noise ratio is defined as the ratio of the received signal power to the total noise power within the receiver bandwidth. In equation form this is expressed as:

$$SNR = \frac{P_R}{P_N} \tag{2.13}$$

where; P_R is the received signal power and P_N is the noise power. Since the received signal power can be estimated using equations 2.1 and 2.2, it remains to obtain a suitable expression for the power delivered by the noise signal. For the simple receiving system shown in Figure 2.6, the noise power can be computed using:

$$P_N = k\left(T_A + T_{eq}\right) B_N \tag{2.14}$$

where, k is Boltzmann's constant, T_A is the antenna temperature and T_{eq} and B_N are the equivalent noise temperature and bandwidth of the receiver. Combining equations (2.1) and (2.14) gives:

$$SNR = \frac{P_T G_T G_R \lambda^3 q^2 r_e^2 A}{16\pi^2 R_T R_R (R_T + R_R) k\left(T_A + T_{eq}\right) B_N} \exp\left(-\frac{32\pi^2 Dt}{\lambda^2 \sec^2 \phi}\right) \tag{2.15}$$

for the underdense condition where A is given by:

$$A = \exp\left(-\frac{8\pi^2 r_0^2}{\lambda^2 \sec^2 \phi}\right). \tag{2.16}$$

Similarly, combining equations 2.2 and 2.14 gives:

$$SNR = \frac{P_T G_T G_R \lambda^2}{32\pi^2 R_T R_R (R_T + R_R) k (T_A + T_{eq}) B_N} \left[\frac{4Dt + r_0^2}{\sec^2 \phi} \ln\left(\frac{r_e \lambda^2 \sec^2 \phi}{\pi^2 (4Dt + r_0^2)}\right)\right]^{1/2} \tag{2.17}$$

for the overdense condition. In both equations (2.15 and 2.17) all parameters are as previously defined. Equations 2.15 and 2.17 clearly show that the SNR varies continuously throughout the duration of the burst. The implication of this effect is that the BER also varies throughout the burst resulting in transmission errors occurring more frequently during those intervals of greatest SNR variation. Typically, these greatest variations would be expected to occur toward the end of a burst.

Realizations of the signal-to-noise ratio described by the preceding equations are shown in Figures 2.7 and 2.8 for receiver noise bandwidths, B_N, of 1, 10 and 100 kHz. For both trail conditions, the link geometry of Figure 2.2, and the receiver parameters of Figure 2.6 were assumed. Additionally, a transmitter power (P_T) of 1kW, an antenna temperature (T_A) of 10,000 Kelvin (Jordan and Balmain [36]) and antenna gains (G_T, G_R) of 10 were employed. In both instances, the result of the bandwidth increase is a corresponding decrease in the overall signal-to-noise ratio which is accompanied by a reduction in the length of time the SNR remains above some minimum acceptable threshold (6 dB for convenience). Fortunately, this reduction in the effective burst duration is relatively small compared to signaling rate increase. For the underdense condition shown in Figure 2.7, the effective duration is decreased by a factor of 4 while the transmission rate has been increased by a factor of 100. In the case of the overdense condition illustrated in Figure 2.8, the reduction in the effective burst duration is negligible for an identical increase in transmission rate. Based on the criterion of peak SNR and

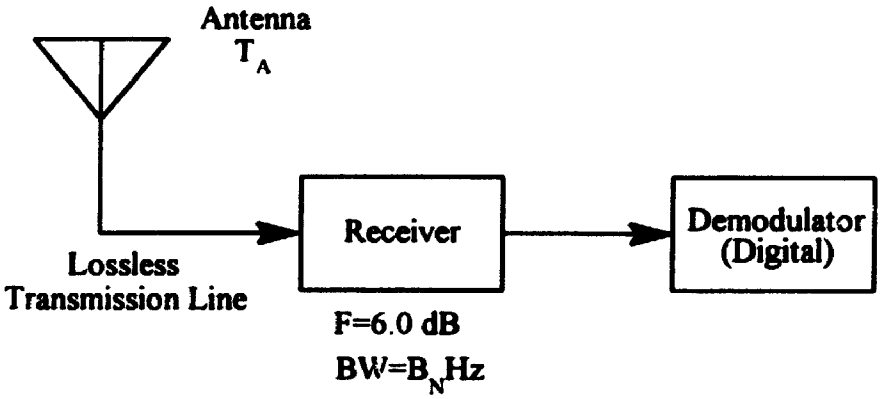


Figure 2.6 Simplified Receiving System Block Diagram

effective burst duration it is clearly advantageous to make use of overdense trails whenever they occur.

Assuming that additive white gaussian noise (AWGN) is the only channel impairment responsible for bit errors, an upper bound on the error rate performance can be established by computing the probability of error assuming a minimum SNR criterion. In the preceding SNR analysis, it was observed that for a relatively wide receiver bandwith of 100 kHz, peak signal-to-noise ratios of between 10 and 20 dB could be achieved, depending on the level of trail ionization. Due to the consistency between the theoretical and experimental results of sections 2.2 and 2.3, it would appear that the parameters assumed for the link geometry are representative of those encountered on a typical 500 km forward scatter link. As a result, it is reasonable to suppose that the SNR obtained under similar assumptions is also representative of those likely to be encountered in a practical operating system. The error probabilities, based on a minimum SNR threshold of 6.0 dB, are presented in Table 2.2 for several commonly used modulation schemes. Although relatively high by digital communication standards, these values represent the upper limit on the error rate since the demodulator in a real system would be designed to ignore the receiver output if the SNR were to drop below the threshold setting. Furthermore, the error rate during the initial portion of the burst has the potential to be much lower, as a

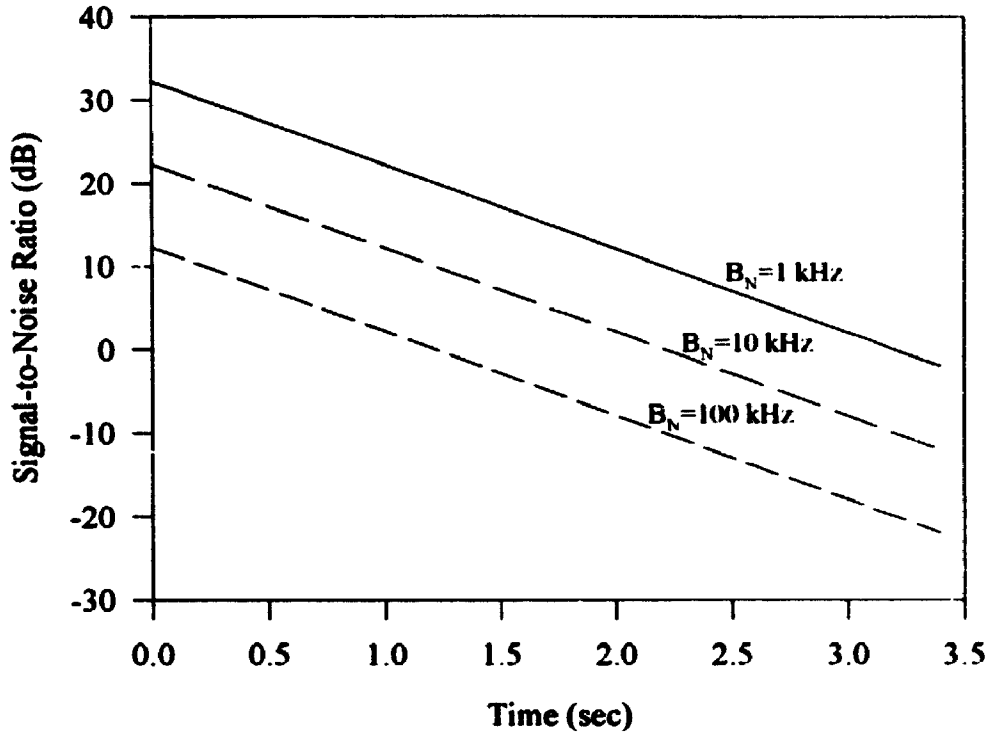


Figure 2.7 Signal-to-Noise Ratio Realization for an Underdense Trail

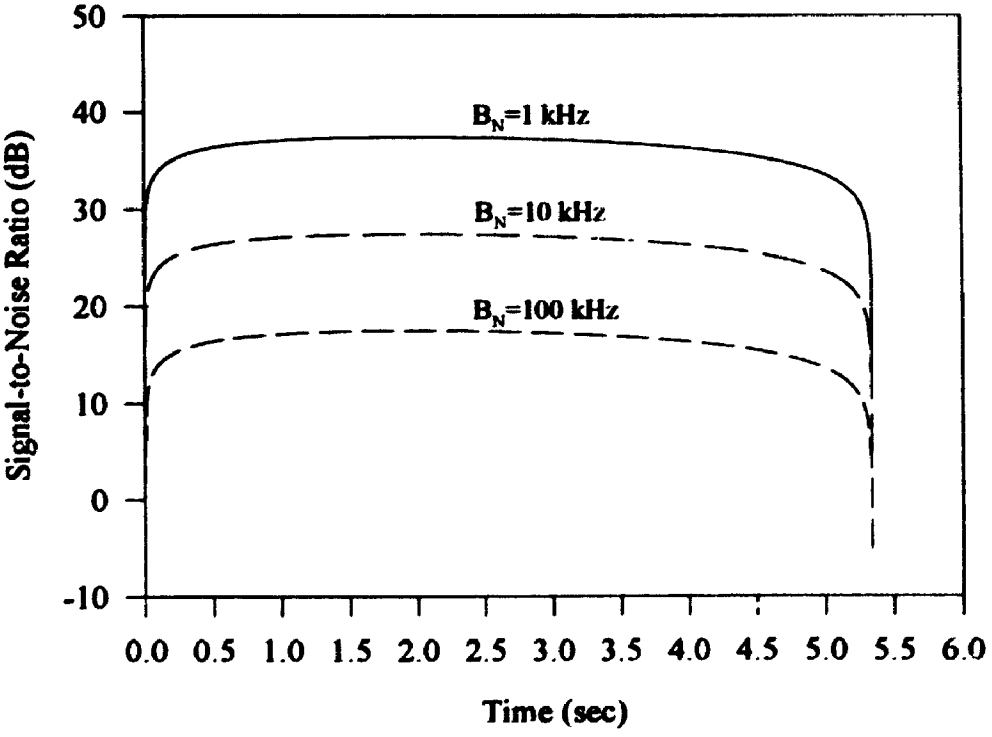


Figure 2.8 Signal-to-Noise Ratio Realization for an Overdense Trail

result of the inherent rise in the received power during the initial phase of the burst. Based on the results of the BER analysis, it is evident that modulation schemes in which coherent detection is employed at the receiver (excluding OOK) are desirable for meteor burst communication. It should be noted that improvements in error rate performance can be achieved through the use of error control coding - a discussion of which can be found in Metzner [16] and Miller [17].

Table 2.2 Error Rates for Common Modulation Schemes (SNR=6.0 dB)

Modulation Scheme	P_e (Coherent Detection)	P_e (Noncoherent Detection)
OOK	0.023	0.068
BPSK	0.002	----
FSK	0.023	0.068
QPSK	0.002	----

2.5 Summary

Calculations using the propagation equations of section 2.1 revealed that the losses on a 500 km link, typical of that spanning the distance from London to Ottawa, could range from 255 to 165 dB depending on the trail ionization. These figures were supported by experimental evidence which showed these losses to range from 185 to 151 dB. Discrepancies between the largest predicted losses and those measured experimentally are attributed to the limitations of the measuring instrument. A statistical treatment of these data found that on average the path loss was 181 dB and that 90% of the time, the path loss was between 185 and 170 dB - an overall variation of 15 dB. In a practical communication system, the effect of these losses could be minimized through the use of high-gain directional antennas, increased transmitter power and, to a limited extent, by a judicious selection of the operating frequency. Due to the inverse-cube relationship

existing between the received power and the operating frequency (see section 2.1) it is advantageous to select a frequency as close to the lower limit of the operating window as possible. Frequently, operating frequencies in the vicinity of 40 MHz are chosen in attempts to capitalize on the rejection of unwanted propagation modes (such as ionospheric scattering), spectrum availability, and the advantages afforded by more compact antennas.

In section 2.3 a signal-to-noise ratio analysis was carried out for a 500 km forward scatter link. The results of this analysis showed that for realistic transmitter, receiver and antenna characteristics acceptable signal-to-noise ratios could be achieved during both underdense and overdense conditions. This analysis also demonstrated that relatively good signal-to-noise ratios could be achieved even when the receiver bandwith is relatively large (typically 100 kHz). Consequently, it is reasonable to assume that relatively high transmission rates can be supported by this mode of propagation. Accepting that during the majority of useful bursts a minimum SNR of 6 dB can be achieved, an upper bound for the error rate was deduced. For standard transmission schemes employing coherent detection at the receiver, the upper limit on the error rate probability was found to be 2.3×10^{-2}. It should be noted that this value constitutes a worst case estimate and that it is implicitly assumed that additive white gaussian noise is the only channel impairment responsible for transmission errors.

The material presented in this chapter has demonstrated that meteor burst communication can be established without the use of exotic and impractical equipment Since this mode of communication relies on meteoroids to function, the material in the following chapter will focus on issues relating to the number and arrival rate of meteors, their location and temporal extent.

Chapter 3

Link Availability

3.1 Introduction

In the previous chapter it was found that a sufficiently large signal-to-noise ratio could be achieved without the use of exotic equipment at the transmitter or receiver locations. To further establish the usefulness of this communications mode it remains to be demonstrated that the channel throughput is sufficiently large to support the information transfer rate required by a specific telecommunications application. The channel throughput is a measure of the amount of data that can be transferred over the channel per unit time and is a function of the time available for transmission (the burst duration) as well as the time elapsed between consecutive bursts (the waiting time). Due to the complexities of the mechanisms governing these parameters, the throughput calculations of this chapter will rely on experimental results extracted from the interferometer system of Webster and Jones [4]. In addition to the throughput assessment, the variability of meteor arrival rates, hot spot locations and flight-times will also be discussed and will be introduced prior to the throughput calculations.

3.2 Meteor Arrivals

The meteor arrival process is a random process which over a short period of time has been found to conform to the statistics associated with a Poisson distribution [35]. For this particular process, the probability of an arrival in the time interval, t_o is given by:

$$\Pr(t \le t_o) = 1 - e^{-Mt_o} \qquad (3.1)$$

where Pr denotes the likelihood of an arrival and M denotes the meteor arrival rate in meteors per hour. Unfortunately, in order to make use of this equation it is necessary to

possess a priori knowledge of the meteor arrival rate, M - a quantity which for a given link geometry and system parameters is usually unknown. Although unknown numerically, the arrival rate is known to exhibit diurnal and seasonal variations which can be explained in terms of the Earth's rotation and motion about the sun. Owing to the distribution of cosmic debris along the orbital path, the periods of most intense meteor activity occur during the early morning hours when the geographical position of the link coincides with the Earth's (see Figure 3.1(a)) apex. Beyond this time the flux begins to diminish and is composed of those meteoroids whose velocities are sufficient for them to catch up with the Earth (see Figure 3.1(b)). As the Earth completes its 24 hour rotation, the flux of meteoroids increases as the geographic location of the link once again coincides with the direction of the Earth's apex. Seasonal variations, on the other hand, result from variations

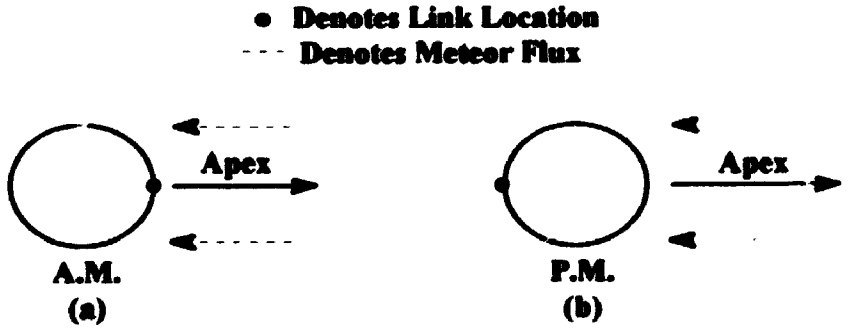


Figure 3.1 Mechanism Responsible for the Diurnal Variation of Meteor Arrivals

in the density of particles encountered throughout the year as well as the tilt of the Earth's rotational axis. As a result, arrival rate increases are expected to occur during the months of June, July and August while increases due to the tilt of the rotational axis are expected to occur in the Northern and Southern hemispheres during the months of September and March respectively [6]. Although not explicitly stated at the outset of this paragraph, the arrival rate, M, represents the arrival rate of meteors that are "visible" to a particular system. The visibility of these meteors is governed by system specific parameters such as the transmitter power, antenna gains, operating frequencies and receiver sensitivity. Consequently, the arrival rates and interarrival times deduced or measured for a particular meteor burst system may not be directly applicable to another.

Using data from the interferometer system of Webster and Jones [4], the hourly arrival rates and daily number of arrivals for the London - Ottawa link were determined on a monthly basis. The usefulness of these measures is that the former provides valuable information about arrival rate variations, from which the hours of peak meteor activity can be identified, while the latter is an indicator of the potential messaging capability of the meteor burst system. Results showing the diurnal variation in arrival rates for the months of March through December 1991 can be found Figures 3.2 and 3.3. The data for each of these curves was obtained by computing the average number of arrivals during a given hour of the day for the month under consideration. These results were then smoothed using a 3 point moving average filter to suppress the high frequency components (i.e. abrupt transitions) from the curves. In all instances, diurnal variations in the arrival rate can be readily observed. Specifically, it should be noted that in all cases the greatest arrival rate occurs during the early morning hours at approximately 6:00 local time, decreases throughout the day reaching its minimum around 18:00 hours beyond which increases in the arrival rate are observed. Seasonal arrival rate variations were observed by plotting the average hourly rate as a function of the month as shown in Figure 3.4. According to these results, increases in meteor activity were encountered during the months of March, July, October and December. Based on the average hourly rates shown in this figure, the daily number of meteors producing ionized trails suitable for communication is estimated to range from approximately 336 for a daily average arrival rate of 14 meteors per hour to 432 for an average rate of 18 meteors per hour. Variations in the magnitude of these estimates are expected for other links.

An interesting feature of the observed meteor trails is the manner in which their locations are distributed with respect to the great circle path (GCP) between transmitter and receiver. As previously discussed in Chapter 1, any ionized meteor trail can be used to provide communication provided that the conditions for specular reflection have been

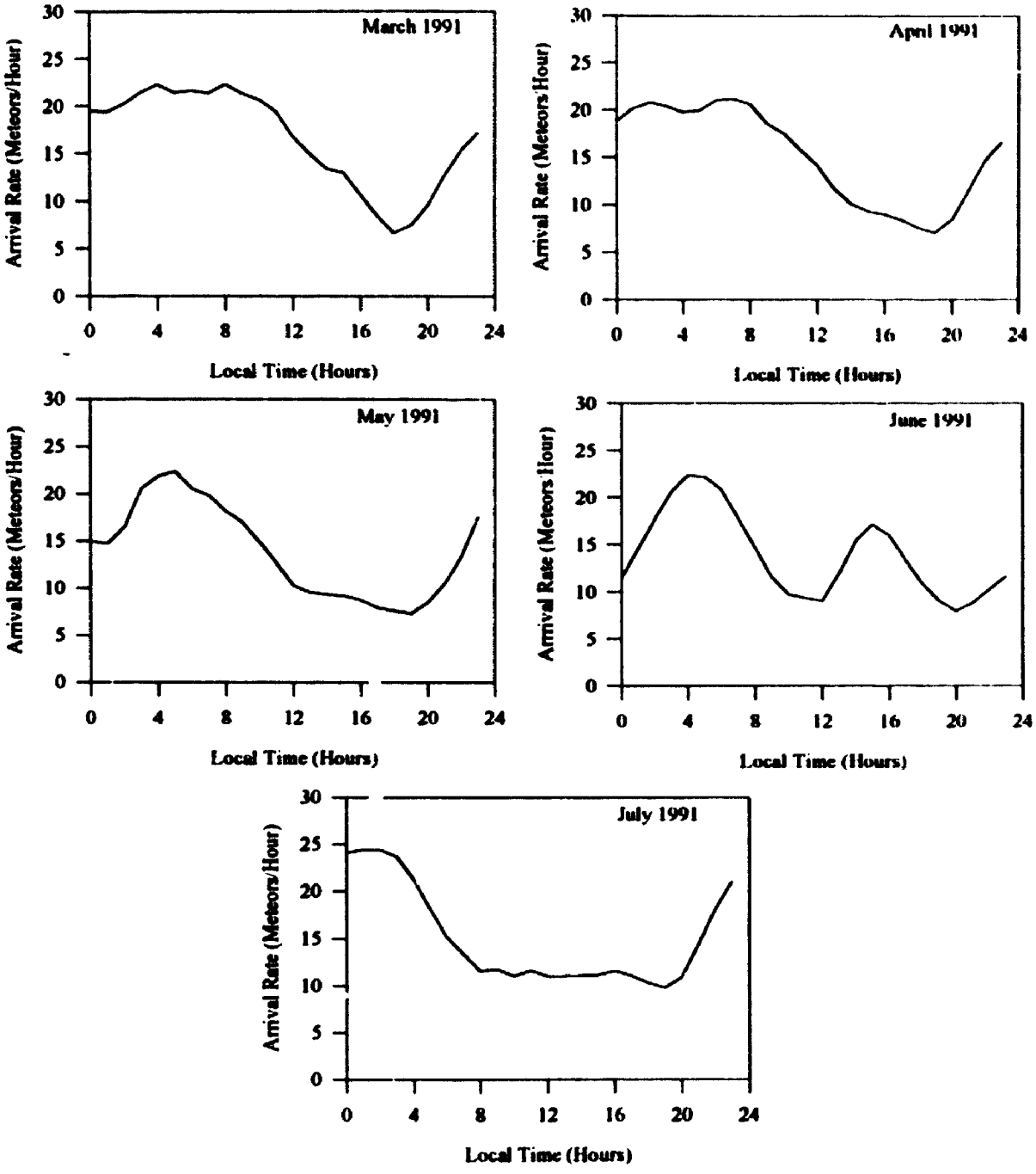


Figure 3.2 Diurnal Arrival Rate Variations for March through July 1991

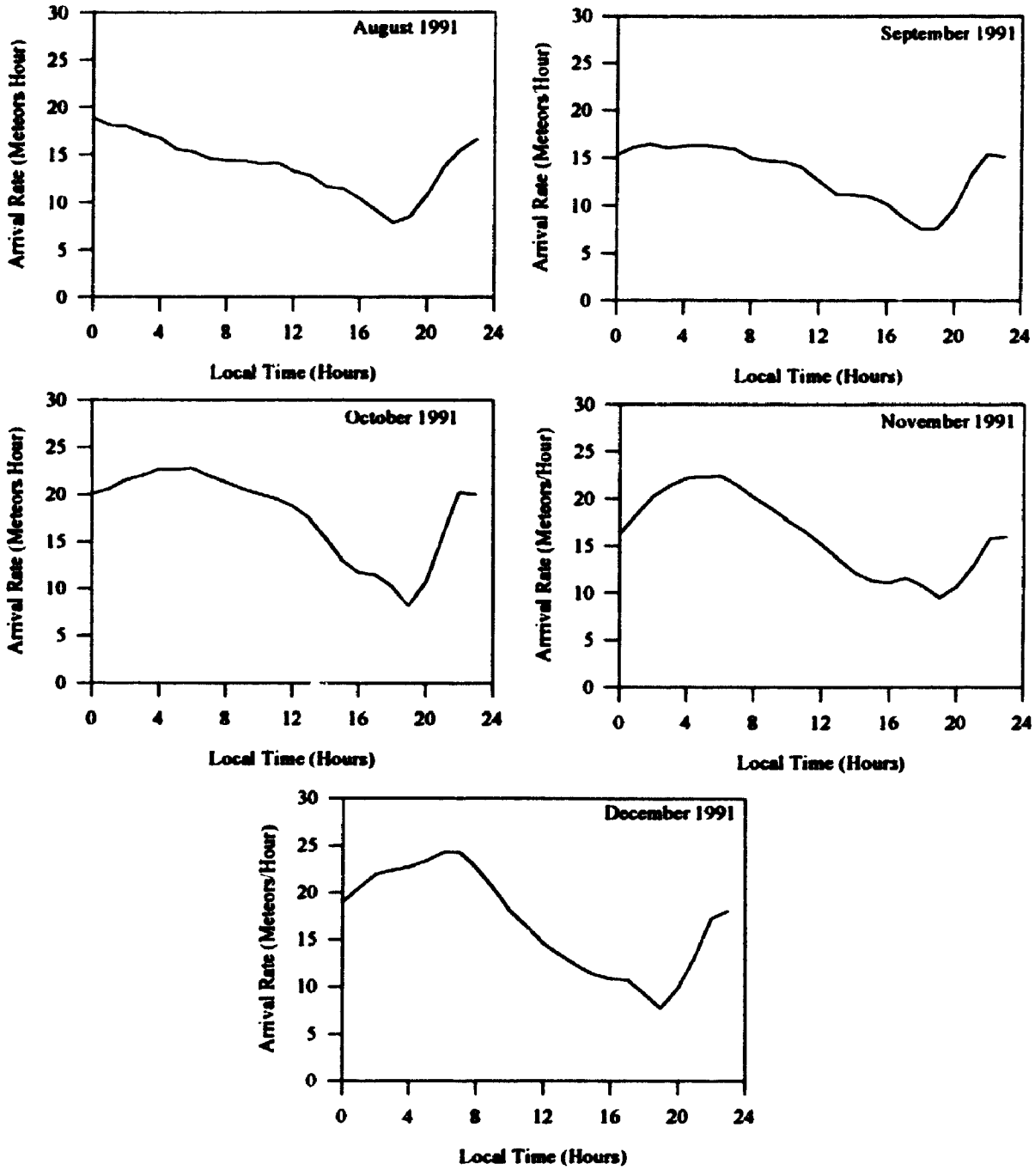


Figure 3.3 Diurnal Arrival Rate Variations for August through December 1991

satisfied. The positions of these trails, relative to the receiver location (see Figure 3.5), is measured by the interferometer system. Specifically, this measurement is that of the unit vector $\vec{T}$:

$$\vec{T} = a\hat{a}_x + b\hat{a}_y + c\hat{a}_z \tag{3.2}$$

where; $\vec{T}$ is parallel to the vector between the receiver and the trail ($\vec{R}_R$)and the coefficients a, b and c are such that

$$|\vec{T}| = 1. \tag{3.3}$$

Given that the vector $\vec{R}_R$ has the form:

$$\vec{R}_R = x_o\hat{a}_x + y_o\hat{a}_y + z_o\hat{a}_z \tag{3.4}$$

and that

$$\vec{R}_R = k\vec{T} \tag{3.5}$$

and by assuming that all meteor trails recorded by the system have an altitude of $z_o = 100$ km, the constant k can be determined and the positions of these trails can be observed by plotting the coordinates (x_o, y_o) for each trail. Figure 3.6 shows the spatial locations of these trails for the month of March 1991 as viewed in 4 hour intervals over a 24 hour period. From the figure it can be seen that communication was established by means of ionized trails occurring on both sides of the great circle path as well as those occurring at positions behind both the transmitter (5,0) and receiver (0,0). Furthermore, the apparent density of meteors occurring on either side of the GCP suggests that these locations of intense meteor activity or hot spots tend to vary throughout the day. Although not shown, similar findings were observed for the months of April through December 1991. To capitalize on the communications potential of this effect, meteor burst systems employing directional antennas capable of illuminating either side of the great circle path are highly desirable.

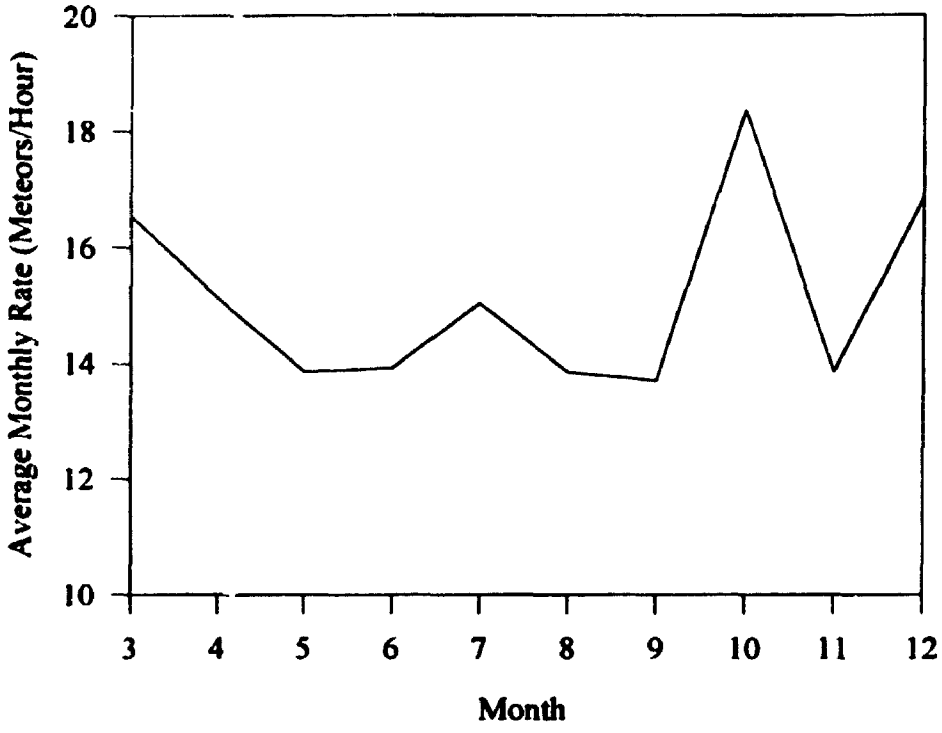


Figure 3.4 Seasonal Arrival Rate Variations for 1991

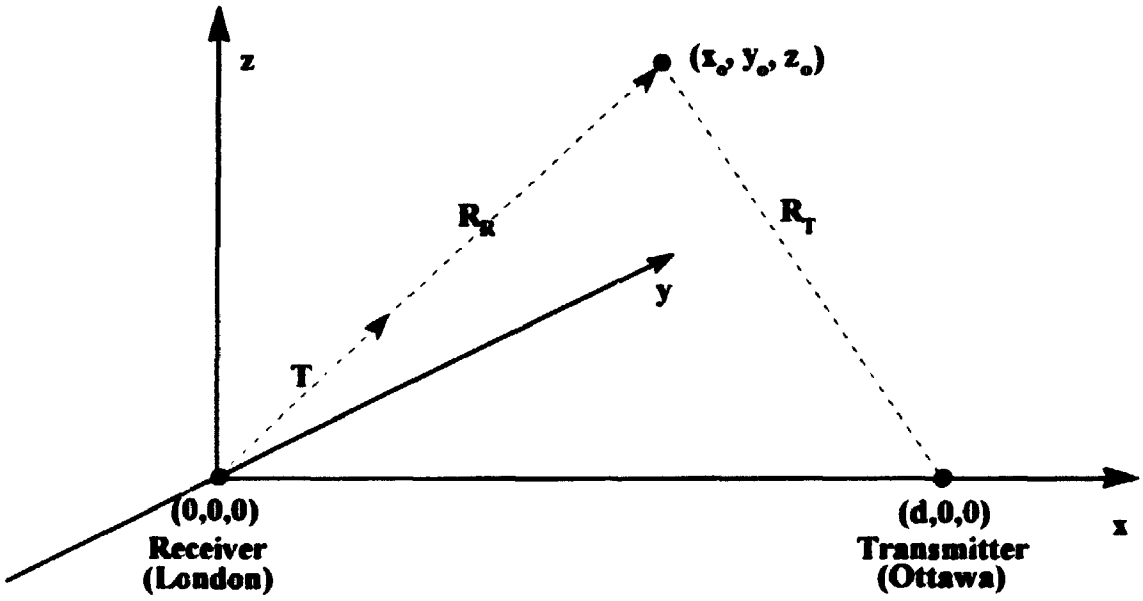


Figure 3.5 Coordinate System for Direction Measurements

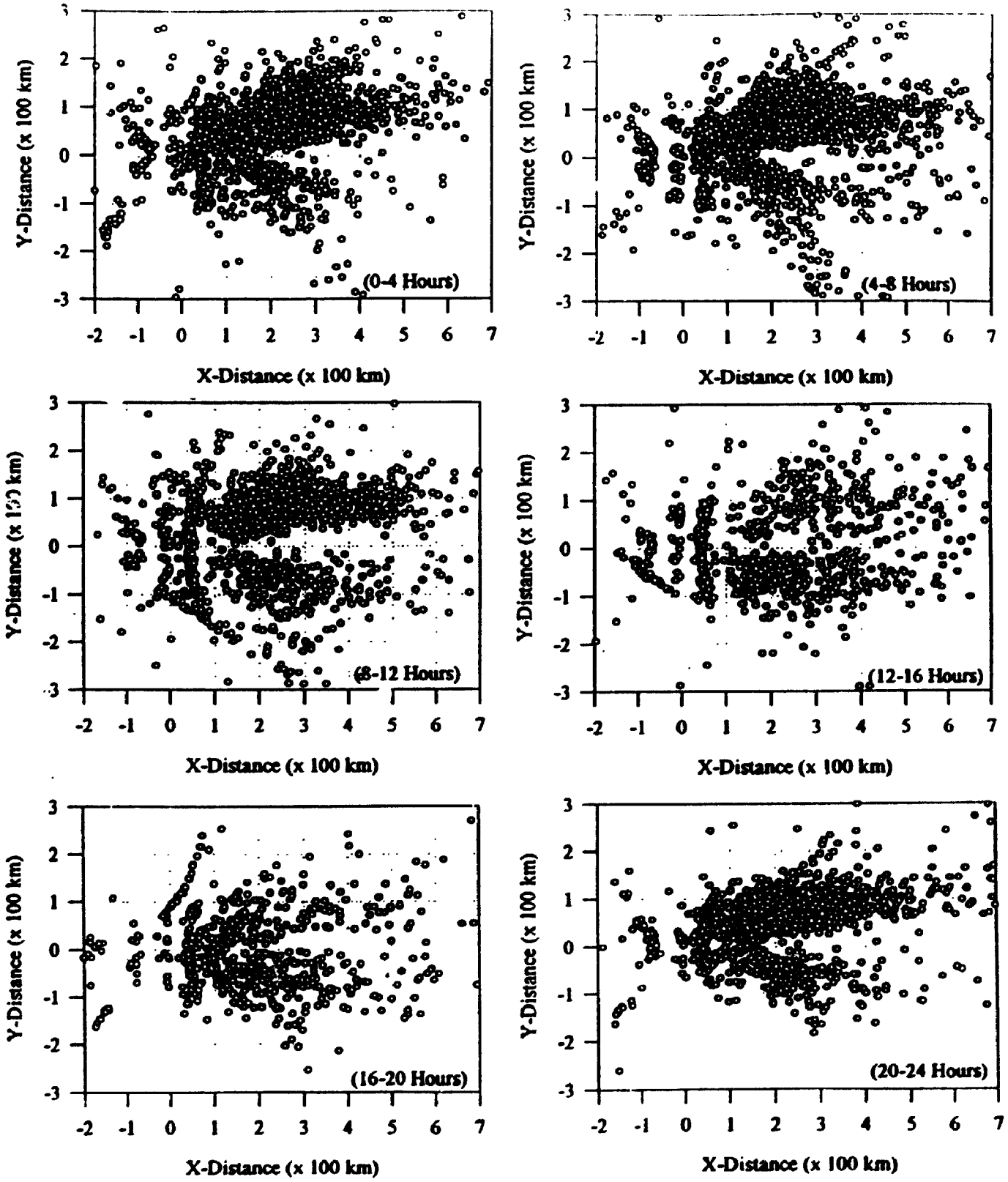


Figure 3.6 Hotspot Locations Viewed in 4 Hour Intervals for March 1991

Since the distance to the meteor trail can be estimated (again assuming a 100 km altitude) and the transmitter location is known, the path length between transmitter and receiver can be computed. Combining equations 3.2, 3.4 and 3.5, results in the equation:

$$\left|\vec{R}_R\right| = \left[\left(\frac{z_o a}{c}\right)^2 + \left(\frac{z_o b}{c}\right)^2 + z_o^2\right]^{1/2}, \qquad \textbf{(3.6)}$$

which gives the length of the vector from the receiver to the trail. The only remaining unknown is the distance from the trail to the transmitter $\vec{R}_T$ which from the geometry is:

$$\left|\vec{R}_T\right| = \left[(d - x_o)^2 + y_o^2 + z_o^2\right]^{1/2} \qquad \textbf{(3.7)}$$

and gives:

$$P_L = \left[\left(\frac{z_o a}{c}\right)^2 + \left(\frac{z_o b}{c}\right)^2 + z_o^2\right]^{1/2} + \left[\left(d - \frac{z_o a}{c}\right)^2 + \left(\frac{z_o b}{c}\right)^2 + z_o^2\right]^{1/2}. \qquad \textbf{(3.8)}$$

Since the path length, P_L and the propagation time, T_P are related by the equation:

$$P_L = cT_P \qquad \textbf{(3.9)}$$

where; c is the velocity of light in a vacuum, the path lengths calculated from the spatial location measurements can be used to determine the statistics of the propagation time. The value of this undertaking is that knowledge of flight-time variations may provide useful insight into methods by which synchronization may be achieved in practical meteor burst communication systems. Based on direction measurements for the months of March through December 1991, the flight-time statistics shown in Figure 3.7 were obtained. According to these results, in 90% of all observations the flight-time ranged from approximately 1.8 ms to 2.5 ms with an average flight-time of approximately 2.1 ms and a standard deviation of 0.5 ms.

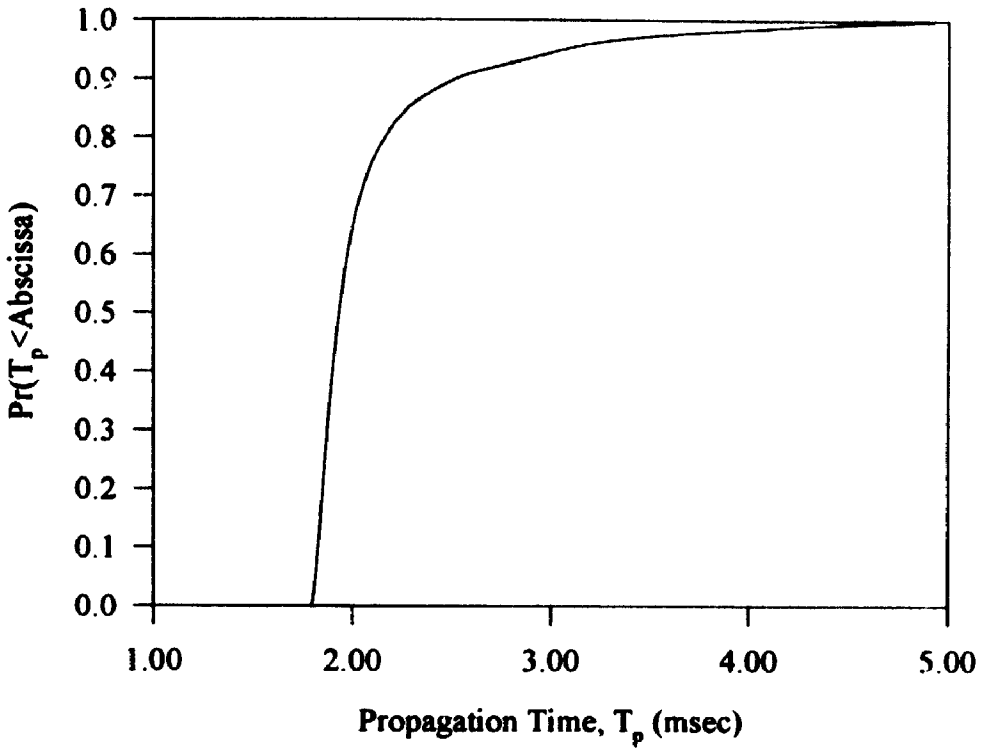


Figure 3.7 Flight-Time Distribution

3.3 Interarrival Statistics

The interarrival time or time between consecutive arrivals is subject to the same factors which influence the number of ionized trails that are visible to a communication system at radio frequencies. In terms of system performance, this parameter represents the length of time a meteor burst station must wait before being able to transmit information over the channel. Due to diurnal variations in the arrival rate, this waiting time varies throughout the day, with the shortest waiting times occurring during the early morning hours (06:00 local time) when the arrival rate is close to its maximum and the largest waiting times occurring in the evening hours (18:00 local time) when the arrival rate is nearest its minimum. A convenient measure of the interarrival time is its average value measured over a period of many days.

Using the time stamps from the interferometer records, the interarrival times were determined for the months of March through December 1991. For each month of recordings, the average value of the interarrival time was computed. The results of these measurements are shown in Table 3.1. In all instances, the average waiting time was found to lie between 3 and 4 minutes. For waiting times of this magnitude it is clear that applications relying on meteor burst communication must be tolerant to transmission delays and not require data exchange in "real" time. It should be noted that an estimate of waiting time could have been obtained through the use of equation 3.1 since some knowledge of the arrival rate, M is known. Assuming a uniform daily arrival rate of M=17 meteors per hour (a representative figure for the month of March), an event likelihood of 90 % and solving equation 3.1 for the waiting time, t_o, yields a value of roughly 8.1 minutes. The waiting time computed in this manner differs significantly from the mean waiting time as equation 3.1 does not take into consideration the diurnal arrival rate variations that are intrinsically included in the measurement of the **mean** waiting time.

3.4 Duration Statistics

The factors affecting burst duration or equivalently, the time available for transmission, can be determined by examining the time constant associated with equations 2.1 and 2.2 which govern the received power level on a forward scatter link. In the case of the underdense trail, the time constant of the received signal can be obtained in the usual manner by computing the time required for the received signal power to decay by a factor of e^{-1}. Applying this definition to equation 2.1 gives:

Table 3.1 Mean Waiting Times (March - December, 1991)

Month	Total Arrivals	Mean Waiting Time (min.)
March	8585	3.4
April	9390	3.7
May	4400	3.9
June	3972	4.0
July	6765	3.6
August	10171	4.1
September	9534	4.3
October	12794	3.2
November	11475	3.6
December	11957	3.4

$$\tau_u = \frac{\lambda^3 \sec^2 \phi}{32\pi^2 D} \tag{3.10}$$

where; τ_u is the time constant and all other parameters are as previously defined It should be noted that others, such as McKinley [2] and Sugar [1], define this time constant as the time required for the signal to decay by a factor of e^{-2} of its initial value. For the overdense trail condition, the time constant is defined as the time required for the argument of the natural logarithmic function to equal unity. Applying this definition to equation 2.2 and solving for the time variable, τ_o, gives:

$$\tau_o = \frac{r_e q \lambda^2 \sec^2 \phi - r_o^2 \pi^2}{4\pi^2 D} \tag{3.11}$$

as the time constant for the overdense trail. In both instances, the time constants given by equations 3.10 and 3.11 are decreasing functions of the operating frequency. This result suggests that in order to fully exploit the communication potential of a particular trail, an operating frequency as low as possible should be selected. Realizations of the time

constant, τ_u, for a half scattering angle, ϕ, of 68.2 degrees and diffusion coefficients, D, of 3 and 12 m²/sec are shown in the graph of Figure 3.8. From these curves it is clear that the magnitude of the diffusion coefficient can have a great impact on the time constant and consequently, the duration of the burst. For example, at an operating frequency of 30 MHz, τ_u is 750 ms for D=3 m²/sec and is reduced to 200 ms for D=12 m²/sec at the same operating frequency. As previously stated, the only way in which to minimize this effect and those due to frequency effects is to choose as low an operating frequency as is practical. Like the underdense case, the time constant associated with overdense trails exhibits similar dependencies with regard to both operating frequency and diffusion coefficient. The time constant, τ_o is also a function of both the electron line density, q, as well as the initial radius r_o. A graph illustrating these dependencies is shown in Figure 3.9. As in the underdense case, best performance is achieved by selecting the lowest possible operating frequency.

In the absence of a model describing the frequency and duration of over and underdense trails it is necessary to rely on actual burst recordings in order to obtain a realistic estimate of the duration of a typical burst and consequently, the length of time available for information transmission. For purposes of this work, the burst duration is defined as the total time over which the signal-to-noise ratio (SNR) remains above a 6 dB threshold. Under conditions such as those encountered during typical underdense trails, the SNR profile appears as shown in Figure 3.10 (a) and the duration measurement is straightforward as illustrated in the diagram. During overdense trail conditions, however, periods of fading may be encountered during which the SNR periodically falls below the threshold (Figure 3.10 (b)). Under these conditions the duration measurement is only slightly more complicated in that it must take into account all time intervals throughout the duration of the burst for which the SNR exceeds the threshold. A problem encountered with this approach is that of obtaining a realistic estimate of the noise power, P_N. One

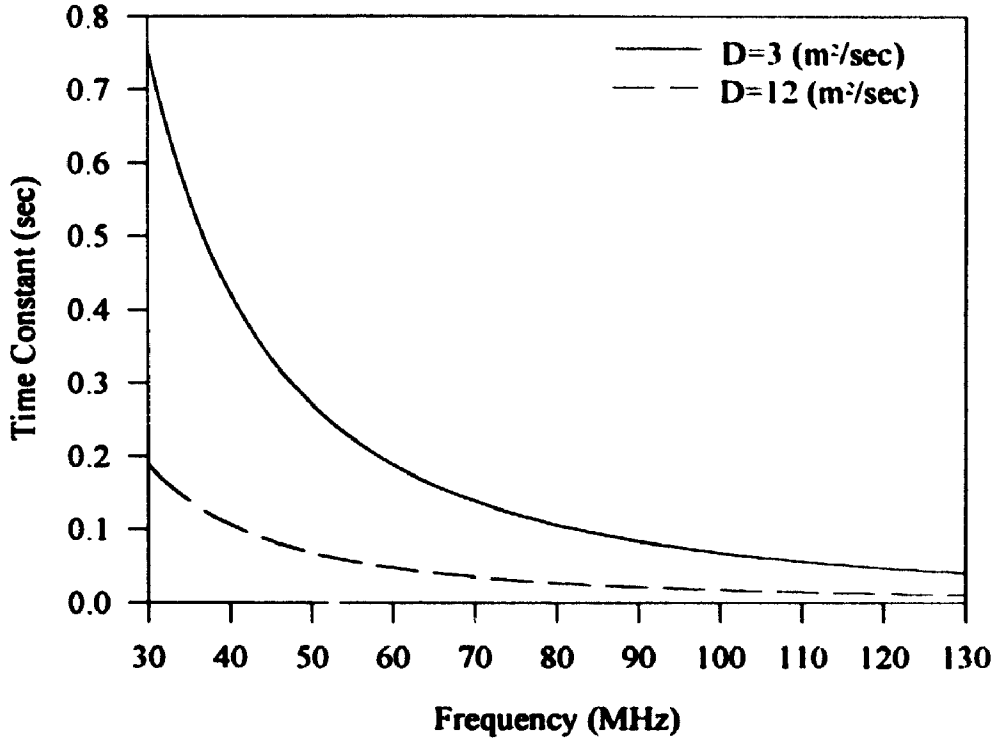


Figure 3.8 Time Constant Dependencies (Underdense Trails)

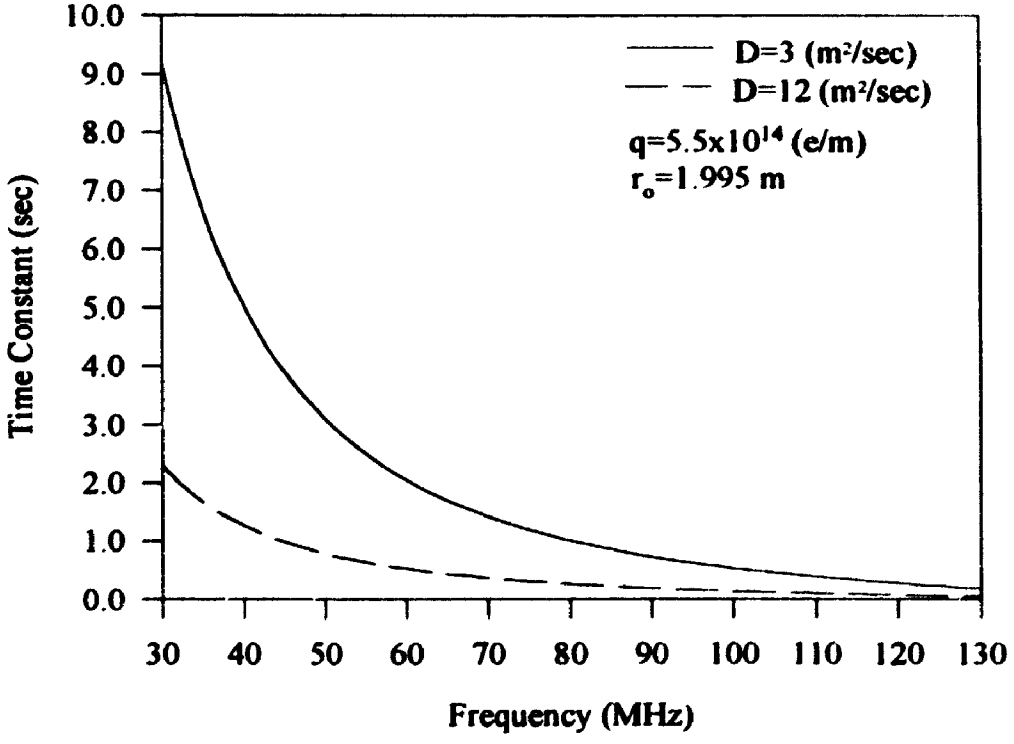


Figure 3.9 Time Constant Dependencies (Overdense Trails)

potential solution to this problem is to estimate the noise power from sections of the burst recording that are known to consist only of noise. The difficulty of this approach however, is that in some instances it is difficult for automatic detection schemes to differentiate between signal and noise, potentially leading to unrealistic values for the noise level. A solution to this problem is to assume that the noise power is constant for all bursts and that it can be determined from either theoretical considerations or empirical results. In the absence of artificial noise, the power delivered by the noise signal can be computed according to the equation:

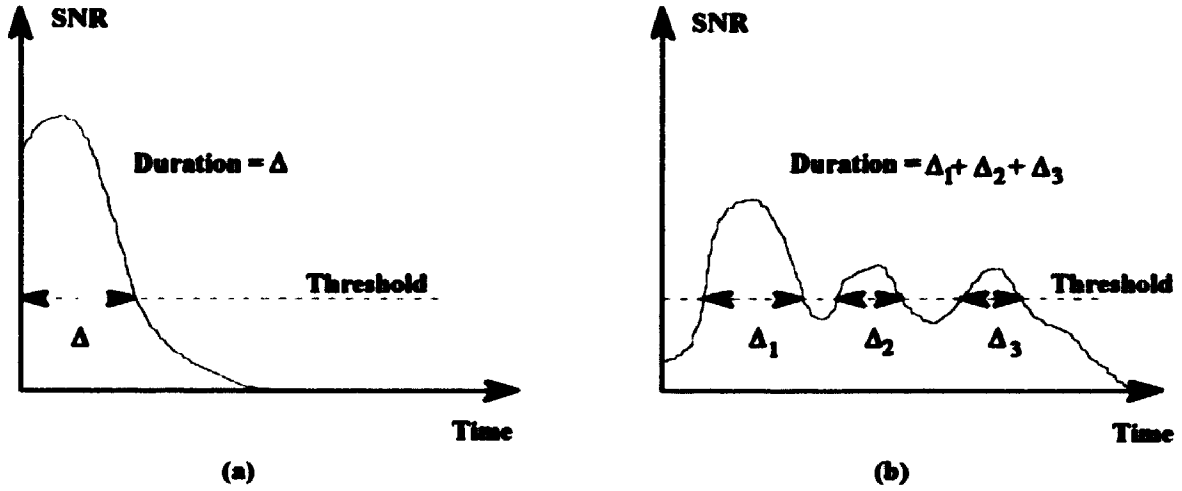


Figure 3.10 Burst Duration Measurements

$$P_N = k\left(T_A + T_{eq}\right)B_N \tag{3.12}$$

where; P_N is the noise power in Watts, k is Boltzmann's constant, T_A is the antenna temperature in Kelvin, and T_{eq} and B_N are the equivalent temperature and bandwidth of the receiver. The equivalent receiver temperature T_{eq} is a function of the receiver noise figure, F and the ambient temperature, T_o and is given by:

$$T_{eq} = T_o(F-1)\,. \tag{3.13}$$

For an operating frequency of 48 MHz, the antenna temperature, T_A, is approximately 10,000 K. Assuming that T_o=290 K, F=4 (6 dB) and given that the interferometer receiver

bandwidth B_N=40 Hz, a noise level of -142 dBm is obtained. It should be noted that the empirical equation given by Schanker [4] for the noise power at a rural location is given by:

$$P_{NR} = -106.8 - 27.7\log_{10} f + 10\log_{10} B \qquad (3.14)$$

where; P_{NR} is the noise power in dBm, f is the operating frequency in MHz and B is the receiver bandwidth in Hz which gives a noise power level of -137 dBm for f = 48 MHz and B=40 Hz. Although the noise levels predicted by the equations 3.12 and 3.14 are similar, suggesting that contributions due to artificial sources are relatively low, it is unlikely that these values are representative of the true noise level at the receiving site Judging from the echo recordings shown in Figures 1.1 and 1.2 it is more likely that the noise power lies between -130 and -125 dBm - a result that is consistent with noise level calculations assuming a business type noise environment. For a business type environment, the noise power given by Schanker [4] is:

$$P_{NB} = -97.2 - 27.7\log_{10} f + 10\log_{10} B \qquad (3.15)$$

and yields a power level of -127 dBm assuming the same parameters as before. Accepting that this level is reasonable, duration measurements were made on all echoes recorded during the month of March 1991.

An analysis of echo recordings for the month of March, 1991 showed the duration of these events to range from a minimum value of 10 ms, to a maximum value of 7.8 seconds with an average duration of 0.46 seconds. Based on the duration statistics shown in Figure 3.11 it is apparent that in the majority of instances the available transmission time is relatively small. Specifically, these statistics show that 75% of bursts possess durations less than 200 ms, 13 % of bursts possess durations ranging between 200 ms and 1 second and the remaining 12% of bursts have durations in excess of 1 second. Because of these transmission time limitations it is clearly advantageous to transmit information at as high a rate as possible to maximize channel throughput.

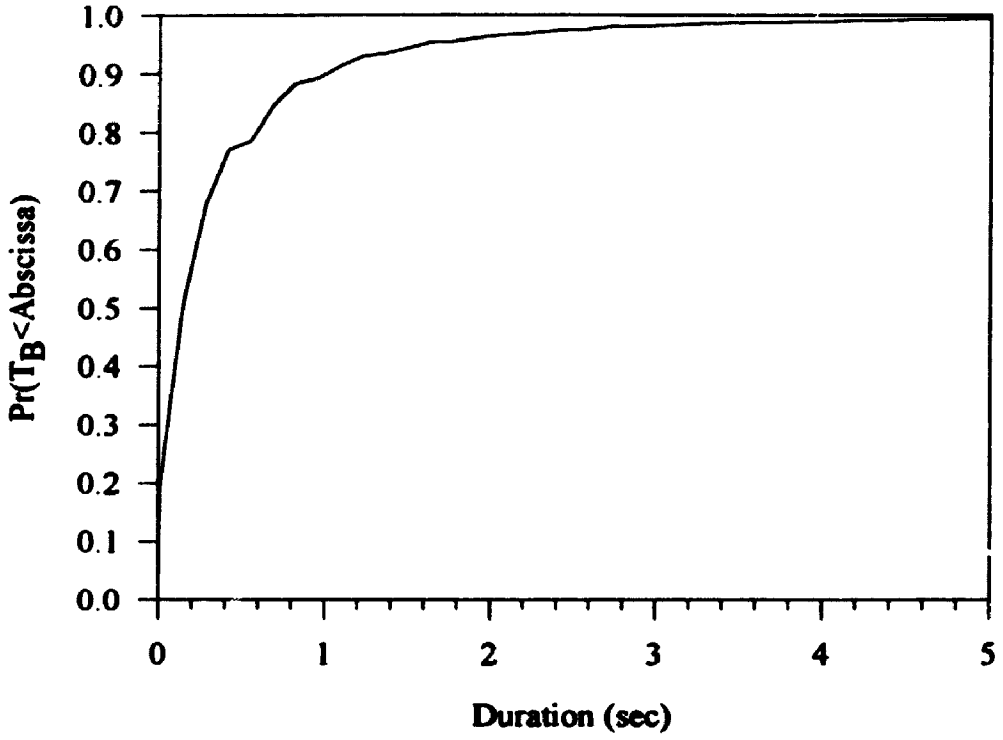


Figure 3.11 Burst Duration Distribution

3.5 Throughput Calculation

Having determined realistic values for the waiting time and burst duration, these results can then be used to assess the performance of a hypothetical meteor burst communication system. A measure that is frequently used in an assessment such as this is the average throughput which is essentially a measure of the data transfer rate that can be supported by the link. The average throughput, after Schanker [4], is defined as:

$$S = \frac{R T_B}{T_W} \tag{3.16}$$

where; S is the average throughput in bits per second (bps), R is the transmission rate (bps), T_B is the average burst duration (sec) and T_W is the average waiting time (sec) between transmissions. Substituting R=4800 bps, T_B=0.5 seconds and T_W=3.5 minutes into equation 3.20 gives an average throughput of 11.4 bps. While this throughput level is

small relative to the Mbps and kbps rates supported by current hardwired and radio systems, the amount of data transferred on a daily basis by a system such as this is roughly 123.1 kbytes which for some applications represents a large amount of information Further increases in throughput can be achieved to a certain degree by decreasing the operating frequency, and increasing the transmitter power thereby taking advantage of weaker trails and reducing the waiting time. Additional gains may also be achieved by increasing the transmission rate, R.

3.6 Summary

Using data from the interferometer system, estimates of the number and arrival rates of meteors for the London-Ottawa link were obtained. The results of this investigation showed that several hundred usable trails are produced on a daily basis and that the arrival rate of suitably ionized trails varies throughout the day with maximum and minimum rates occurring at 06:00 and 18:00 hours (local time) respectively. For this particular system the average arrival rate was found to be 17 meteors per hour and the average interarrival or waiting time was found to be approximately 3.5 minutes Burst duration calculations derived from actual echo recordings found the average burst to be approximately 0.5 seconds in length.

Measurements in which the direction of meteor echoes were recorded were used to identify spatial regions in the vicinity of the transmitter and receiver in which meteor activity was most intense. These locations or hotspots were observed to occur on both sides of the great circle path between transmitter and receiver and were noted to vary in intensity throughout the day. The usefulness of this information is that by knowing the hot spot locations as well their diurnal variations the potential exists to develop a steerable or perhaps a multi antenna system to selectively illuminate those areas of the sky having the greatest meteor activity. Direction measurements were also used to assess flight-time

variations arising from the vast range of trail locations observed. The statistics derived from these measurements showed that in 90% of trails examined, the estimated flight-times varied from 1.8 to 2.5 ms with an average flight-time of 2.1 ms and a standard deviation of 0.5 ms. Although not immediately obvious, the magnitude of these flight-time variations may have an impact on system synchronization issues.

A throughput analysis of a hypothetical meteor burst communication system was carried out assuming the burst duration and waiting times derived from interferometer data. Assuming a transmission rate of 4800 bps, the average system throughput was found to be 11.4 bps. Even though the system throughput is low the amount of information that can be transferred on a daily basis is 123.1 kbytes which for some applications, such as remote monitoring or vehicle tracking, represents a considerable amount of information. Increases in system throughput are possible and may be achieved by increases in transmitter power, antenna gains and signaling rate. The maximum transmission rate that can be used however, may be limited by channel impairments such as those arising from multipath interference.

Chapter 4

An Impulse Response Measurement System

4.1 Introduction

An important factor limiting the overall performance and consequently the usefulness of a meteor burst communication system is the maximum data rate that can be supported by the channel. This maximum rate, however, is constrained by bandwidth limitations arising from channel impairments such as multipath interference. In the majority of cases, these limitations are usually deduced from knowledge of the channel impulse response which is ultimately obtained using empirical techniques. To date, little research with the notable exception of that conducted by Akram [22], Weitzen [23,24] and Eriksson [25] has been undertaken to measure systematically the channel impulse response. The necessity of an endeavor such as this is justified by the need to analyze and develop communication strategies that fully exploit the limited resources of the meteor burst channel. In pursuit of this knowledge, a novel system incorporating chirp radar techniques was developed to obtain measurements of the channel impulse response, path-loss, flight-times and path-lengths. Details of the instrument as well as the designs of the transmitter and receiver constitute the subject matter of this chapter.

4.2 Principles of Operation

The instrument devised for the impulse response measurements operates on the same fundamental principals employed in modern F.M. pulse compression radars. Systems such as these rely on the transmission and subsequent demodulation of a sinusoid whose instantaneous frequency is a linearly increasing or decreasing function of time. These

signals are described collectively as "chirp" waveforms or individually as up-chirp or down-chirp waveforms, depending on the time-frequency relationship of the modulating signal. An up-chirp waveform representative of that employed in the measurement system is illustrated in Figure 4.1. Signal characteristics such as chirp rate and duration as well as starting and stopping frequencies can be illustrated in a more intuitive manner through the use of a frequency-time diagram as shown in Figure 4.2.

The process of recovering range information from the chirp transmission is illustrated by means of the frequency-time diagram shown in Figure 4.3. For the range measuring system under consideration, the up-chirp signal emitted by the transmitter arrives at the receiver after having been delayed by the propagation time T_p. At some time prior to the arrival of this signal, T_d, an up-chirp signal identical in all respects but its starting frequency is generated at the receiver. Under these conditions, the frequency difference between these two signals, Δf_1, is constant as illustrated in Figure 4.3. In the event that the propagation time, T_p, is increased by an amount δ due to a possible change in path length, the difference frequency is again constant but has increased to Δf_2. Consequently, for a system of this type, measurements of propagation time or equivalently range, can be deduced from the frequency difference between the locally generated and received chirp signals.

An adaptation of this scheme specifically tailored to the measurement of multipath effects is shown in Figure 4.4. The signal emitted by the transmitter is an up-chirp waveform and is described by the equation:

$$s(t) = p(t)\cos\left(2\pi f_0 t + \pi\mu t^2\right) \qquad \textbf{(4.1)}$$

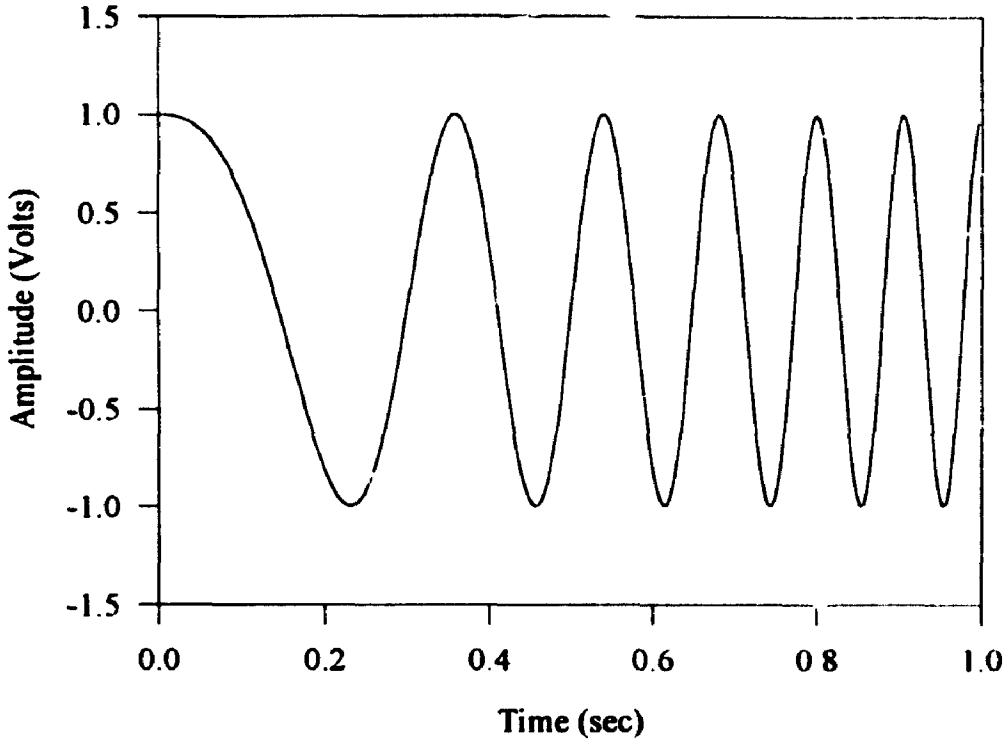


Figure 4.1 An Up-Chirp Signal Viewed in the Time Domain

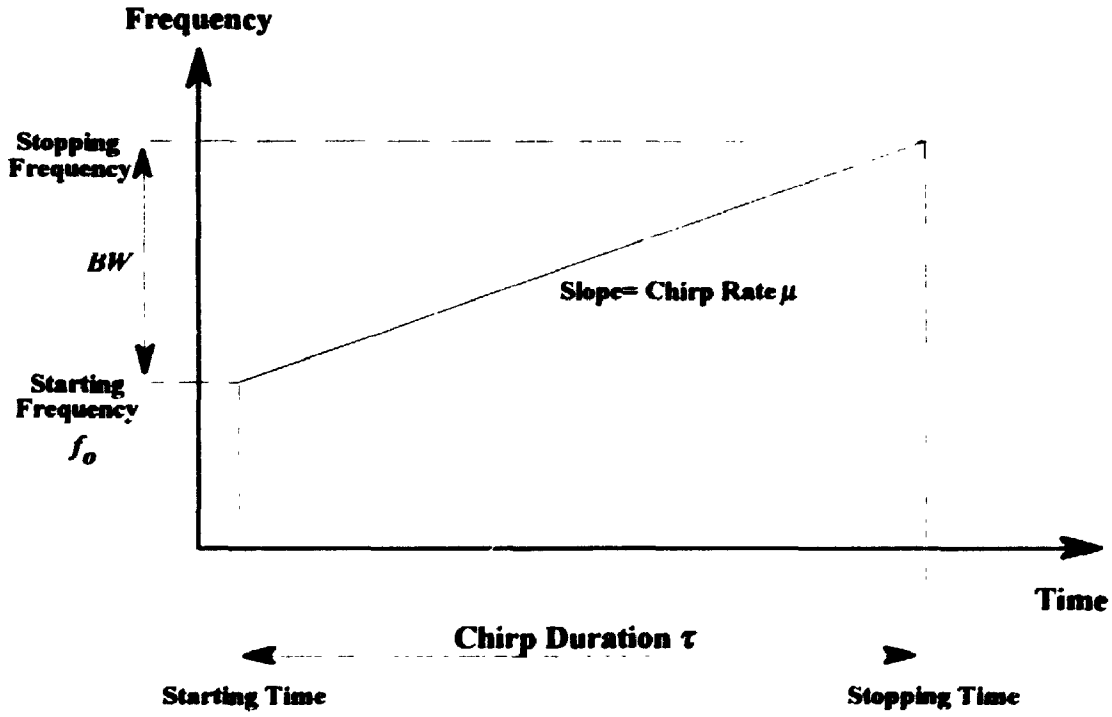


Figure 4.2 Frequency-Time Diagram for an Up-Chirp Signal

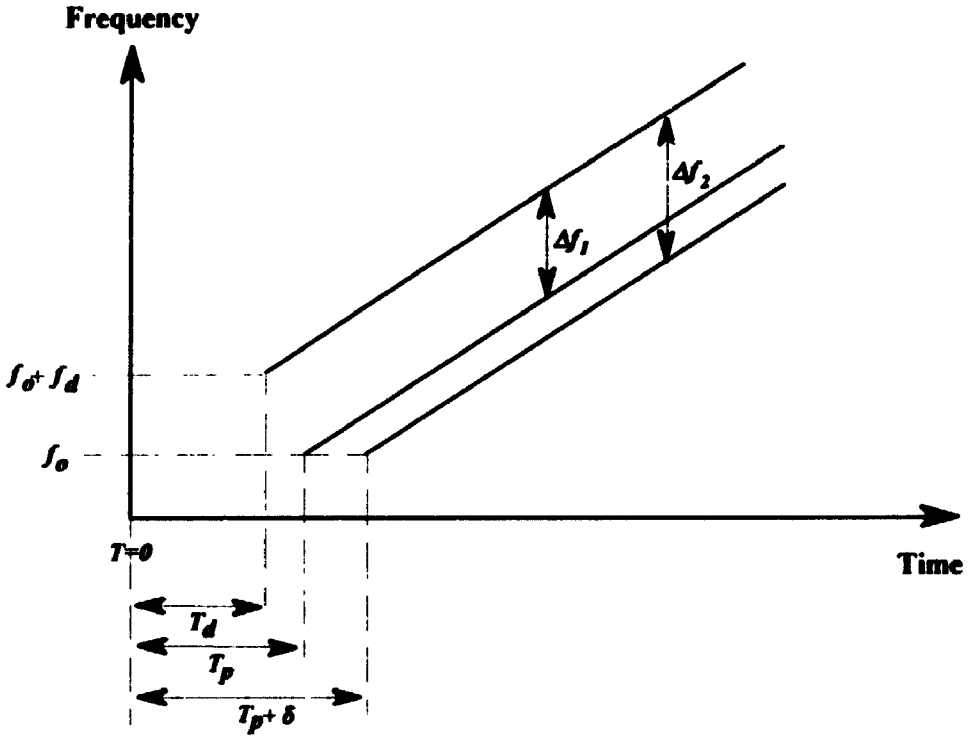


Figure 4.3 Frequency-Time Diagram for the Measurement Process

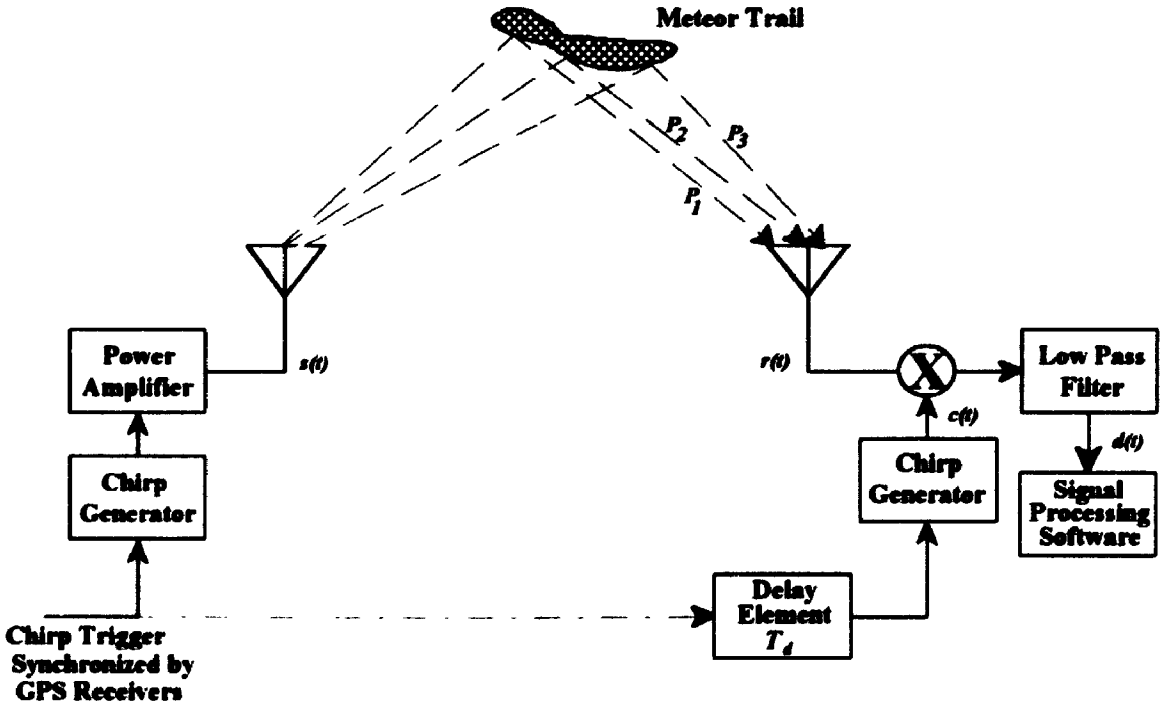


Figure 4.4 Simplified Block Diagram of the Measurement System

where; f_o is the starting frequency of the chirp (Hz), μ is the chirp rate (Hz/sec.) and $p(t)$ is a unipolar rectangular switching waveform which controls the duration and repetition rate of the transmitted signal. The chirp signal generated at the receiver is similar to that generated at the transmitter and is described by the equation:

$$c(t) = p(t - T_d)\cos\left(2\pi(f_0 + f_d - \mu T_d)t + \pi\mu t^2 + \theta\right) \qquad (4.2)$$

where; f_d is the frequency offset (Hz), T_d (sec.) is the delay between the starting times of the chirp waveforms generated at the transmitter and receiver, θ is the phase angle (rad.) and all other symbols have their usual meanings. When the signal emitted by the transmitter arrives at the receiver after having taken one or potentially several paths, it is mixed with the locally generated chirp, $c(t)$, and filtered producing the receiver output signal, $d(t)$, given by:

$$d(t) = \sum_{k=1}^{N} \beta_k p\left(t - T_{pk}\right)\cos\left(2\pi\left(f_d + \mu\left(T_{pk} - T_d\right)\right)t + \theta_1\right) \qquad (4.3)$$

where; N is the number of paths between the transmitter and receiver, β and T_{pk} represent the attenuation and flight-times for the k^{th} path and all other parameters have their usual meanings. All measurements of interest (time-of-flight etc.) are derived from the spectrum of the output signal, $d(t)$. A sketch of an idealistic spectrum is shown in Figure 4.5 for the case in which 3 paths exist between transmitter and receiver. In this case, the flight-times for each path can be estimated from knowledge of the frequency components, f_k, which from equation 4.3 are given by:

$$f_k = f_d + \mu\left(T_{pk} - T_d\right). \qquad (4.4)$$

Since the parameters f_k, μ, f_d, and T_d are known or measurable, the preceding equation can be solved in terms of the flight-times giving:

$$T_{pk} = T_d + \frac{f_k - f_d}{\mu}. \qquad (4.5)$$

It should be noted that the delay time, T_d, must remain constant throughout the measurements and this is achieved in practice by synchronizing the transmitter and

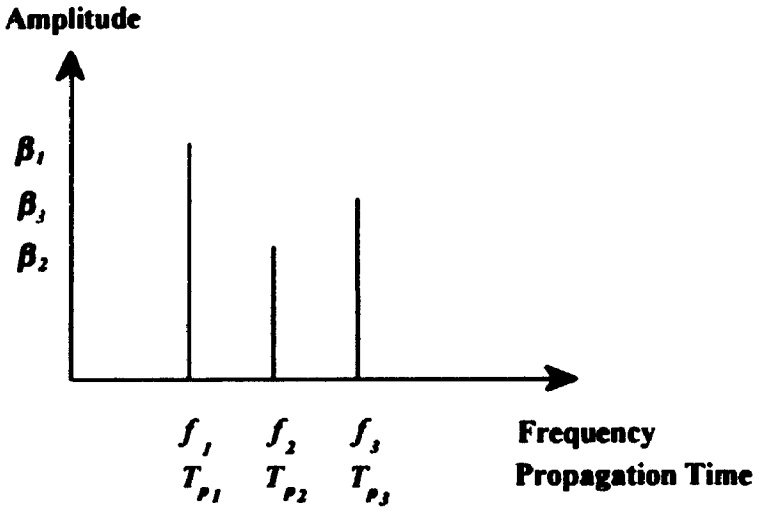


Figure 4.5 Ideal Spectrum Representing Impulse Response

receiver timing through the use of Global Positioning Satellite (GPS) receivers located at both ends of the link. Having obtained the flight-times, the corresponding path lengths, P_k, can then be computed using:

$$P_k = cT_{pk} \tag{4.6}$$

where; c is the velocity of light in a vacuum. In addition to providing time-of-flight information, the amplitude spectrum of *d(t)* represents the impulse response of the channel and can be used in conjunction with the Fourier transform to estimate the channel transfer function. Furthermore, if the receiver is properly calibrated to measure the power delivered by each of the frequency components, f_k, the path losses between transmitter and receiver can also be measured.

4.3 Chirp Waveform Characteristics

A complete characterization of the chirp waveform requires the specification of various signal parameters. These parameters include duration, bandwidth and chirp rate in addition to the specification of the starting and stopping frequencies. The discussion presented in this section seeks to determine reasonable values for these parameters in as rigorous a manner as possible. Once the parameter values have been established they will

then be used in conjunction with theoretical results to estimate the spectral characteristics of the chirp signal.

In order to attain a detection probability on the order of 90%, it is necessary to ensure that the received signal energy is sufficient to yield a signal-to-noise ratio of between 20 and 30 dB. Assuming that the software filters contained in the signal processing module approximate the performance of a matched filter, the signal-to-noise ratio (Cook and Bernfeld [26]) can be evaluated using:

$$SNR = \frac{2E_g}{N_o} \quad , \tag{4.7}$$

where; E_g is the received signal energy and N_o is the noise power spectral density In this case the received signal energy, E_g, can be estimated (assuming the underdense condition) using:

$$E_g = \frac{P_T G_T G_R \lambda^3 q^2 r_e^2}{16\pi^2 R_T R_R (R_T + R_R)} \exp\left(-\frac{8\pi^2 r_o^2}{\lambda^2 \sec^2 \phi}\right) \tau \tag{4.8}$$

where; τ is the chirp duration and all other parameters have their usual meanings It should be noted that the first two factors in equation 4.8 represent the peak received signal power as given by equation 2.1. Assuming the link parameters of Figure 2.2, a transmitter power, P_T, of 1.0 kW, antenna gains G_T and G_R of 5 dB, an electron line density of 1.0×10^{14} e/m, a signal-to-noise ratio of 20 dB, and a noise power spectral density, N_o, of 4×10^{-17} Watts/Hz (measured), equations 4.8 and 4.7 can be combined and solved in terms of the pulse width, τ, giving:

$$\tau = 79.6 \text{ ms} \quad . \tag{4.9}$$

To simplify implementation without incurring a serious degradation in SNR, a chirp duration of 50.0 ms was selected.

The bandwidth of the chirp signal is constrained by two conflicting objectives; the need to achieve good resolution and the desire to minimize interference to nearby stations occupying the same frequency band. Based on the results of experimental work by Eriksson [25] a resolution capability of approximately 100 meters or less is necessary to investigate the multipath characteristics of the channel. Defining the resolution as the smallest flight-time differential, Δt, that can be measured by the instrument results in the equation:

$$\Delta t = \frac{f_2 - f_1}{\mu} \quad , \tag{4.10}$$

which can be arrived at through the use of equation 4.4 assuming $T_{p2} > T_{p1}$ (for simplicity). When the frequency difference:

$$f_2 - f_1 = \Delta f \tag{4.11}$$

is greater than the detection filter bandwidth given by:

$$\Delta f = \frac{1}{\tau}, \tag{4.12}$$

both sinusoids can be resolved. By expressing the chirp rate, μ, as:

$$\mu = \frac{BW}{\tau} \tag{4.13}$$

and substituting equations 4.11 through 4.13 into equation 4.10, gives the overall system resolution as:

$$\Delta t = \frac{1}{BW} \tag{4.14}$$

where; BW represents the bandwidth of the chirp signal. The relationship between the spatial resolution, Δd, and the flight-time resolution, Δt is:

$$\Delta d = c\Delta t \quad . \tag{4.15}$$

Therefore, to achieve a spatial resolution on the order of Δd=100 m a bandwidth of approximately 3.0 MHz is required. For convenience, a chirp bandwidth of 2.5 MHz was chosen yielding path-length and flight-time resolutions of 120 m and 400 ns respectively.

Given the values for the chirp duration and bandwidth, the corresponding chirp rate, μ, is 5×10^7 (Hz/sec). Based on the arguments presented in Chapter 2 regarding operating frequency selection, starting and stopping frequencies of 38.75 MHz, 41.25 MHz were chosen. The frequency offset (f_d) between the starting and stopping frequencies of the transmitter and receiver chirps was chosen arbitrarily to be 12.0 kHz. A summary of the chirp waveform parameters is presented in Table 4.1.

Table 4.1 Summary of Chirp Waveform Parameters

Parameter	Transmitter	Receiver (at I.F Stage)
Starting Frequency	38.75 MHz	1.012 MHz
Stopping Frequency	41.25 MHz	3.512 MHz
Bandwidth (BW)	2.5 MHz	2.5 MHz
Duration (τ)	50.0 ms	50.0 ms
Chirp Rate (μ)	5.0×10^7 Hz/s	5.0×10^7 Hz/s

Throughout the calculations in the preceding paragraphs it was assumed that the signal bandwidth was equal to the difference between the starting and stopping frequencies of the chirp. The goal of the following discussion is to demonstrate the validity of this assumption by calculating the amplitude spectrum of the chirp signal having those characteristics summarized in Table 4.1. From Cook and Bernfeld [26], the amplitude spectrum of a chirp signal is given by the equation:

$$|S(\omega)| = \frac{1}{2}\sqrt{\frac{\pi}{\mu_1}}\left\{\left[C(X_1)+C(X_2)\right]^2+\left[S(X_1)+S(X_2)\right]^2\right\}^{1/2} \qquad \textbf{(4.16)}$$

where the functions $C(X)$ and $S(X)$ are the Fresnel integrals given by:

$$C(X) = \int_0^X \cos\left(\frac{\pi y^2}{2}\right) dy \tag{4.17}$$

and

$$S(X) = \int_0^X \sin\left(\frac{\pi y^2}{2}\right) dy. \tag{4.18}$$

By making use of several substitutions, also suggested by Cook and Bernfeld [26], the variables X_1 and X_2 can be simplified and expressed in terms of the pulse compression parameter $\tau\Delta f$ and a normalized frequency parameter, n. Thus the variables X_1 and X_2 become:

$$X_1 = \sqrt{\tau\Delta f}\left(\frac{1+n}{\sqrt{2}}\right) \tag{4.19}$$

and

$$X_2 = \sqrt{\tau\Delta f}\left(\frac{1-n}{\sqrt{2}}\right). \tag{4.20}$$

Using these substitutions, equation 4.16 can be plotted in terms of the frequency parameter n, with n=0 corresponding to the center of the spectrum. It should be noted that the parameter μ_1 appearing in equation 4.16 represents the chirp rate in units of rad/sec^2 and that the parameter Δf appearing in equations 4.19 and 4.20 is equal to the difference between the starting and stopping frequencies of the chirp. In cases where the pulse compression parameter $\tau\Delta f$ is relatively small ($\tau\Delta f$ =13) no numerical problems are encountered and the spectrum shown in Figure 4.6 is readily obtained. As the pulse compression parameter is increased to much larger values representative of those in Table 4.1 for which $\tau\Delta f$ 125×10^3, arguments of the Fresnel integrals become very large and numerical problems are encountered. Fortunately, for large arguments both the Fresnel sine and cosine integrals approach the limiting value of 0.5. By taking advantage of this

approximation in the evaluation of these integrals, the spectrum shown in Figure 4.7 is obtained. It should be noted that for both values of the pulse compression parameter, the signal spectrum is generally rectangular and has a bandwidth approximately equal to the frequency difference between the starting and stopping frequencies of the chirp.

4.4 System Operation

Because of the intermittent nature of the meteor burst channel, the impulse response measurement system was designed to collect data on both a continuous and triggered basis. In the continuous mode of operation, the receiver output is recorded continuously on video tape for a period of several hours. The disadvantages of this approach are that large portions of the tape will contain no useful information during the absence of suitably ionized trails and furthermore the entire tape must be processed to determine if any trails were present. The attractive feature of this mode is that a permanent record of the signal is always available and can be scrutinized for weak signals using digital processing techniques. In the triggered mode of operation the system records data only when trails having suitable ionization are present. This is accomplished by transmitting a 10 ms carrier, or "probe" signal at 39.75 MHz, 20 ms in advance of the chirp transmission. When the "probe" signal is detected by the receiver, its output is digitized and stored in a convenient form. A diagram showing the transmission cycle is given in Figure 4.8. Regardless of the recording mode, repetitive transmission of the chirp signal at intervals of 100 ms permit variations in the channel characteristics to be recorded over the duration of a burst. For typical durations of 1.0 second, approximately 10 chirp measurements can be obtained.

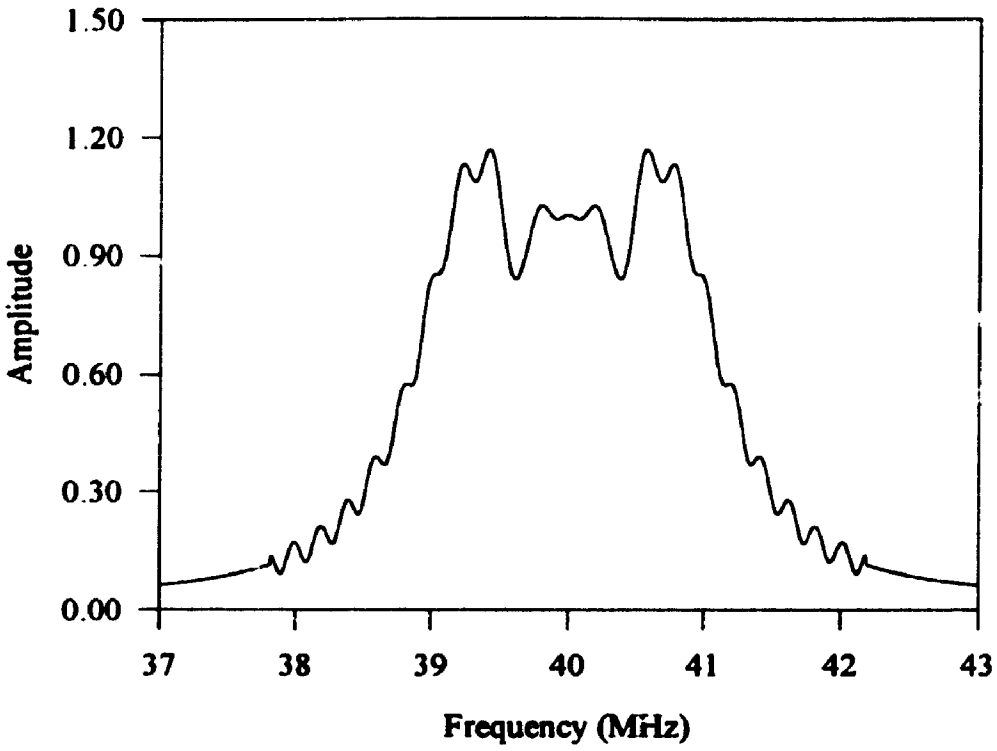


Figure 4.6 Chirp Amplitude Spectrum for $\tau\Delta f = 13$

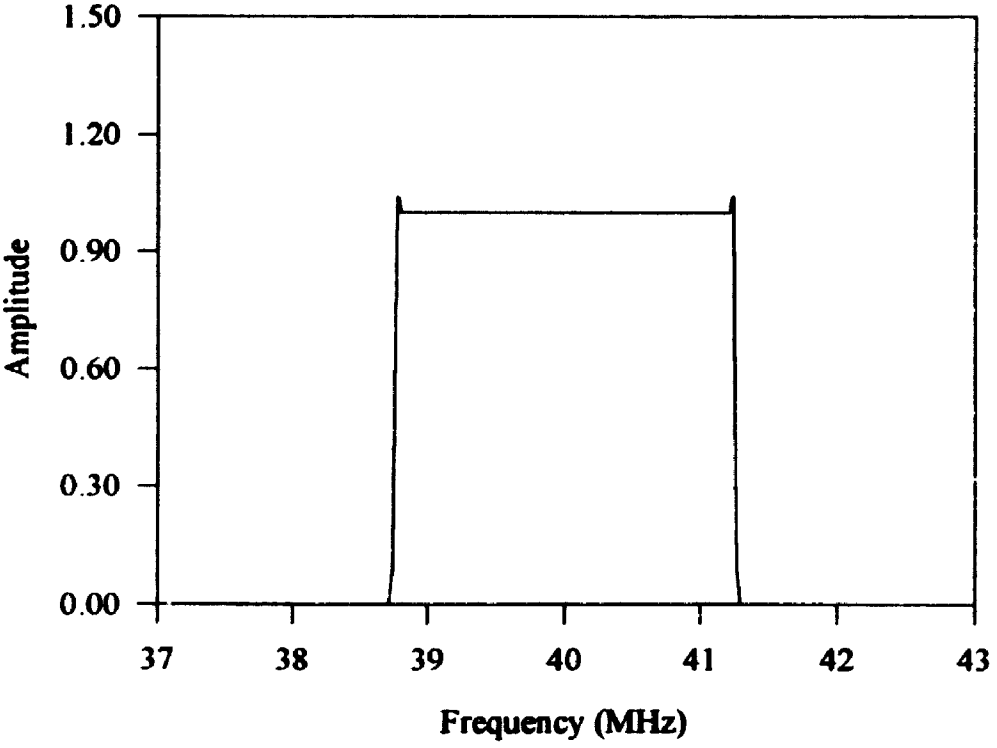


Figure 4.7 Chirp Amplitude Spectrum for $\tau\Delta f = 125\times10^3$

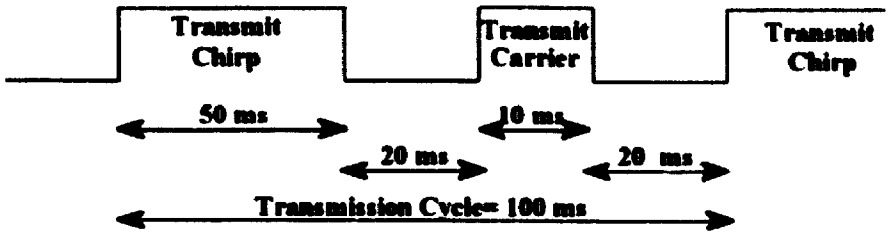


Figure 4.8 Transmission Cycle

4.5 The Transmitter

In order to simplify presentation, the complete transmission system has been divided into five modules. These modules are: the chirp generator, composite signal generator, transmit sequencer, up-converter, and the final RF power amplifier. Each of the paragraphs that follow outline the function and operation of these units. Where appropriate, signal levels, attenuation and filter cutoff frequencies have been included to assist in the discussion. Detailed schematic diagrams for each of the transmitter modules can be found in Appendix A.

Due to the complexities of chirp generation, it was necessary to synthesize the transmitter chirp signal in a two-step process. During this process, a replica of the desired signal is first generated at baseband by the chirp generator unit and then translated to the desired frequency range by the up-converter unit. A block diagram of the chirp generator module is shown in Figure 4.9. The chirp signal is initially generated by software running on a personal computer (PC), sampled at a rate of 8.0 MHz and then uploaded to the chirp generator memory (512 k $\times$ 8 bytes) through the computers' parallel printer port. Once the samples have been loaded into RAM, they are clocked out at the sampling rate to the D/A converter where the analog signal is reconstructed. In the final stage of this module the reconstructed analog signal is applied to a low pass filter which passes the desired chirp spectrum ranging from 1.0 to 3.5 MHz and attenuates unwanted high frequency harmonics above 4.0 MHz. In addition to the parallel port connection required for data transfer,

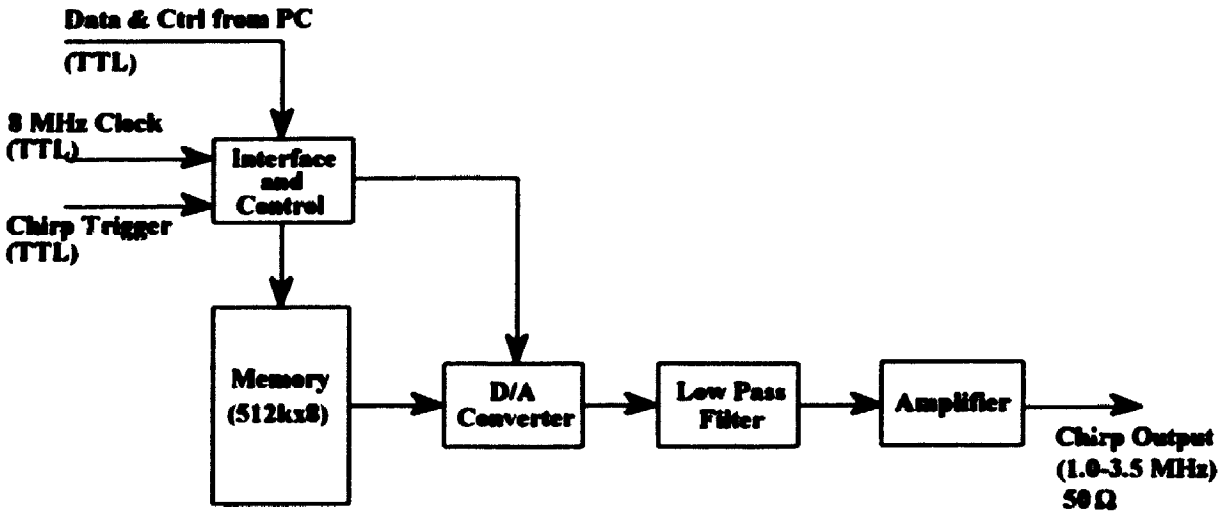


Figure 4.9 Chirp Generator Module

connections are also provided for an 8.0 MHz sample clock (TTL) and a control line for the trigger signal (TTL). Chirp transmission is initiated on the rising edge of the signal applied to this control line. The source code necessary for generating and uploading the chirp is included with the schematic diagrams in Appendix A. It should be noted that the two programs included in the Appendix have identical functions and generate baseband chirps with starting and stopping frequencies differing by 12.0 kHz. Program chirp1.c generates the chirp signal for the transmitter and chirp2.c generates the signal used at the receiver.

Prior to the up-conversion process, the baseband chirp and carrier signals must be combined to produce a composite signal which, when applied to the up-converter, will result in an output signal having the desired spectrum. The block diagram of this module is shown in Figure 4.10. Modulation of the carrier or "probe" signal is accomplished by ANDing the 2.0 MHz TTL oscillator output with the logic level on the carrier transmit control line. This control signal is produced by the transmit sequencer module whose function is to govern the timing and duration of the carrier transmission according to the transmission cycle illustrated in Figure 4.8. After modulation, the carrier signal is filtered

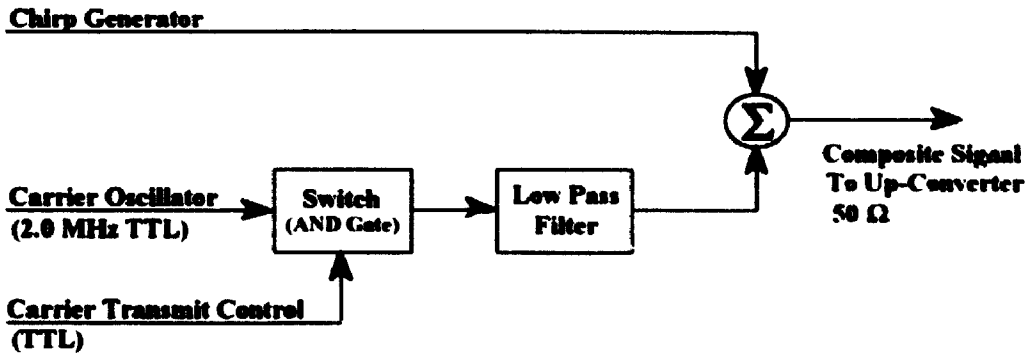


Figure 4.10 Composite Signal Generator Module

to attenuate unwanted high frequency harmonics before being combined with the baseband signal from the chirp generator output.

As previously indicated, the function of the transmit sequencer is to generate the keying waveform needed to control carrier transmission. The transmit sequence is initiated by the rising edge of a 10 Hz timing signal applied to the trigger input. This 10 Hz signal is derived from the GPS receiver (to be discussed later) and is also used to initiate chirp transmission. Once triggered, the combined effect of the two one-shot devices is to generate a 10 ms control signal that is delayed 70 ms from the onset of the chirp transmission. A block diagram of this module is shown in Figure 4.11.

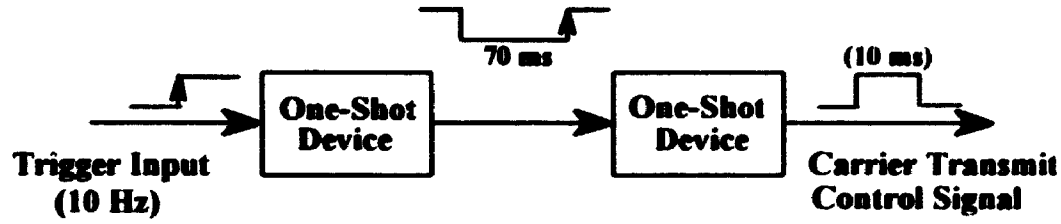


Figure 4.11 Transmit Sequencer Module

Since the signal produced by the composite signal generator module is a baseband signal, its spectrum must be translated in order to obtain the desired chirp and carrier signals. This function is accomplished by the up-converter module shown in Figure 4.12 which translates the composite baseband spectrum and produces a chirp signal having the

desired starting and stopping frequencies of 38.75 and 41.25 MHz, as well as a carrier signal having a frequency of 39.75 MHz. After each translation, band pass filters are used to attenuate unwanted sidebands and spurious high frequency components which may be present in the system. Although not explicitly shown in the block diagram, amplifiers are used throughout the up-converter module to develop the desired output power level. Both carrier oscillator signals, required in the up-conversion process, are derived from two Rockland model 5600 frequency synthesizers.

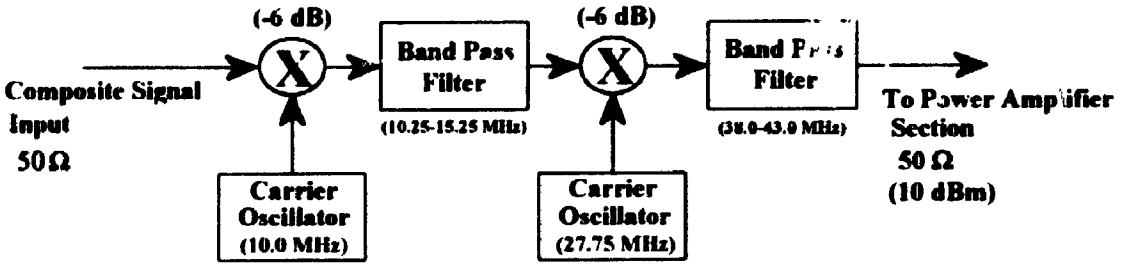


Figure 4.12 Up-Converter Module

Following the up-conversion process, the signal must be further amplified to achieve a power level suitable for transmission. The power amplifier section shown in Figure 4.13 amplifies the signal from a level of 10 dBm to approximately 60 dBm (1.0 kW) As in the up-converter section, band pass filters are required to attenuate spurious frequency components occurring outside the desired passband from 38.0 to 43.0 MHz. After amplification, the transmitter signal is then fed to the antenna system.

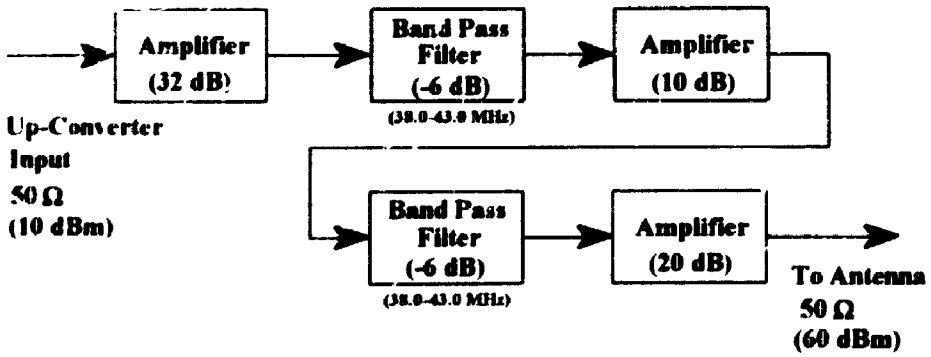


Figure 4.13 Power Amplifier Module

4.6 Transmitter Synchronization

In order to ensure that transmissions occur in the proper time sequence, all control signals are synchronized via the timing waveforms generated by the GPS receiver The Magellan Brain™ OEM receiver provides 1 Hz and 1 kHz signals from which all necessary timing/synchronization signals can be obtained. In order to begin a transmission cycle once every 100 ms, the 1.0 kHz squarewave signal from the GPS receiver is divided by a value of 100 in a 3 step (÷10, ÷5, ÷2) division process. To ensure that the 10 Hz signal remains synchronized with the GPS signals, the counter and flip-flop are reset every second by the 1Hz clock. A block diagram showing the details of the transmitter synchronization module is shown in Figure 4.14.

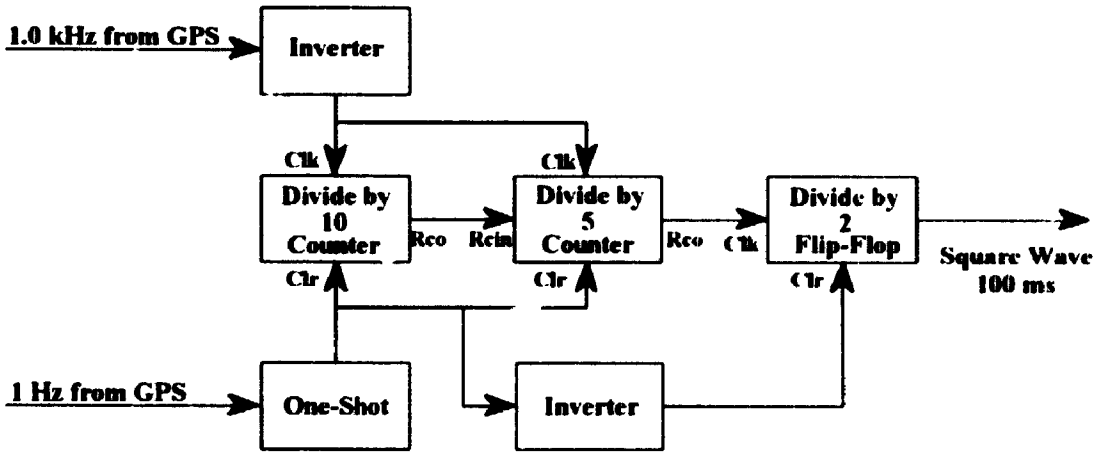


Figure 4.14 Transmitter Synchronization Circuit (after Ohno [27])

4.7 The Receiver

As in the transmitter discussion, the receiver has been divided into several modules in order to simplify presentation The three modules comprising the receiver are, the carrier detector/down-converter section, the I.F. section and the data recording/storage unit. The function and operation of each of these modules is outlined in the following paragraphs As in the transmitter discussion, signal levels, attenuation and filter cutoff

frequencies have been included wherever appropriate. Detailed schematic diagrams for the receiver modules can be found in Appendix A.

The two functions of the carrier detect/down-converter module are to detect the presence of the carrier or "probe" signal at 39.75 MHz and to translate the spectrum of the received chirp signal to baseband for demodulation and recording by the I.F. and data storage units. A block diagram showing the details of the carrier detector/down-converter module is shown in Figure 4.15.

Whenever the carrier signal is received, it is amplified and demodulated by the Eddystone 990R VHF receiver. The carrier detect circuitry is connected to the A.M. detector output of this receiver and consists of a 10 Hz active low pass filter followed by an additional stage of amplification. A signal is "detected" when the amplitude of this waveform exceeds a preset threshold level. When such an event occurs a one-shot device is triggered generating a pulse, which in turn, triggers the A/D converter and its associated data recording system. In order to prevent the receiver from drifting off frequency, use was made of the rear panel connection into which a stable local oscillator signal was injected. For reception at 39.75 MHz, an external local oscillator having a frequency of 50.444 MHz and a power level of approximately -11 dBm is required. Amplifiers within the receiver amplify this signal to obtain the desired level for mixing.

When the chirp signal is received, it is translated to baseband in a dual-conversion process Before and after each of the conversions, the signal is filtered and amplified so as to reduce the effects of unwanted signals resulting from the mixing process. The local oscillator frequencies required to achieve this overall translation are 27.75 MHz and 10.0 MHz, both of which are obtained from stable sources. The waveform produced at the I.F. output port is the desired baseband chirp signal whose starting and stopping frequencies

are 1.0 and 3.5 MHz respectively. It should be noted that interference between the carrier and chirp signals doesn't occur since the two signals are never broadcast simultaneously

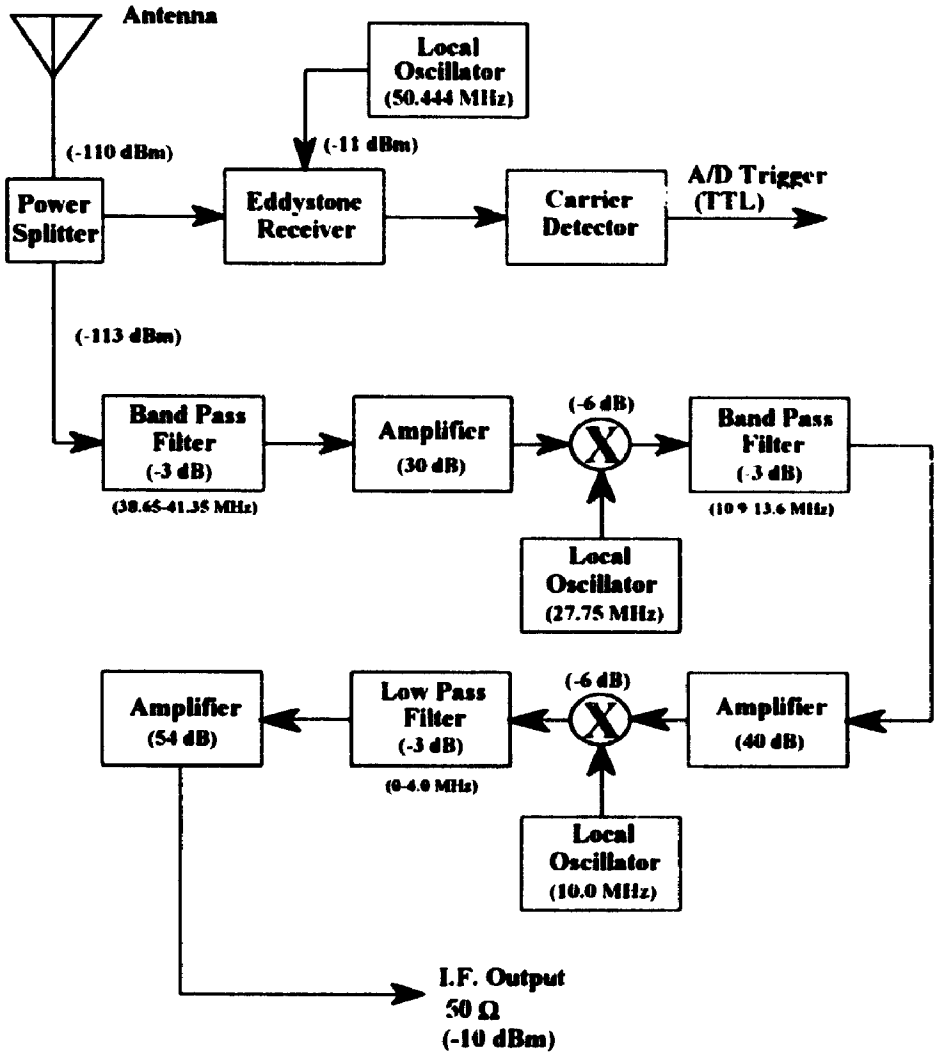


Figure 4.15 Carrier Detector/Down-Converter Module

When the I.F. signal is mixed with the locally generated chirp and filtered as shown in Figure 4.16, beats are produced as discussed in Section 4.2 At this point in the receiver, the signal path is split and applied to both the A/D converter and quadrature detection circuits. Signals applied to the A/D converter are recorded only when triggered by the detection of the "probe" signal while those applied to the quadrature detector are

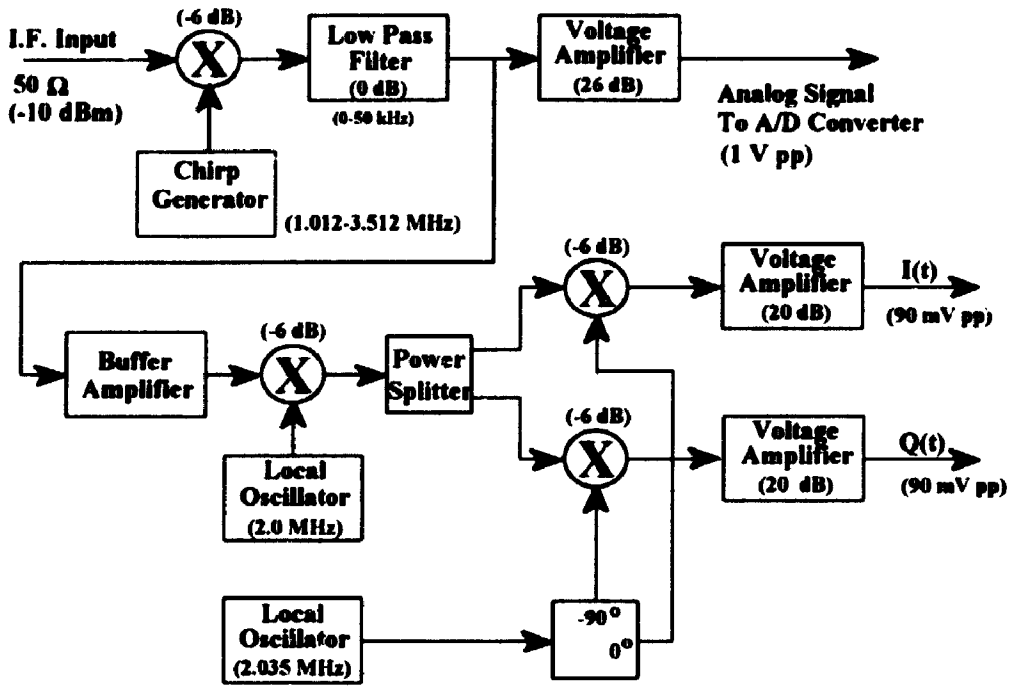


Figure 4.16 I.F. Section

recorded continuously. Quadrature detection is necessary in this system to permit the spectrum of the demodulated signal, which has a bandwidth of approximately 40 kHz, to be recorded on the two 20 kHz channels of the digital audio processing unit. It should be noted that the main A/D channel as well as both the I and Q channels have been filtered to reduce the effects of aliasing. In the case of the main A/D channel, extensive filtering is performed with an active filter section having a corner frequency of 50 kHz and roll-off rate of 80 dB/decade. Fortunately, the filter requirements for the I/Q channels are less stringent due to the filtering employed in the front end of the SONY PCM-F1 unit. For these filters, a roll-off rate of 40 dB/decade and corner frequencies of 30 kHz were selected

A block diagram of the data recording and storage module is shown in Figure 4.17 The analog signal from the I.F. section is applied to the A/D converter, sampled at a rate of 200 kHz and stored in binary format on disk. In-phase and quadrature signals applied to the SONY digital audio processor are converted into a composite video signal

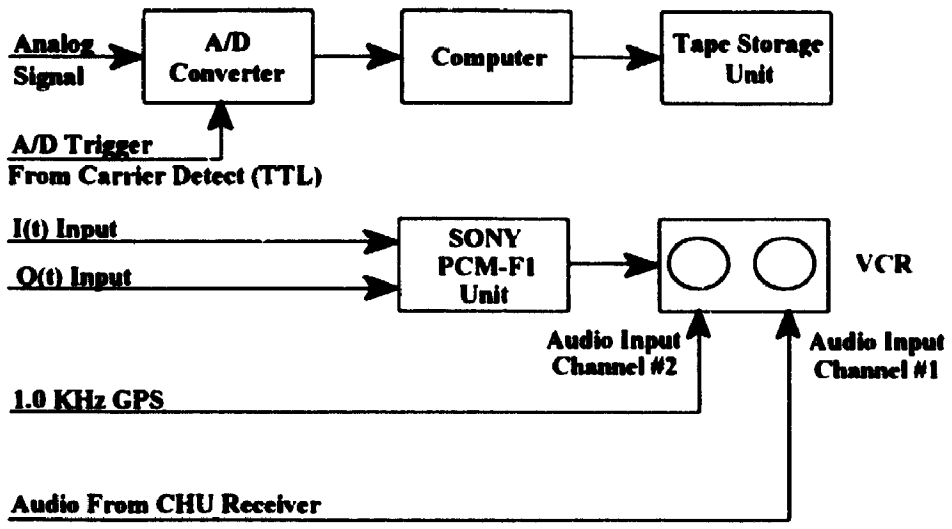


Figure 4.17 Data Recording and Storage Module

that is recorded by the VCR. Provisions were made in this system to permit timing information from the time station CHU as well as the 1.0 kHz time signal from the GPS receiver to be recorded on the audio tracks of the video tape for indexing purposes.

4.8 Receiver Synchronization

To properly demodulate transmissions, the starting time of the receiver chirp must be delayed by the expected propagation time, T_d, relative to the onset of chirp transmission. A simple method for generating such a delay is to apply a clock signal of known frequency to a series of counters and wait for the "count" to reach a certain value The success of an approach such as this requires that counter initialization and other associated control functions be performed in synchronism with similar activities at the transmitter. Since the transmission cycle is derived from GPS timing signals, system-wide synchronization can be achieved through the use of a second GPS module located at the receiver.

A simplified block diagram of the delay element is shown in Figure 4.18 Synchronism is achieved by resetting the counter once every 100 ms in an identical manner

to that outlined in Section 4.6 for the transmitter. The 100 kHz clock signal controlling the count rate is derived from the 1.0 kHz GPS timing signal using a PLL frequency multiplication circuit. In the first stage of multiplication, the 1.0 kHz signal is multiplied by a factor of 10 producing a 10 kHz signal that serves as the frequency reference for the following stage. In the final multiplication stage, the 10 kHz signal from the previous stage is multiplied by a factor of 10, producing the desired 100 kHz clock signal. Using this approach, the delay element can generate delays of up to several milliseconds with a minimum time resolution of 10 μs. Circuit diagrams for the delay unit can also be found in Appendix A.

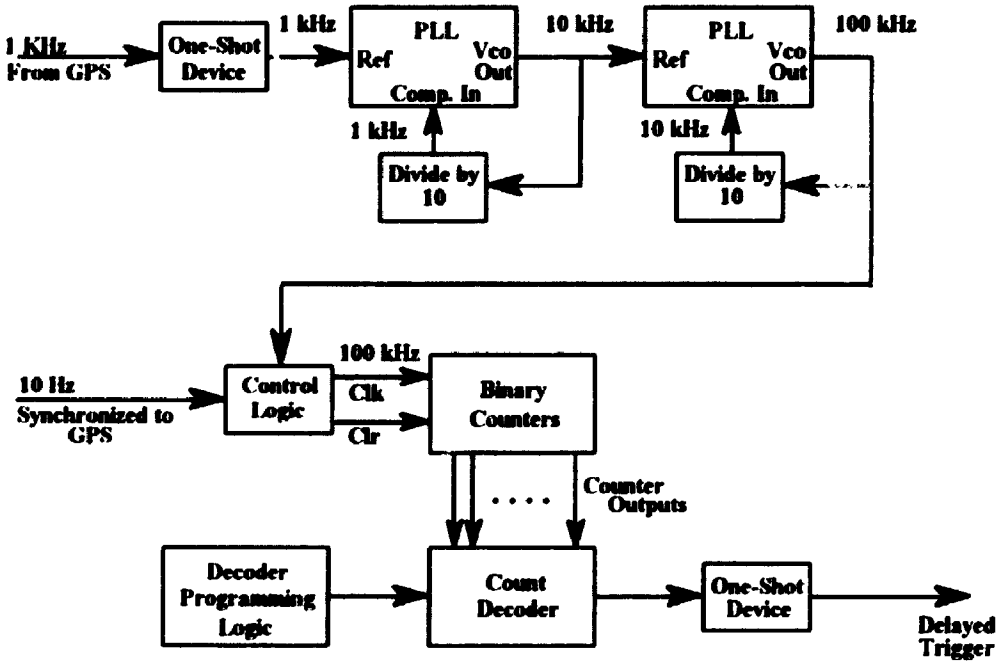


Figure 4.18 Synchronized Delay Element (after Ohno[27])

4.9 Summary

A complex system for measuring the impulse response of the meteor burst channel was developed. This system relies on chirp radar techniques to gain insight into channel impairments such as multipath interference which affect communication system performance. In addition to impulse response measurements, this system is also capable of measuring flight-time, path-length and propagation loss. The material presented in this

chapter discusses all aspects of the instrument which include system fundamentals and operating modes as well as the hardware required for transmission, reception, synchronization and recording. Throughout this discussion, block diagrams were employed to simplify presentation. Detailed schematic diagrams for all hardware components can be found in Appendix A. Although technically a component of the receiver, a discussion of the signal processing approach is deferred to the following chapter.

Chapter 5

Signal Processing Approach

5.1 Introduction

Measurements of flight-time, path-length, impulse response and path loss all rely on information obtained from the spectrum of the receiver output signal as previously discussed in Chapter 4. Having elected to make use of software techniques, the aim of the work presented here is to justify decisions regarding details of the analysis process. Since software spectral analysis techniques invariably depend on the use of the Fast Fourier Transform (FFT) algorithm, issues relating to sequence padding and window selection will be addressed. The discussion of window selection, however, will be restricted to the Hamming and Dolph-Chebyshev windows whose performance will be assessed on the basis of resolution capabilities. Throughout this chapter it is assumed that the data to which the analysis is applied was obtained from the main A/D channel. Data from the in-phase and quadrature (I(t) and Q(t)) channels are processed in a similar manner.

5.2 Data Padding

The signal available on the main receiver output channel is sampled at a rate of f_s=200 kHz by the A/D converter. Since the signal has a duration τ of 50 ms, this results in a total of 10,000 samples per sweep. The spectrum of a time-series such as this can be computed using the Discrete Fourier Transform algorithm (DFT) without alterations to the sequence length. Unfortunately, the drawback of this approach is that computation of the Fourier coefficients requires a relatively large amount of computing time - approximately 10 to 15 minutes per sweep on a 386 personal computer. Reductions in processing time can be attained through the use of a Fast Fourier Transform (FFT)

algorithm. However, for the particular algorithmic implementation used in this thesis, the length of the time series must be a power of 2. Padding the original time series to satisfy this requirement causes the FFT to interpolate the amplitudes of spectral components at frequency increments smaller than the unpadded resolution of the signal thereby producing a spectrum that is "smoother" in appearance [28]. The advantage of this smoother spectrum is that peaks and valleys tend to be well defined As a consequence, relatively simple peak detection algorithms can be employed to extract the desired measures from the signal spectrum.

Figures 5.1 through 5.4 illustrate the effects of padding on a synthetic receiver output signal representative of the case in which two paths exist between transmitter and receiver. To facilitate comparison, power and frequency values in each of these graphs were normalized by their peak power and Nyquist frequencies respectively This series of Figures shows the effect of padding the original sequence (of length 10,000) to lengths of 16,384 (2^{14}), 32,768 (2^{15}) and 65,536 (2^{16}) points in order to capitalize on the speed advantage offered by the FFT algorithm. For simplicity, a Hamming window was assumed in all cases. As the length of the series was increased, the two peaks were observed to have become increasingly well defined in the sense that their contours possessed fewer rough edges. In those cases where the spectral features are well defined, such as those of Figures 5.3 and 5.4, power/frequency measurements can be obtained by simply examining the power spectrum in overlapping three point contiguous sections looking for the instance in which the middle sample is greater than the two end samples as illustrated in Figure 5.5. When an event of this type occurs, the desired measurement(s) can be obtained from the power/frequency information associated with the medial sample.

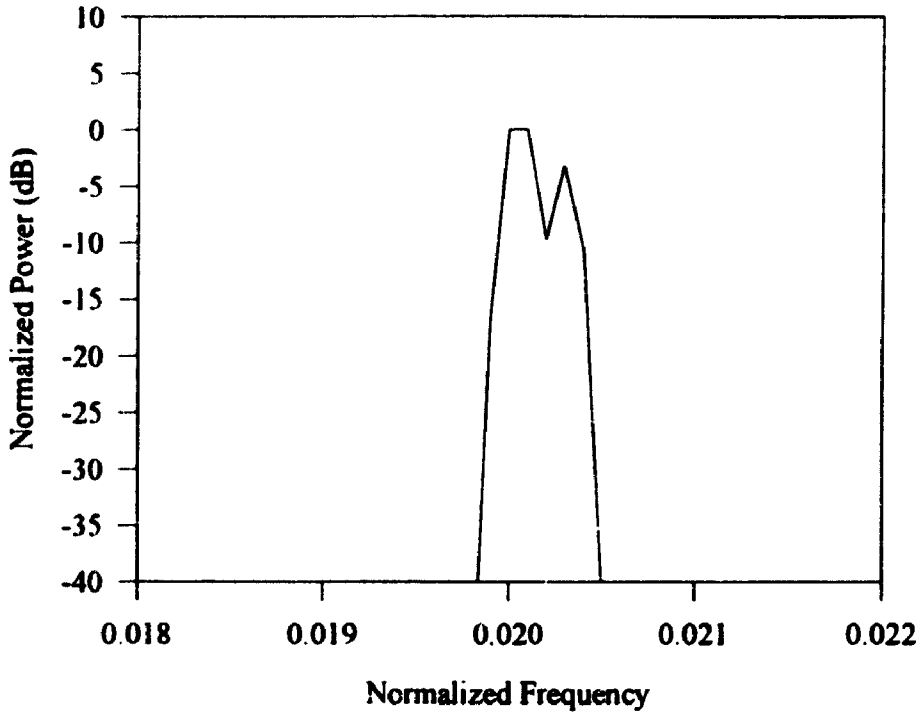


Figure 5.1 Length = 10,000 Samples with Hamming Window (using DFT) Power Normalized to Peak Value, Frequency Normalized to Nyquist Rate.

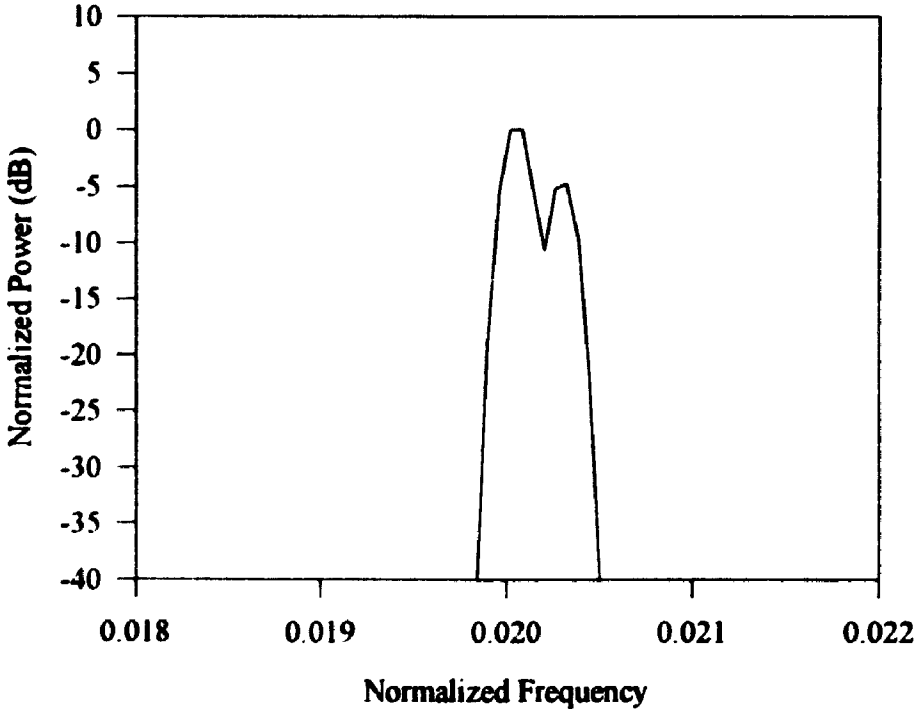


Figure 5.2 Length = 16,384 Samples with Hamming Window (using FFT) Power Normalized to Peak Value, Frequency Normalized to Nyquist Rate.

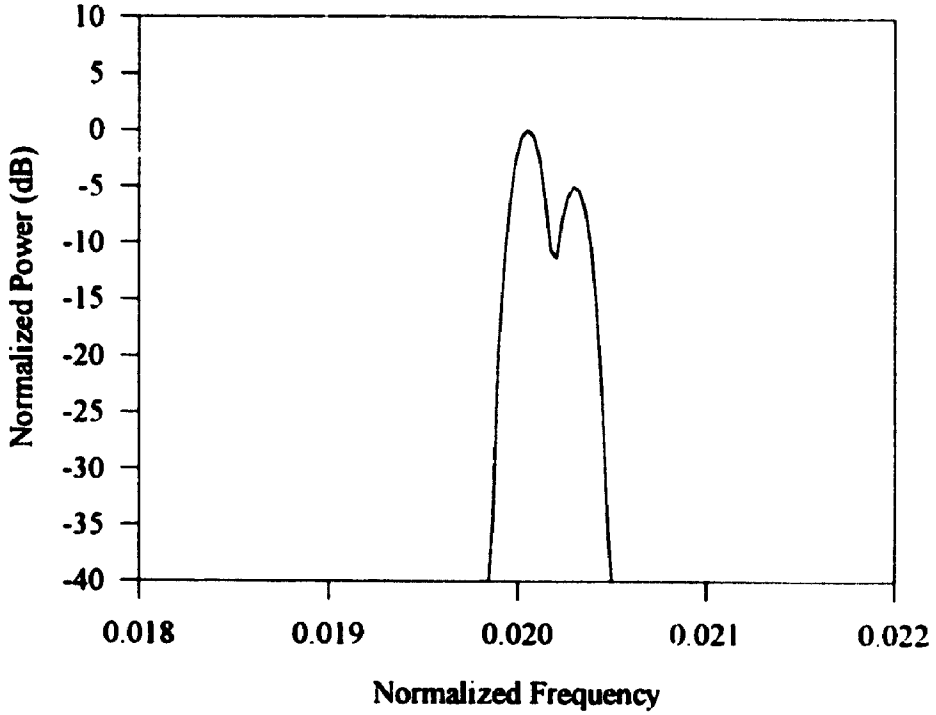


Figure 5.3 Length = 32,768 Samples with Hamming Window (using FFT)
Power Normalized to Peak Value, Frequency Normalized to Nyquist Rate

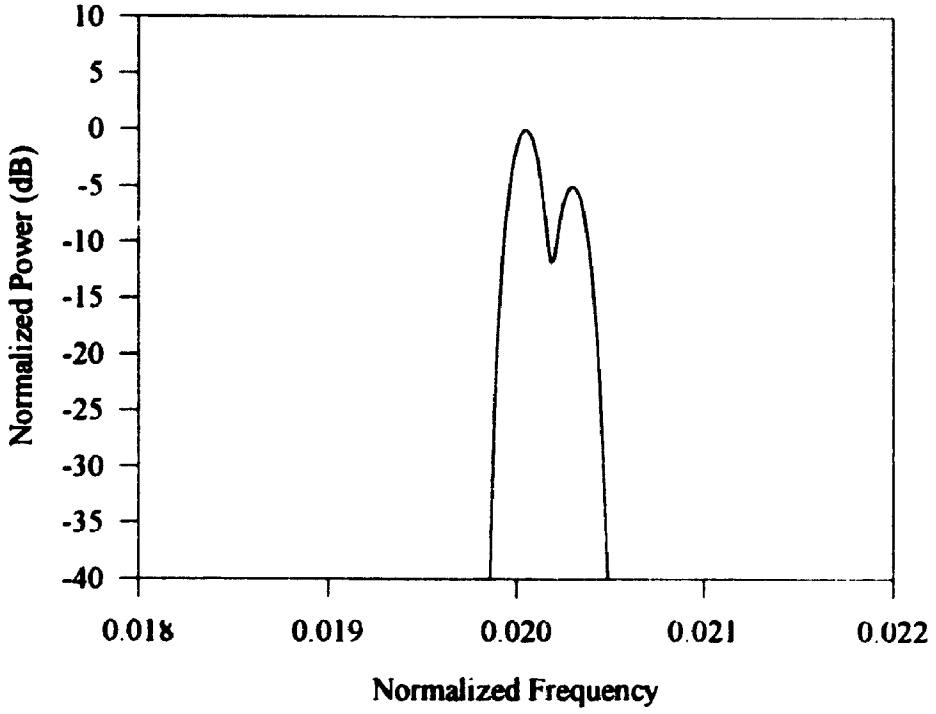


Figure 5.4 Length = 65,536 Samples with Hamming Window (using FFT)
Power Normalized to Peak Value, Frequency Normalized to Nyquist Rate

A visual comparison of Figures 5.3 and 5.4 reveals only minor differences between their spectral features and consequently there appears to be little advantage in padding the time series to a length of 65,536 points. Since one of the objectives is to minimize the processing time it is desirable to select as short a sequence length as possible. Given the conflicting needs for minimizing processing time and obtaining well defined spectral features, a series length of 32,768 samples was chosen. Although not explicitly measured, the processing time required to compute the transform for the 32,768 point series using the FFT algorithm was observed to be less than that for processing the 10,000 point series using the DFT algorithm.

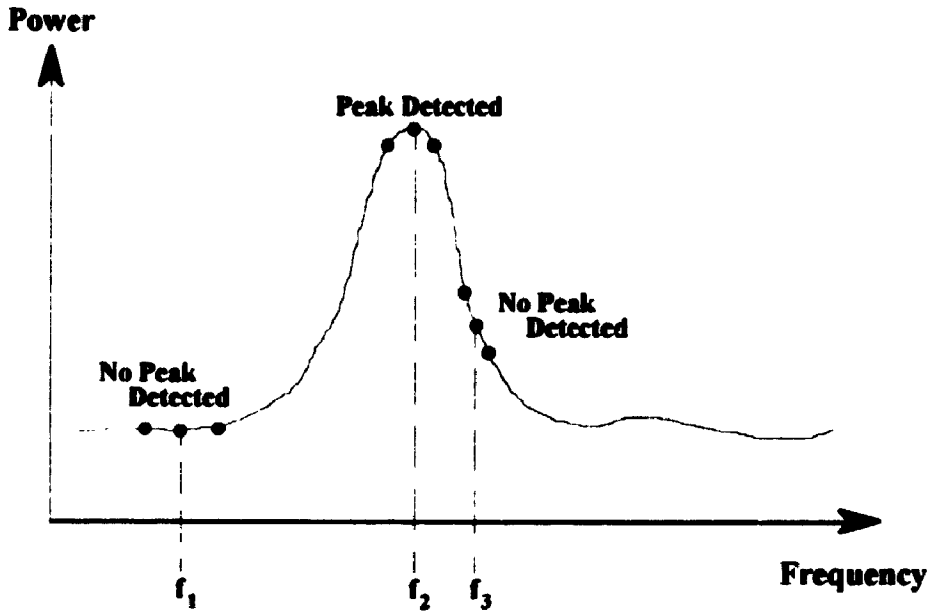


Figure 5.5 Illustration of the Peak Detection Strategy

5.3 Candidate Window Functions

For the signal processing application being considered, the most important characteristic of the windowing function is its ability to clearly resolve nearby spectral components. Failure to select such a window reduces the overall resolution capabilities of the impulse response measurement system, thereby limiting the flight-time and path-length

differentials that can be observed. In order to achieve good resolution, potential window functions must possess a narrow main lobe and have low sidelobe levels. Based on the results of an in-depth comparison of window characteristics by Harris [29], the Hamming and Dolph-Chebyshev window functions were selected as suitable candidates. In addition to its desirable spectral characteristics, an attractive feature of the Hamming window is its ease of implementation. Although considerably more difficult to implement than the Hamming window, Edde [30] and Harris [29] suggest that windows belonging to the Dolph-Chebyshev family are very well suited to the application under consideration. In the remainder of this section, implementation details for the Hamming and Dolph-Chebyshev windows will be presented and will be accompanied by a discussion and comparison of their spectral characteristics.

The equation describing the Hamming window function in the time or sample domain is given by Ludeman [28] as:

$$w[n] = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \le n \le N-1 \tag{5.1}$$

where; $w[n]$ is the weight of the n^{th} sample and N is the number of samples (length) in the time series. A graph of the window function in the sample domain is shown in Figure 5.6. To simplify presentation, a series length of N=50 samples was chosen. The spectral characteristics of this window were computed by first padding the window function to a length of 1024 samples and then computing its amplitude spectrum using the DFT algorithm. To facilitate future comparisons with windows from the Dolph-Chebyshev family, a normalized amplitude and frequency scaling was adopted as illustrated in Figure 5.7. The noteworthy features of this spectrum are the main lobe width and the peak sidelobe level. Employing a -6 dB criterion to the main lobe measurement gives a main lobe width (single-sided) of 0.0371 units. A measurement of the sidelobe levels reveals that the peak sidelobe level is approximately 43 dB down from the main lobe.

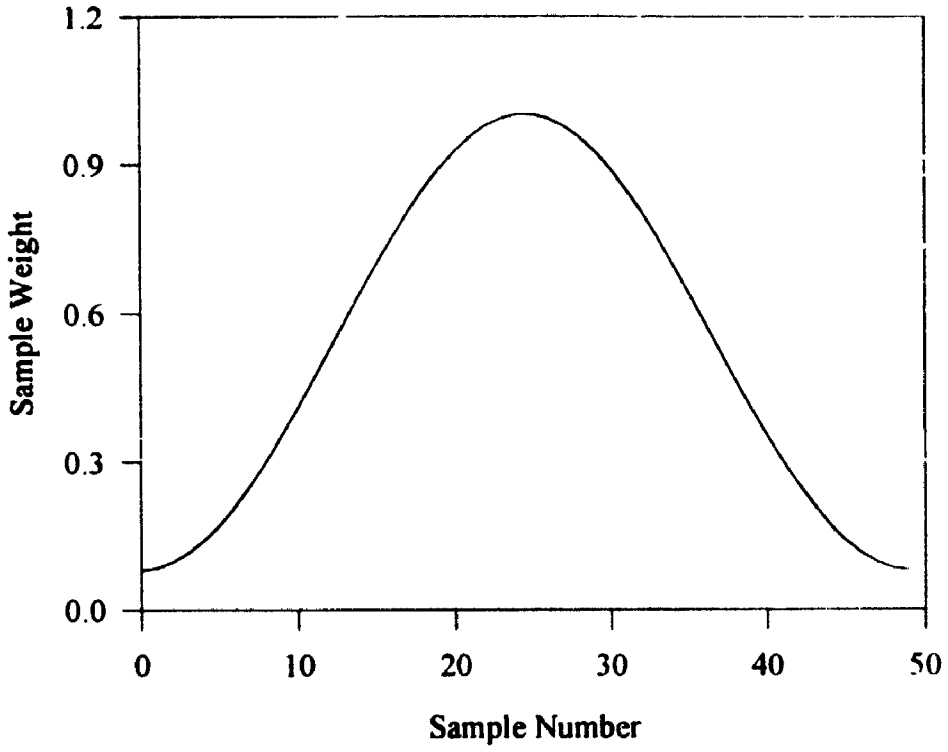


Figure 5.6 The Hamming Window Function in the "Sample" Domain

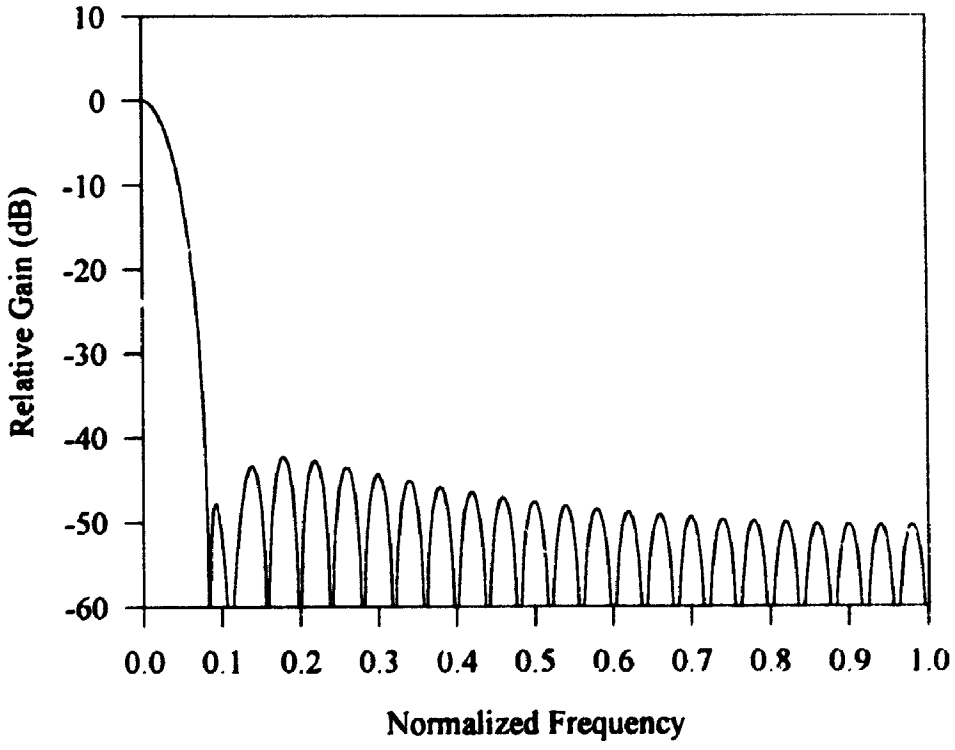


Figure 5.7 Amplitude Spectrum of the Hamming Window

The procedure for generating the sample domain weighting coefficients for the Dolph-Chebyshev family of windows is considerably more involved than that required for the Hamming window. Following the method of Nuttal [31] the weighting coefficients in the sample domain *w[n]* can be obtained from a series of complex coefficients according to the equations:

$$w_{2n} = \mathrm{Re}\{\alpha_n\} \quad 1 \le 2n \le \frac{N}{2} \tag{5.2}$$

and

$$w_{2n-1} = \mathrm{Re}\{\alpha_{N+1-n}\} \quad 1 \le 2n-1 \le \frac{N}{2} \tag{5.3}$$

where; the complex coefficients, α_n are given by:

$$\alpha_n = \mathrm{F}\left\{ \mathrm{r_n} \exp\left(\frac{j\pi n}{N} \right) \right\} \quad n = 0, 1, \ldots, \frac{N}{2} - 1 \tag{5.4}$$

In equation 5.4, *F{}* denotes the Fourier transform operation and r_n is a real-valued coefficient given by:

$$r_n = \frac{2}{N} \varepsilon_n T_{2N-1}\left(z_o \cos\left(\frac{\pi n}{N} \right) \right) \quad n = 0, 1, \ldots, \frac{N}{2} - 1 \tag{5.5}$$

where ε_n, is the constant:

$$\varepsilon_n = \begin{cases} \frac{1}{2}, & n = 0 \\ 1, & n \ge 1 \end{cases}, \tag{5.6}$$

$T_n(z)$ is the Chebyshev polynomial (Spiegel [32]) of the first kind given by:

$$T_n(z) = \begin{cases} \cos\left(n \cos^{-1}(z) \right), & |z| \le 1 \\ \cosh\left(\cosh^{-1}(z) \right), & |z| > 1 \end{cases} \tag{5.7}$$

and z_o is a constant given by:

$$z_o = \cosh\left[\frac{1}{N-1} \cosh^{-1}\left(10^{\alpha} \right) \right] \tag{5.8}$$

In equations 5.2 through 5.8, the variable n represents the index number in the sample domain and N represents the series length. The remaining parameter α (different than α_n), in equation 5.8, controls the ratio of the peak sidelobe level to the main lobe. This ratio in dB can be computed according to the equation:

$$dB = 20\alpha \quad . \tag{5.9}$$

In practice, the weighting coefficients are obtained by applying equations 5.2 through 5.9 in reverse order. Thus the process begins by selecting the parameter α and then working backwards through the equations computing r_n, α_n and finally the desired window coefficients, $w[n]$.

Figures 5.8 and 5.9 show the Dolph-Chebyshev window in both the sample and frequency domains for α=2.0. As in the case of the Hamming window, the window coefficients were generated for a time series 50 samples in length and were padded to a length of 1024 samples for spectral calculations. In the spectral plot of Figure 5.9, it should be observed that the sidelobe level is constant and is equal to the expected level of -40 dB as predicted by equation 5.9. Although not explicitly shown, an additional noteworthy feature of the Dolph-Chebyshev window spectra is that the width of the main lobe increases as the peak sidelobe level decreases. Assuming a -6 dB width criterion, as in the case of the Hamming window, the main lobe widths for Dolph-Chebyshev windows having α=1.75, 2.0 and 2.25 are given in Table 5.1. Thus, depending on the sidelobe level selected, main lobe widths narrower than that of the Hamming window can be obtained, thereby affording the potential to resolve more closely spaced signals (see Table 5.1). Given that the dynamic range of the receiver is on the order of 30 dB, the Dolph-Chebyshev window shown in Figure 5.8 with α=2.0 will be assumed for the comparative resolution study in the following section. Since this window has a narrower main lobe than that of the Hamming window, better resolution performance is expected. Software for

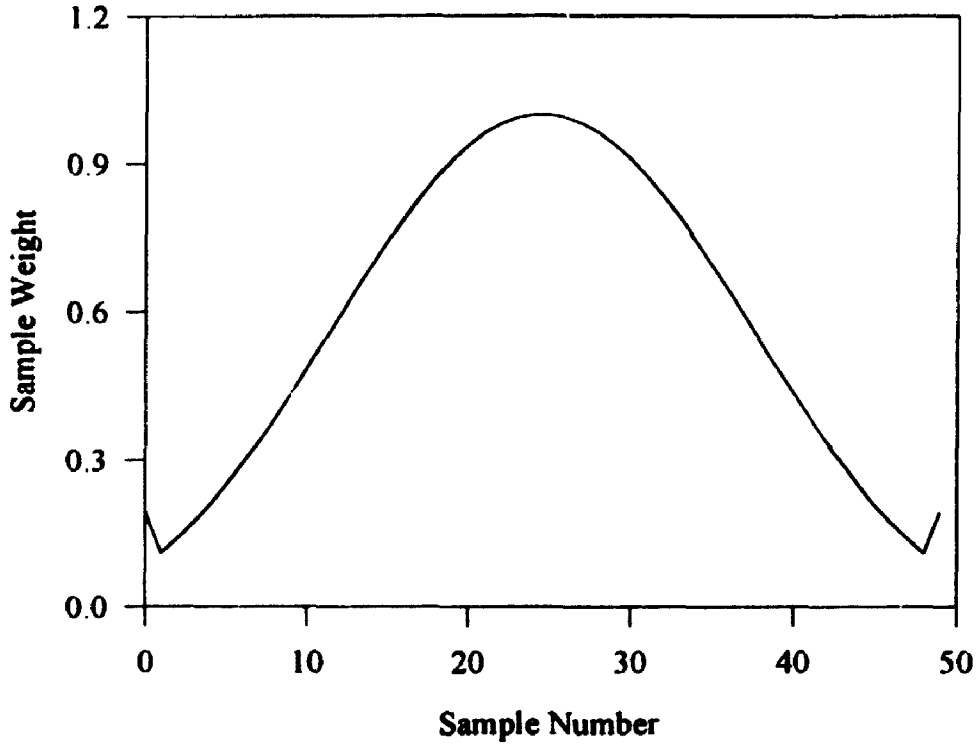


Figure 5.8 Dolph Chebyshev Window (α=2.0) in the "Sample" Domain

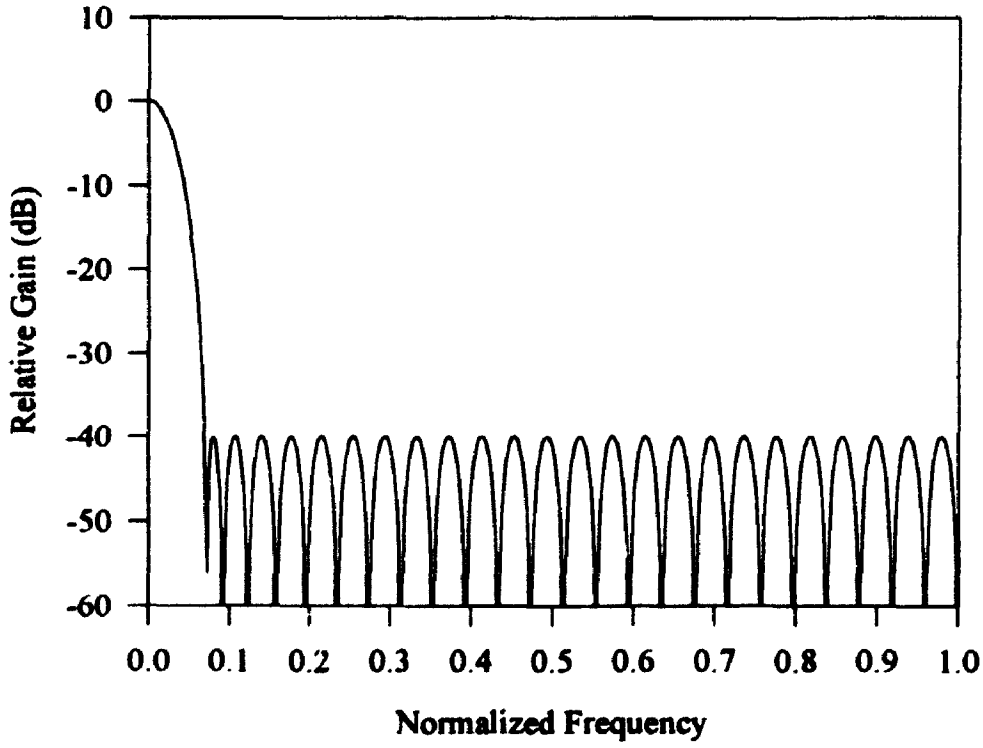


Figure 5.9 Amplitude Spectrum of the Dolph-Chebyshev Window for α=2.0

generating the weighting coefficients for these window functions can be found in Appendix B.

Table 5.1 Window Characteristics

Window Name and Parameters	Peak Sidelobe Level (dB)	-6 dB Bandwidth (Normalized Units)
Hamming	-43	0.0371
Dolph-Chebyshev (α=1.75)	-35	0.0332
Dolph-Chebyshev (α=2.0)	-40	0.0352
Dolph-Chebyshev (α=2.25)	-45	0.0371

5.4 Resolution Testing

The procedure for evaluating the resolution capabilities of the Hamming and Dolph-Chebyshev windows traces that outlined by Harris[29]. According to this approach, the spectrum of a signal containing two sinusoids of differing amplitudes and frequencies is computed after weighting the synthetic signal by the desired windowing function. After each series of computations, the spectrum is examined to determine if peaks corresponding to the two sinusoids are clearly distinguishable. In the event that both peaks are resolved, the frequency difference between the sinusoids is decreased and the procedure repeated. The smallest frequency difference for which the peaks can still be distinguished is a measure of the resolution capability of the window.

A series of synthetic test signals each consisting of two sinusoids with power levels differing by 5 dB, frequency differentials of 60, 40 and 20 Hz and random phases were generated and analyzed using the Hamming and Dolph-Chebyshev windows respectively. For a frequency differential of 60 Hz, both windows were found to be capable of resolving

the two sinusoids in all instances investigated. When the frequency differential was reduced to 40 Hz however, the analysis program employing the Dolph-Chebyshev window was found to be capable of resolving both peaks reliably while the program employing the Hamming window could resolve both peaks in only 89% of the cases investigated. By increasing the frequency differential to approximately 45 Hz, reliable detection of both peaks could be obtained with the Hamming window. In the final series of tests where the spectral interval was further reduced to 20 Hz, neither window was found to be capable of reliably detecting the two peaks. Since the resolution of the Dolph-Chebyshev window with α=2.0 was found to be marginally superior to that of the Hamming window, it will be used exclusively in the analysis of data from the impulse response measurement system. From the analysis of Chapter 4, a frequency resolution of 40 Hz translates into a differential flight-time measurement capability of 0.8 μsec which corresponds to a path-length differential of 240 meters.

5.5 Summary

In an effort to speed up spectral calculations, the DFT algorithm was replaced by the more computationally efficient FFT. However, the particular version of the FFT algorithm used in this analysis requires that the time series be padded such that its length is a power of 2. Although increased sequence lengths result in smoother spectral features, a penalty in the form of increased processing time is incurred. As a compromise, a sequence length of 32,768 samples was selected. Records of this length which were processed using the FFT algorithm, showed a speed advantage over the DFT approach.

An evaluation of the Hamming and Dolph-Chebyshev window functions revealed that marginally better resolution could be achieved through the use of the Dolph-Chebyshev window with α=2.0. This evaluation showed the window resolution to be approximately 40 Hz, which corresponds to a differential flight-time measurement of 0.8 μ

sec or, equivalently, a differential path length measurement of 240 meters. The Dolph-Chebyshev window with α=2.0 will be used exclusively in the analysis of data obtained from the impulse response measurement system as discussed in the following chapters.

Chapter 6

Testing and Calibration

6.1 Introduction

Due to the complex nature of the impulse response measurement system, extensive testing procedures were undertaken to verify instrument operation. In addition to a thorough testing of the integrated modules comprising the transmitter, receiver, signal processing and recording systems, procedures were also undertaken to calibrate the receiver for both path loss and flight-time measurements. Since this system must ultimately function in a noisy environment, additional tests were carried out to evaluate the integrity of measurements made under normal operating conditions. Further assurances were derived from qualitative experiments in which the "bursty" conditions of the meteor burst channel were simulated using simple test equipment. Detailed descriptions of these procedures and their results, are presented in this chapter.

6.2 Transmitter Measurements

A block diagram of the apparatus used to test the transmitter is shown in Figure 6.1. The "transmitter" block in this diagram represents the power amplifier module of Figure 4.13 while the block labeled "chirp or carrier" represents the chirp generator, composite signal generator, transmit sequencer and up-converter modules of Figures 4.9 through 4.12. Using this setup, one of several test signals (chirp, carrier or sweep) can be applied to the transmitter and the output spectrum observed. It should be noted that the dummy load used in this apparatus permits the input signal to monitored through an

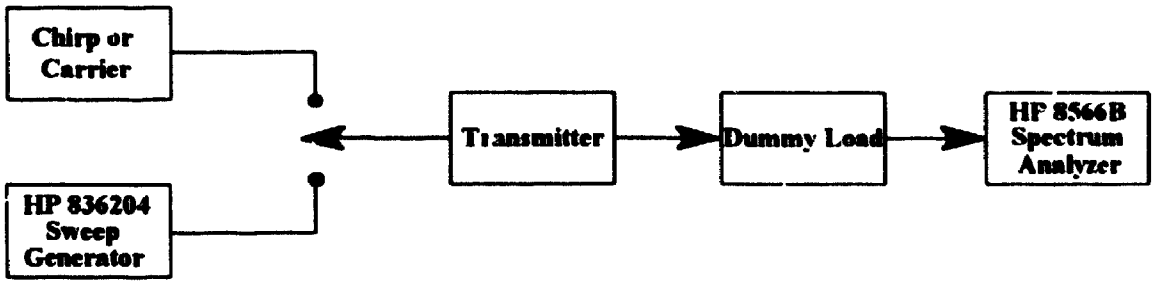


Figure 6.1 Transmitter Test Apparatus

internal 30 dB attenuator thereby enabling the transmitter to be operated at maximum power without overloading the input of the spectrum analyzer.

Tests were first conducted with the sweep generator connected to the transmitter input in order to confirm that the filters and power amplifiers were functioning as designed. For this test, the sweep generator was configured to sweep the frequency band from 38.75 to 41.25 MHz with an output power level of 10 dBm. The spectrum of the output signal observed under these conditions is shown in Figure 6.2. As expected, the transmitter gain remains relatively constant over the bandwidth of interest (38 to 42 MHz) and was found to vary by approximately 2.0 dB. According to the graph, the output power at the band-center (39.75 MHz) is 23.5 dBm. Compensating for the 30 dB attenuation provided by the dummy load, gives the true transmitter output power as 53.5 dBm or 224 Watts. Although the output power level is 6.5 dB below the desired 60 dBm level called for in the design, the spectral measurements of Figure 6.2 indicate that the power amplifier module was otherwise functioning as designed.

Having verified the operation of the power amplifier module, the sweep generator was disconnected and replaced with the unit capable of generating the chirp or carrier signals. For the case in which only the chirp signal was applied to the transmitter (carrier signal disabled), the spectrum shown in Figure 6.3 was obtained. Like the spectrum produced using the sweep generator, the spectral characteristics of the chirp source were

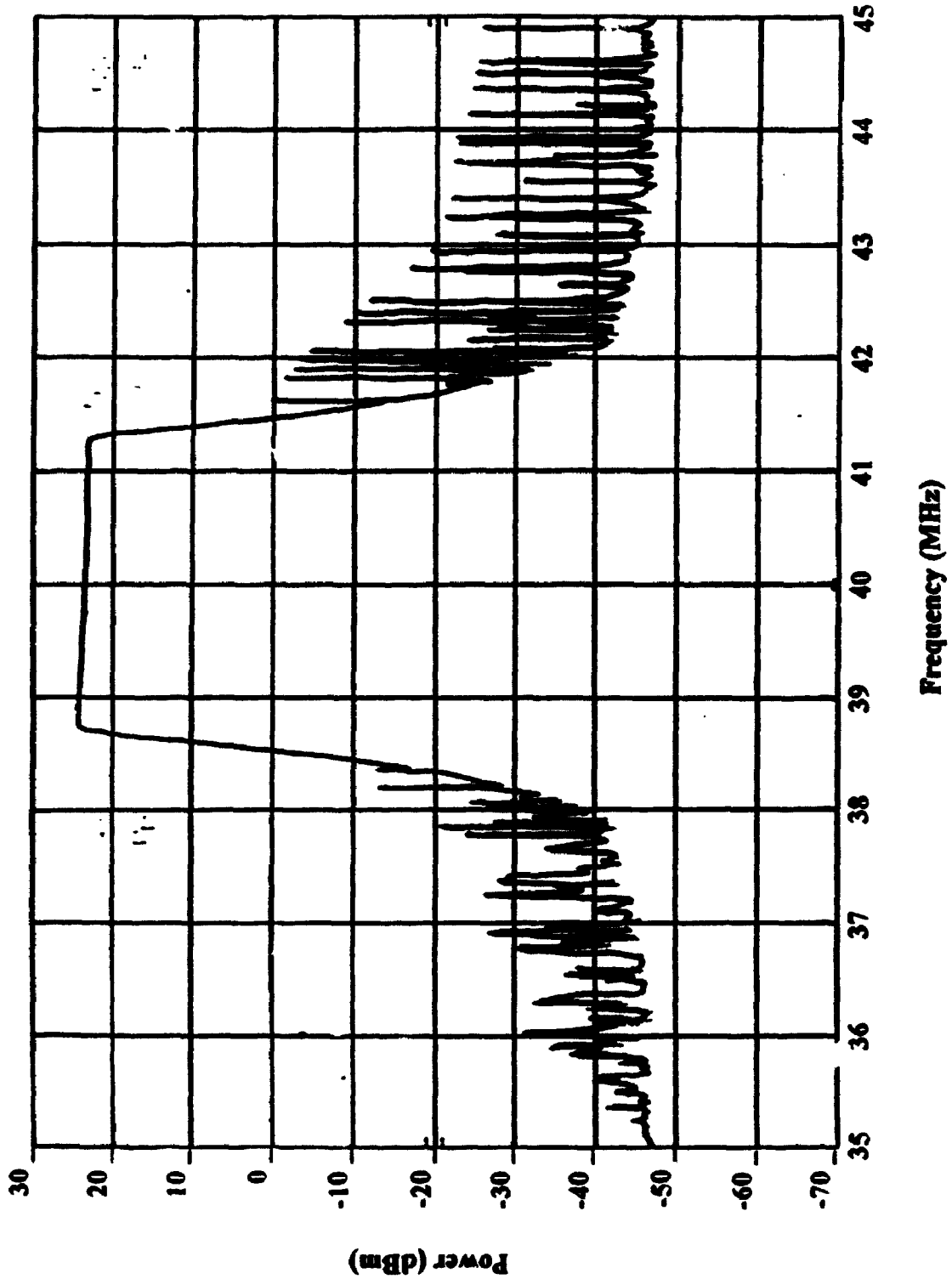


Figure 6.2 Transmitter Output Spectrum (Using Sweep Generator)

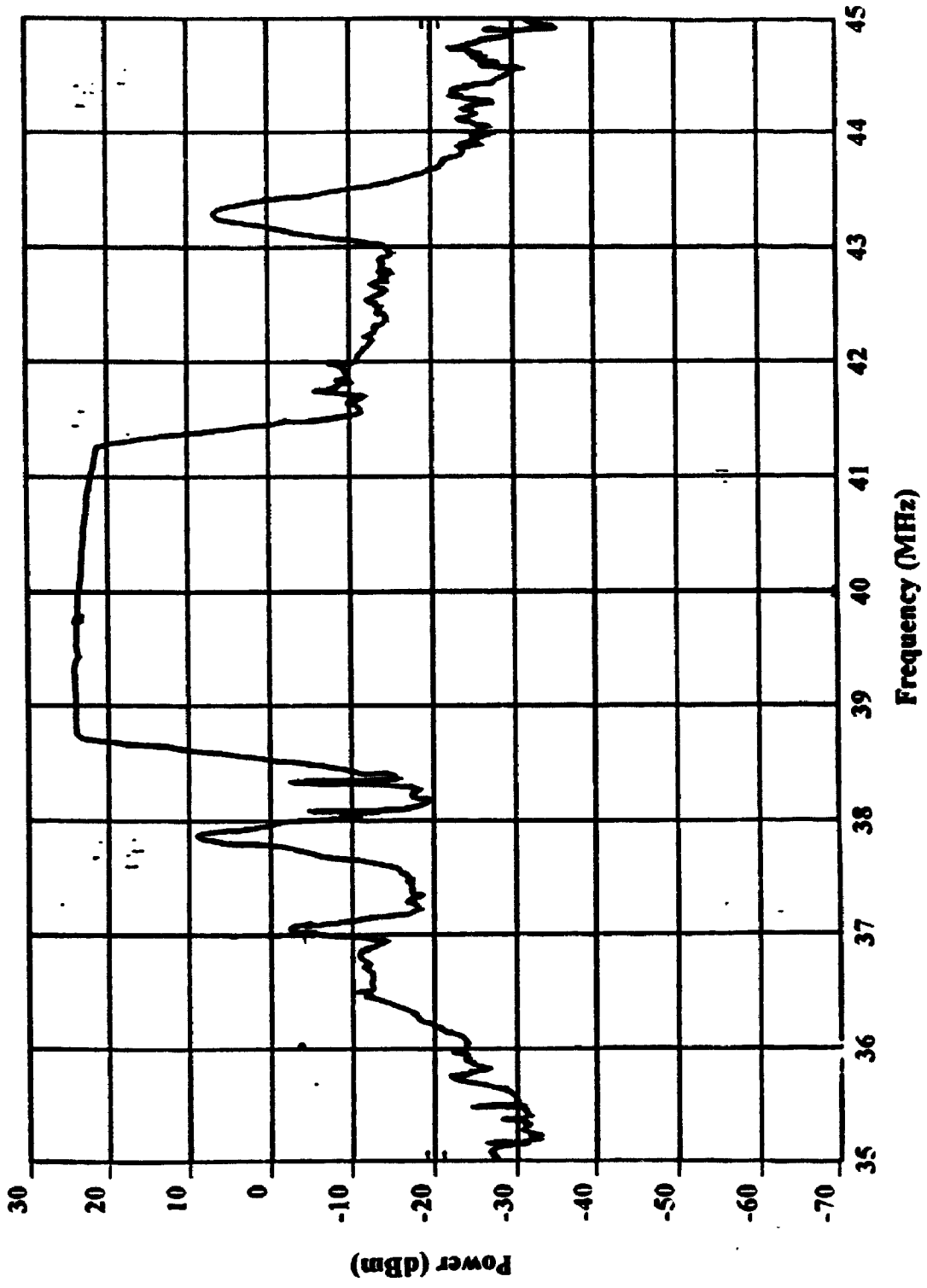


Figure 6.3 Transmitter Output Spectrum (Using Chirp Generator)

found to be similar in that both possessed nearly identical gain characteristics in the frequency band from 38.75 MHz to 41.25 MHz. The single differentiating feature of these two spectra is the power level of the spurious emissions produced by the chirp source. Fortunately, however, these out-of-band emissions occurred only intermittently and when present, as shown in Figure 6.3, are roughly 20 dB down from the level of the desired output signal. In the final case in which only the carrier signal was applied to the transmitter (chirp disabled) the spectrum shown in Figure 6.4 was obtained. It should be noted that this spectrum is consistent with previous observations in that the levels of the undesired emissions were again roughly 20 dB below the level of the desired spectral component. Aside from the previously noted reduction in the output power level and the presence of low-level spurious emissions, the test results demonstrate that the integration of the modules comprising the transmission system results in the generation of signals possessing the desired characteristics.

6.3 Chirp Receiver Measurements

As in the assessment of the previous section, test procedures were carried out to ensure that the receiver was operating in accordance with the design specifications. Due to the overall complexity of the receiver these tests were carried out in two stages. In the first stage of testing, the down-converter module of the chirp receiver was evaluated in terms of its gain and frequency response characteristics while in the second stage of testing the I.F. and synchronization modules were integrated with the down-converter in order to conduct a qualitative evaluation of the entire receiving system.

Gain and bandwidth measurements for the down-converter section were obtained using the test set-up illustrated in Figure 6.5. By sweeping the generator frequency over the range from 38 to 42 MHz, the response characteristics of the amplifiers, filters and mixers was obtained as shown in Figure 6.6. From the graph, the overall gain of this

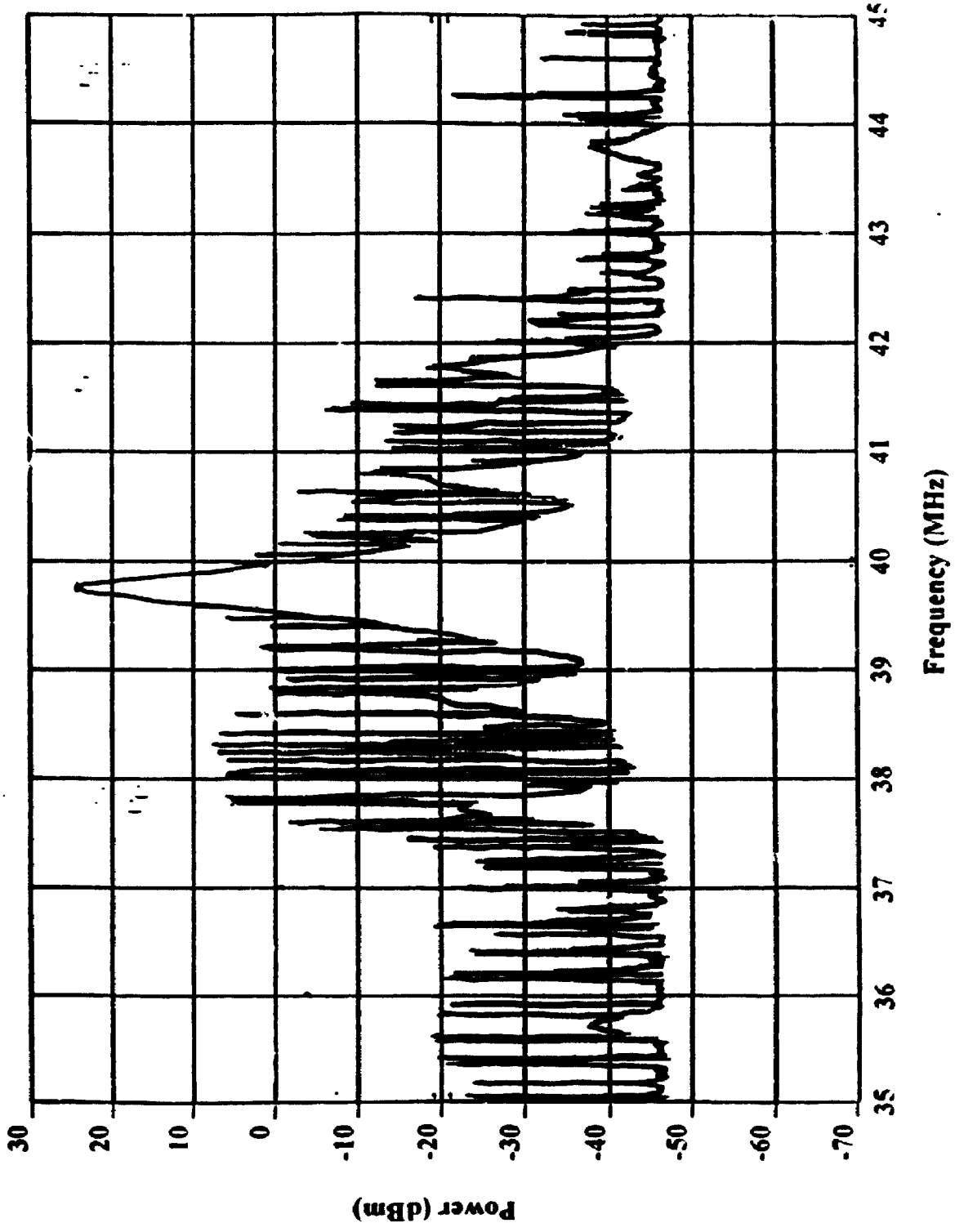


Figure 6.4 Transmitter Output Spectrum (Carrier Only)
Note: The larger sidelobes in this spectrum are due to the presence of an intermittent noise signal

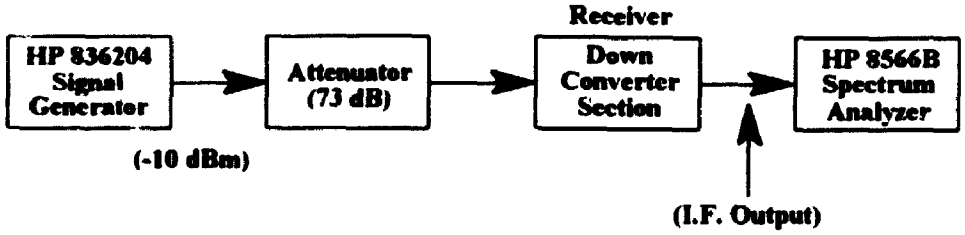


Figure 6.5 Receiver Test Apparatus

section was found to be approximately 88 dB. Although not explicitly shown in the diagram, a variable attenuator was installed in the amplifier chain of the down-converter section to prevent amplifier saturation under large signal conditions. For the measurements shown in Figure 6.6, this attenuator was set to a level 30 dB but can be reduced as necessary in order to increase the receiver gain. The response characteristics of this stage show the gain to be relatively constant across the entire pass-band with minor gain variations of only 1 to 2 dB occurring throughout. According to the graph, the 3 dB bandwidth of this module is roughly 2.8 MHz which is slightly larger than the desired 2.5 MHz bandwidth. Furthermore, it should be noted that the frequency components of the input signal ranging from 38.75 to 41.25 MHz have been successfully translated down to the 1.0 to 3.5 MHz range as required. These results confirm that the down-converter section is operating in accordance with the design specifications.

In order to qualitatively assess the operation of the entire receiver, the I.F. and synchronization modules were integrated with the down-converter section and evaluated through on-air tests. The single objective of these tests was to determine if chirp transmissions could be properly demodulated by the I.F. section. Due to the combined effects of high output power and stray coupling it was not possible to test the system with both the transmitter and receiver located within the same room. To surmount these difficulties the transmitter was installed in a building 1.005 km in distance from the receiver location (experiment conducted at the CRC complex in Ottawa) as shown in Figure 6.7. At the transmitter site a horizontally polarized log periodic dipole antenna

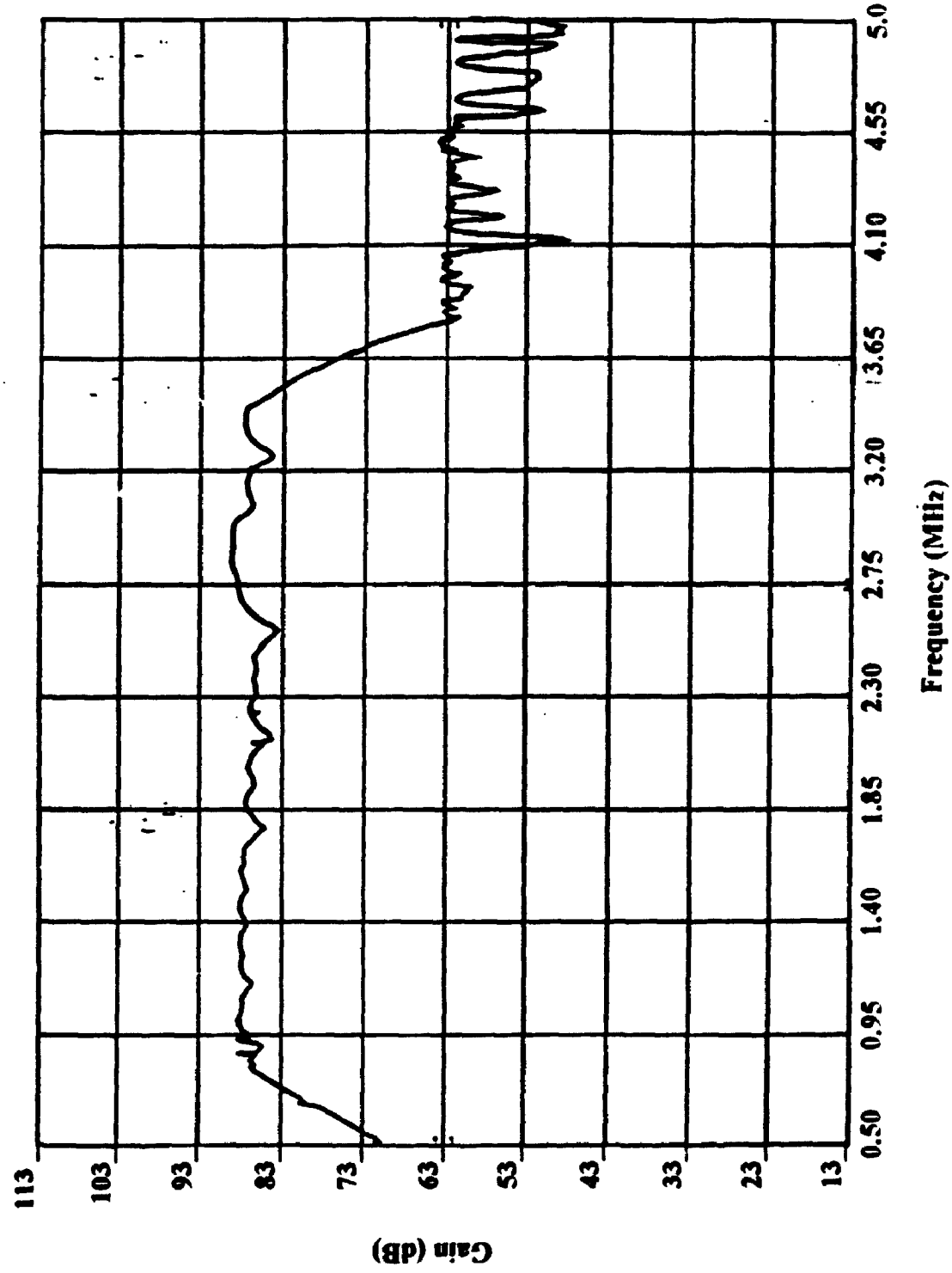


Figure 6.6 Down-Converter Gain Characteristics (Sweep Generator Input)

(LPDA) was mounted at a height of approximately 2.5 meters and connected to the transmitter output via a short length RG8-J coaxial cable. The antenna used at the receiving site was a discone antenna and was mounted on the roof of building 2B - a distance of approximately 30 meters from the receiver. This particular antenna was used at the receiver because of its broad frequency coverage (25 to 1200 MHz) and compact size. In spite of the physical isolation a residual feedthrough level was observed. A modest reduction in this level was gained by bonding all receiving system chassis to ground potential.

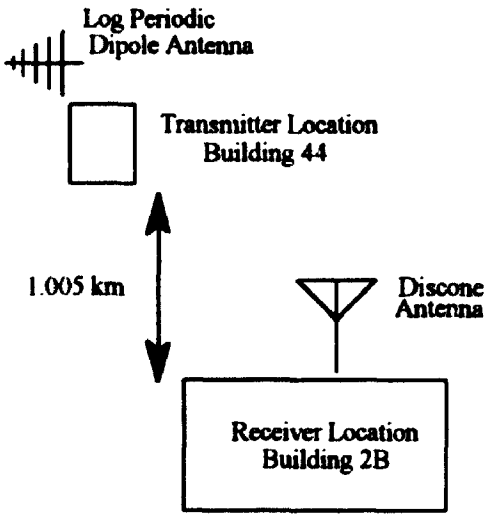


Figure 6.7 Test Setup for "On Air" Receiver Testing

Using this setup, qualitative measurements were made on the main, I(t) and Q(t) analog channels using an oscilloscope. Sinusoidal signals were observed on all channels and the frequency of these sinusoids were observed to change in accordance with variations in the delay time which was manually altered. Although only qualitative in nature, these observations were sufficient to determine that the I.F. section was correctly demodulating the chirp signal. Quantitative results in which the amplitude and frequency of these signals are measured is the subject of the following sections in which the

procedures used to calibrate the receiver for path loss and flight-time measurements are discussed.

6.4 Path Loss Calibration

The path loss between transmitter and receiver can be estimated through the use of the link budget equation given by:

$$L = P_T + G_T + G_R - L_T - L_R - P_{In} \qquad (6.1)$$

where; L is the path loss (dB), P_T is the transmitter power (dBm), G_T is the gain of the transmitting antenna (dB), G_R is the gain of the receiving antenna (dB), L_T represents the transmission line losses at the transmitter (dB), L_R represents the transmission line losses at the receiver (dB) and P_{In} is the power at the antenna terminals of the chirp receiver. With the exception of the receiver input power, P_{In}, all quantities on the right-hand side of equation 6.1 are readily known or can be estimated from existing knowledge of the transmission system parameters. Since direct measurements of P_{In} in an experimental setting are impractical, it is necessary to establish the relationship existing between the receiver input and output powers. This, of course, requires measurement of both quantities during the calibration process.

The experimental setup used to determine the receiver transfer characteristics is shown in Figure 6.8. For completeness, calibration measurements were taken for the main analog output channel as well as the in-phase and quadrature (I/Q) channels although data from the latter will not be used in the analysis of Chapter 7. To ensure reliable triggering at high levels of attenuation, the carrier detect receiver was disconnected from the main power splitter and connected to a separate antenna so that carrier transmissions could still be received. In this configuration it was necessary to reduce the gain of the RF and IF sections to prevent the VHF carrier detect receiver from overloading. This was accomplished using the front panel controls provided for this function. With the chirp

receiver connected to the discone antenna through the variable attenuator, and the main output channel connected to the A/D converter as shown in the diagram, a series of measurements of the receiver output signal were made. For measurements involving the in-phase and quadrature channels, the A/D converter was disconnected from the main channel and reconnected to the headphones output of the SONY PCM-F1 unit as illustrated in Figure 6.8. For all I/Q channel measurements, the output attenuator on the PCM unit was set to 0 dB and both left and right channel gains were set at maximum (10).

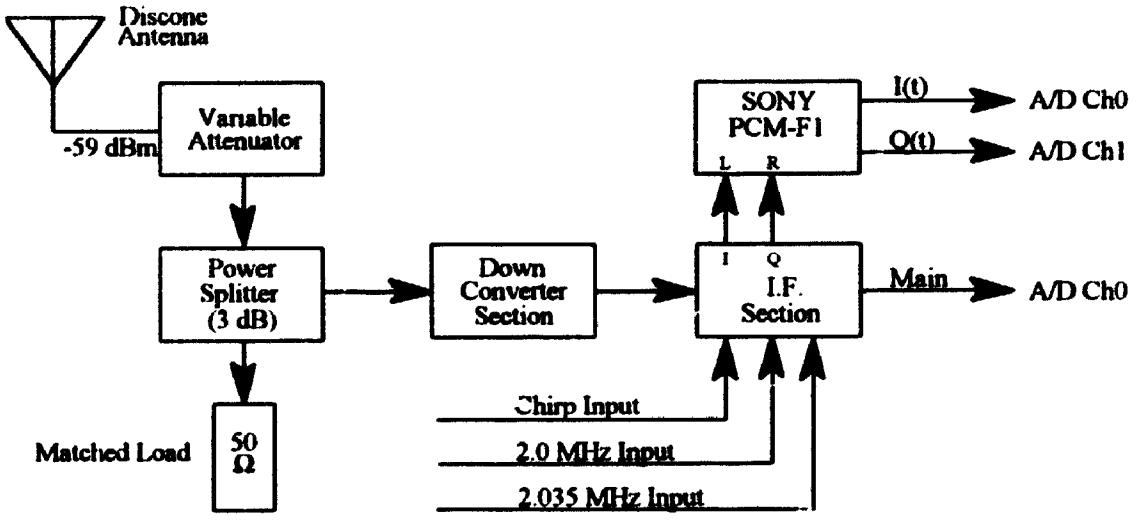


Figure 6.8 Test Setup for Measurement of Receiver Transfer Characteristics

At each external attenuator setting, signals on either the main A/D channel or the in-phase and quadrature (I/Q) channels were digitized and recorded when triggered by the carrier. Signals on the main channel were sampled at a rate of 200 kHz as previously discussed in Chapter 5, while those present on the I/Q channels were each sampled at a rate of 50 kHz. Data representing the signal on the main channel were acquired using the program rddata4.c (see Appendix C) while that from the I/Q channels was collected using a slightly modified version of the same program. Due their similarity, the operation of both data programs can be discussed with the assistance of the simplified flow-chart shown in

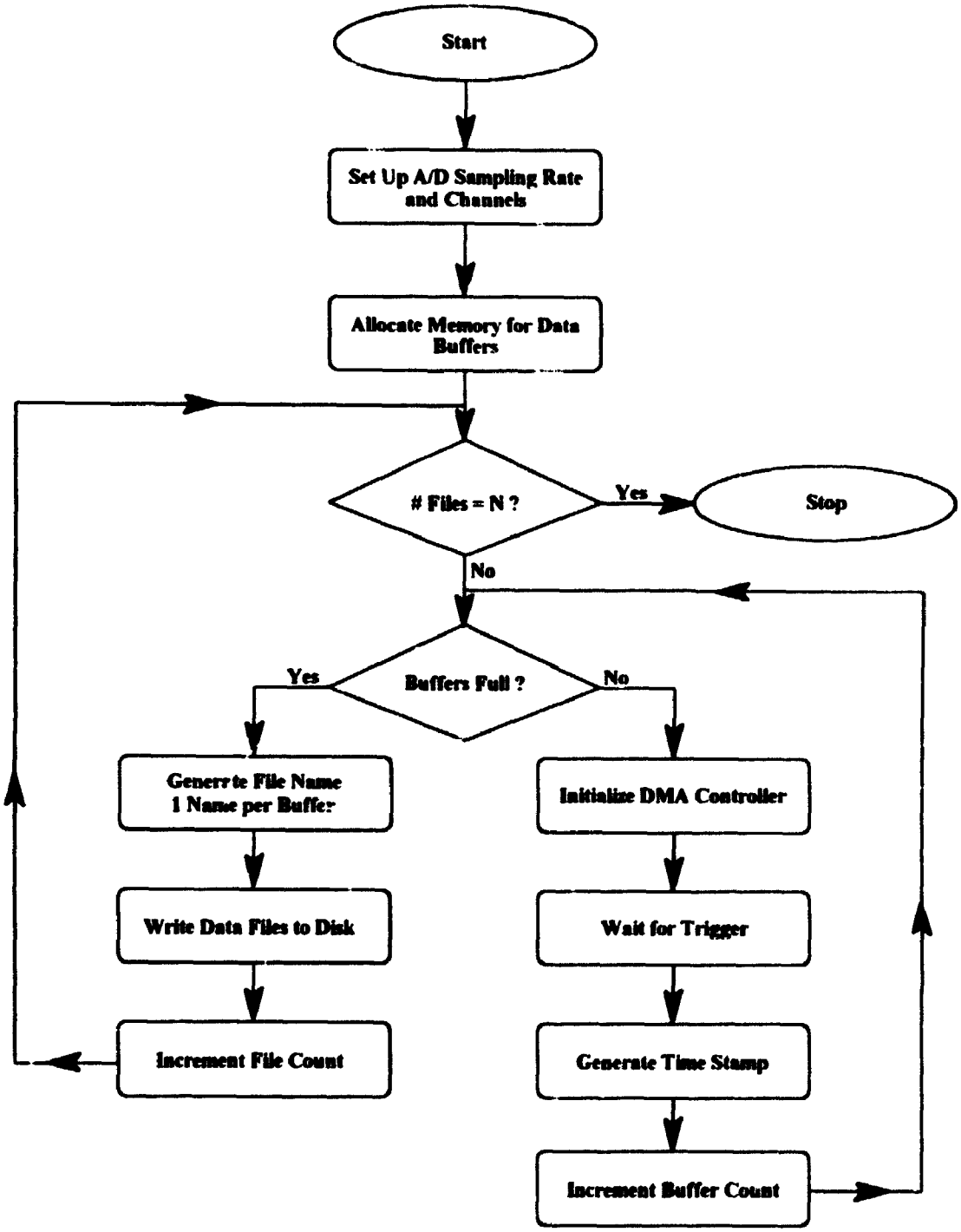


Figure 6.9 Simplified Flow-Chart of the Data Acquisition Program(s)

Figure 6.9. Once initiated, both programs acquire data until the prescribed number of data buffers are filled. On completion of this event, the data is written to disk in binary format. Regardless of source from which the data were recorded, the first 8 bytes of these files contain the date and time information associated with the recording and have the format shown in Figure 6.10. The remainder of the information contained in the file represents the signal samples. In the case of the I/Q channels, the data is stored in multiplexed form beginning with a sample from the in-phase channel and alternating between Q and I samples throughout the remainder of the record as illustrated in the diagram. Since the A/D converter has a resolution of 12 bits, 2 bytes of storage are required for each signal sample. Combined with the 8 bytes required for the time stamp information, this results in data files of 20,008 and 10,008 bytes in length for the main and I/Q channels respectively. Program execution is terminated when the desired number of files have been recorded.

Measurements of the receiver output power, P_{Out}, were obtained from the spectral analysis of the receiver output signal. In all cases, these estimates were derived from the power spectrum and were computed by windowing the data with a Dolph-Chebyshev window (with α=2.0), computing its Fourier transform and finally, computing the square of the resulting complex values. The receiver output power was then estimated from the level associated with the largest spectral peak. Source code listings for the main (pcqld.c) and I/Q channel (iqpwr2.for) analysis programs can be found in Appendix C. It should be noted that in order to use the analysis program iqpwr2.for, the data in the binary file must first be converted into ASCII format using the program iqbin.c. This conversion program is also included in the Appendix.

Before proceeding with the calibration measurements, it was necessary to estimate the power level at the receiver input. Since the antenna was connected to the receiver through a variable attenuator, the receiver input power can be calculated provided the

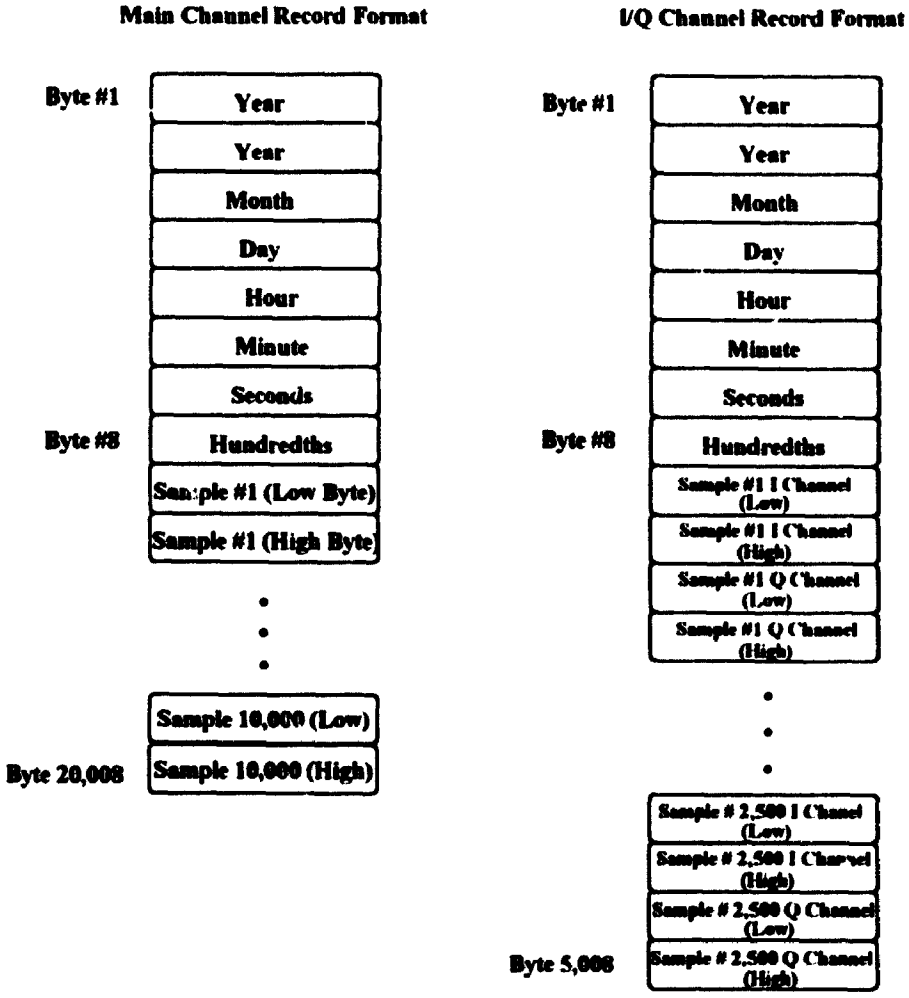


Figure 6.10 Binary Record Formats for Main and I/Q Channels

power at the receiving antenna is known. Knowledge of this quantity was obtained by connecting the receiving antenna directly to the spectrum analyzer input. Under the conditions shown in Figure 6.7 the power at the antenna terminals was measured to be -59 dBm. In order to simulate the range of signal levels likely to be encountered on the link, external attenuator settings between 34 and 70 dB were used which resulted in received power levels ranging from -96 to -132 dBm.

For each external attenuator setting, 10 measurements of the receiver output were recorded and analyzed as previously discussed. During the processing of the data inconsistent measurements were found in the data set and were discarded. These measurements were easy to detect since in all cases, either the power in the main peak was extremely small or the frequency associated with this peak was found to differ significantly from the other frequency measurements. In all cases however, the minimum number of usable measurements at a given attenuator setting was 7 for the main channel and 9 for the I/Q channels. Figures 6.11 and 6.12 show the receiver transfer characteristics for the main and I/Q channels respectively. For both sets of measurements, the circles on the graph indicate the average value of the 10 measurements made at that particular signal level while the solid curves indicate the lines of best fit. For the main channel, the equation relating the output and input powers was found to be:

$$P_{Out} = 111.9 + 0.9 P_{In} \quad , \tag{6.2}$$

where, P_{Out} and P_{In} are the receiver output and input power in dBm. Similarly, the output/input relationship for the I/Q channels was found to be:

$$P_{Out} = 126.2 + P_{In} \quad , \tag{6.3}$$

where, all symbols are as previously defined. It should be noted that for receiver input levels less than -126 dBm, the receiver output level remained relatively constant as a result of feedthrough effects which could not be eliminated in the test setup. Data corresponding to these points were dropped from the data sets for both the main and I/Q channels prior

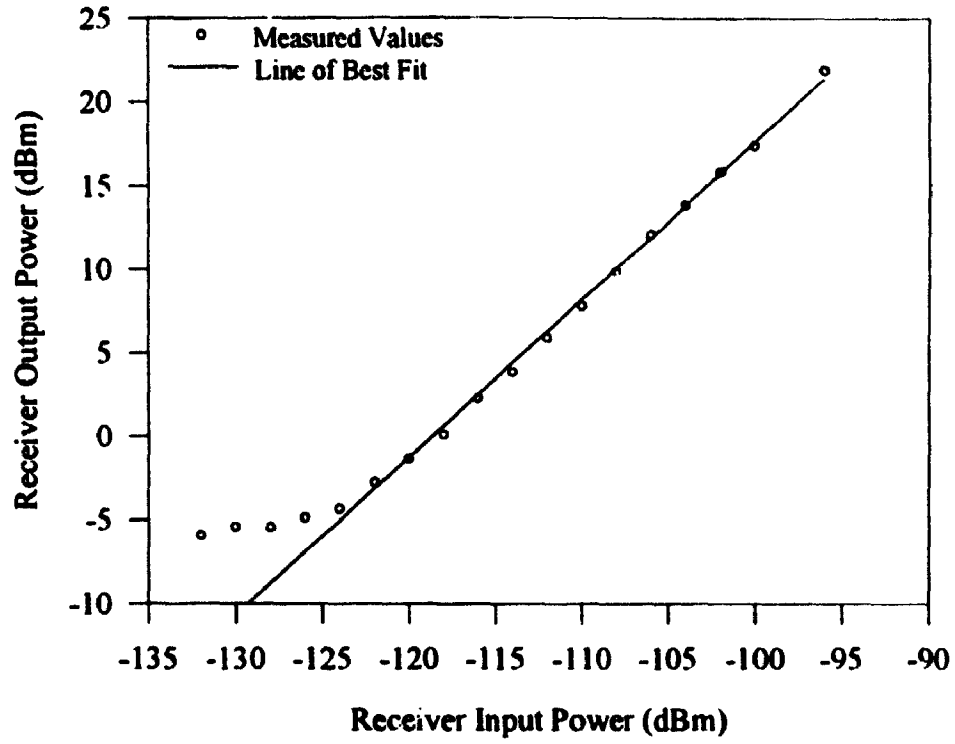


Figure 6.11 Receiver Transfer Characteristic (Main Channel)

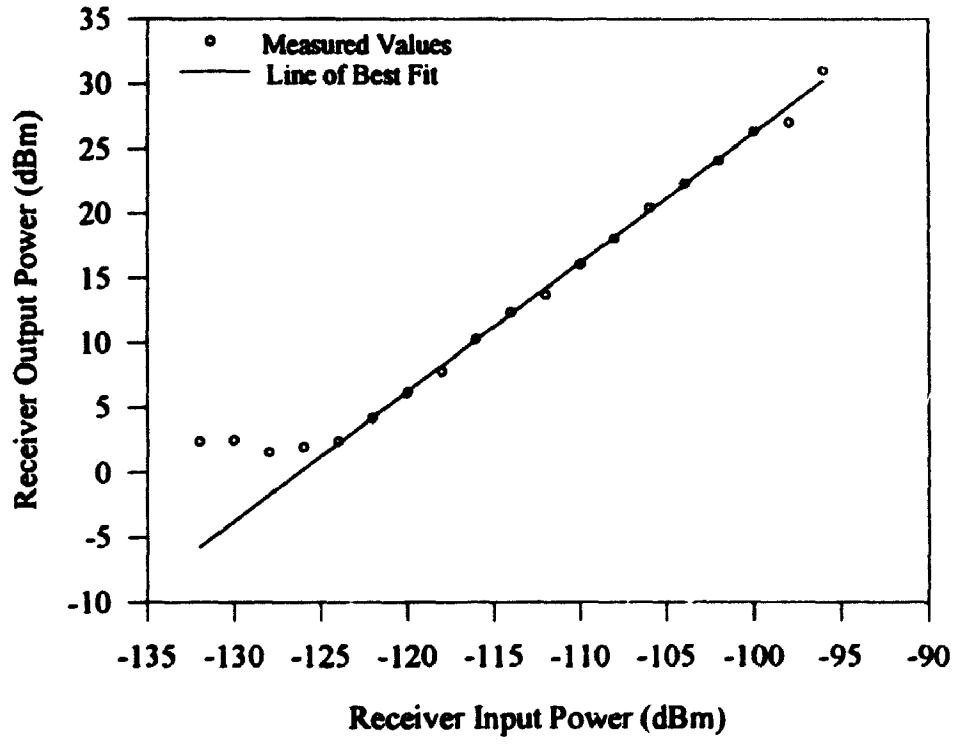


Figure 6.12 Receiver Transfer Characteristic (I/Q Channels)

to the curve fitting process. It should also be noted that the absolute level of the receiver output power is **fictitious** since the actual load resistance was large (typically tens to hundreds of kilohms), but for processing simplicity unity resistance was assumed and the power levels normalized by 10^{-3} watts giving units of "dBm". Since the relationships between the input and output powers are now known, equations 6.2 and 6.3 can be solved in terms of the input power, P_{In}, and substituted into equation 6.1 to obtain equations for the path loss, L. Carrying out these algebraic manipulations gives:

$$L = P_T + G_T + G_R - L_T - L_R - \left(\frac{P_{Out} - 111.9}{0.9}\right) \tag{6.4}$$

for the main channel and:

$$L = P_T + G_T + G_R - L_T - L_R - (P_{Out} - 126.2) \tag{6.5}$$

for the I/Q channels. Realistic values for system parameters of equations 6.4 and 6.5 are P_T=54 dBm (measured), G_T=G_R=7 dB (estimated gain for the LPDA), L_T=1 dB and L_R=3.6 dB. Assuming these values, equations 6.4 and 6.5 can be expressed in simplified form as:

$$L = 188.1 - 1.1P_{Out} \tag{6.6}$$

for the main channel and:

$$L = 189.6 - P_{Out} \tag{6.7}$$

for the I/Q channels. It should be noted that there will always be some degree of uncertainty regarding the "true" path loss as the values assumed for the antenna gains, feedline losses etc. are only estimates of the quantities involved.

Throughout the calibration procedure, all attenuation measurements were made with T_d held constant at 695 μs. To determine if the calibration equations were applicable for measurements made at other delay times, the response of the overall system was tested using the apparatus shown in Figure 6.13. By replacing the receiver chirp generator unit with a 1.0 MHz signal source and applying a C.W. signal to the front-end of the chirp

receiver, frequencies spanning the bandwidth of the I.F. unit from 2.0 kHz to 60.0 kHz were generated. As the source frequency was incremented in 2.0 kHz steps starting from 38.75 MHz, the output voltages on the main, I(t) and Q(t) channels were recorded. The corresponding frequency domain characteristics for these channels are shown in Figures 6.14 and 6.15. For the main channel, the gain was observed to be relatively constant over a 40 kHz bandwidth while the gains of both the I(t) and Q(t) channels were observed to be constant over a 20 kHz bandwidth. As a consequence of these results, the calibration equations for the main and I/Q channels are valid for measurements made at other delay times.

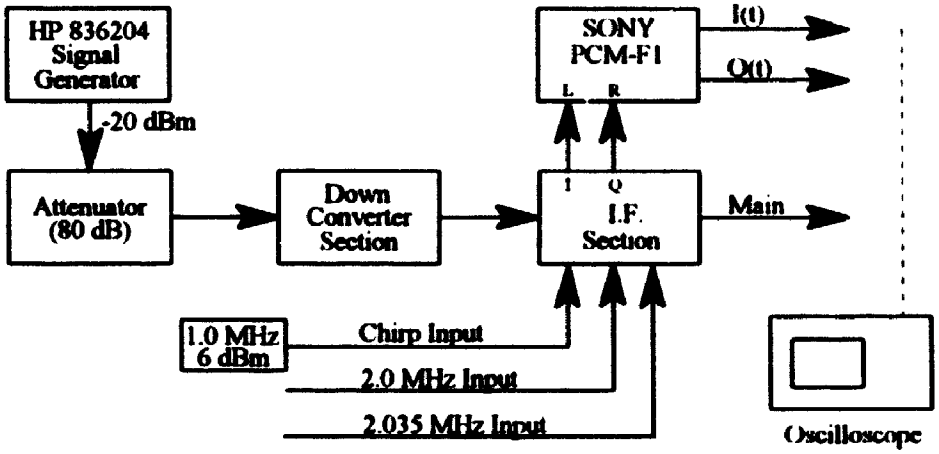


Figure 6.13 Test Setup for I.F. Section Bandwidth Measurements

6.5 Flight-Time Calibration

The flight-time between transmitter and receiver is given by (see Chapter 4).

$$T_p = T_d + \frac{f - f_d}{\mu} \tag{6.8}$$

whose application requires knowledge of the system parameters T_d, f_d and μ as well as the receiver output frequency, f. In order to verify the time/distance measurements of the instrument, an experiment was conducted wherein the delay times (T_d) between the

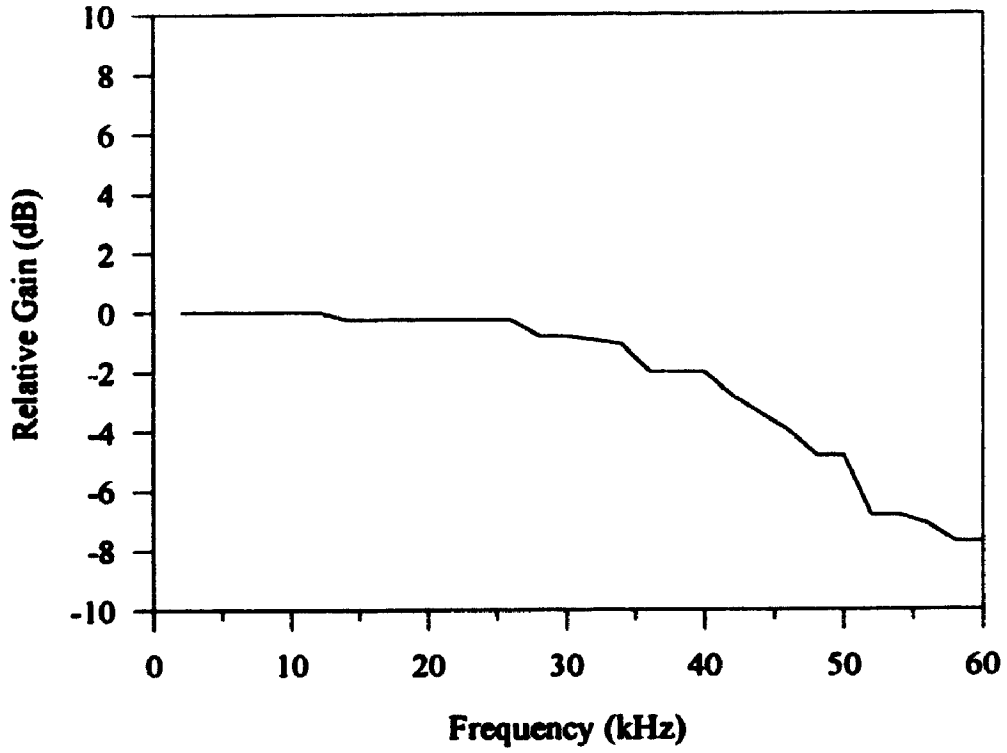


Figure 6.14 Receiver Frequency Response as Viewed on the Main Channel

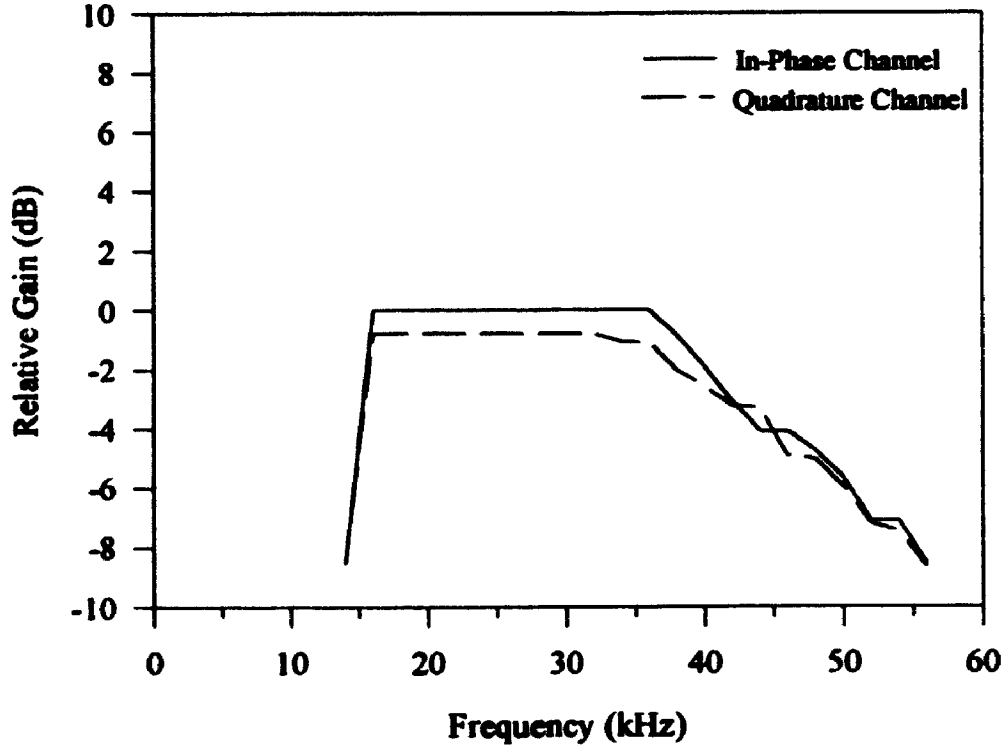


Figure 6.15 Receiver Frequency Response as Viewed on the I&Q Channels

start of the chirps of the transmitter and receiver were varied. As these delay times were changed, the frequency of the receiver output signal also changed and the variations were recorded. For this experiment, the apparatus and geographical setups shown in Figures 6.8 and 6.7 were used. Under these conditions the distance between transmitter and receiver is constant and hence the propagation times computed using equation 6.8 should be identical for all delay time settings.

For each of the delay time settings a total of ten frequency measurements were made and averaged to obtain a single estimate of the "true" output frequency from which, an estimate of the propagation time was deduced. The propagation times determined in this manner are shown in Figures 6.16 and 6.17 for the main and I/Q channels respectively. It should be noted that in the case of the I/Q channels, values for delays ranging between 595 μsec and 1195 μsec were used since delays above and below these values produced signals that were outside the bandwidth of the recording system. Since the distance between the transmitter and receiver was measured to be 1.005 km, the true propagation time can be computed using:

$$T_{po} = \frac{d}{c} = 3.35(\mu\text{sec}) \ , \qquad (6.9)$$

where; T_{po} is the true propagation time, d is the separation distance and c is the speed of light in a vacuum. From the graphs showing propagation time as a function of the delay time (Figures 6.16 and 6.17) it is obvious that the measured propagation times are greater than the true propagation time and this difference is due to the delay time introduced by the receiver. Therefore, a receiver delay time can be associated with each propagation time measurement and computed according to the equation:

$$T_{Rk} = T_{Pk} - T_{po} \qquad (6.10)$$

where the index, k, indicates the flight-time at the k^{th} delay time setting. Since the receiver delay times, T_{Rk}, change very little over the range of delay times (T_d) the T_{Rk} values were

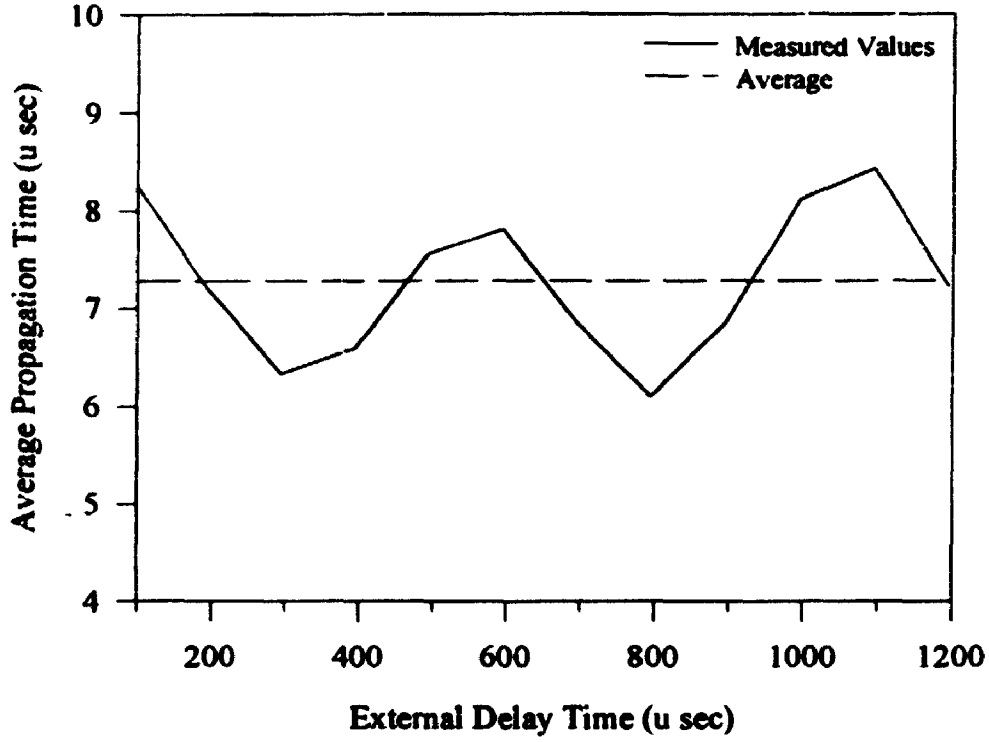


Figure 6.16 Propagation Time Variations (Main Channel)

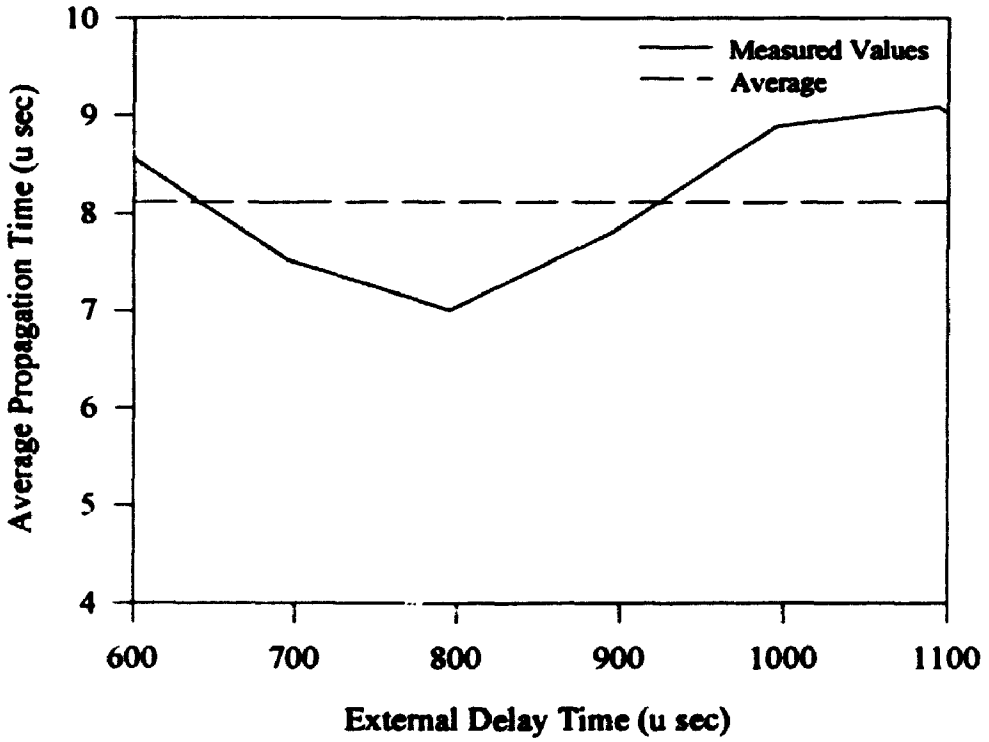


Figure 6.17 Propagation Time Variations (I/Q Channels)

averaged to obtain a single estimate of the receiver delay time. That is:

$$\overline{T}_R = \frac{1}{N}\sum_{k=1}^{N}\left(T_{pk} - T_{po}\right) \quad . \tag{6.11}$$

where; N is the number of delay times for which measurements were made and all other symbols have their usual meanings. Following this analysis, the receiver delay times were found to be 3.923 μsec and 4.764 μsec for the main and I/Q channels respectively Recognizing that the propagation time is simply the sum of the receiver delay time and the true flight-time, equation 6.8 can be modified and rewritten as:

$$T_p = T_d - \overline{T}_R + \frac{f - f_d}{\mu} \quad . \tag{6.12}$$

Thus flight-time measurements can be obtained using equation 6.12 in conjunction with the system parameters μ=5×10^7 (Hz/sec), f_d=12 kHz, $\overline{T}_R$=3.923 or 4.764 μsec and the delay time setting, T_d. It should be noted that the oscillations in the data of Figures 6.16 and 6.17 are likely the result of small delay differentials introduced by the delay element.

The analysis software from which the frequency measurements were obtained operates in an identical manner to that used in the previous calibration procedure. For the measurements taken on the main channel, the program pcald.c is used and the output frequency was determined by manually averaging the 10 individual frequency measurements made at each of the delay time settings. Depending on the magnitude of the delay setting, the receiver output frequency given by:

$$f = f_d + \mu\left(T_p - T_d\right) \tag{6.13}$$

can take on negative values for delay times, T_d, greater than ≈243 μ sec. Since the time series is real valued, the analysis program cannot distinguish between positive and negative frequencies and the sign associated with a particular frequency measurement must be taken care of manually when computing the propagation times during the analysis. For measurements made on the I/Q channels the time series is complex-valued and the analysis

program iqfrq.for performs the frequency averaging automatically and returns a signed frequency measurement. The sign associated with each of these measurements indicates whether the frequency component is greater or less than the center frequency (f_c) of the spectral range recorded on the in-phase and quadrature channels. In order to determine the propagation time, the frequency measurements must first be transformed through the use of the equation:

$$f = f_c + f_M \tag{6.14}$$

where; f is the "true" receiver output frequency, f_c is the center frequency (35 kHz for this experiment) and f_M is the frequency measured by the analysis program. Once the true receiver output frequency is known, its sign must be accounted for as previously discussed and the remainder of the analysis follows that carried out for data collected from the main output channel. Source code listings for both analysis programs as well as a discussion of the quadrature recording system can be found in Appendix C.

6.6 Burst Testing

Throughout the test procedures in the previous sections paths between transmitter and receiver were continuously available and the system was constantly supplied with carrier pulses of sufficient strength to trigger the recording instrument. In the actual operating environment, however, the transmission path only exists for a short period of time - typically 0.5 to 1.0 second - during which the system must be capable of triggering and recording information. To test this aspect of system operation, a relay controlled by a pulse generator was used to simulate the brief duration of a path between the transmitter and receiver. A block diagram of the setup is shown in Figure 6.18. When manually activated by the push-button switch, the one-shot device generates a pulse having a width of approximately 600 ms which in turn energizes the relay thereby connecting the antenna to both the carrier detect and chirp receivers. In the quiescent state, a 50 Ω load is

connected to the attenuator input representing the absence of a suitable path between transmitter and receiver.

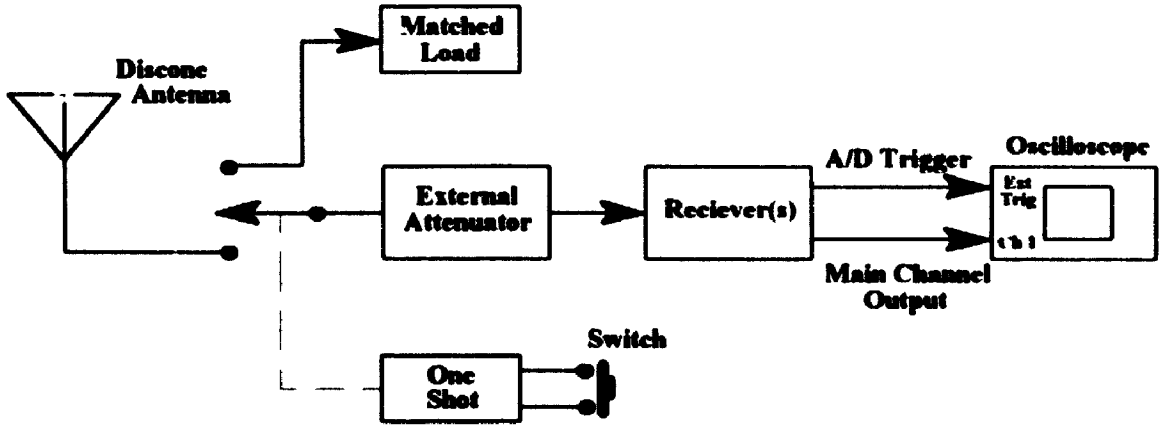


Figure 6.18 Burst Simulation Apparatus

During this qualitative experiment, the relay was energized several times. Upon detection of the carrier signal, the oscilloscope was triggered and the receiver output signal from the main channel was displayed on the oscilloscope. These observations were sufficient to deduce that the system was capable of triggering and recording data under conditions likely to be encountered on a typical meteor burst link.

6.7 Noise Floor and Measurement Reliability

Measurement reliability is closely related to the signal-to-noise ratio (SNR) at the receiver output. In instances when the SNR is large, typically greater than 10 dB. measurements may be interpreted with great confidence, while those made under conditions of low SNR must be treated with caution. In this section, the procedure used to estimate noise floor of the system will be discussed and the results used to establish an upper bound on the path loss that can be measured with the CHIRP system.

Due to localized variations in the radio noise environment, it was necessary to measure the receiver output noise with the receiver located at Elginfield - the intended

operating location. Since the objective of these measurements was to measure the background noise, the transmitter (still located in Ottawa) was modified to transmit only the "probe" signal. When trails of suitable ionization and orientation were present, the recording system was triggered and the noise signals present on the main channel were digitized and recorded on disk. Since only data from the main receiver output channel will be analyzed in Chapter 7, no noise measurements were made for the I/Q channels.

A total of 50 noise measurements were made on the main output channel for purposes of establishing the receiver noise floor. The power spectra of four such measurements are shown in Figure 6.19. From these graphs, the noise power was observed to be relatively constant over the receiver bandwidth and a noise threshold of -17 dBm was chosen so as to be greater than the largest spectral peak encountered in the ensemble of noise recordings. Using this figure as a representative value of the noise power and combining it with the expression for the path loss (equation 6.6) the signal-to-noise floor ratio was computed as a function of the path loss as shown in Figure 6.20. It should be noted that in the analysis of the data in following chapter, an additional 3 dB is added to the noise floor to ensure that all spectral components used in the analysis possess a minimum signal-to-noise floor ratio of 3.0 dB. Thus the largest path loss that can be measured while maintaining a 3 dB signal-to-noise floor ratio is approximately 202 dB.

6.8 Summary

Extensive testing of the transmitter and receiver showed both systems to be operating within the boundaries of the design specifications. Measurements of the transmitter output spectrum showed the transmitter output power to be lower than desired and also revealed the presence of low-level spurious emissions. Neither of these shortcomings however, are expected to have any discernible effect on system operation.

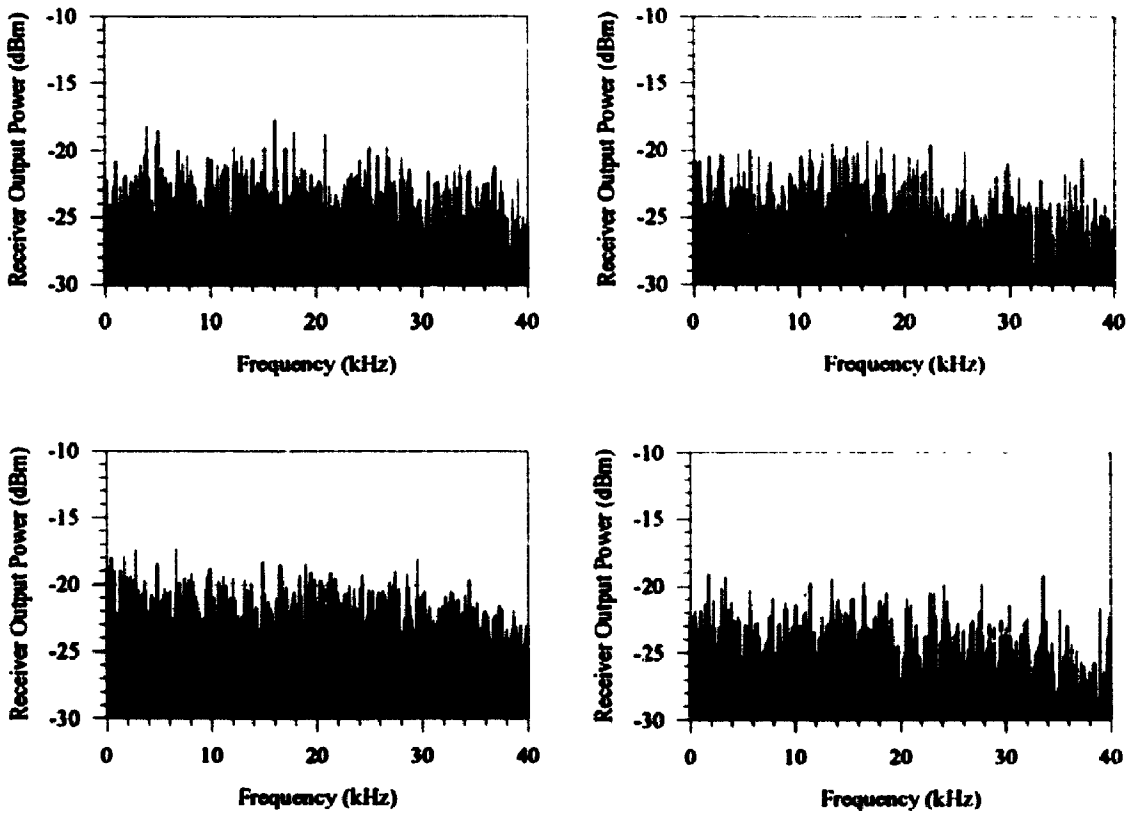


Figure 6.19 Receiver Noise Measurements at the Elginfield Site

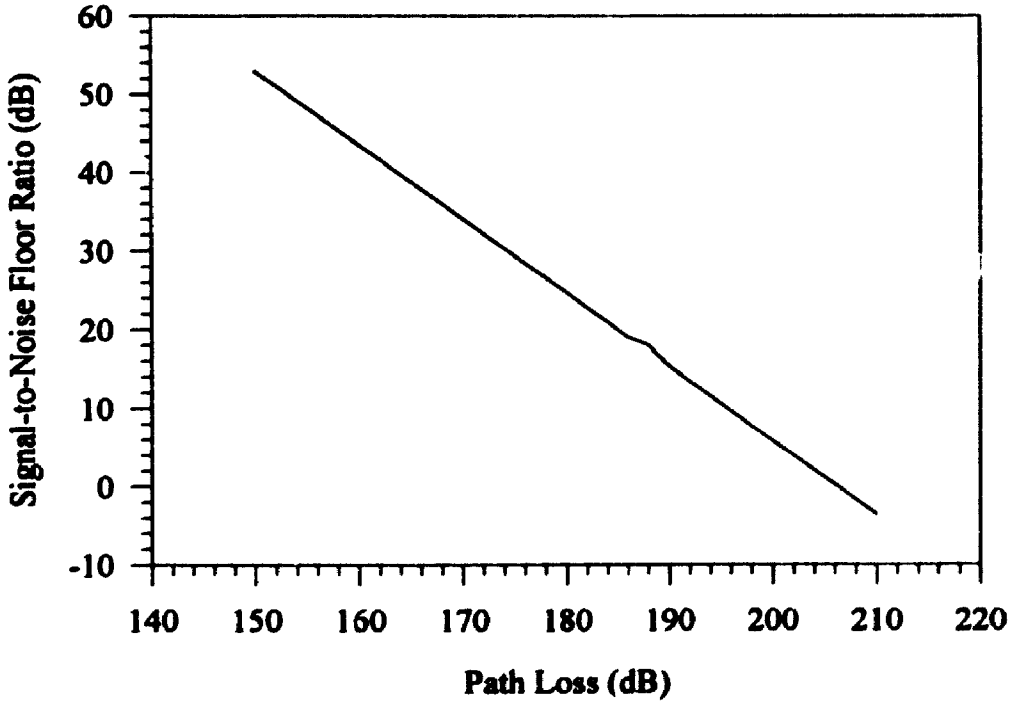


Figure 6.20 Receiver Measurement Reliability as a Function of Path Loss

Tests of receiver performance indicated that the receiver possessed good overall gain and frequency response characteristics.

In order to attach significance to the data collected by the system it was necessary to undertake a series of experiments aimed at calibrating the receiver for both path-loss and flight-time measurements. From these experiments, relationships were developed for transforming power and frequency measurements at the receiver output into path-loss and flight-time information. These experiments also served to verify the operation of the entire measurement system. Measurements of the channel characteristics and their interpretation will be the topic of the following chapter.

Chapter 7

Experimental Results

7.1 Introduction

In this chapter, data from the impulse response measurement system will be analyzed to assess the characteristics of the meteor burst channel. For this experiment, the measurement system was deployed with its transmitter located on the CRC complex in Ottawa and its receiver located at the interferometer receiving site at Elginfield, near London. Due to similarities in the link geometry of the CHIRP and interferometer systems, path loss and flight-time measurements from the two systems will be compared (where possible) to test the validity of the experimental results. Additionally, the multipath characteristics of the channel will be assessed in terms of the temporal extent of the power delay profile as well as the number of paths between the transmitter and receiver. Parameters extracted from the power delay profile will then be used in conjunction with the discrete multipath model to appraise the effects of multipath interference on the frequency response characteristics of the channel. As stated previously in Chapter 6, all CHIRP system data used in these analyses was obtained exclusively from the main receiver output channel.

7.2 Path Gain and Flight-Time Measurements

Path loss and flight-time measurements can be obtained from the raw data of the CHIRP system, through the application of equations 6.6 and 6.12 from the previous chapter and knowledge of the delay time setting, T_d, which for this experiment, was set to a value of 2.075 ms. Throughout this series of tests, data from the main receiver output

channel was collected using a modified version of the data collection program previously discussed in Chapter 6. Analysis of the propagation data was carried out using the program plftnp.c whose operation is outlined in the flow chart of Figure 7.1. As in the analysis of the calibration data, the power spectrum of the receiver output signal is computed after windowing the data with the Dolph-Chebyshev window having α=2.0. Unlike the analysis of the previous chapter, however, the spectral components are compared against a noise floor threshold prior to the application of the peak detection algorithm. Spectral components whose power levels are less than the noise floor threshold (-14 dBm) are ignored while those exceeding this level remain unaltered. At this point in the analysis, the peak detection algorithm of Chapter 5 is applied and the path gain and flight-time information associated with each of the spectral peaks is computed and written to the analysis output file. A source code listing of the data analysis program (plftnp.c) can be found in Appendix D.

Since several measurements can be recorded during the passage of a single burst, consecutive path gain/flight-time profiles may be combined to obtain a three-dimensional representation of the channel behavior over the lifetime of the ionized trail as shown in Figures 7.2 through 7.5. In each of these plots, the profile orientation has been adjusted so as to provide the best possible view of the multipath features. The values along the sweep number axis in these diagrams serve as an index through which the individual profiles may be identified. Because of the cyclic nature of the chirp transmissions, the time interval between consecutive indices along the sweep axis is 100 ms. From the graphs it can clearly be seen that the dominant path is usually accompanied by one or several weaker paths which firmly establishes the presence of multipath interference. In the majority of cases, the gains associated with these additional paths are on the order of 10 dB below that of the main path. However, in rare instances the gains associated these secondary

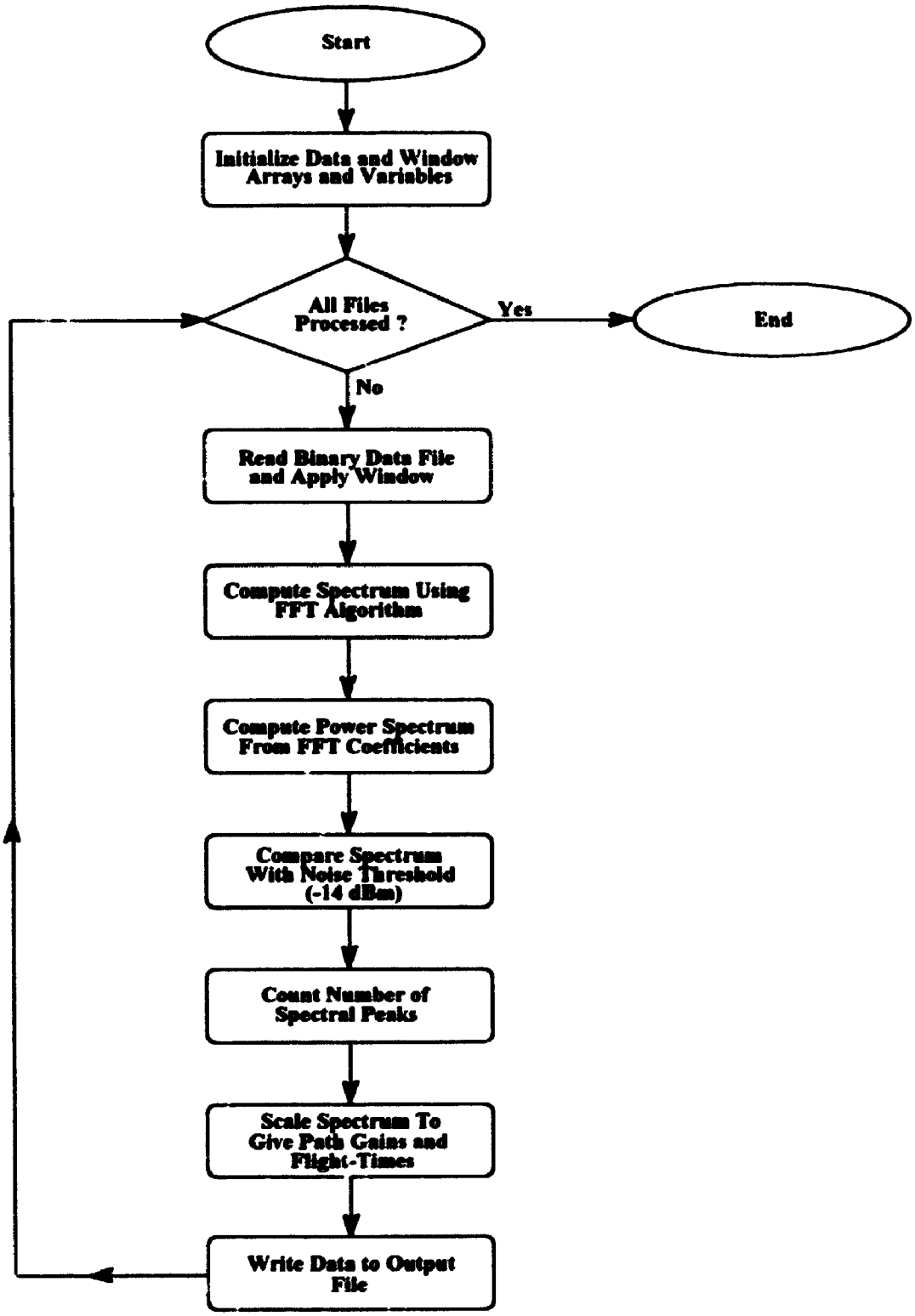


Figure 7.1 Flowchart of the Data Analysis Program

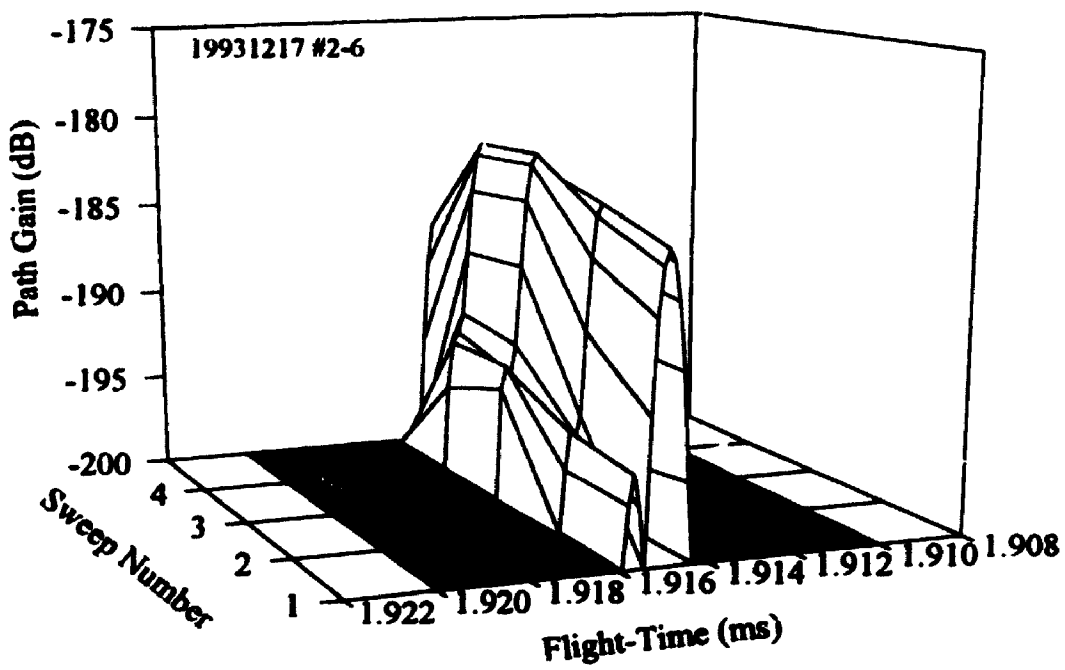


Figure 7.2 Gain/Flight-Time Profile for Burst #1
(Flight-Time Sample Spacing = 100 ns)

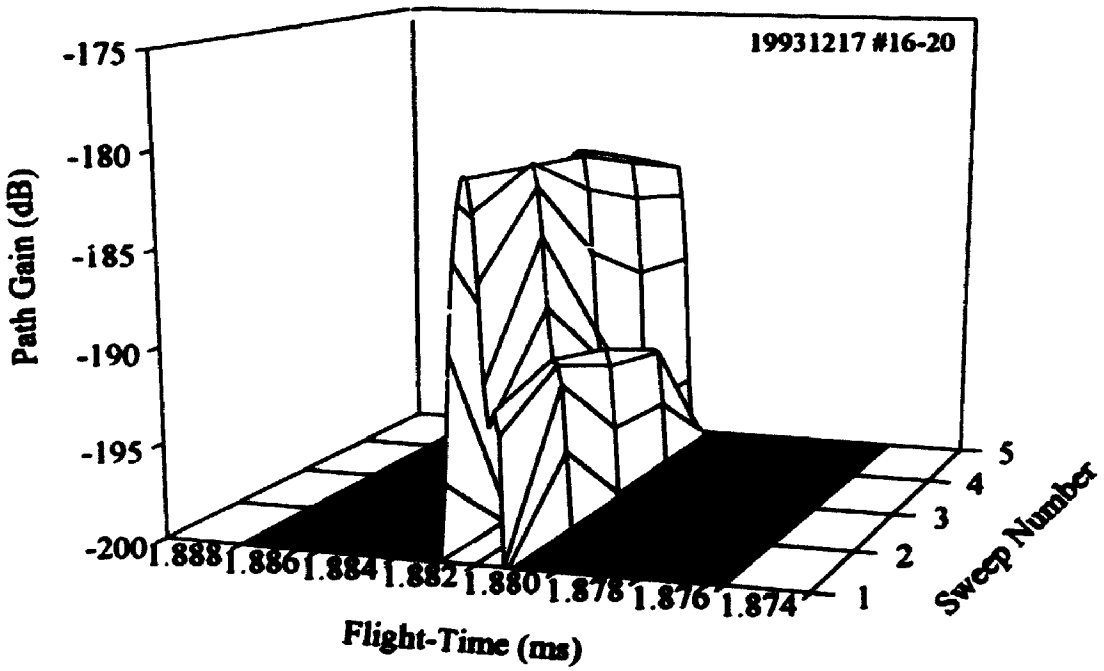


Figure 7.3 Gain/Flight-Time Profile for Burst #2
(Flight-Time Sample Spacing = 100 ns)

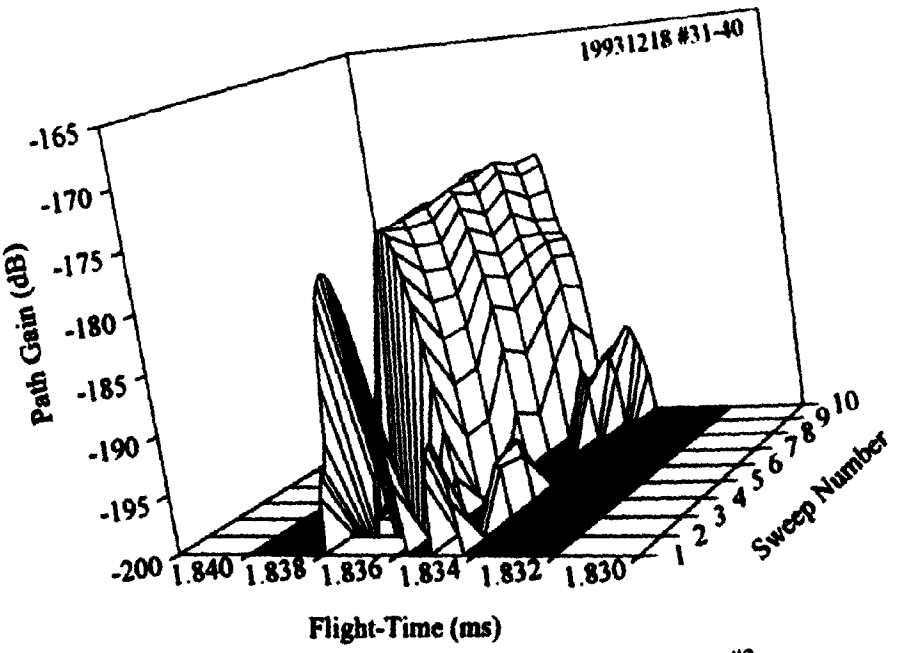


Figure 7.4 Gain/Flight-Time Profile for Burst #3
(Flight-Time Sample Spacing = 100 ns)

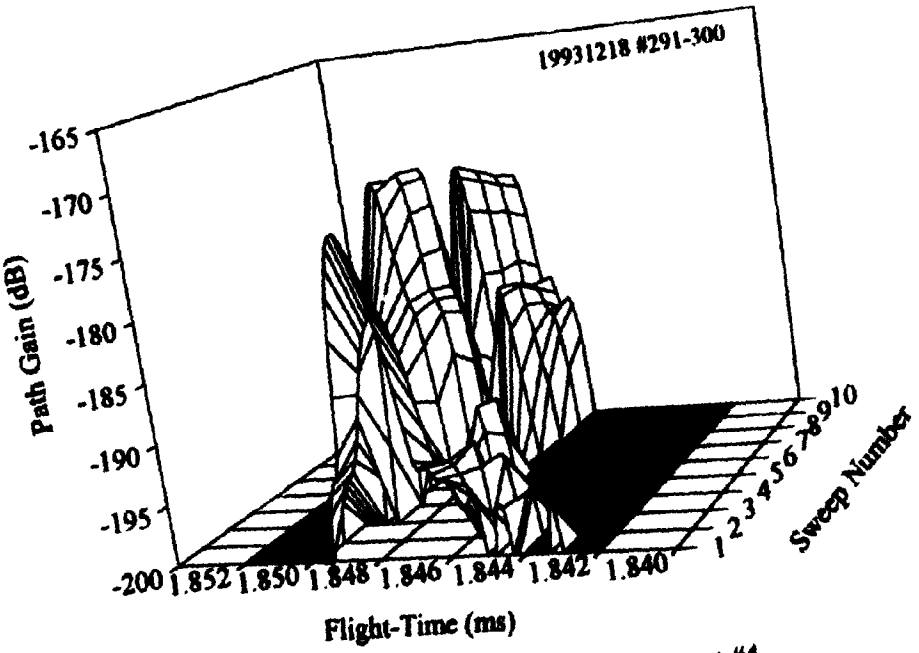


Figure 7.5 Gain/Flight-Time Profile for Burst #4
(Flight-Time Sample Spacing = 100 ns)

paths have been observed to differ by much smaller amounts typically 3 to 4 dB. A more detailed investigation of multipath properties follows in Section 7.3.

The measurement of flight-times associated with the dominant path of each profile (within an individual burst) revealed this quantity to be invariant throughout the duration of a particular burst. A summary of the flight-times and corresponding path-lengths for the dominant path of each of the four bursts is given in Table 7.1. Assuming a link geometry with a midpoint reflection at an altitude of 100 km and a separation distance of 500 km, gives a path-length of 539 km and a flight-time of:

$$T_p = \frac{539 \times 10^3}{3 \times 10^8} = 1.797 \quad (ms) \qquad (7.1)$$

which is consistent with the flight-time and path-length measurements given in the Table. These flight-time measurements are also consistent with those estimated from the interferometer data which were found to range from approximately 1.8 ms to 2.5 ms. Although not shown in the table, a manual examination of the flight-times revealed all values to be in excess of 1.8 ms with a maximum flight-time of roughly 2.01 ms. Path loss measurements for the bursts of Figures 7.2 through 7.5 yielded values that were consistent with the range of path losses (150 to 185 dB) calculated using the data from the interferometer system. This observation was found to be true of all loss values measured using the CHIRP system.

To further establish the validity of these measurements, burst recordings common to both the CHIRP and interferometer systems were scrutinized for similarities. Due to differences in power levels and antenna systems, meteor trails were not always visible to both instruments at the same time thereby resulting in only a few simultaneous recordings. Identification of these recordings was further complicated by time differences in the clocks

Table 7.1 Summary of Dominant Path Measurements

Burst #	Flight-Time (ms)	Path-Length (km)
1	1.915	574.5
2	1.882	564.6
3	1.837	551.1
4	1.848	554.4

used by the two systems. Prior to February 12, 1994, the CHIRP system clock was noted to be roughly 1.5 minutes in advance of the interferometer system clock. Taking this into account, Figures 7.6 through 7.9 show simultaneous path gain measurements for four separate bursts; the first three were recorded on January 26, 1994 and the fourth was recorded on January 30, 1994. In each of these recordings, the path gain measurements from the CHIRP system were observed to be roughly 3 to 5 dB greater than those obtained from the interferometer system. This discrepancy is attributed to 8.7 MHz difference in the operating frequencies of the two systems. For a frequency differential of this magnitude, the relative performance curve of Figure 2.2 predicts that a gain penalty on the order of 5 dB should be incurred by the measurement system operating at the higher frequency. An additional advantage of these simultaneous recordings is that the trail altitude can be deduced by combining direction information from the interferometer system with path length measurements from the CHIRP system Since echo direction (a,b,c), separation distance (d) and path length (P_L) are known, equation 3 5 can be solved in terms of the altitude (z_o) giving:

$$z_o = \frac{P_L^2 - d^2}{2\left(P_L\left(\frac{a^2}{c^2} + \frac{b^2}{c^2} + 1\right)^{1/2} - \frac{da}{c}\right)} \tag{7.2}$$

Applying equation (7.2) to the experimental data for the bursts shown in Figures 7.6 through 7.8 and assuming a separation distance of 500 km resulted in altitudes ranging from 92.5 km to 118 km all of which lie within the currently accepted altitude range of 80 to 120 km. A summary of parameters for these three bursts can be found in Table 7.2.

Table 7.2 Parameters and Altitudes for Bursts 5-7

Burst #	P_R (dBm)	P_t (km)	a	b	c	z_o (km)
5	-122	550.4	0.8100	0.2924	0.5084	92.5
6	-121	558.6	0.8186	0.0676	0.5704	118.4
7	-116	556.4	0.7705	-0.1773	0.6123	106.5

Unfortunately, in the case of the fourth burst (Figure 7.9) the interferometer measurements contained no direction information and consequently, an altitude calculation was not possible. Based on these findings and those discussed in the previous paragraph, it would appear that the CHIRP system measurements are reasonable in that they are consistent with previously known theoretical and empirical results.

7.3 Multipath Characterization

A visual inspection of Figures 7.2 through 7.5 clearly reveals the presence a dominant path accompanied by one or more secondary paths that are separated by relatively small flight-time differentials ranging from several hundreds of nanoseconds to several microseconds. The easiest way in which to characterize the multipath nature of these recordings is by simply counting the number of paths resolved by the analysis software. It should be noted that the true number of paths may actually be greater than this number due to the resolution limitations of the analysis program. For the propagation data collected on the first two days of a ten day recording period (December 17 and 18, 1993), the results shown in Table 7.3 were obtained. These results show that in 97% of

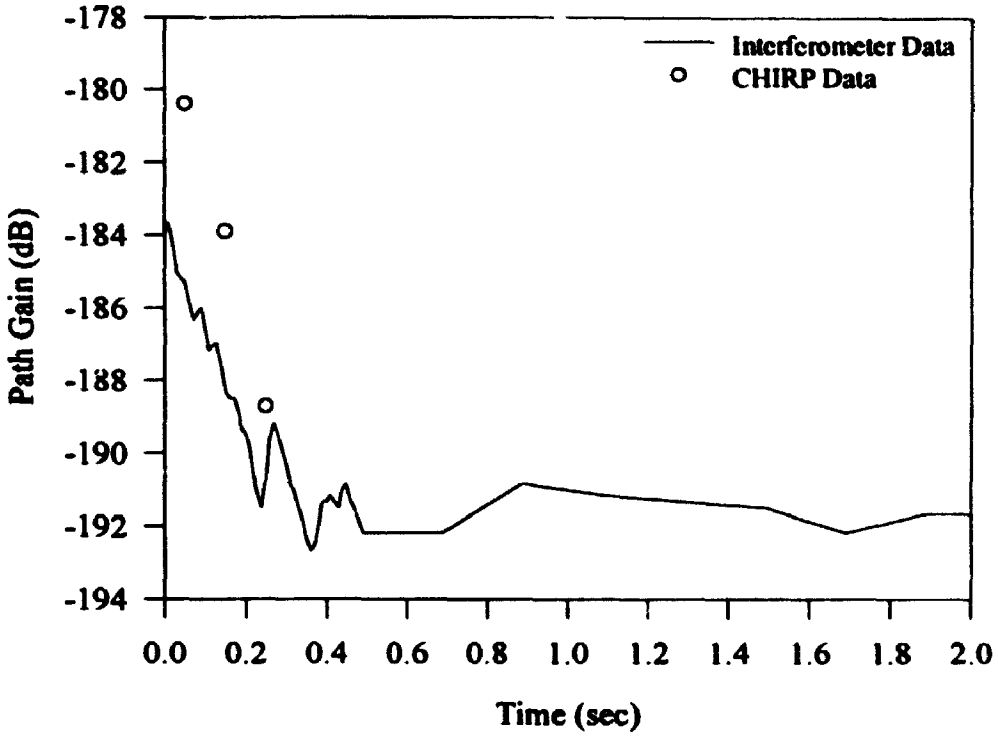


Figure 7.6 Simultaneous Recordings for Burst #5

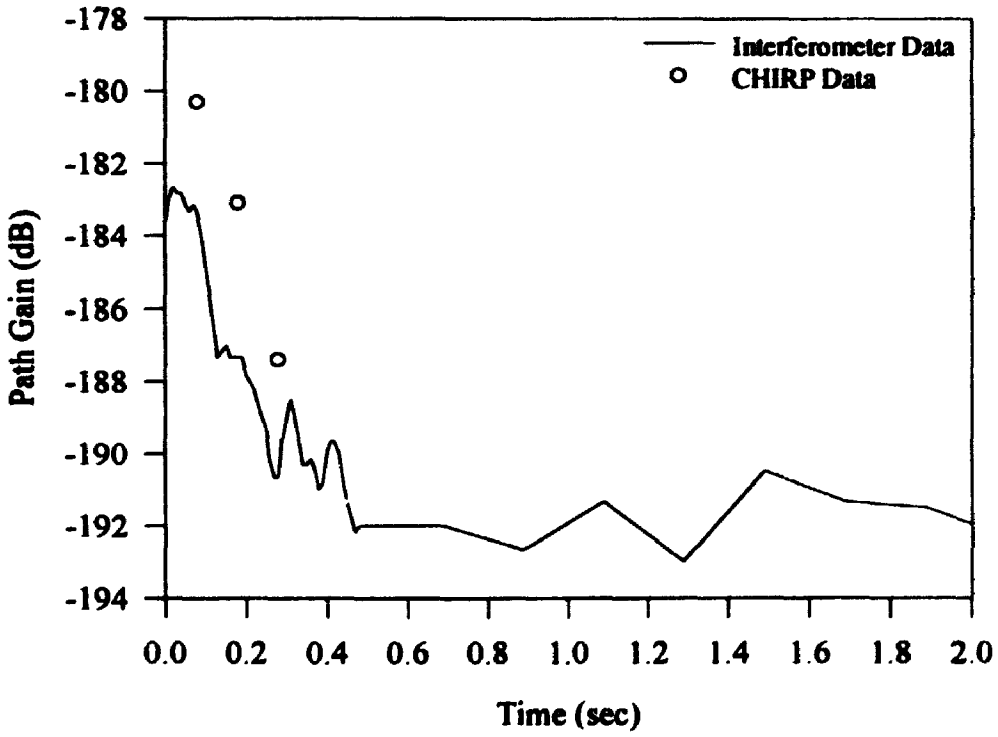


Figure 7.7 Simultaneous Recordings for Burst #6

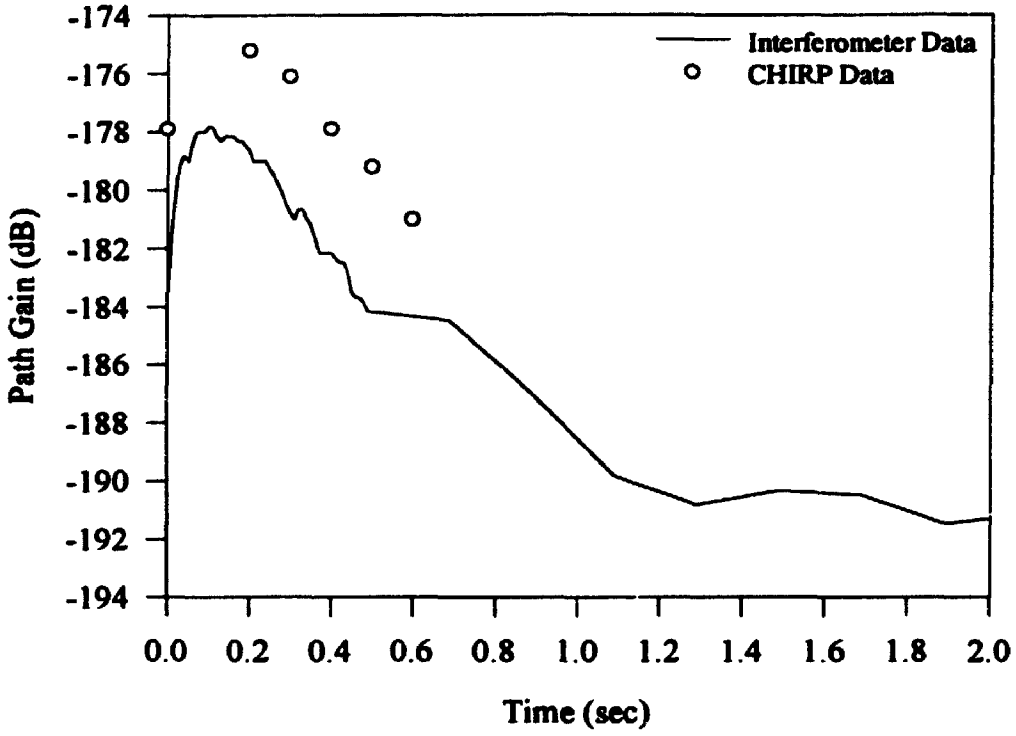


Figure 7.8 Simultaneous Recordings for Burst #7

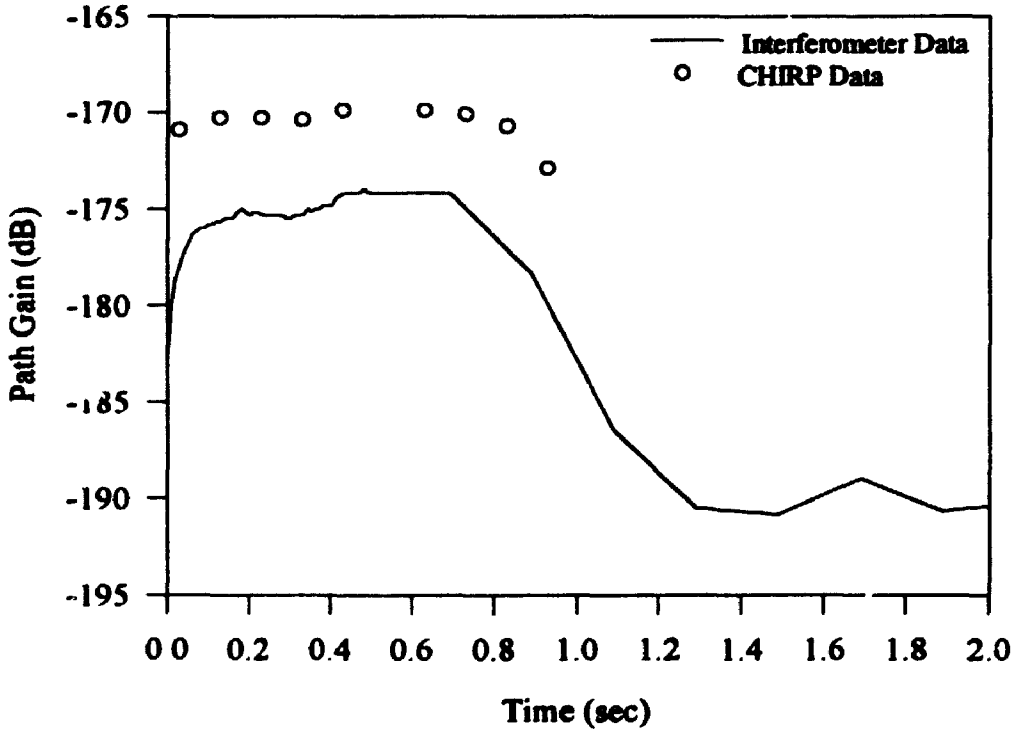


Figure 7.9 Simultaneous Recordings for Burst #9

the recordings investigated the channel conditions were such that there were no more than two paths between the transmitter and receiver which suggests that adverse effects due to multipath interference may be relatively infrequent.

Table 7.3 Summary of Path Count Measurements

Date	1 Path	2 Paths	3 Paths	4 Paths	5 Paths	Total
19931217	113	38	2	1	0	154
19931218	168	69	8	1	0	246
Combined	281	107	10	2	0	400

The multipath nature of a communications channel is frequently described in terms of its rms delay spread [25,33,34] which is in essence a measure of the width of a power delay profile. In the context of the work presented here, a power delay profile is the power spectrum of the receiver output signal and is intimately related to the gain/flight-time profiles shown in Figures 7.2 through 7.5. Studies investigating the effects of multipath interference on the performance of communication systems in both indoor and mobile environments have found the signaling rate to be primarily dependent on the delay spread of the channel [34,39,40]. The delay spread, S (seconds), is defined as the second central moment of a power delay profile and is computed according to the equation (Eriksson [25], Cox [33]):

$$S = \left(\frac{\sum_{k=1}^{N} (\tau_k - D)^2 P(\tau_k)}{\sum_{k=1}^{N} P(\tau_k)} \right)^{1/2} \qquad (7.3)$$

where; $P(\tau_k)$ is the power level at the k^{th} delay sampling time and D is the first moment of the profile, or average delay time (seconds), and is given by:

$$D = \frac{\sum_{k=1}^{N} \tau_k P(\tau_k)}{\sum_{k=1}^{N} P(\tau_k)} \quad . \tag{7.4}$$

Once representative values of the delay spread are known, the maximum transmission rate that can be supported by the channel can be roughly estimated using (Saleh [38]):

$$R_{Max} = \frac{1}{10\,S} \tag{7.5}$$

where R_{Max} is the maximum transmission rate (bps) and S is as previously defined.

In order to develop a "feel" for the factors influencing the magnitude of the delay spread measurement, consider the simplified example in which the power delay profile describing a two path link is given by:

$$P(\tau_k) = P_1\,\delta(\tau_k - \tau_1) + P_2\,\delta(\tau_k - \tau_2) \tag{7.6}$$

where; $P(\tau_k)$ is the power delay profile, P_1 and P_2 represent the power levels at delay times τ_1 and τ_2 respectively. Applying equations 7.3 and 7.4, to this particular profile results in the equation:

$$S^2 = \left(\frac{P_1}{P_T} - \frac{P_1^2}{P_T^2}\right)\tau_1^2 + \left(\frac{P_2}{P_T} - \frac{P_2^2}{P_T^2}\right)\tau_2^2 - 2\frac{P_1 P_2}{P_T^2}\tau_1\tau_2 \tag{7.7}$$

where the symbol P_T represents the total power which is simply the sum of the individual powers (P_1 and P_2) associated with each of the ray paths. By letting:

$$A = \frac{P_1}{P_T} - \frac{P_1^2}{P_T^2} \tag{7.8}$$

$$B = \frac{P_2}{P_T} - \frac{P_2^2}{P_T^2} \tag{7.9}$$

$$C = \frac{2P_1P_2}{P_T^2} \tag{7.10}$$

and

$$\tau_2 = \tau_1 + \delta \tag{7.11}$$

equation 7.7 can be simplified and rewritten as:

$$S^2 = B\left[\delta^2 + \frac{(2B\tau_1 - C\tau_1)}{B}\delta + \frac{(A+B-C)\tau_1^2}{B}\right] \tag{7.12}$$

where δ represents the time differential between the path flight-times, τ_1 and τ_2. Fortunately, the constant and linear terms in equation 7.12 both reduce to zero giving:

$$S = \sqrt{B}\,\delta\,. \tag{7.13}$$

By making use of one additional substitution, namely:

$$P_2 = kP_1 \tag{7.14}$$

it can be shown that the constants A and B are equal and can be expressed in terms of the ratio k as:

$$A = B = \frac{k}{(1+k)^2} \tag{7.15}$$

which allows equation 7.13 to be expressed in its final form as:

$$S = \frac{\sqrt{k}}{(1+k)}\delta. \tag{7.16}$$

From this equation, it can easily be seen that the delay spread is dependent only on the ratio of the ray powers, k, as well as the flight-time differential δ. Consequently, for a fixed k value the delay spread increases linearly with increases in the flight-time differential δ. While the preceding discussion is true for simple power delay profile assumed at the outset, it is also applicable to more realistic delay profiles such as those illustrated in Figure 7.10 (a-c). Using this measure to compare profiles (a) and (b) would result in profile (a) having a larger delay spread than profile (b) due to the larger total power contained in the region about the secondary peak in profile (a). A comparison of profiles (a) and (c) however, would result in profile (c) having the larger delay spread than (a)

simply due to the fact that the secondary peak in profile (c) is separated from the dominant peak by a larger flight-time differential than in profile (a).

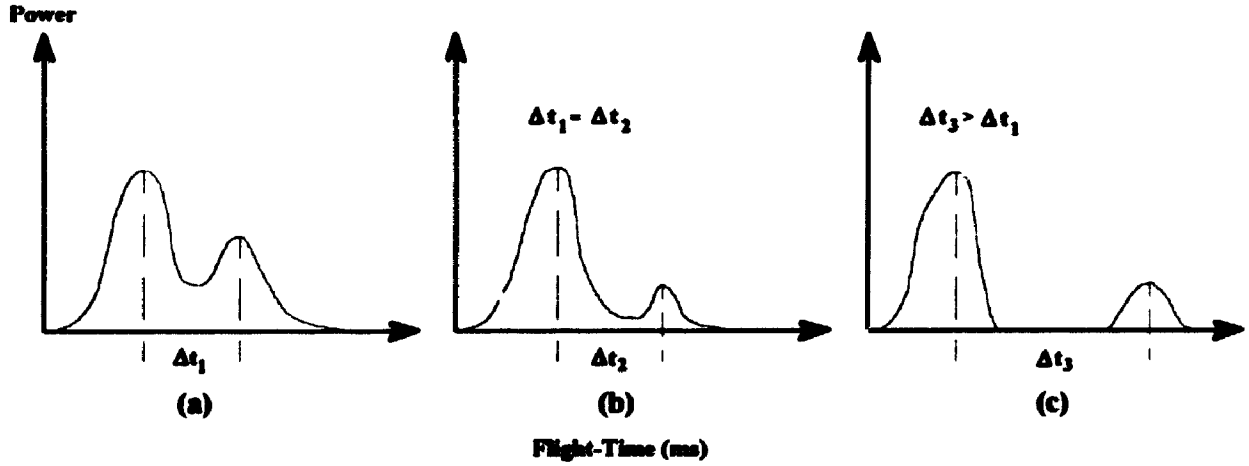


Figure 7.10 Power Delay Profiles and Delay Spread

In order to provide useful information about the multipath nature of the channel, measurements obtained through the application of equations 7.3 and 7.4 to the CHIRP system data must yield consistent results under similar conditions. For example, under those conditions where only a single path exists between transmitter and receiver, all delay spread measurements should be identical and independent of the absolute values of the received power and flight-times. To assess the consistency of this measurement a computer simulation approach was adopted in which the delay spread for a power delay profile representing a single path between transmitter and receiver was evaluated under conditions of varying power and times-of-flight. A flow chart outlining the simulation strategy is shown in Figure 7.11. At each power level selected, the frequency of the sinusoid representing the path between transmitter and receiver was increased in 10 small steps so that spreading induced by FFT leakage effects could be assessed. The total power and delay spread measurements at each of these frequencies were then averaged and written to the output data file. Based on the simulation results shown in Figure 7.12, direct application of the delay spread measurement according to equations 7.3 and 7.4 results in

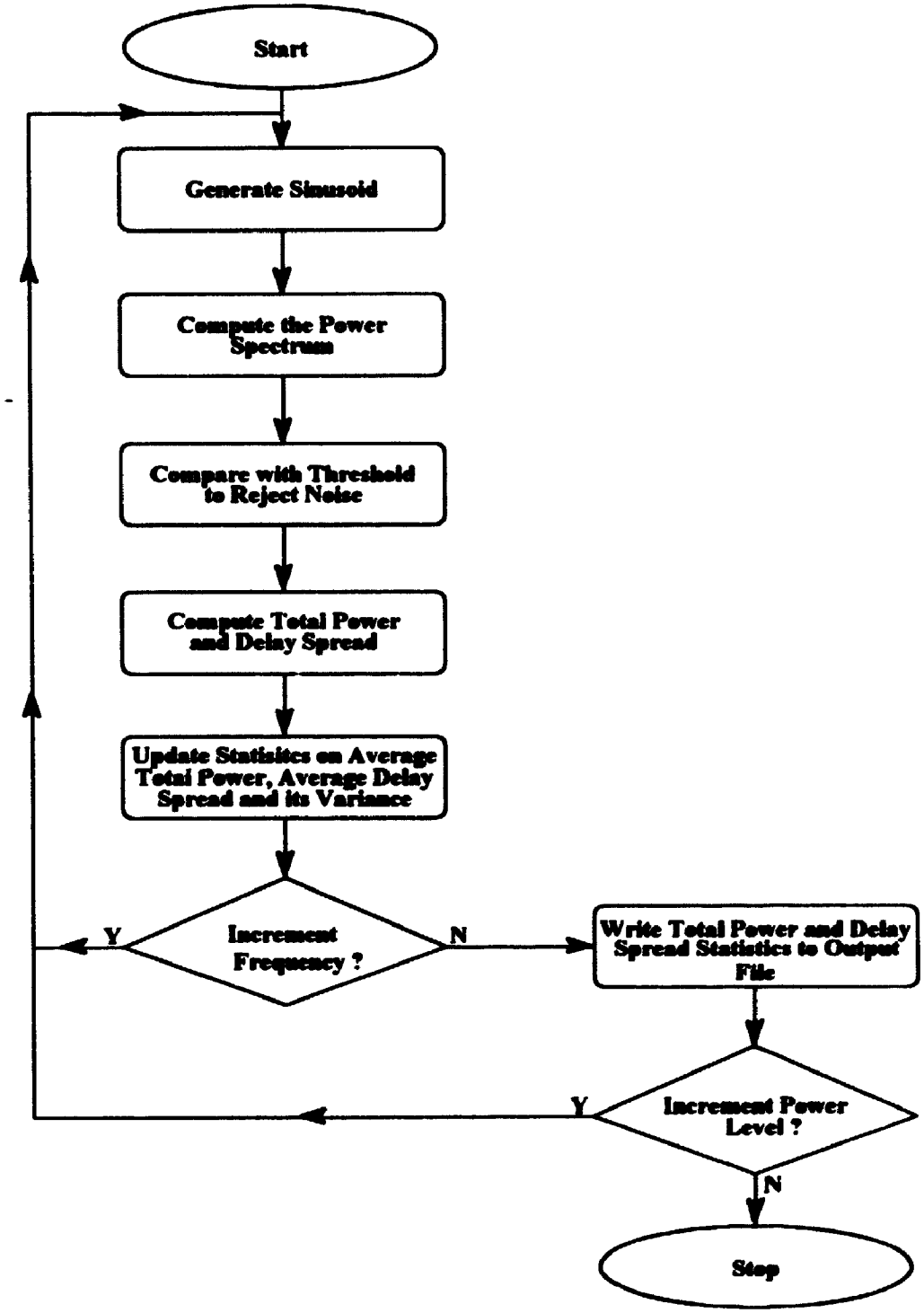


Figure 7.11 Flow Chart of the Delay Spread Simulation Program

delay spread measurements that are clearly dependent on the total power of the delay profile. As a result of this dependency, additional precautions must be taken to ensure that comparisons made on the basis of delay spread measurements reflect a fair assessment of the channel impairments arising from multipath effects.

One of the implications of equation 7.16 is that regardless of the k value or the total power, P_T, the entire family of curves intersect at a common point for which S=0 when the flight-time differential is zero. When this occurs, the power delay profile of equation 7.6 can be treated as having a single path with a total power equivalent to the sum of the individual powers P_1 and P_2. The simulation results for the single path case (see Figure 7.12) which are representative of this situation show that the delay spread measurements take on values other than S=0 and vary depending on the power level. The implication of this result is that delay spread measurements made on experimental data should be referenced to the simulated delay spreads having the same power level. That is:

$$S_E = S_M - S_S \tag{7.17}$$

where; S_E is the estimate of the delay spread due to the channel, S_M is the delay spread measured by the analysis software and S_s is the delay spread of the simulated signal possessing the same total power as the signal from which S_M was measured. Since it is likely that there is always some level of multipath interference present, albeit small, it is expected that all experimentally measured delay spreads should be greater than the simulated delay spread for the single path case having the same total power. For ease of comparison, a piecewise linear approximation of the simulation results of Figure 7.12 has been plotted along with the delay spread measurements for the London-Ottawa link made on December 17, 1993 as shown in Figure 7.13. In all but a few cases, the experimental delay spreads were found to be greater than the simulated values supporting the notion that there is always some degree of multipath interference being introduced by the channel. For those few instances where the delay spreads were found to lie below the simulated

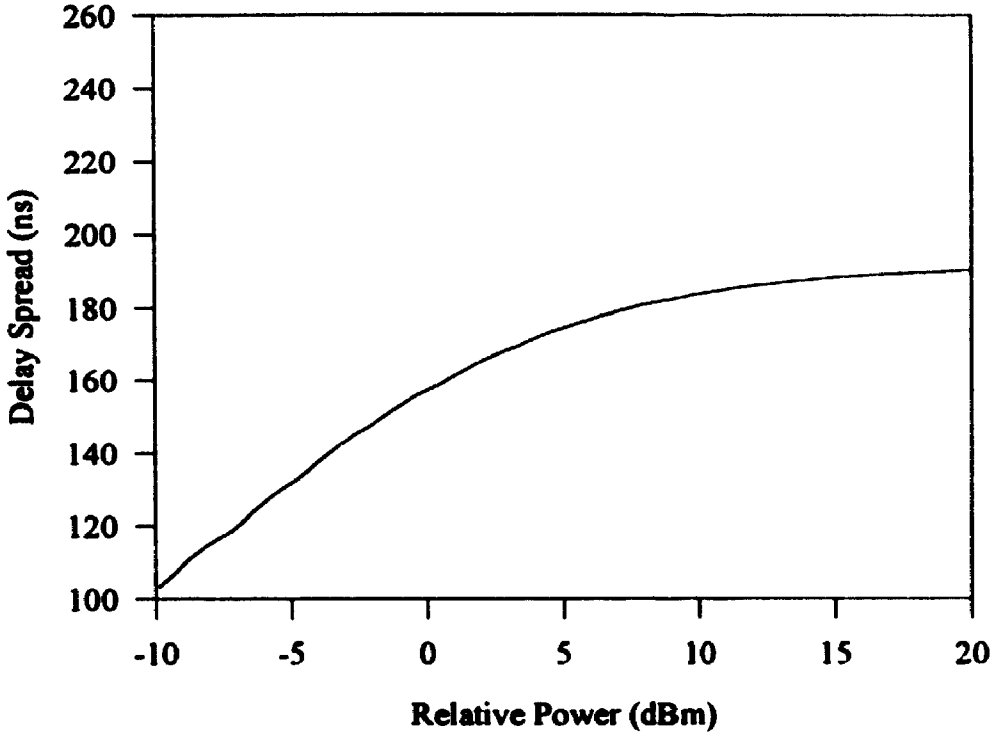


Figure 7.12 Simulated Delay Spreads (Single Path)

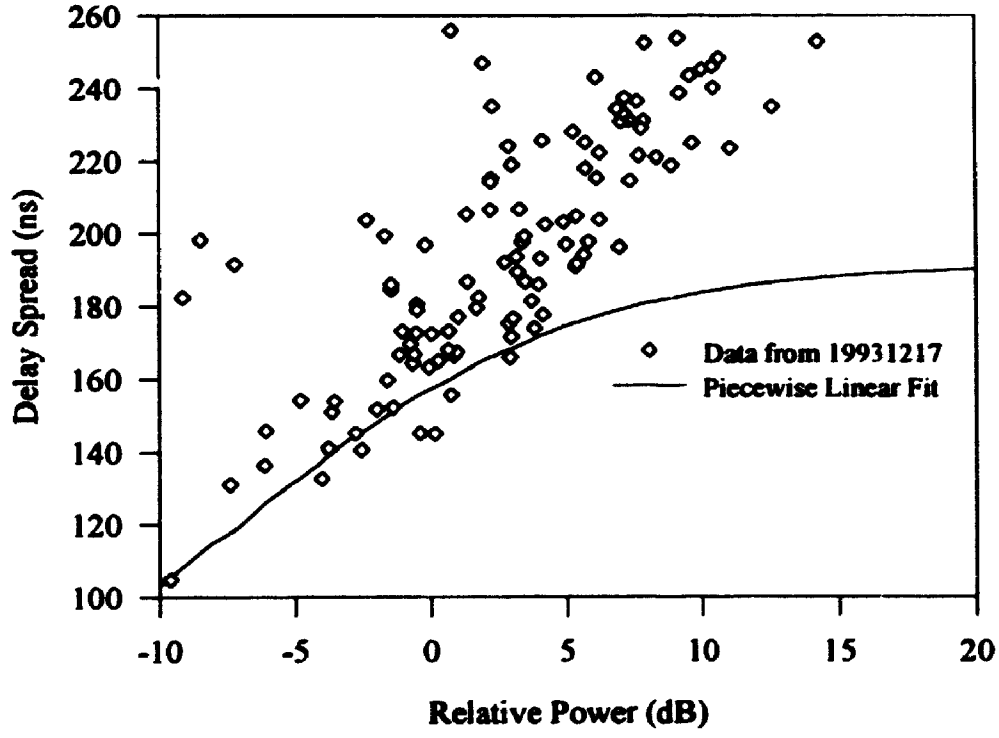


Figure 7.13 Measured Delay Spreads (December 17, 1993)

delay spread curve, the differences between the measured and simulated values were found to be on the order of 10.0 ns - a level of error that is acceptable given that the delay spreads associated with serious multipath interference are expected to be on the order of several hundred nanoseconds.

While the applicability of equation 7.17 appears to be intuitively obvious, further evidence was sought to defend its use. Using a simulation strategy similar to that outlined in Figure 7.11, a series of synthetic multipath signals were generated having 2 paths, different k values and total powers. The delay spreads of the resulting power delay profiles were then computed according to equation 7.17. Since the parameters of the simulation were known, it was also possible to compute the theoretical delay spreads of these profiles using equation 7.16 which could then be compared with the delay spread measurements returned by the simulation. The results of this experiment are shown in Figure 7.14 for three different k and four different power levels. Simulation results shown in the graph suggest that for power levels greater than 3 dBm, the delay spread measurements are independent of both the total power and k values and consequently the degree of spreading introduced by the channel can be estimated by means of equation 7.17. An additional benefit of these simulation results is that the relationship between the measured and theoretical delay spreads can be determined if desired. In this work however, no use will be made of such a relationship as the main interest is that of observing the variability of the delay spread measurements.

Statistics describing variations in the delay spread were computed from the measured values S_M after first subtracting the simulated delay spread (at the same power level) from the measured value as discussed in the previous paragraph. With the delay spread measurements (S_E) in hand, the delay spread distribution for the days of December 17-20, 1993 and January 26-27, 1994 was computed and is shown in Figure 7.15.

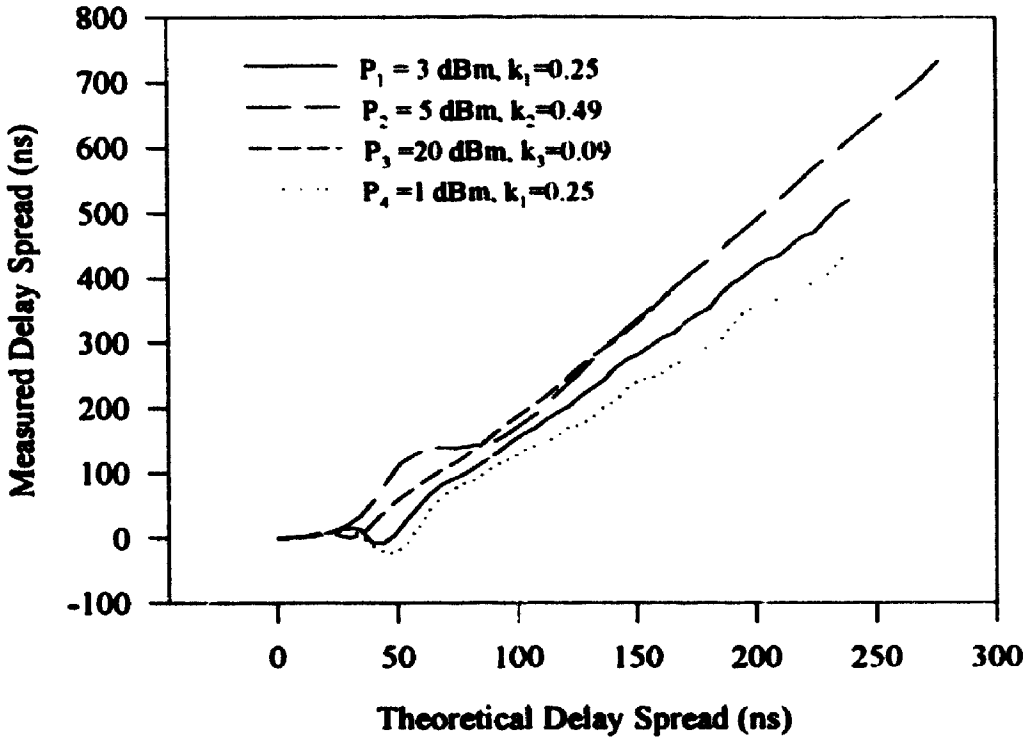


Figure 7.14 Measured and Theoretical Delay Spreads (from Simulation)

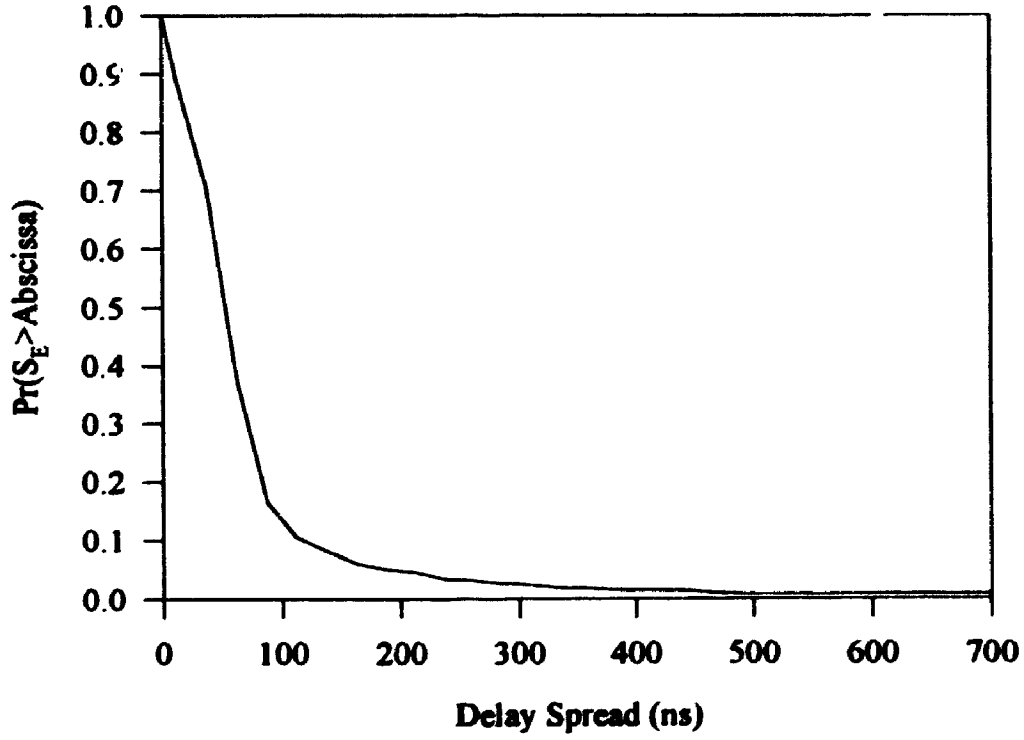


Figure 7.15 Delay Spread Statistics

According to these statistics, in 90% of the measurements the delay spread was found to be less than 100 ns and in 99% of all measurements the delay spread was found to be less than 400 ns. While the majority of all delay spread measurements tend to be less than 400 ns, in several rare instances delay spreads ranging from 1.0 to 7.0 μs have been observed. Statistics indicating similar findings have been reported by Eriksson [25]. In light of this evidence, it would appear that trail conditions conducive to large degrees of multipath interference are relatively infrequent.

7.4 The Impulse and Frequency Response Functions

A model frequently used to describe the multipath channel is the discrete multipath model for which the complex impulse response is given by:

$$h(t) = \sum_{k=1}^{N} \beta_k \, \delta(t - \tau_k) \, e^{j\phi_k} \tag{7.18}$$

where β_k represents the path attenuation, τ_k is the path delay and ϕ_k is the phase shift due to the scattering/reflection process, N represents the number of paths from transmitter to receiver, and k is the path number. Using this model, the frequency domain transfer function for the multipath channel can be obtained by computing the Fourier Transform of equation 7.18. That is:

$$H(\omega) = F\{h(t)\}. \tag{7.19}$$

where the $F\{\}$operator denotes the Fourier Transform operation. By making the appropriate substitutions, $H(\omega)$ becomes:

$$H(\omega) = \int_{-\infty}^{\infty} \left(\sum_{k=1}^{N} \beta_k \, \delta(t - \tau_k) e^{j\phi_k} \right) e^{-j\omega t} \, dt. \tag{7.20}$$

which, after simplification, can be written as:

$$H(\omega) = \sum_{k=1}^{N} \beta_k e^{j(\phi_k - \omega\tau_k)}. \tag{7.21}$$

When only a single path (N=1) exists between transmitter and receiver (i.e. no multipath interference) the amplitude and phase responses of the channel are:

$$|H(\omega)| = \beta_1 \tag{7.22}$$

and

$$\angle H(\omega) = \phi_1 - \omega\tau_1. \tag{7.23}$$

Under conditions such as these, the amplitude response is independent of the operating frequency and consequently the channel imposes no bandwidth limitations. Furthermore, since the amplitude response is constant and the phase response is a linear function of frequency, distortionless transmission can be achieved. When a second path between transmitter and receiver is introduced (N=2), the amplitude and phase responses for the channel become:

$$|H(\omega)| = \left[\beta_1^2 + \beta_2^2 + 2\beta_1\beta_2 \cos\left(\phi_1 - \phi_2 + \omega(\tau_2 - \tau_1)\right)\right]^{\frac{1}{2}} \tag{7.24}$$

and

$$\angle H(\omega) = \tan^{-1}\left(\frac{\beta_1 \sin\eta_1 + \beta_2 \sin\eta_2}{\beta_1 \cos\eta_1 + \beta_2 \cos\eta_2}\right) \tag{7.25}$$

where; η_1 and η_2 are given by the expressions:

$$\eta_1 = \phi_1 - \omega\tau_1 \tag{7.26}$$

and

$$\eta_2 = \phi_2 - \omega\tau_2 \tag{7.27}$$

Under these conditions the channel gain function contains periodic oscillations, the frequency of which is proportional to the time differential between the propagation times (τ_2-τ_1) of the two paths. The magnitude of these oscillations is related to the product of the gains associated with each of the two paths. Depending on the magnitude of these fades, errors in transmission may occur. In the general case in which potentially N paths exist between transmitter and receiver, the amplitude and phase responses of the channel can be computed using:

$$|H(\omega)| = \left[\sum_{k=1}^{N} \beta_k^2 + 2 \sum_{\substack{k=1 \\ l=1 \\ k \neq l}}^{N} \beta_k \beta_l \cos(\eta_k - \eta_l) \right]^{\frac{1}{2}} \tag{7.28}$$

and

$$\angle H(\omega) = \tan^{-1} \left(\frac{\sum_{k=1}^{N} \beta_k \sin \eta_k}{\sum_{k=1}^{N} \beta_k \cos \eta_k} \right) \tag{7.29}$$

where all symbols are as previously defined.

Transfer function calculations based on the discrete multipath model of equation 7.18 require that the model parameters β_k, τ_k, and ϕ_k be extracted from the experimental data. Fortunately, the first two parameters are readily available from the CHIRP measurements and the remaining parameter, ϕ_k, is assumed to be a random variable uniformly distributed on the interval $[0,2\pi)$. The justification for this assumption is that phase shifts resulting from the reflection/scattering process are likely to take on a particular phase shift with equal likelihood thereby resulting in a phase angle that is uniformly distributed. For presentation purposes the frequency response profiles shown in Figures 7.16 through 7.19 have been normalized to the gain at the channel center which corresponds to a frequency of 0 Hz. Since the theoretical resolution of the impulse response measurement system is 400 ns, the corresponding bandwidth over which these calculations are representative of the true channel behavior is 2.5 MHz, or equivalently, a frequency range of -1.25 MHz to 1.25 MHz about the channel center.

Using the multipath measurements from the path gain flight-time profiles (Figures 7.2 through 7.5), the frequency response profiles shown in Figures 7.16 through 7.19 were obtained. In the first two instances, the gains of the secondary rays were noted to be approximately 10 dB below that of the main path and consequently the magnitude of the variation observed in the frequency domain was observed to be on the order of 5 dB. Similar results can be observed in Figure 7.18. In this case however, a large null occurs in the transfer function profile due to what appears to be a large fade in signal strength. The magnitude of this fade however is constant over the entire frequency band of interest. Figure 7.19 illustrates the range of frequency selective fading that can occur under multipath conditions. Frequency domain variations on the order of several dB (excluding broadband fades) typical of those observed in the previous figures dominate the profile, however, at the time corresponding to the 6th sweep two paths with nearly identical gains have evolved, producing nulls on the order 15 to 20 dB to appear in the profile. It is likely that during fades of this order of magnitude a relatively large increase in transmission error would be observed. It should be noted that at the start of this profile, the presence of 4 paths has only a minor effect on the degree of frequency selective fading due to the relatively small gains associated with these secondary paths. Given that in the majority of cases only a single path exists between transmitter and receiver, it is unlikely that multipath effects will seriously limit the data rate that can be supported by the meteor burst channel.

7.5 Doppler Effects

The influence of Doppler shift on measurements from the CHIRP system is such that it will serve to cause small variations in the frequency(ies) present in the receiver output signal. Since the receiver output frequency and the flight-times are intimately related, the presence of a Doppler shift should manifest itself (if sufficiently large) in the form of flight-time variations over the period of a particular burst. Because of

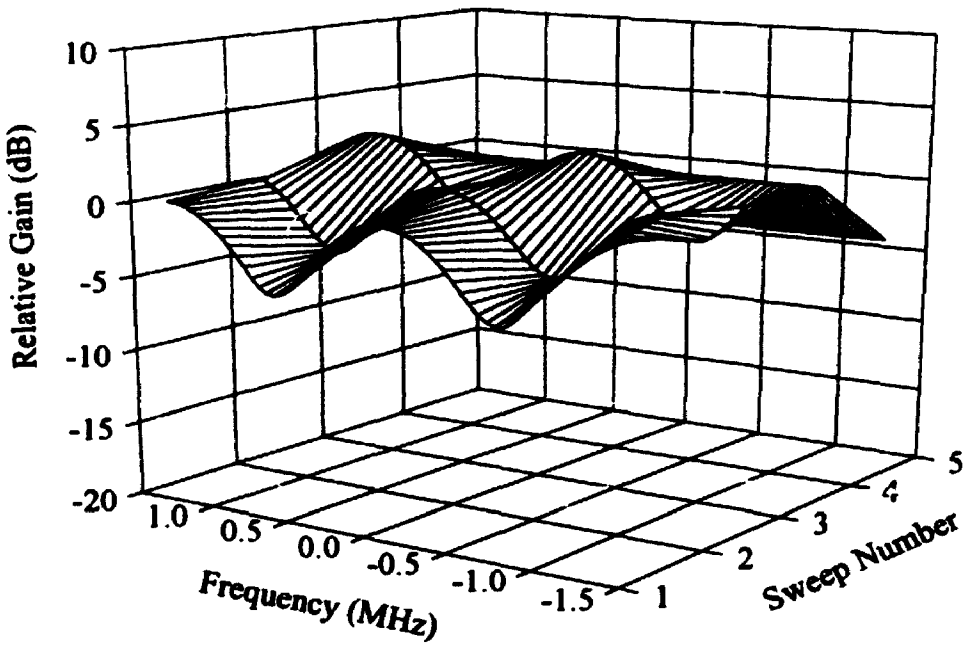


Figure 7.16 Transfer Function Profile for Burst #1
(Frequency Increment = 50 kHz)

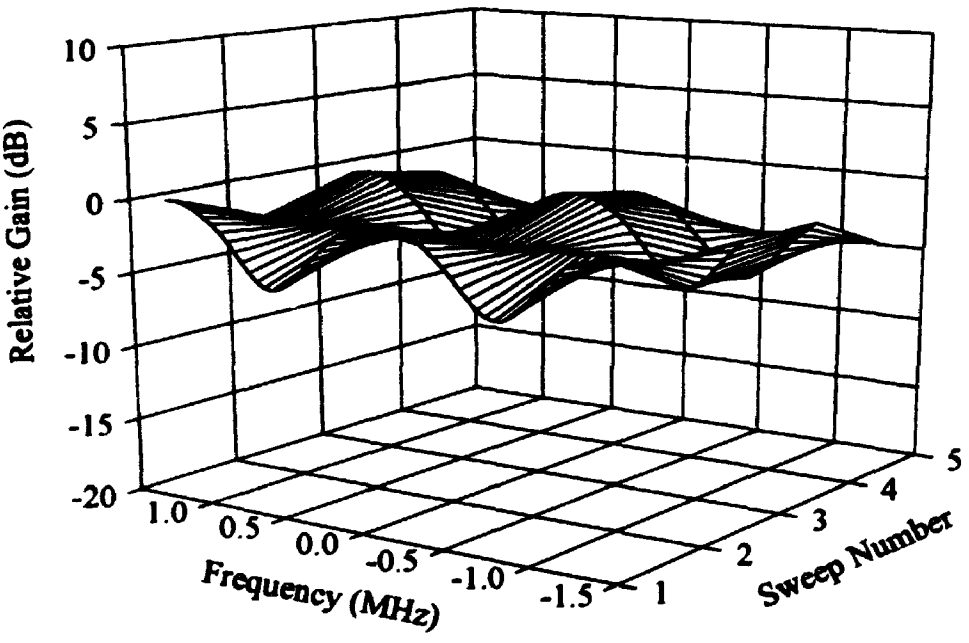


Figure 7.17 Transfer Function Profile for Burst #2
(Frequency Increment = 50 kHz)

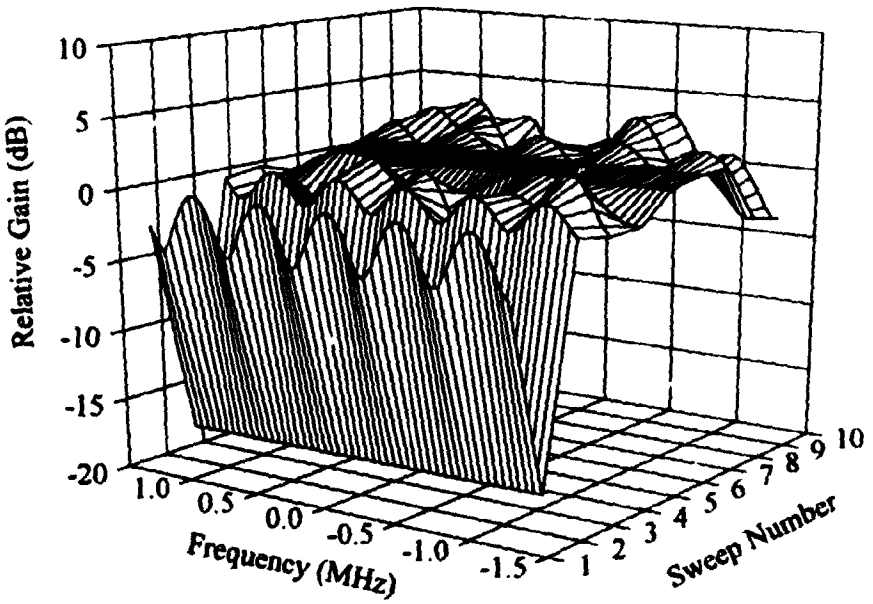


Figure 7.18 Transfer Function Profile for Burst #3
(Frequency Increment = 50 kHz)

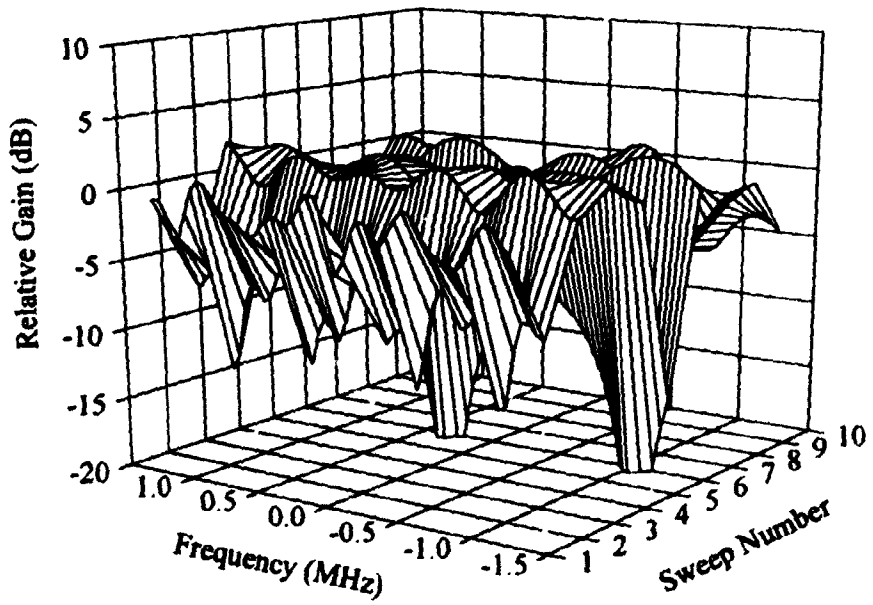


Figure 7.19 Transfer Function Profile for Burst #4
(Frequency Increment = 50 kHz)

measurement limitations, Doppler shifts of the magnitude reported by Eriksson [25] (±5 Hz) cannot be reliably detected. Observations of flight-time variation during those bursts for which multiple recordings were obtained revealed little variation in the flight-time and this suggests that the magnitude of the shift (if any) was beyond the measurement capabilities of this system.

7.6 Summary

Path loss and flight-time measurements obtained from the impulse response measurement system were found to be consistent with similar measurements derived from the interferometer data. In those instances where simultaneous recordings were available, path loss measurements from the CHIRP system were found to be 3 to 5 dB better than those computed from the Elginfield data. The differences between these loss values is reasonable and is attributed to the 8 MHz difference in the operating frequencies of the two measurement systems. Additionally, flight-time measurements from both systems were found to agree.

Measurements aimed at determining channel impairments revealed that in the majority of instances, the degree of multipath interference appears to be relatively small. This conclusion is supported by experimental results which show that the number of instances in which two or more paths occur are relatively infrequent relative to the number of instances in which only a single path was resolved. It should be noted that in those cases where only a single path was resolved, the potential exists for unresolved rays to exist in close proximity to the dominant ray. To further identify and compare the effects of multipath interference, the delay spreads of the power delay profiles were computed. The results of this analysis showed that in the majority of cases the delay spreads for most bursts is relatively small ranging from 10's of nanoseconds to as much 400 ns. Only in rare instances were large delay spreads on the order of several microseconds observed.

Using the discrete multipath model, the frequency domain transfer functions for several bursts were computed. In all instances, small nulls several dB in depth were observed. Under severe multipath conditions, however, the depths of these nulls were found to increase to between 15 and 20 dB. Since the multipath conditions responsible for such events appear to be relatively infrequent, it is unlikely that these impairments pose a significant threat to the success of digital communications systems. It should be noted that in the four bursts examined in detail, the channel characteristics for the most part appear to be stationary throughout the duration of the burst. A complete summary of results and their significance are the subject of the following chapter.

Chapter 8

Summary, Conclusions and Recommendations

The propagation losses encountered on the London-Ottawa link were determined experimentally to range between 185 and 151 dB - a result which was found to lie within the range of values calculated for a 500 km path. Although these values are reasonable, it should be noted that their range is artificially limited by the equipment sensitivity. A statistical treatment of these measurements found that on average, the path loss was 181 dB and that in 90% of all observations the path loss was found to range between 185 and 170 dB. In practical systems, the effects of these losses could be reduced through the use of high-gain directional antennas, increased transmitter power and to a certain extent, by selecting as low an operating frequency as possible for the application under consideration. It should also be noted that the range of losses measured for the London-Ottawa link are expected to be typical of those experienced on other links spanning the same distance.

Further analysis of the interferometer data showed the meteor arrival rate to be variable throughout the day with peak meteor activity occurring in the early morning hours, around 6:00 local time and the lowest meteor activity occurring in the evening, around 18:00 hours. For the months of March through December 1991, the mean hourly rate was estimated to range from 14 to 18 meteors per hour which equates to between 336 to 408 usable meteors on a daily basis. In addition to arrival rate variations, diurnal trends were also observed in the hotspot locations which were noted to form on either side of the great circle path between the transmitter and receiver. To capitalize on these locations,

directional antennas capable of selectively illuminating either side of the GCP are clearly advantageous. Due to the range of spatial locations associated with these trails, the flight-time between transmitter and receiver is variable and may, depending on the magnitude of the variations, affect the manner in which link synchronization is achieved. The flight-time statistics for this particular link showed that in 90% of all observations the flight-times varied from 1.8 ms to 2.5 ms with an average flight-time of 2.1 ms and a standard deviation of 0.5 ms.

In the absence of a suitable engineering model describing the arrival rate and duration of both overdense and underdense trails, estimates for the average waiting time and burst duration were deduced from experimental results. Using recordings from the interferometer system, the mean time between arrivals and average transmission time were found to be 3.5 minutes and 0.5 seconds respectively. By assuming these values and a transmission rate of 4800 bps, the average throughput for the current London-Ottawa link was estimated to be 11.4 bps. Although a throughput of this magnitude is small, on a daily basis a total of 123.1 kbytes can be transferred over the link. In applications such as remote monitoring where perhaps, a maximum of 10 measurements of some slowly varying quantity are taken per hour, the throughput of this meteor burst link is more than sufficient to satisfy the telecommunications needs for this particular service. An additional application well suited to meteor burst communication is that of long distance vehicle tracking, where the information to be exchanged consists only of the vehicle location (as determined by LORAN or GPS) and identification. Throughput increases are achievable and can be realized by increasing the transmitter power and signaling rate.

To assess the effects of multipath interference, a system was designed and constructed to permit measurements of the channel impulse response, path loss and flight-times. The path loss and flight-time measurements obtained from this system were found

to be consistent with those estimated from the interferometer data. Furthermore, by combining direction measurements from the interferometer system with simultaneous path length measurements from the CHIRP system, the trail altitudes could be deduced. For the three simultaneous recordings investigated, these measurements yielded altitudes of 92.5, 106.5 and 118.4 km which are within the accepted altitude range of 80 to 120 km. A multipath characterization of the impulse response measurements revealed that in the majority of instances only a single path exists between the transmitter and receiver. Delay spread measurements confirmed the presence of multipath interference and showed that for 90% of all trails, the delay spread is less than 100 ns. Using parameters extracted from the impulse response measurements, the frequency response profiles for several bursts were computed. In three of the profiles investigated, the effects of multipath interference were relatively small, and produced gain variations on the order of 3 to 5 dB within the 2.5 MHz passband. In the fourth profile however, the effects of multipath interference were more pronounced and deep nulls on the order of 20 dB were observed. Given that the majority of measurements indicated only a single path, it is unlikely that the channel imposes any serious limitations on the available bandwidth and consequently it is likely that signaling rates ranging from several hundred kbps to perhaps 1 Mbps can be used.

As a result of the experimental work undertaken in this thesis much new information regarding the variability of channel conditions has been obtained. Among the contributions of this work are the compilation of statistics for the path loss, flight-time, arrival rate, interarrival time and burst duration, all of which are vital to the design and implementation of practical meteor burst communication systems. Although it is likely that statistics for these parameters have been compiled by others, publications detailing such findings appear to be rare - most probably due to the commercial value attached to this information. In addition to the statistics compiled for the aforementioned parameters, a major contribution of this work has been in the characterization of multipath effects

through the measurement of the channel impulse response function. As formerly mentioned, little experimental work in this area has been undertaken with the notable exception of that conducted by Weitzen [24] and Eriksson [25]. Although the findings of this work confirm those of Eriksson, the impulse and frequency response measurements resulting from this study are unique in that the mutability of these functions can be observed over time. Observations to date suggest that the degree of multipath interference (i.e. the magnitudes of the delay spreads) encountered in the majority of instances is relatively small and imposes no serious limitation on the signaling rate supported by the channel. Based on the combined experimental work presented in this thesis, it is certain that on a daily basis hundreds of suitably ionized trails capable of supporting relatively high rate digital communication are produced. Moreover, these chance events can be utilized using existing equipment and techniques to provide long distance, cost effective communication for low volume applications such as remote monitoring and vehicle tracking.

In light of the experiences gained through the operation of the CHIRP measurement system, there are several modifications which if implemented would improve the overall performance of the instrument. Once such modification is to reduce the bandwidth of the main receiver output channel from 40 kHz to approximately 20 kHz. The effect of such a change would permit the use of a much lower sampling rate, and consequently reduce the amount of data recorded per sweep. In conjunction with this alteration, changes to the data collection program should be made to develop a more effective scheme by which data is transferred from the memory buffers to disk, thereby reducing the amount of information lost due to insufficient space. In the event that studies requiring simultaneous observations are desired, a method should be devised whereby the two systems may be synchronized at periodic intervals. Alternatively, modifications to

both systems may be considered whereby the interferometer and CHIRP recordings are initiated by a common trigger.

Future studies, in which this work is continued should begin with a series of measurements aimed at demonstrating that the multipath effects observed on the London-Ottawa link are typical of those encountered on other links. Using the results of these studies, an appropriate model for the meteor burst channel can then be developed. With such a model in hand, the performance of various modulation and coding strategies can then be evaluated in a systematic manner to determine those schemes that are best suited for this mode of communication. Studies such as these can then serve as the foundation from which improved protocols and networking strategies for meteor burst communication systems can be developed.

Appendix A

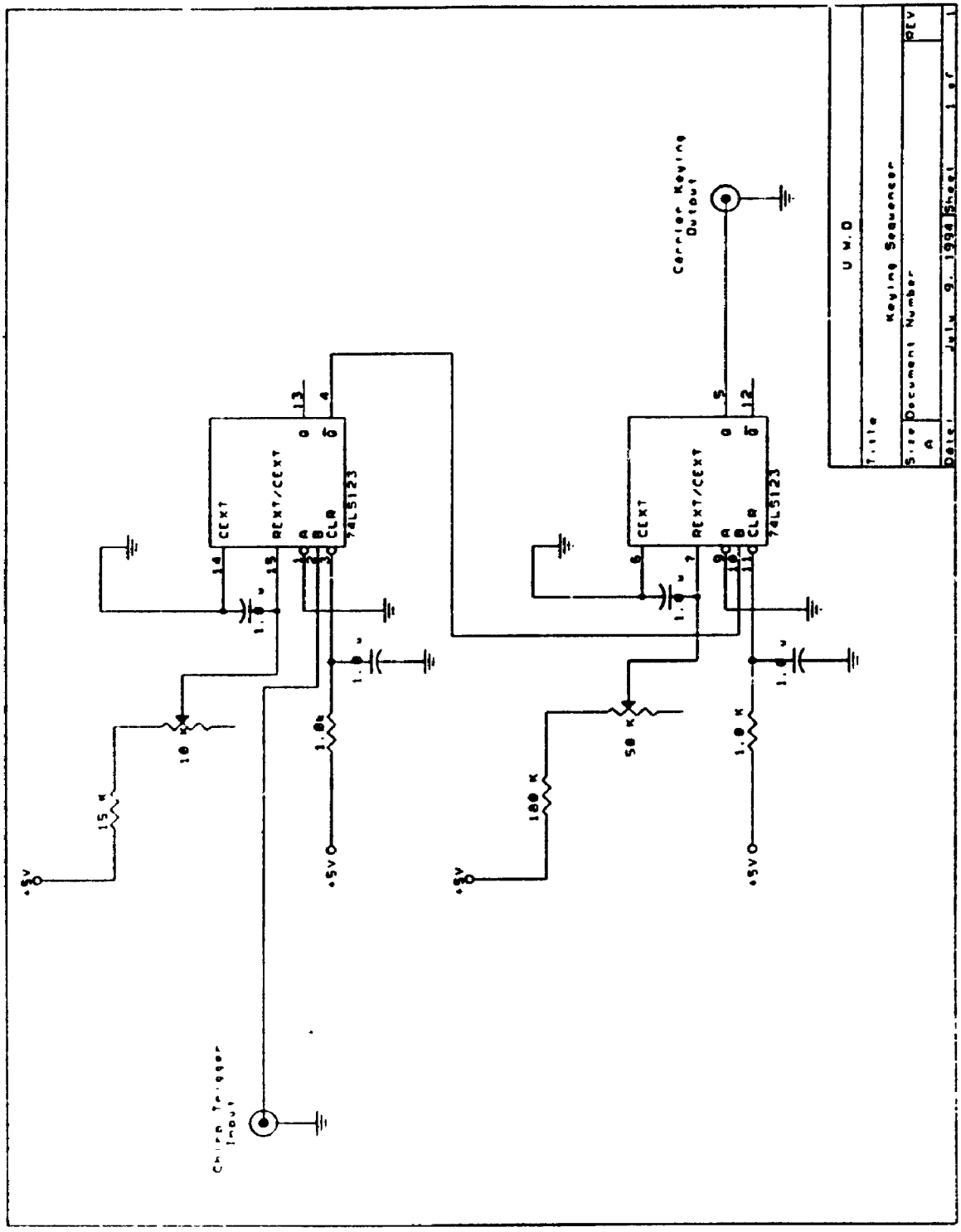
Appended Material for Chapter 4

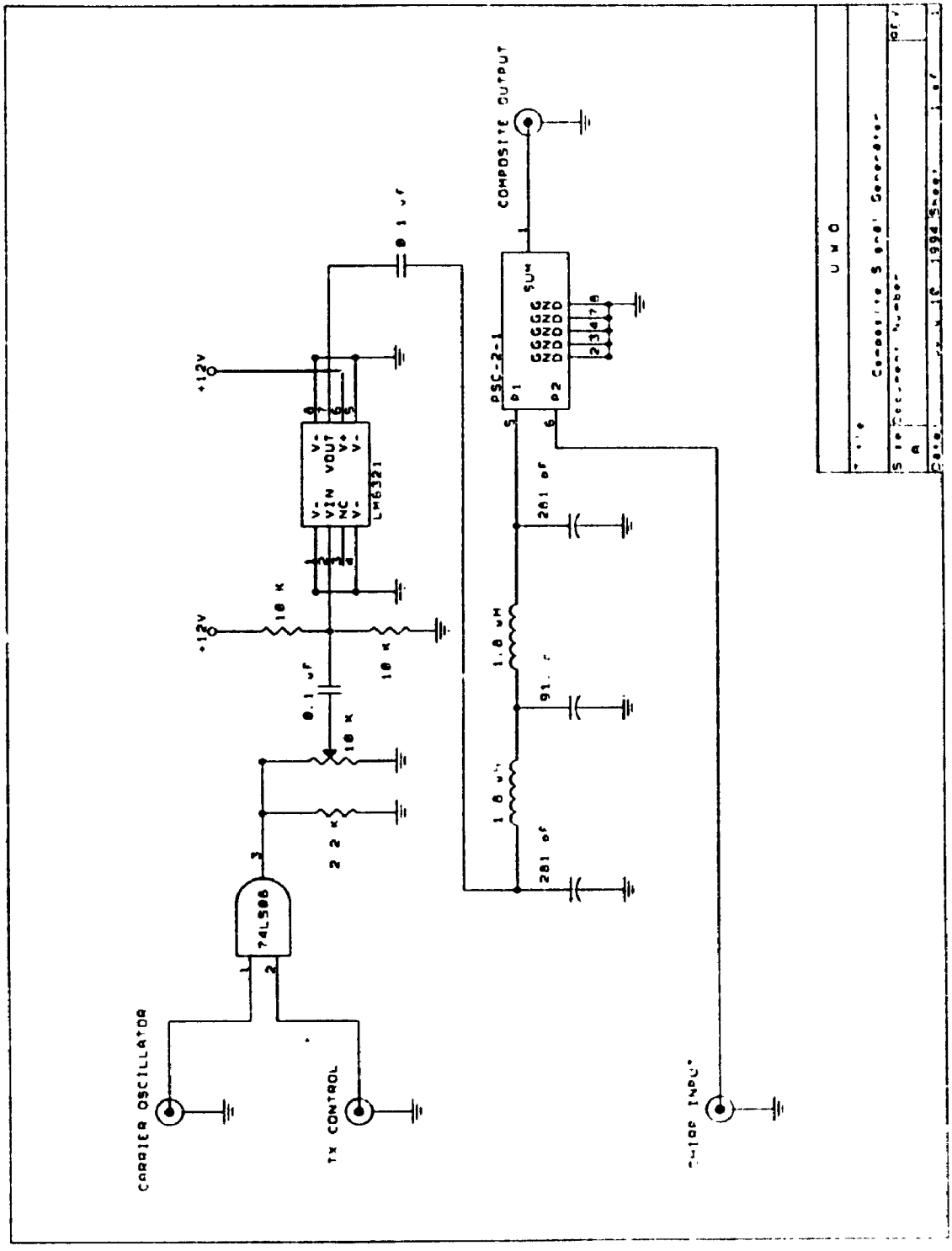
```
/*************************************************************************/
/*                                                                       */
/*    File Name: chirp1.c                                                */
/*    Created: 19930809                                                  */
/*    Created By: Kerry Ellis                                            */
/*                                                                       */
/*    This program generates the sample values for a digitized           */
/*    chirp waveform and loads them into the chirp generator via         */
/*    the parallel printer port. The entire process takes approx-        */
/*    imately 10 minutes.(Transmitter)                                   */
/*                                                                       */
/*    Variables:                                                         */
/*                                                                       */
/*      data     - sample value to be stored in the chirp generator.     */
/*      i,j      - index variables.                                      */
/*      nsamps   - number of sample values to be generated.              */
/*      port1    - address of the data register for the parallel         */
/*                 printer port.                                         */
/*      port2    - address of the control register for the parallel      */
/*                 printer port.                                         */
/*      pi       - the value of pi.                                      */
/*      fr       - starting frequcny of the chirp in Hz.                 */
/*      wr       - angular frequency in rad/sec.                         */
/*      fs       - sampling frequency in Hz.                             */
/*      ts       - sampling interval.                                    */
/*      dur      - duration of the chirp.                                */
/*      nbits    - number of bits used by the D/A converter (8).         */
/*      delta    - amplitude increment.                                  */
/*      vmin     - minimum amplitude value.                              */
/*      arg1     - intermediate result.                                  */
/*      arg2     - intermediate result.                                  */
/*      arg3     - intermediate result.                                  */
/*      v        - sampled signal amplitude.                             */
/*      bw       - chirp bandwidth.                                      */
/*      u1,u2    - intermediate results.                                 */
/*                                                                       */
/*************************************************************************/

#include  <stdio.h>
#include  <math.h>

main()
  {
     int                data;
     long int           i,j,nsamps;
     unsigned int       port1,port2;
     double             pi,fr,wr,fs,ts,dur,nbits,delta,vmin,arg1,v;
     double             bw,u1,u2,arg2,arg3;

 /*-------------------------------------------------------------------*/
 /*   Initialize the parallel port addresses for the data (port1)     */
 /*   and control signals (port2).                                    */
 /*-------------------------------------------------------------------*/

     port1=888;
     port2=890;

/*--------------------------------------------------------------------*/
/*   Initialize chirp parameters and other necessary constants.       */
/*--------------------------------------------------------------------*/

     pi=acos(-1.0);
     fr=1.0e6;
     wr=2.0*pi*fr;
     fs=8.0e6;
     ts=1.0/fs;
     dur=50.0e-3;
```

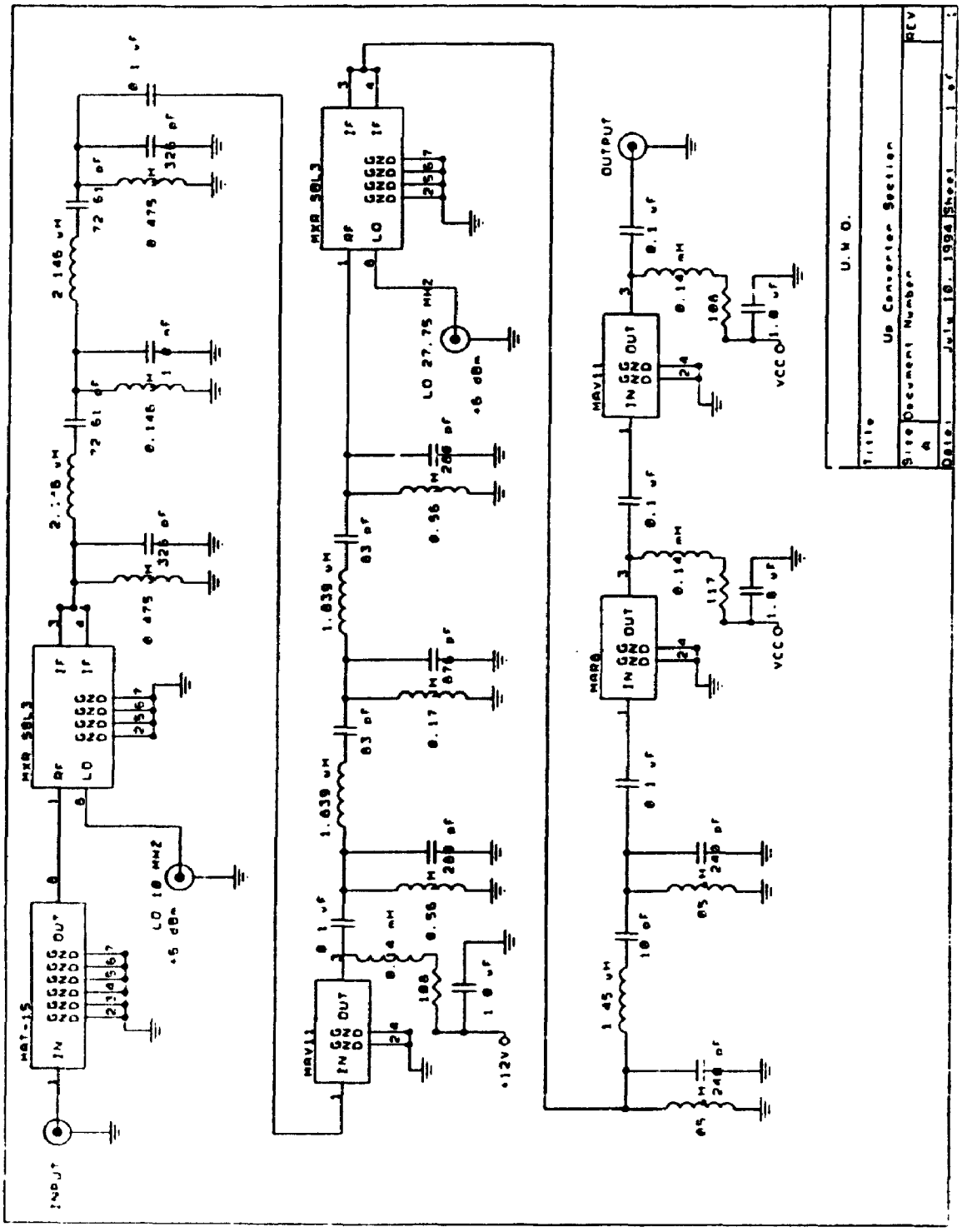
```
     bw=2.5e6;
     ul=bw/dur;
     u2=2.0*pi*ul;
     nbits=8.0;
     nsamps=(long int) (dur*fs);

/*-----------------------------------------------------------------*/
/*   Toggle the mode switch of the chirp generator several times   */
/*   to ensure that the system resets properly. The final toggle   */
/*   will set the chirp generator into the proper mode for         */
/*   accepting data from the computer.                             */
/*-----------------------------------------------------------------*/

     for(i=1;i<=7;++i)
        {
          outp(port2,1);
          for(j=0;j<75;++j);
          outp(port2,0);
        }

     delta=2.0/(double)(pow(2.0,nbits)-1);
     vmin=-1.0-(delta/2.0);

/*-----------------------------------------------------------------*/
/*   Generate and write data values to the chirp generator.        */
/*-----------------------------------------------------------------*/

     for(i=0;i<524288;++i)
        {
          if (i < nsamps)
             {
                arg1=wr*(double)(i)*ts;
                arg2=(double)(i)*ts;
                arg3=u2*pow(arg2,2)/2.0;

                v=cos(arg1+arg3);

                data=(int)((v-vmin)/delta);
             }
          else
             data=128;

          outp(port1,data);

          outp(port2,2);
          for(j=0;j<75;++j);
          outp(port2,10);
          outp(port2,0);

        }

/*-----------------------------------------------------------------*/
/*   Toggle the mode switch in order to permit the chirp generator */
/*   to be triggered by the external trigger signal.               */
/*-----------------------------------------------------------------*/

     outp(port2,1);
     for(j=0;j<75;++j);
     outp(port2,0);

  }
```

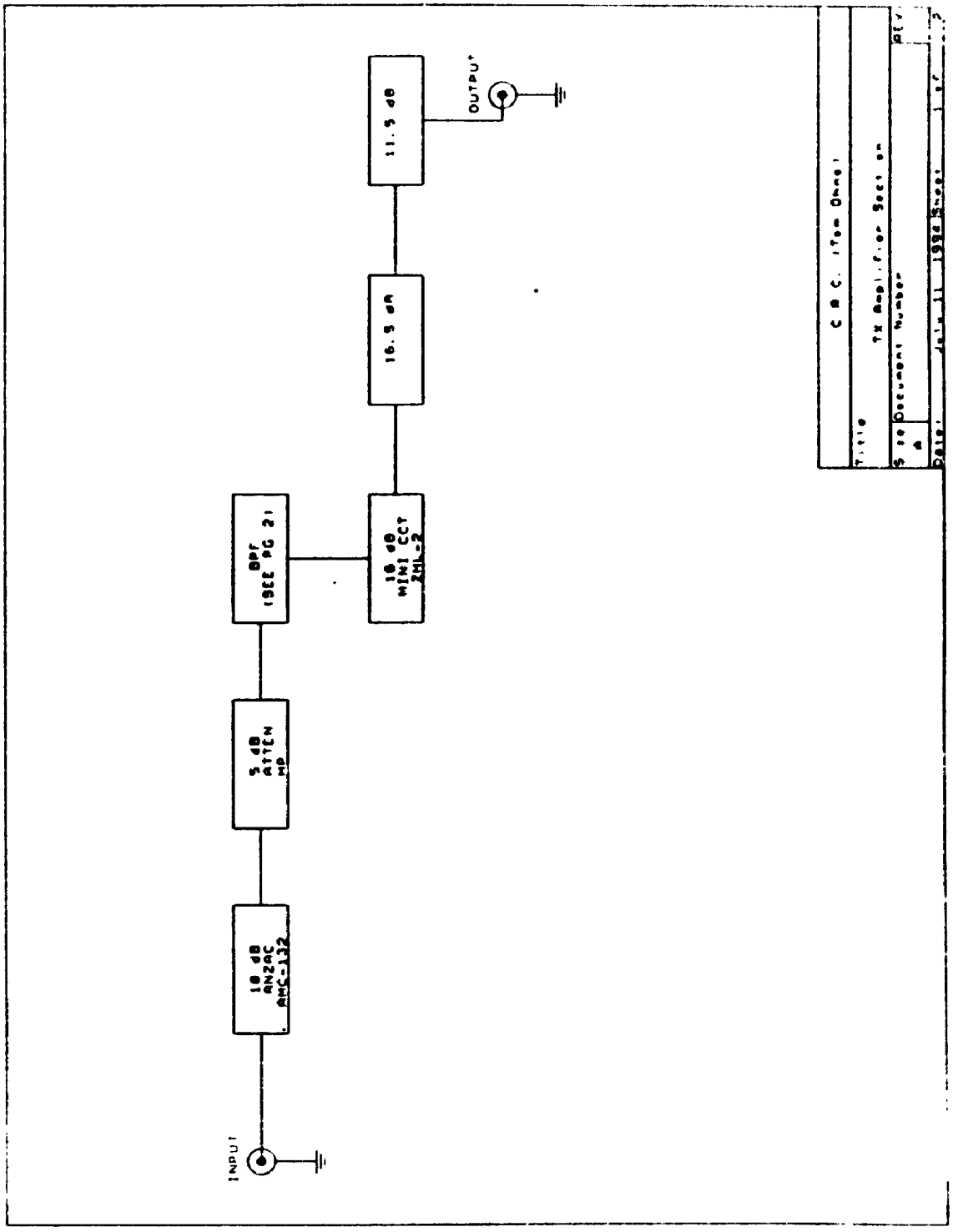
```
/*****************************************************************/
/*                                                               */
/*    File Name: chirp2.c                                        */
/*    Created: 19930809                                          */
/*    Created By: Kerry Ellis                                    */
/*                                                               */
/*    This program generates the sample values for a digitized   */
/*    chirp waveform and loads them into the chirp generator via */
/*    the parallel printer port. The entire process takes approx-*/
/*    imately 10 minutes. (Receiver)                             */
/*                                                               */
/*    Variables:                                                 */
/*                                                               */
/*      data     - sample value to be stored in the chirp generator. */
/*      i,j      - index variables.                              */
/*      nsamps   - number of sample values to be generated.      */
/*      port1    - address of the data register for the parallel */
/*                 printer port.                                 */
/*      port2    - address of the control register for the parallel */
/*                 printer port.                                 */
/*      pi       - the value of pi.                              */
/*      fr       - starting frequcny of the chirp in Hz.         */
/*      wr       - angular frequency in rad/sec.                 */
/*      fs       - sampling frequency in Hz.                     */
/*      ts       - sampling interval.                            */
/*      dur      - duration of the chirp.                        */
/*      nbits    - number of bits used by the D/A converter (8). */
/*      delta    - amplitude increment.                          */
/*      vmin     - minimum amplitude value.                      */
/*      arg1     - intermediate result.                          */
/*      arg2     - intermediate result.                          */
/*      arg3     - intermediate result.                          */
/*      v        - sampled signal amplitude.                     */
/*      bw       - chirp bandwidth.                              */
/*      u1,u2    - intermediate results.                         */
/*                                                               */
/*****************************************************************/

#include  <stdio.h>
#include  <math.h>

main()
  {
     int               data;
     long int          i,j,nsamps;
     unsigned int      port1,port2;
     double            pi,fr,wr,fs,ts,dur,nbits,delta,vmin,arg1,v;
     double            bw,u1,u2,arg2,arg3;

  /*---------------------------------------------------------------*/
  /*   Initialize the parallel port addresses for the data (port1) */
  /*   and control signals (port2).                                */
  /*---------------------------------------------------------------*/

     port1=888;
     port2=890;

  /*---------------------------------------------------------------*/
  /*   Initialize chirp parameters and other necessary constants.  */
  /*---------------------------------------------------------------*/

     pi=acos(-1.0);
     fr=1.012e6;
     wr=2.0*pi*fr;
     fs=8.0e6;
     ts=1.0/fs;
     dur=50.0e-3;
```

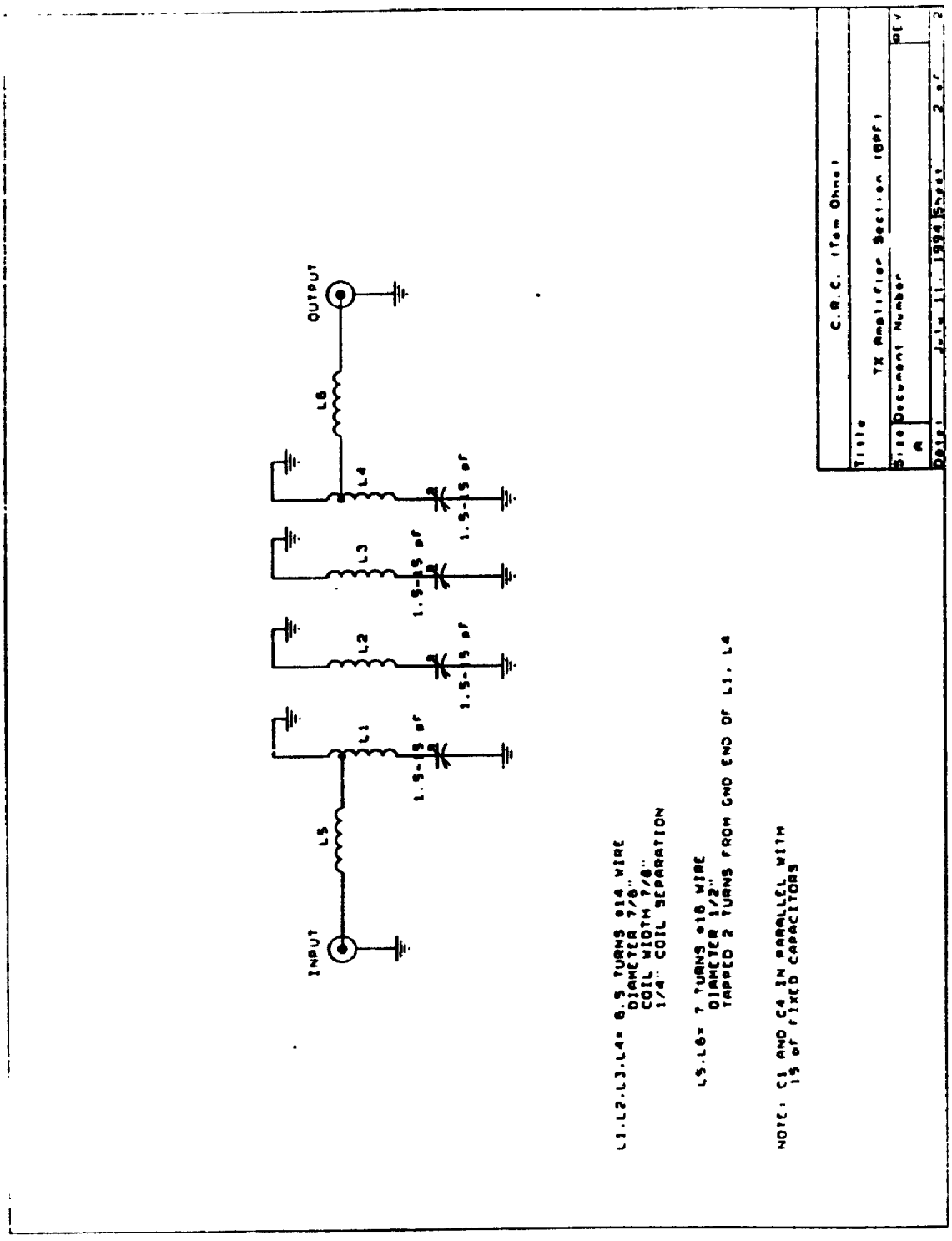
```
     bw=2.5e6;
     u1=bw/dur;
     u2=2.0*pi*u1;
     nbits=8.0;
     nsamps=(long int) (dur*fs);

/*-----------------------------------------------------------------*/
/*   Toggle the mode switch of the chirp generator several times   */
/*   to ensure that the system resets properly. The final toggle   */
/*   will set the chirp generator into the proper mode for         */
/*   accepting data from the computer.                             */
/*-----------------------------------------------------------------*/

     for(i=1;i<=7;++i)
        {
           outp(port2,1);
           for(j=0;j<75;++j);
           outp(port2,0);
        }

     delta=2.0/(double)(pow(2.0,nbits)-1);
     vmin=-1.0-(delta/2.0);

/*-----------------------------------------------------------------*/
/*   Generate and write data values to the chirp generator.        */
/*-----------------------------------------------------------------*/

     for(i=0;i<524288;++i)
        {
          if (i < nsamps)
             {
                arg1=wr*(double)(i)*ts;
                arg2=(double)(i)*ts;
                arg3=u2*pow(arg2,2)/2.0;

                v=cos(arg1+arg3);

                data=(int)((v-vmin)/delta);
             }
          else
             data=128;

          outp(port1,data);

          outp(port2,2);
          for(j=0;j<75;++j);
          outp(port2,10);
          outp(port2,0);

        }
/*-----------------------------------------------------------------*/
/*   Toggle the mode switch in order to permit the chirp generator */
/*   to be triggered by the external trigger signal.               */
/*-----------------------------------------------------------------*/

     outp(port2,1);
     for(j=0;j<75;++j);
     outp(port2,0);

  }
```

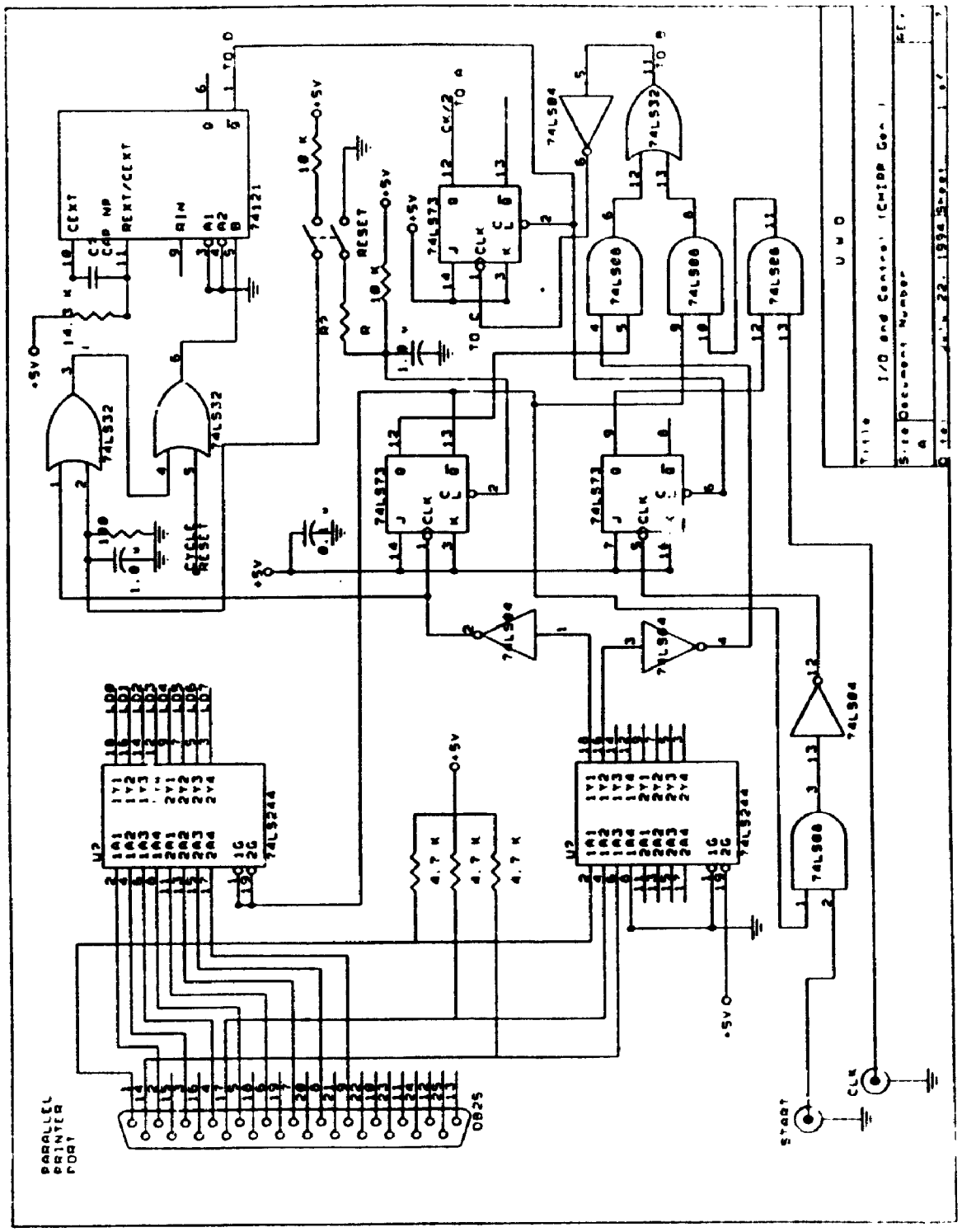

U.M.D
Keying Sequencer
Document Number
July 9, 1994
74LS123
CEXT
REXT/CEXT
CLR
+5V
15 K
10 K
1.0 K
100 K
50 K
Carrier Keying Output

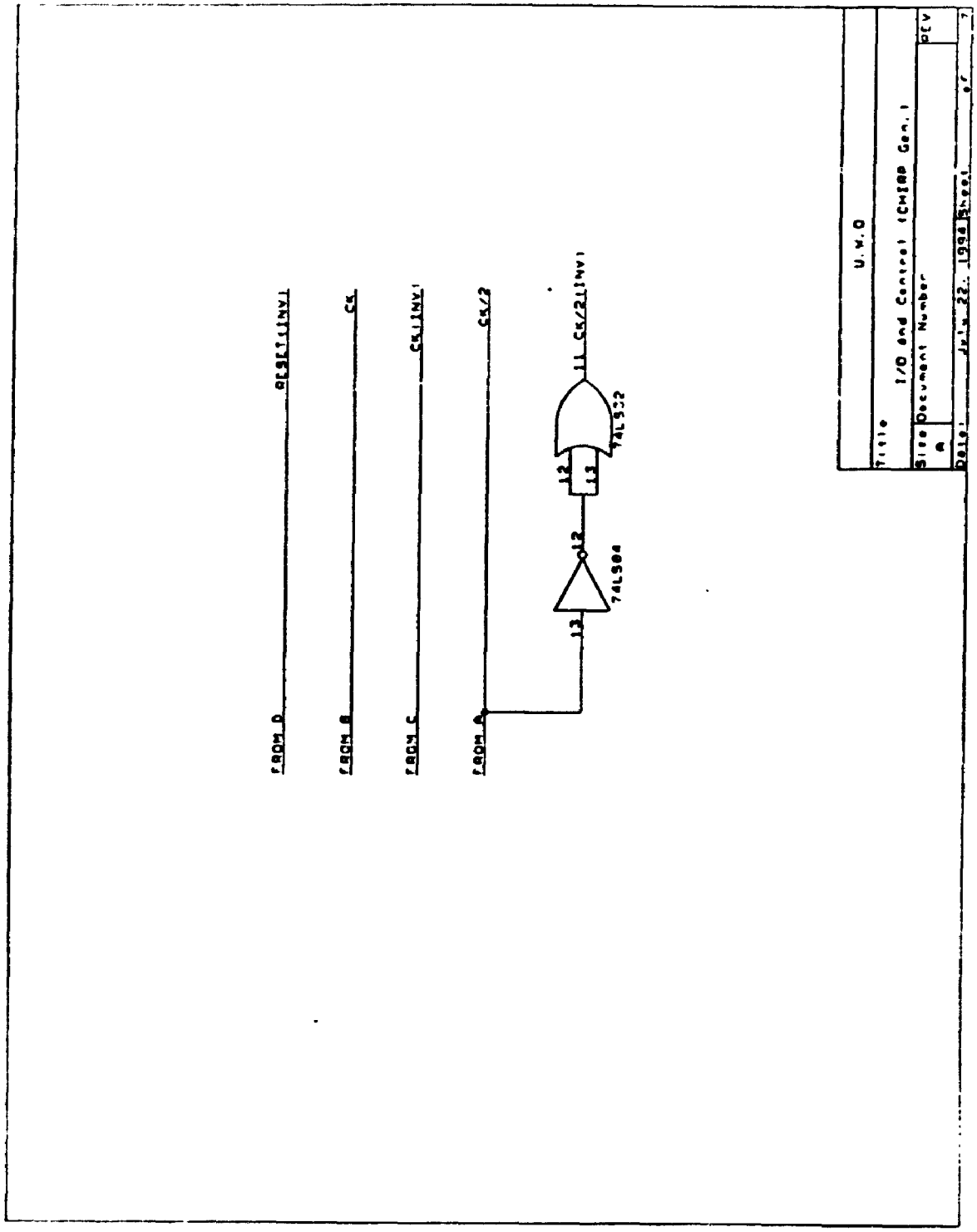

COMPOSITE OUTPUT
PSC-2-1
LM6321
74LS08
+12V
10 K
2.2 K
0.1 uF
CARRIER OSCILLATOR
TX CONTROL
U W O

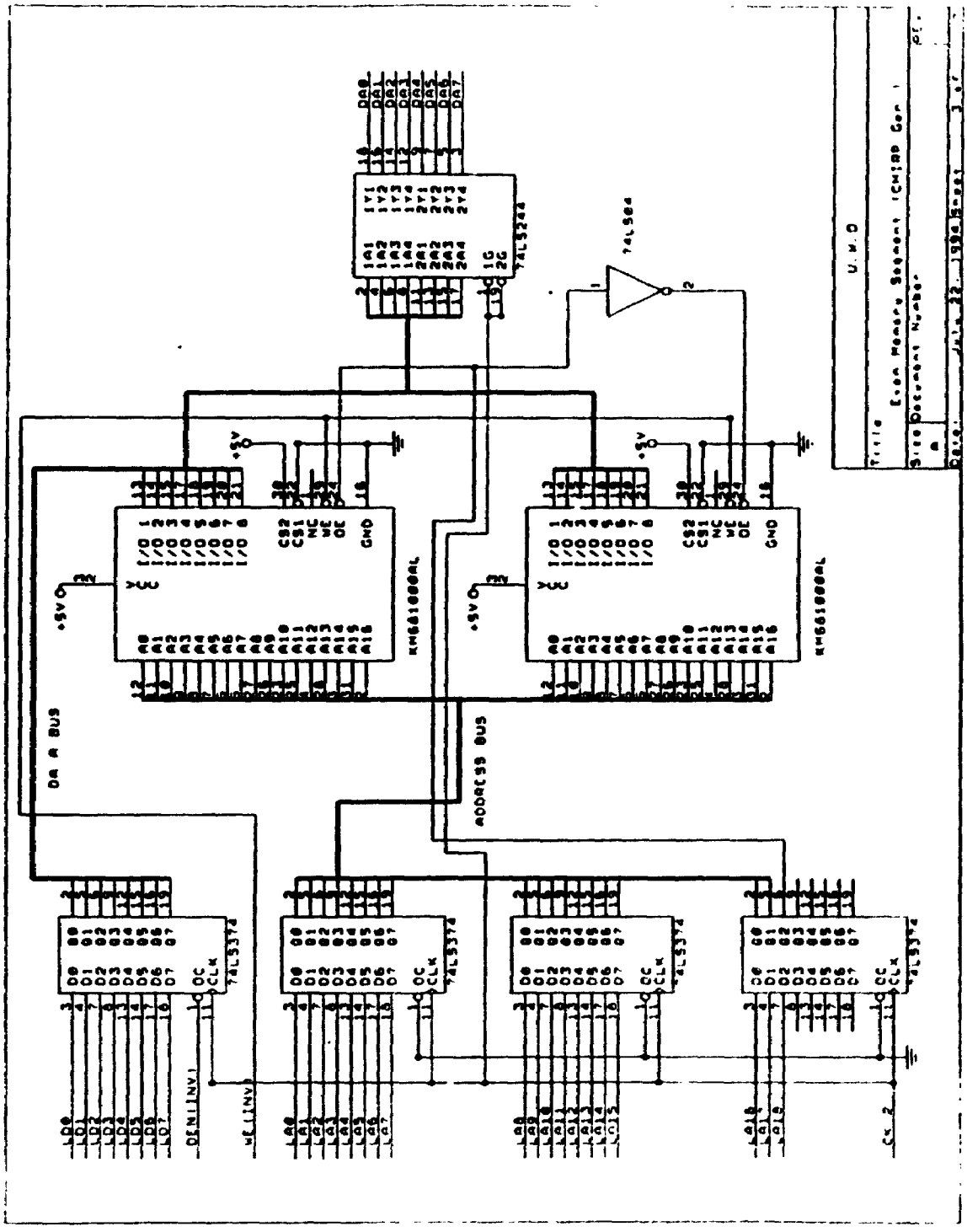

U.W.O.
Up Converter Section
Title
Size
Document Number
REV
A
Date: July 18, 1994 Sheet 1 of 1
INPUT
MAT-15
MXR SDL3
LO 10 MHZ
+5 dBm
MAV11
+12V
MXR SDL3
LO 27.75 MHZ
+5 dBm
MAV11
MAR8
OUTPUT
VCC

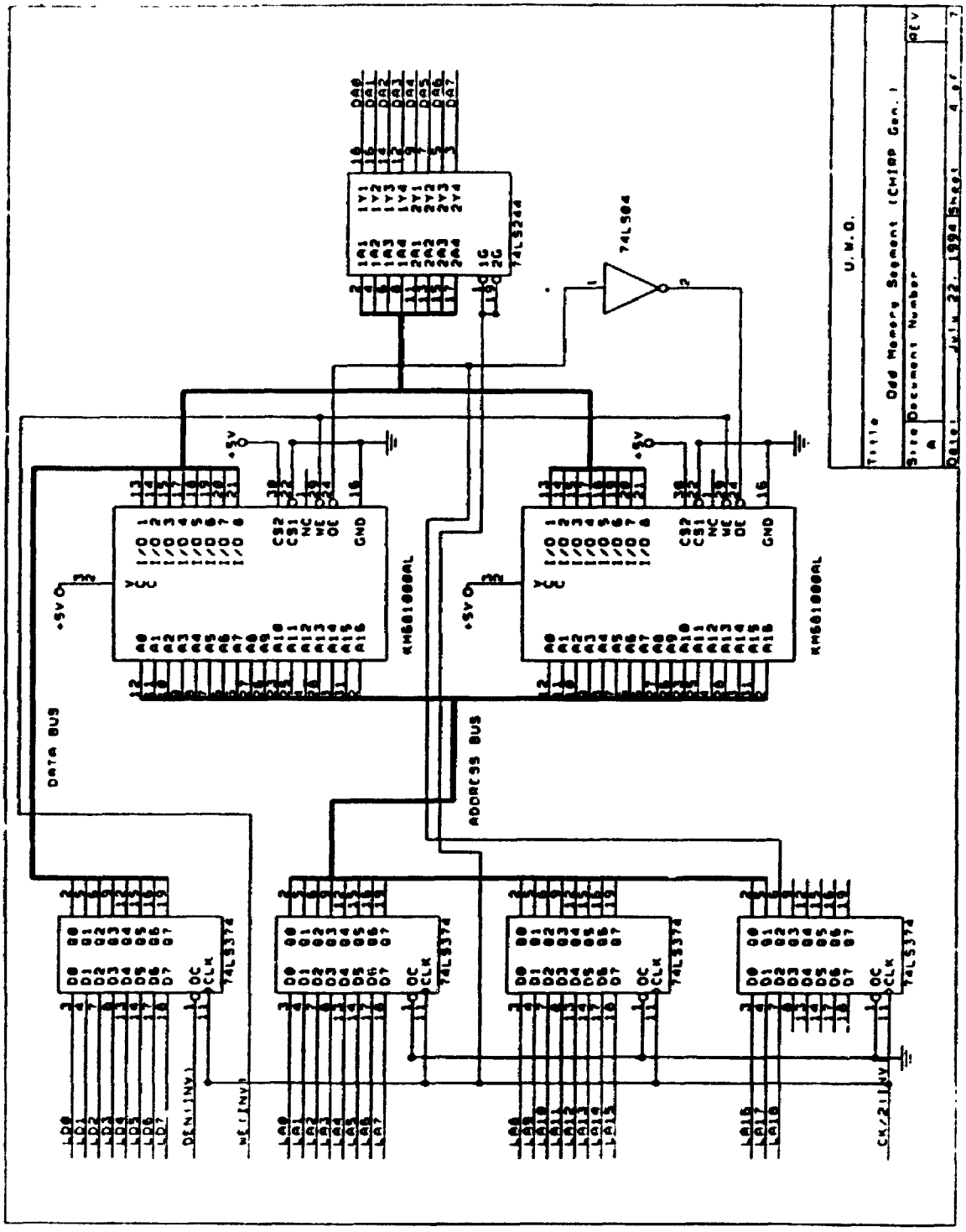

INPUT
10 dB
ANZAC
AMC-132
5 dB
ATTEN
MP
BPF
(SEE PG 21
10 dB
MINI CCT
ZHL-2
16.5 dB
11.5 dB
OUTPUT
C B C. (Ton Ohm)
TX Amplifier Section
Document Number
Date: June 11 1994

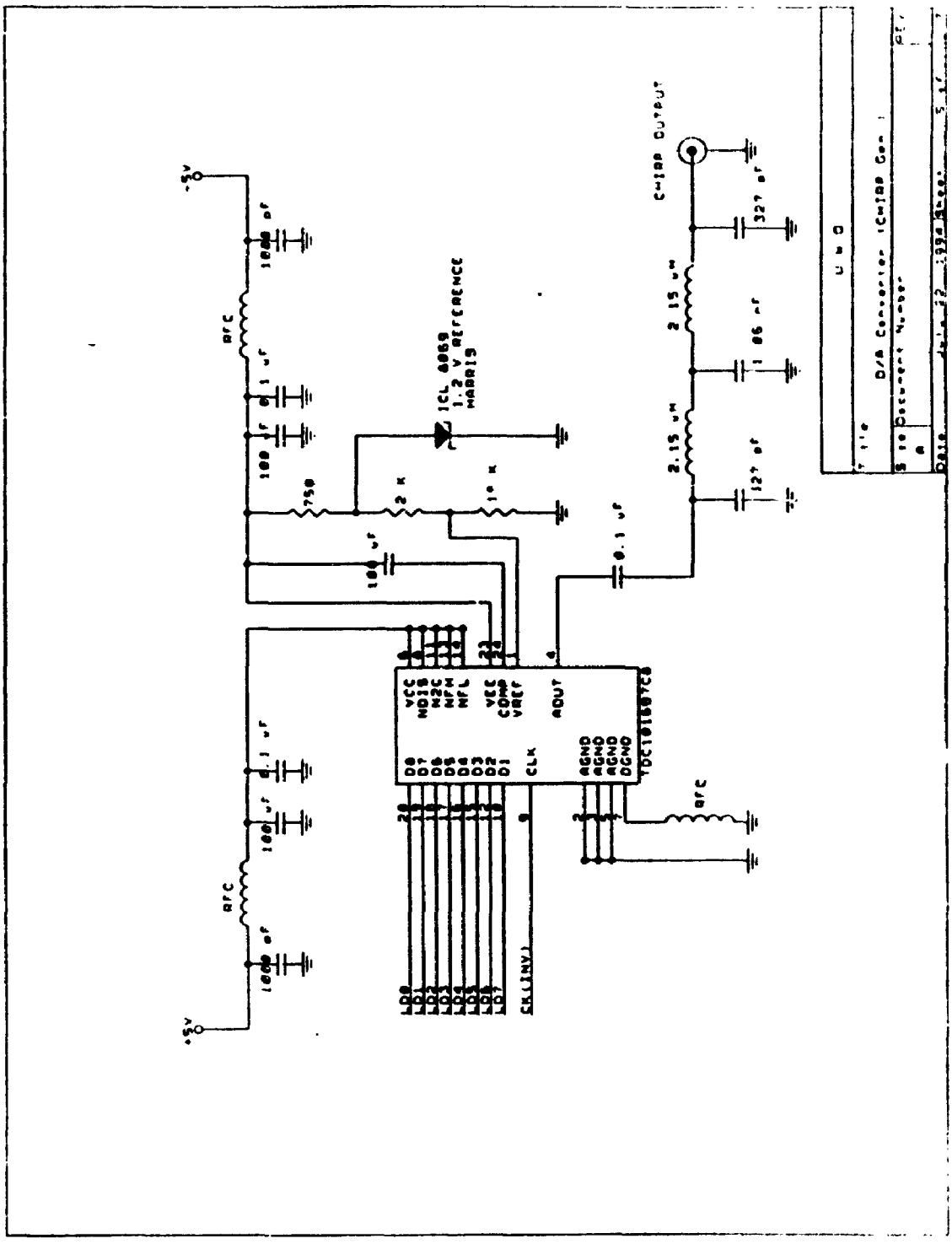

INPUT
OUTPUT
L5
L6
L1
L2
L3
L4
1.5-15 pf
1.5-15 pf
1.5-15 pf
1.5-15 pf
L1.L2.L3.L4= 6.5 TURNS #14 WIRE
DIAMETER 7/8"
COIL WIDTH 7/8"
1/4" COIL SEPARATION
L5.L6= 7 TURNS #18 WIRE
DIAMETER 1/2"
TAPPED 2 TURNS FROM GND END OF L1. L4
NOTE: C1 AND C4 IN PARALLEL WITH
15 pf FIXED CAPACITORS
C.R.C. (Ten Ohms)
Title
TX Amplifier Section (80M)
Size
A
Document Number
REV
Date: June 11, 1994 Sheet 2 of 2

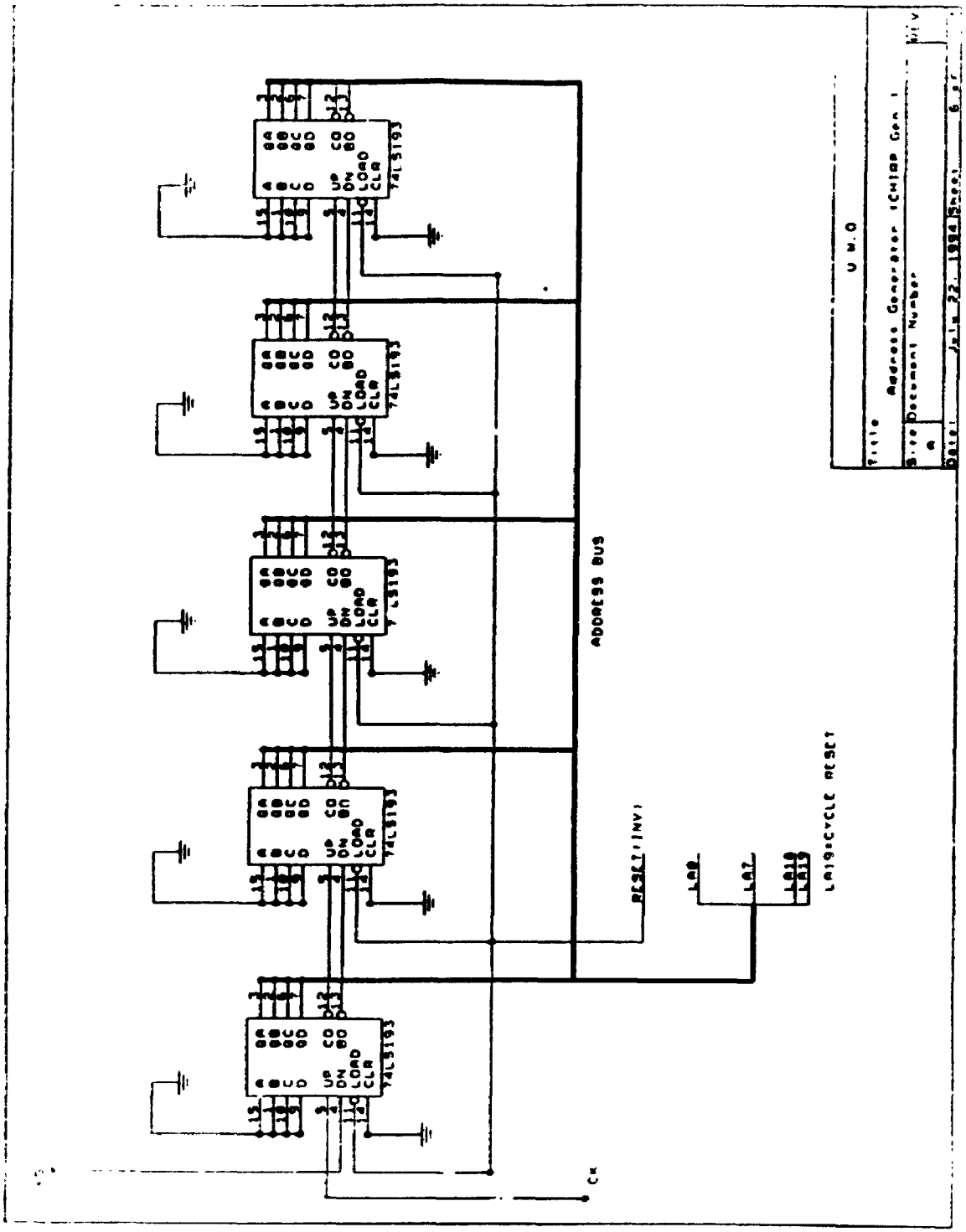

74121
74LS32
74LS32
74LS123
74LS123
74LS123
74LS04
74LS32
74LS08
74LS08
74LS08
74LS04
74LS04
74LS04
74LS08
74LS244
74LS244
RESET
RESET
+5V
+5V
+5V
+5V
+5V
10 K
10 K
4.7 K
4.7 K
4.7 K
PARALLEL PRINTER PORT
DB25
START
CLK
U W O
I/O and Control (CHIDP Gen.)

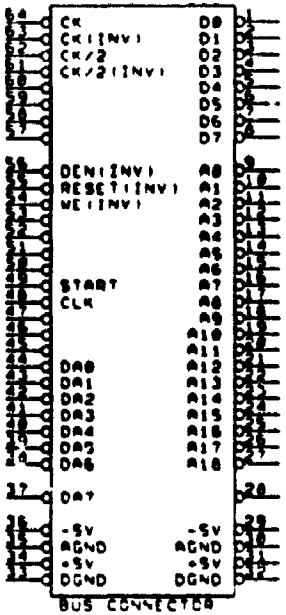

U.W.O
I/O and Control (CMOS Gen.)
Document Number
REV
Size
A
Date: July 22, 1994
Sheet
of
Title
FROM D
FROM B
FROM C
FROM A
CS/2
CS/INT1
CS
INT1/SS
INT1/SS/2
74LS04
74LS32


DA A BUS
ADDRESS BUS
74LS244
74LS04
74LS374
U.W.D
Title
Eeom Memory Segment (CH:80 Cor
Size Document Number
A

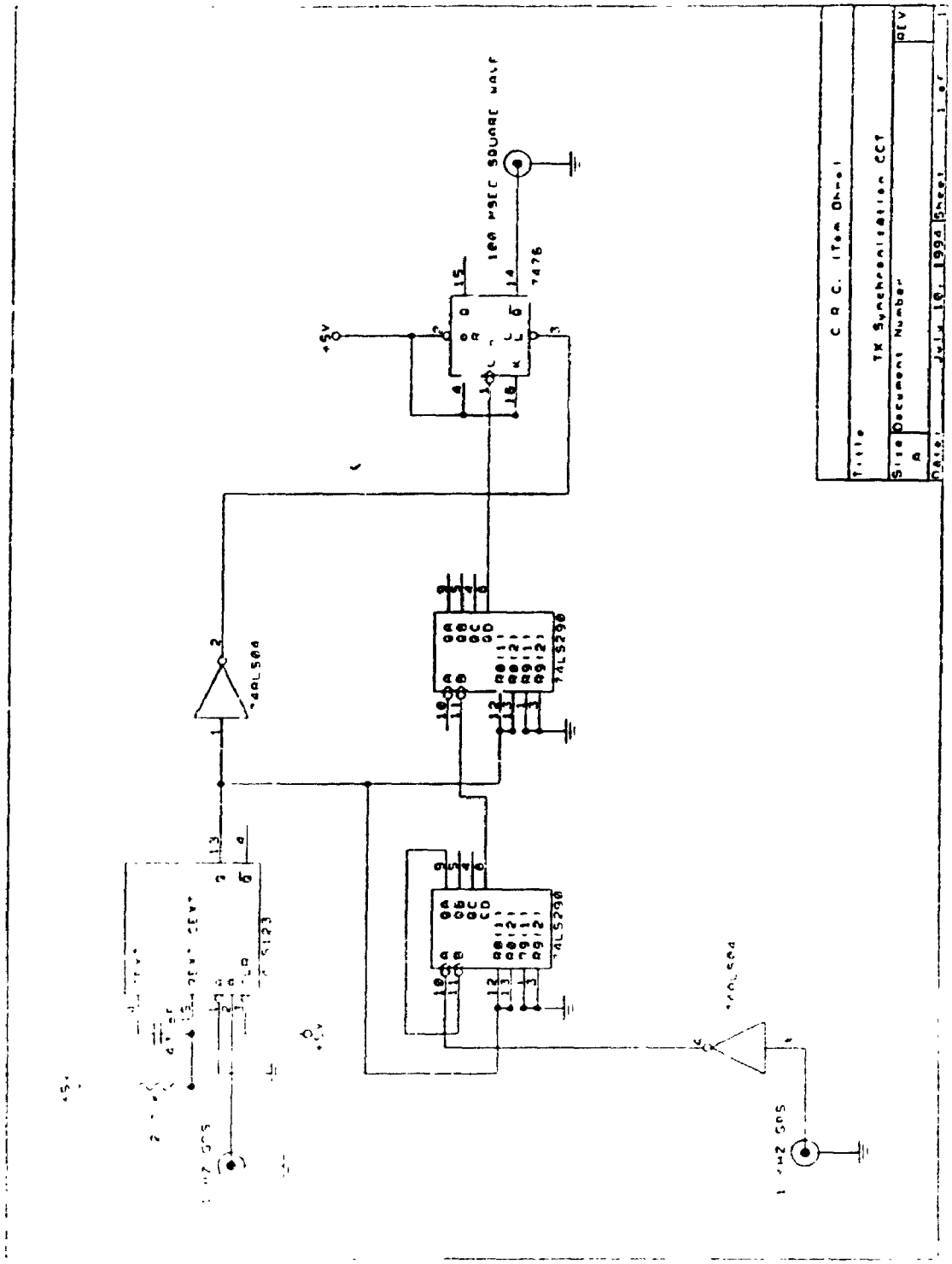

U.W.O.
Odd Memory Segment (CMIBP Gen.)
74LS244
74LS04
KM681000AL
DATA BUS
ADDRESS BUS
74LS374

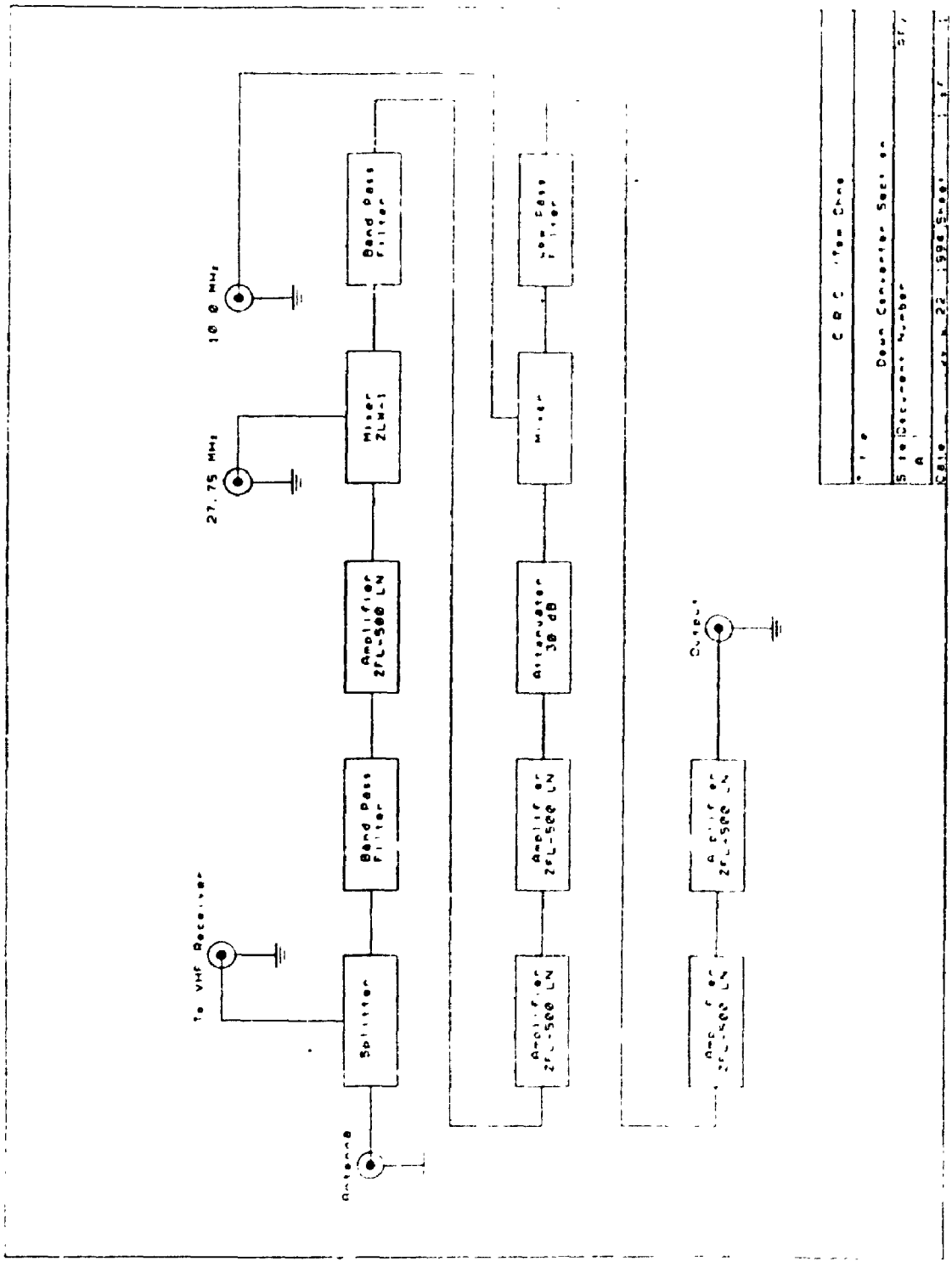

-5V
+5V
RFC
ICL 8069
1.2 V REFERENCE
HARRIS
750
2 K
1 K
2.15 uH
327 pF
1.86 nF
127 pF
0.1 uF
100 uF
1000 pF
VCC
NC15
N2C
NFH
NFL
VEE
COMP
VREF
AOUT
D8
D7
D6
D5
D4
D3
D2
D1
CLK
AGND
DGND
EXTINV
D/A Converter (CM7128 Gen 1)

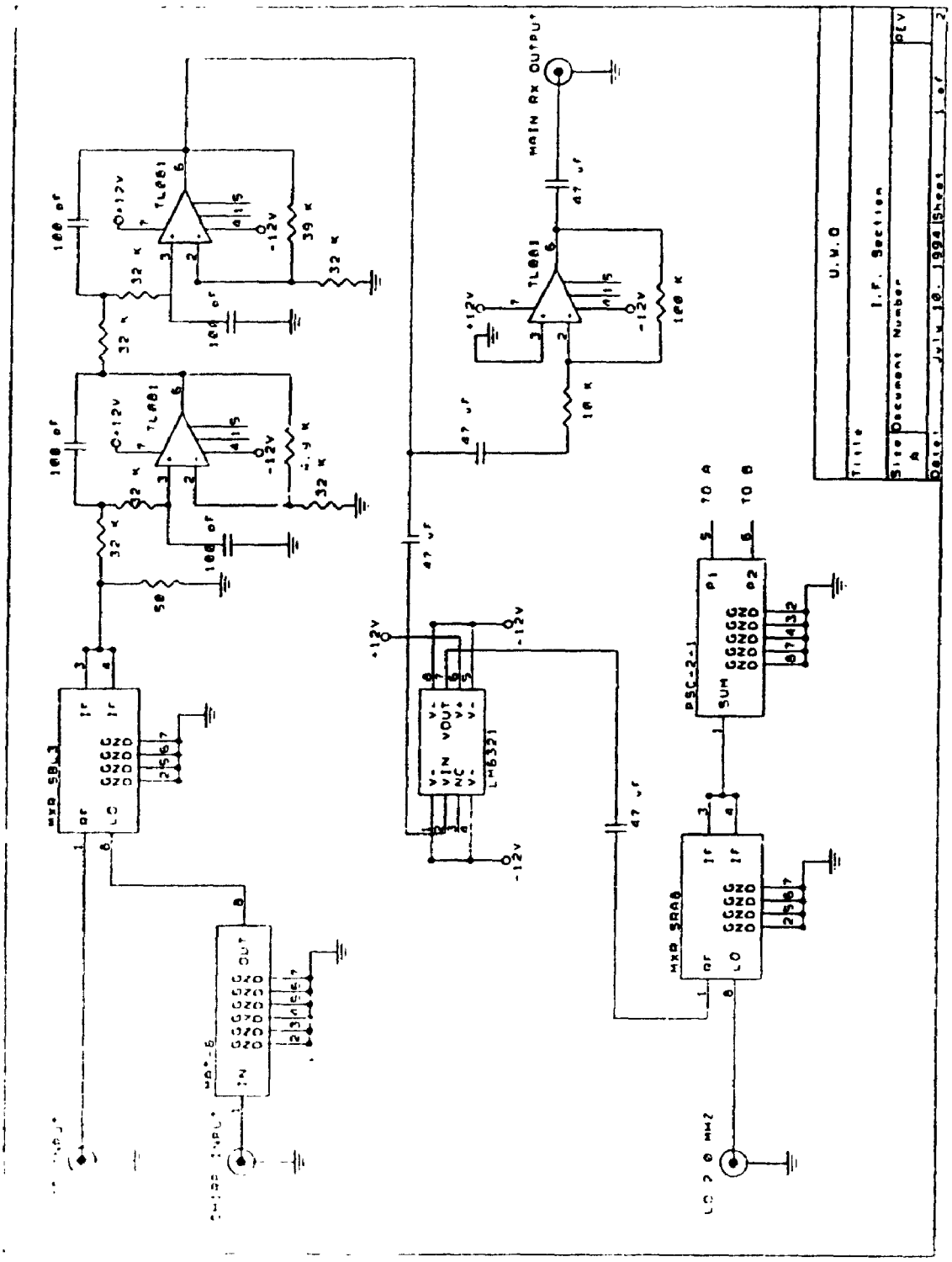

74LS193
ADDRESS BUS
RESET (INV)
LA19
LA18
LA19=CYCLE RESET
U W.O
Address Generator (CHIP Gen.)
Document Number
Date: June 22, 1994 Sheet 6 of
Title
Size
A
Rev

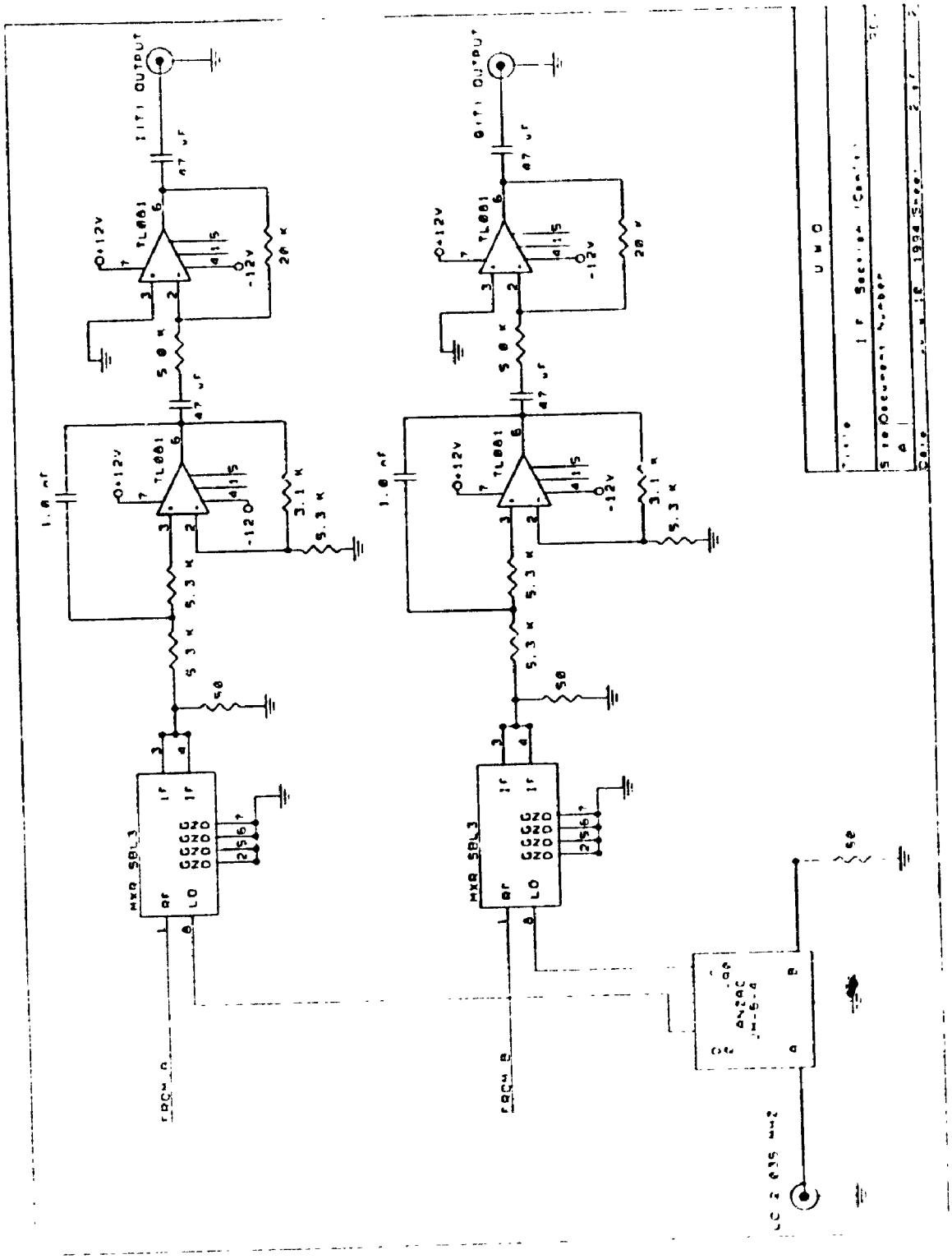


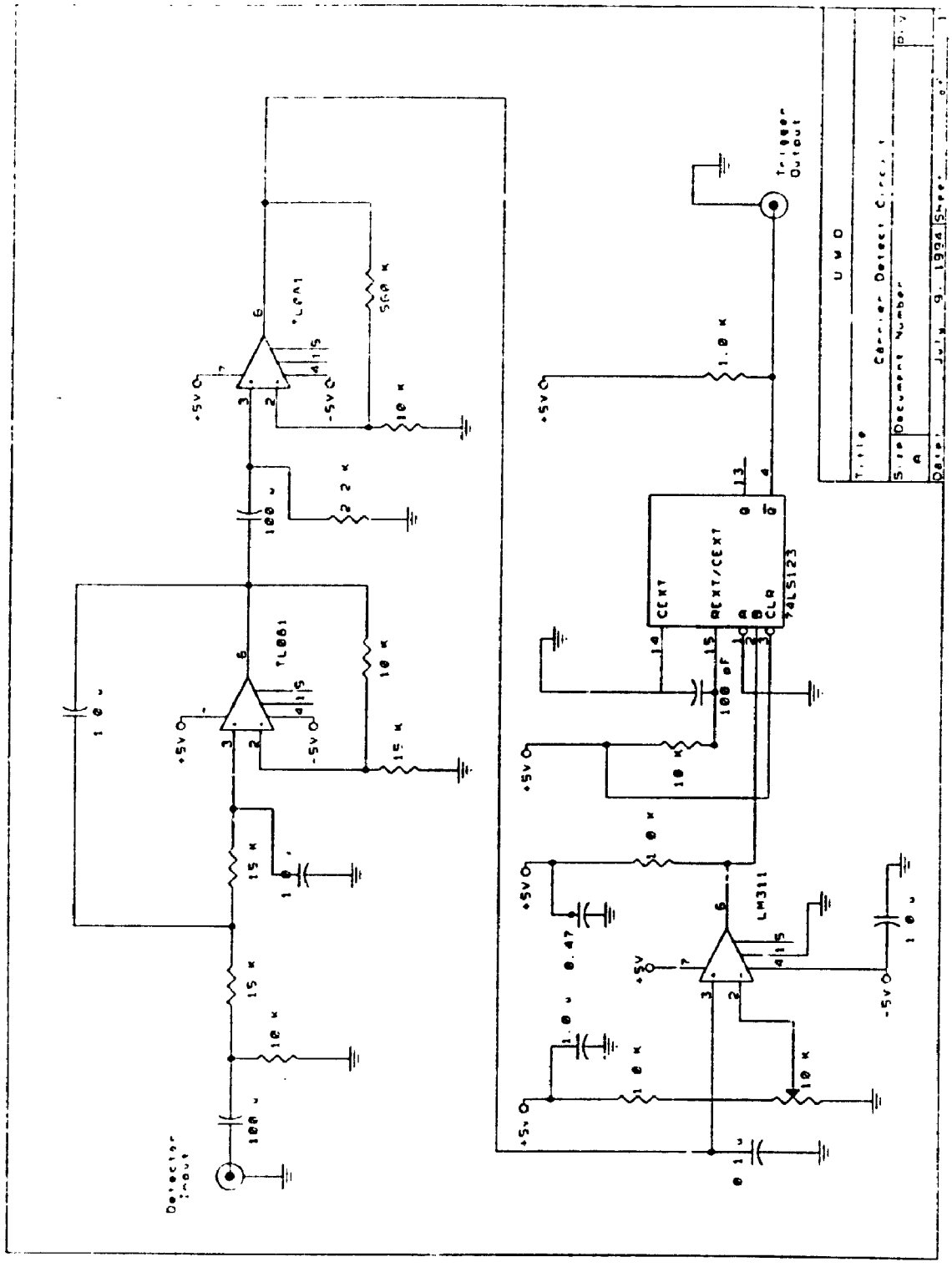
Note Backplane system consists of
5 such connectors wired in parallel.

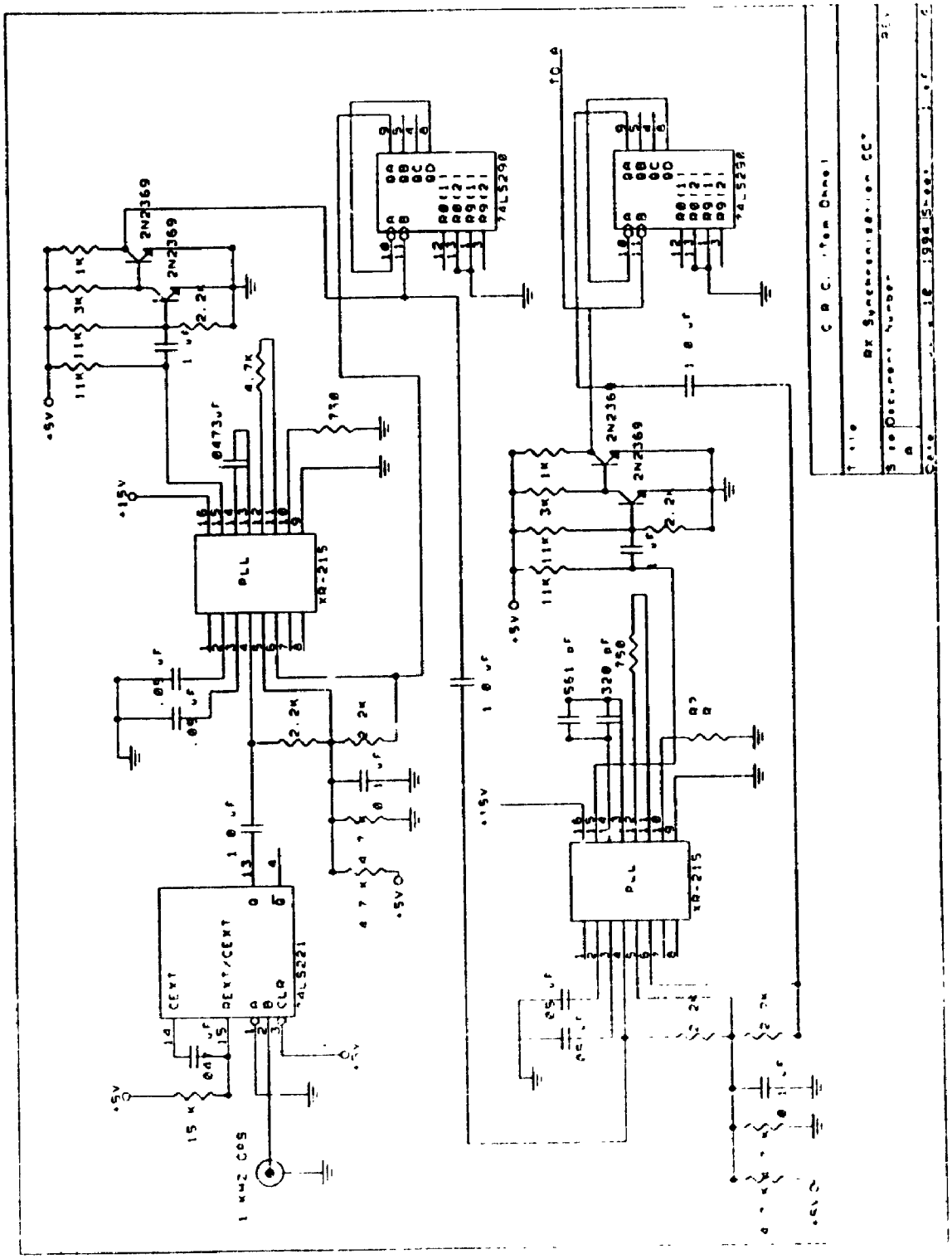

C R C. Ithaca Ohio
TX Synchronization CCT
Document Number
Sheet
100 MSEC SQUARE WAVE
7476
74LS04
74LS290
74LS123
74LS04
1 MHZ SQS

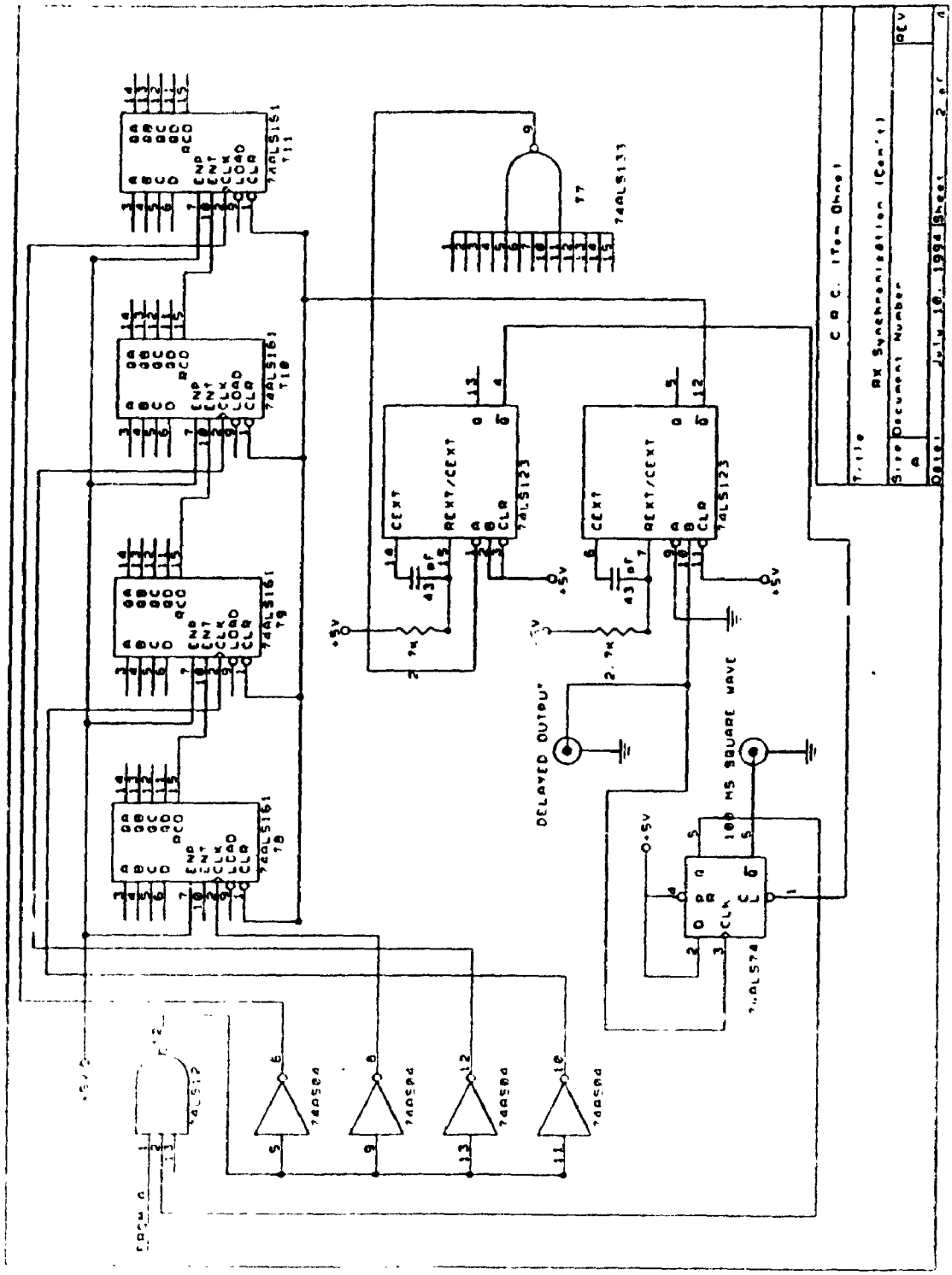

To VHF Receiver
Antenna
Splitter
Band Pass Filter
Amplifier ZFL-500 LN
Mixer ZLW-1
27.75 MHz
10.0 MHz
Band Pass Filter
Amplifier ZFL-500 LN
Attenuator 30 dB
Mixer
Low Pass Filter
Output
Amplifier ZFL-500 LN
Amplifier ZFL-500 LN
C R C
Down Converter Number

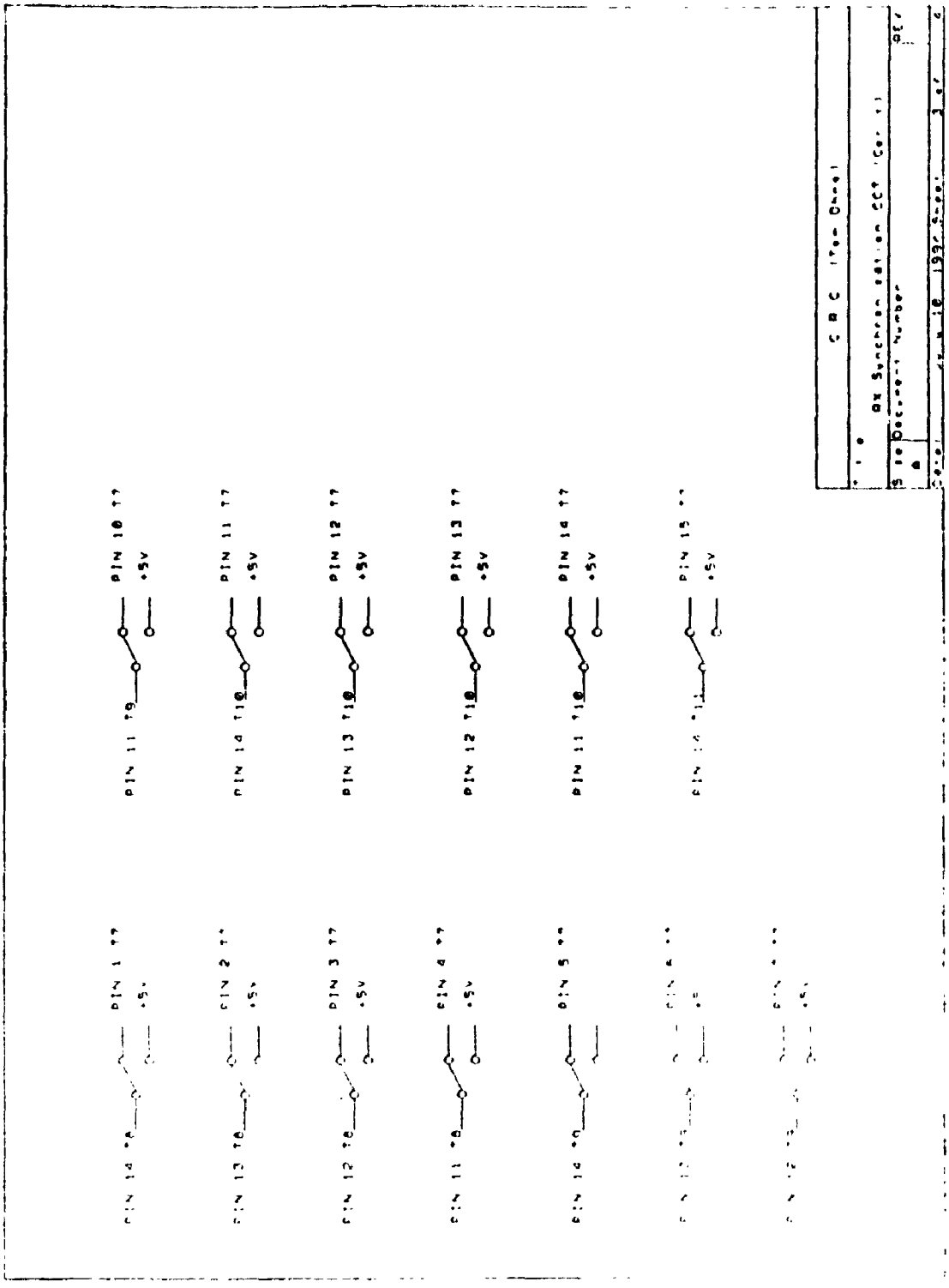

MAIN RX OUTPUT
TL081
LM6321
PSC-2-1
MXR SB-1
MXR SRA-8
MD-16
U.W.O
I.F. Section
Title
Size
Document Number
Date: July 10, 1994
Sheet 1 of 2
REV

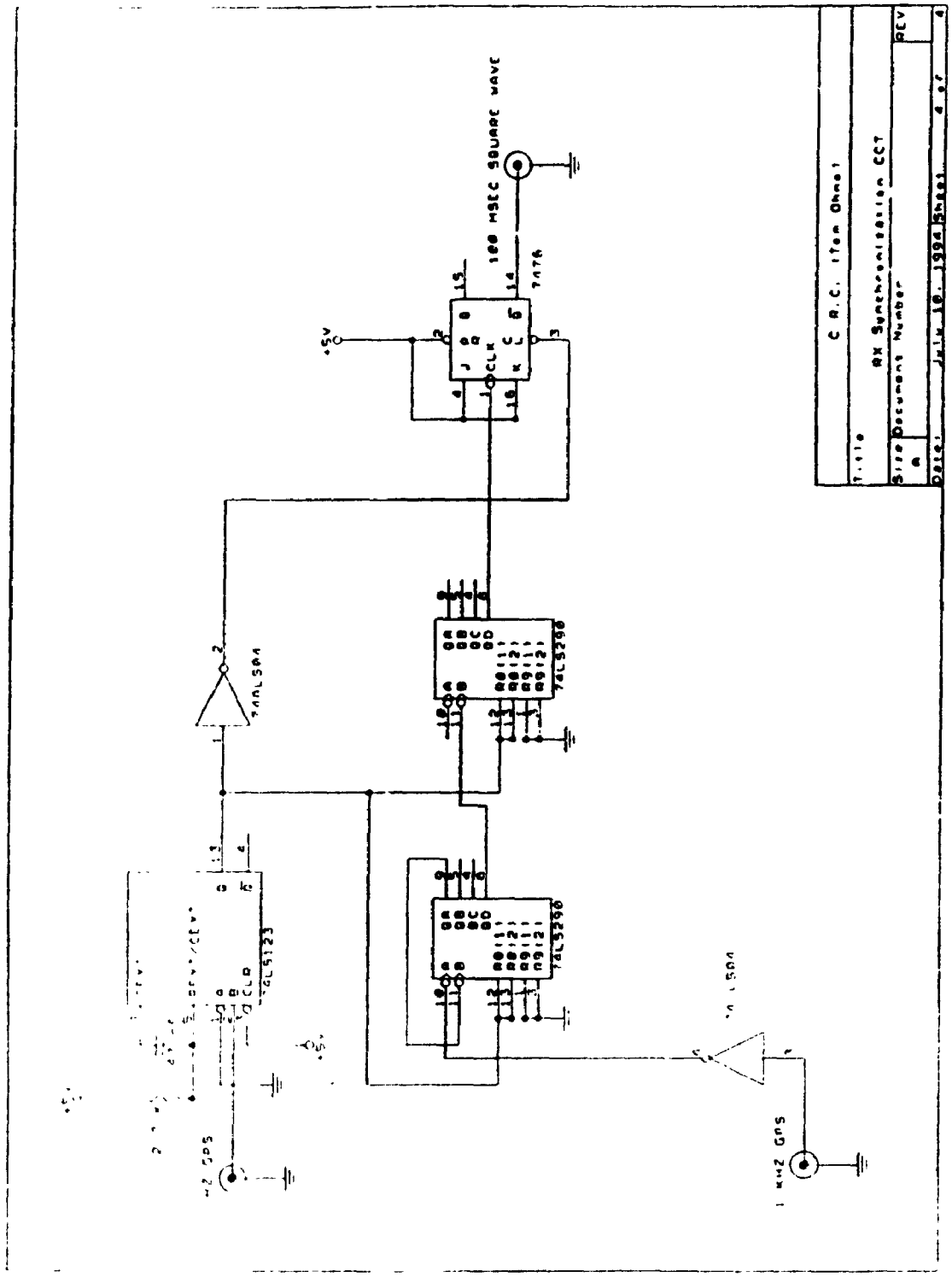



U W D
Carrier Detect Circuit
Document Number
Sheet
July 9, 1994
Trigger Output
Detector Input
TL081
LM311
74LS123
CEXT
REXT/CEXT
CLR
+5V
-5V
100 pF
0.47
10 K
15 K
2.2 K
1.0 K
100 µ
1.0 µ
0.1 µ


2N2369
2N2369
1K
3K
11K
11K
2.2K
1 uF
+5V
+15V
.0473uF
4.7K
750
PLL
XR-215
.05 uF
.01 uF
2.2K
2K
1 uF
1.0 uF
4.7K
+5V
74LS221
DEXT/CEXT
CEXT
A
B
CLR
Q
.01 uF
.15 uF
+5V
15 K
1 KHZ COS
74LS298
TO A
1.0 uF
2N2369
2N2369
3K
1K
11K
2.2K
+5V
.0561 uF
320 pF
750
+15V
PLL
XR-215
.05 uF
2.2K
2K
C B C . . . Rom Panel
BX Synchronization CC
BX Document Number


74ALS161
74ALS123
74ALS74
74ALS04
74ALS12
74ALS133
DELAYED OUTPUT
100 NS SQUARE WAVE
+5V
43 pf
2.7K
C R C iTech Ohio
RX Synchronization (Cont'd)
Size
Document Number
Date: July 10, 1994 Sheet


PIN 10
+5V
PIN 11
+5V
PIN 12
+5V
PIN 13
+5V
PIN 14
+5V
PIN 15
+5V
PIN 1
+5V
PIN 2
+5V
PIN 3
+5V
PIN 4
+5V
PIN 5


100 MSEC SQUARE WAVE
+5V
74LS74
74LS04
74LS123
74LS290
74LS00
1 KHZ GPS
C.R.C. Ottawa
RX Synchronization CCT
Size
Document Number
REV
Date: July 18, 1994 Sheet 4 of 4

Appendix B

Appended Material for Chapter 5

```
C*****************************************************************
C     HAMMING WINDOW TEST PROGRAM
C*****************************************************************
C
C     FILE NAME: HAMTST.FOR
C     CREATED: 19940322
C     CREATED BY: KERRY ELLIS
C
C     THIS PROGRAM IS USED TO GENERATED A HAMMING WINDOW OF A GIVEN
C     SIZE N (SAMPLES). THE WINDOW SAMPLES GENERATED BY THIS PROGRAM
C     ARE STORED IN THE FILE NAME JUNK.DAT. ONCE THE WINDOW
C     COEFFICIENTS ARE WRITTEN TO THE OUTPUT FILE, THE DATA SEQUENCE
C     IS PADDED TO A LENGTH OF 1024 SAMPLES AND THE AMPLITUDE
C     SPECTRUM OF THE WINDOW IS COMPUTED. SPECTRAL INFORMATION IS
C     WRITTEN TO THE OUTPUT FILE JUNK1.DAT.
C
C     VARIABLES:
C       N        - NUMBER OF SAMPLES IN THE WINDOW.
C       NN       - NUMBER OF SAMPLES TO WHICH THE SEQUENCE IS PADDED
C                  FOR SPECTRAL COMPUTATIONS.
C       I        - INDEX VARIABLE.
C       IFACT    - TEMPORARY STORAGE ARRAY USED BY DFT SUBROUTINE.
C       DR       - ARRAY CONTAINING THE REAL PART OF THE DATA.
C       DI       - ARRAY CONTAINING THE IMAGINARY PART OF THE DATA.
C       VR,VI    - TEMPORARY ARRAYS FOR REAL AND IMAGINARY PARTS.
C       MAX      - LARGEST AMPLITUDE IN THE SPECTRUM. USED FOR
C                  SPECTRAL NORMALIZATION.
C       FRQ      - NYQUIST FREQUENCY. USED FOR FREQUENCY
C                  NORMALIZATION.
C
C*****************************************************************
C
      INTEGER N,I,IFACT(10000)
      DOUBLE PRECISION DR(10000),DI(10000),VR(10000),VI(10000)
      DOUBLE PRECISION MAX,FRQ
C
C-----------------------------------------------------------------
C     INITIALIZE VARIABLES AND ARRAYS.
C-----------------------------------------------------------------
C
      N=1024
C
      DO 10 I=1,N
         IFACT(I)=0
         DR(I)=0.0D0
         DI(I)=0.0D0
         VR(I)=0.0D0
         VI(I)=0.0D0
  10  CONTINUE
C
C-----------------------------------------------------------------
C     GENERATE THE WINDOW SAMPLES.
C-----------------------------------------------------------------
C
      N=50
C
      DO 12 I=1,N
         DR(I)=1.0D0
  12  CONTINUE
C
      CALL HAM2(N,DR)
C
C-----------------------------------------------------------------
C     WRITE THE WINDOW SAMPLES TO THE OUTPUT FILE.
C-----------------------------------------------------------------
C
      OPEN(UNIT=1,FILE='JUNK.DAT')
```

```
C
      DO 13 I=1,N
         WRITE(1,100) I-1,DR(I)
  13  CONTINUE
C
      CLOSE(1)
C
C-----------------------------------------------------------------------
C     PAD THE SEQUENCE TO A LENGTH OF 1024 SAMPLES AND COMPUTE THE
C     AMPLITUDE SPECTRUM.
C-----------------------------------------------------------------------
C
      N=1024
C
      DO 15 I=1,N
         DI(I)=0.0D0
         VR(I)=0.0D0
         VI(I)=0.0D0
         IFACT(I)=0
  15  CONTINUE
C
      CALL DFOUR2(DR,DI,N,-1,VR,VI,IFACT)
C
      DO 17 I=1,N
         DR(I)=DSQRT(DR(I)**2+DI(I)**2)
  17  CONTINUE
C
C-----------------------------------------------------------------------
C     NORMALIZE THE SPECTRAL COMPONENTS BY THE LARGEST AMPLITUDE IN
C     THE SPECTRUM.
C-----------------------------------------------------------------------
C
      MAX=DR(1)
C
      DO 18 I=1,N
         IF (DR(I).GT.MAX) THEN
            MAX=DR(I)
         ENDIF
  18  CONTINUE
C
      DO 19 I=1,N
         DR(I)=DR(I)/MAX
C
         IF (DR(I).GT.0.0D0) THEN
            DR(I)=20.0D0*DLOG10(DR(I))
         ELSE
            DR(I)=-100.0D0
         ENDIF
C
  19  CONTINUE
C
C-----------------------------------------------------------------------
C     WRITE THE SPECTRUM TO THE OUTPUT DATA FILE. FREQUENCY
C     NORMALIZATION IS DONE AT THIS TIME.
C-----------------------------------------------------------------------
      OPEN(UNIT=1,FILE='JUNK1.DAT')
C
      DO 20 I=1,N
         FRQ=DBLE(I-1)/DBLE(N/2)
         WRITE(1,200) FRQ,DR(I)
  20  CONTINUE
C
      CLOSE(1)
C
  100 FORMAT(I4,5X,F16.6)
  200 FORMAT(F16.6,5X,F16.6)
      END
```

```
C*********************************************************************
      SUBROUTINE HAM2(N,SIG)
C*********************************************************************
C
C     FILE NAME: HAM2.FOR
C     CREATED: 19931019
C     CREATED BY: KERRY &LLIS
C
C     THIS SUBROUTINE MULTIPLIES THE CONTENTS OF THE SIGNAL VECTOR
C     BY A HAMMING WINDOW.
C
C     VARIABLES:
C       N      - LENGTH OF THE SIGNAL VECTOR
C       J      - INDEX VARIABLE
C       SIG    - SIGNAL VECTOR
C       PI     - THE VALUE OF PI
C
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      INTEGER  N,J
      DOUBLE PRECISION  PI,TEMP,SIG(N)
C
      PI=DACOS(DBLE(-1.0))
C
      DO 10 J=1,N
         TEMP=DBLE(0.54-0.46*(DCOS(2.0*PI*(J-1)/(N-1))))
         SIG(J)=SIG(J)*TEMP
  10  CONTINUE
C
      RETURN
      END
```

```
C*********************************************************************
      SUBROUTINE DFOUR2(DR,DI,N,ISIGN,VR,VI,IFACT)
C*********************************************************************
C
C     FILE NAME: DFOUR2.FOR
C
C     DOUBLE PRECISION VERSION. THIS VERSION IS A MODIFIED VERSION
C     OF THE DFT SUBROUTINE DFOURG.FOR.
C
C     VARIABLES:
C       DR      - REAL PART OF THE DATA TO BE TRANSFORMED
C       DI      - IMAGINARY PART OF THE DATA TO BE TRANSFORMED
C       N       - NUMBER OF POINTS TO BE USED IN THE TRANSFORM
C       ISIGN   - DIRECTION OF THE TRANSFORM
C                   -1 => FORWARD TRANSFORM
C                    1 => REVERSE TRANSFORM
C       VR      - REAL PART OF TEMPORARY STORAGE VECTOR
C       VI      - IMAGINARY PART OF TEMPORARY STORAGE VECTOR
C*********************************************************************
C
         IMPLICIT DOUBLE PRECISION (A-H,O-Z)
         IMPLICIT INTEGER (I-N)
       DIMENSION DR(N),DI(N),VR(N),VI(N),IFACT(N)
       PI=DACOS(DBLE(-1.0))
       TWOPI=2.D0*PI*DBLE(ISIGN)
       IF=0
       NPART=N
       DO 50 ID=1,N,2
       IDIV=ID
       IF (ID-1) 10,10,20
 10    IDIV=2
 20    IQUOT=NPART/IDIV
       IF (NPART-IDIV*IQUOT) 40,30,40
 30    IF=IF+1
       IFACT(IF)=IDIV
       NPART=IQUOT
       GO TO 20
 40    IF (IQUOT-IDIV) 60,60,50
 50    CONTINUE
 60    IF (NPART-1) 80,80,70
 70    IF=IF+1
       IFACT(IF)=NPART
 80    NFACT=IF
       IP0=1
       IP3=IP0*N
       IWORK=1
       I3REV=1
       DO 110 I3=1,IP3,IP0
       VR(IWORK)=DR(I3REV)
       VI(IWORK)=DI(I3REV)
       IP2=IP3
       DO 100 IF=1,NFACT
       IP1=IP2/IFACT(IF)
       I3REV=I3REV+IP1
       IF (I3REV-IP2) 110,110,90
 90    I3REV=I3REV-IP2
 100   IP2=IP1
 110   IWORK=IWORK+IP0
       IWORK=1
       DO 120 I3=1,IP3,IP0
       DR(I3)=VR(IWORK)
       DI(I3)=VI(IWORK)
 120   IWORK=IWORK+IP0
       IF=0
       IP1=IP0
 130   IF (IP1-IP3) 140,240,240
 140   IF=IF+1
```

```
      IFCUR=IFACT(IF)
      IP2=IP1*IFCUR
      THETA=TWOPI/DBLE(IFCUR)
      SINTH=DSIN(THETA/2.D0)
      ROOTR=-2.D0*SINTH*SINTH
      ROOTI=DSIN(THETA)
      THETA=TWOPI/DBLE(IP2/IP0)
      SINTH=DSIN(THETA/2.D0)
      WSTPR=-2.D0*SINTH*SINTH
      WSTPI=DSIN(THETA)
      WMINR=1.D0
      WMINI=0.D0
      DO 230 I1=1,IP1,IP0
      IF (IFCUR-2) 150,150,170
  150 DO 160 I3=I1,IP3,IP2
      J0=I3
      J1=I3+IP1
      TEMPR=WMINR*DR(J1)-WMINI*DI(J1)
      TEMPI=WMINR*DI(J1)+WMINI*DR(J1)
      DR(J1)=DR(J0)-TEMPR
      DI(J1)=DI(J0)-TEMPI
      DR(J0)=DR(J0)+TEMPR
  160 DI(J0)=DI(J0)+TEMPI
      GO TO 220
  170 IWMAX=IP0*IFCUR
      DO 210 I3=I1,IP3,IP2
      I2MAX=I3+IP2-IP1
      WR=WMINR
      WI=WMINI
      DO 200 IWORK=1,IWMAX,IP0
      I2=I2MAX
      SUMR=DR(I2)
      SUMI=DI(I2)
  180 I2=I2-IP1
      TEMPR=SUMR
      SUMR=WR*SUMR-WI*SUMI+DR(I2)
      SUMI=WR*SUMI+WI*TEMPR+DI(I2)
      IF (I2-I3) 190,190,180
  190 VR(IWORK)=SUMR
      VI(IWORK)=SUMI
      TEMPR=WR
      WR=WR*ROOTR-WI*ROOTI+WR
  200 WI=TEMPR*ROOTI+WI*ROOTR+WI
      IWORK=1
      DO 210 I2=I3,I2MAX,IP1
      DR(I2)=VR(IWORK)
      DI(I2)=VI(IWORK)
  210 IWORK=IWORK+IP0
  220 TEMPR=WMINR
      WMINR=WMINR*WSTPR-WMINI*WSTPI+WMINR
  230 WMINI=TEMPR*WSTPI+WMINI*WSTPR+WMINI
      IP1=IP2
      GO TO 130
  240 CONTINUE
C
C     THIS NEXT PART DIVIDES BY N REAL AND IMAGINARY PARTS BY N
C
      Q=1.D0
      IF (ISIGN.EQ.-1) Q=1.D0/DBLE(N)
      DO 35 J=1,N
      DR(J)=DR(J)*Q
   35 DI(J)=DI(J)*Q
C
      RETURN
      END
```

```
C*****************************************************************
C     DOLPH-CHEBYSHEV WINDOW TEST PROGRAM
C*****************************************************************
C
C     FILE NAME: DCTST1.FOR
C     CREATED: 19940322
C     CREATED BY: KERRY ELLIS
C
C     THIS PROGRAM IS USED TO GENERATED A DOLPH-CHEBYSHEV WINDOW
C     OF A GIVEN SIZE N (SAMPLES) AND HAVING SIDELOBES 20*ALPHA
C     DOWN FROM THE MAIN LOBE. WINDOW SAMPLES ARE CALCULATED USING
C     THE TECHNIQUE OF: A.H. NUTTAL, "GENERATION OF DOLPH-CHEBYSHEV
C     WEIGHTS VIA A FAST FOURIER TRANSFORM", PROCEEDINGS OF THE
C     IEEE, VOL. 62, PG. 1396, 1974. THE WINDOW SAMPLES GENERATED BY
C     THIS PROGRAM ARE STORED IN THE FILE NAME JUNK.DAT. ONCE THE
C     WINDOW COEFFICIENTS ARE WRITTEN TO THE OUTPUT FILE, THE
C     DATA SEQUENCE IS PADDED TO A LENGTH OF 1024 SAMPLES AND THE
C     AMPLITUDE SPECTRUM OF THE WINDOW IS COMPUTED. SPECTRAL
C     INFORMATION IS WRITTEN TO THE OUTPUT FILE JUNK1.DAT.
C
C     VARIABLES:
C       N        - NUMBER OF SAMPLES IN THE WINDOW.
C       NN       - NUMBER OF SAMPLES TO WHICH THE SEQUENCE IS PADDED
C                  FOR SPECTRAL COMPUTATIONS.
C       I        - INDEX VARIABLE.
C       IFACT    - TEMPORARY STORAGE ARRAY USED BY DFT SUBROUTINE.
C       DR       - ARRAY CONTAINING THE REAL PART OF THE DATA.
C       DI       - ARRAY CONTAINING THE IMAGINARY PART OF THE DATA.
C       VR,VI    - TEMPORARY ARRAYS FOR REAL AND IMAGINARY PARTS.
C       MAX      - LARGEST AMPLITUDE IN THE SPECTRUM. USED FOR
C                  SPECTRAL NORMALIZATION.
C       FRQ      - NYQUIST FREQUENCY. USED FOR FREQUENCY
C                  NORMALIZATION.
C       ALPHA    - WINDOW PARAMETER CONTROLLING THE PEAK SIDELOBE
C                  LEVEL.
C
C*****************************************************************
C
      INTEGER N,I,IFACT(10000),NN
      DOUBLE PRECISION DR(10000),DI(10000),VR(10000),VI(10000)
      DOUBLE PRECISION MAX,ALPHA,FRQ
C
C-------------------- ----------------------------------------------
C     INITIALIZE VARIABLES AND ARRAYS.
C-----------------------------------------------------------------
C
      NN=1024
C
      DO 10 I=1,NN
         IFACT(I)=0
         DR(I)=0.0D0
         DI(I)=0.0D0
         VR(I)=0.0D0
         VI(I)=0.0D0
  10  CONTINUE
C
C-----------------------------------------------------------------
C     GENERATE THE WINDOW SAMPLES AND WRITE THEM TO THE OUTPUT FILE
C     JUNK.DAT.
C-----------------------------------------------------------------
C
      N=50
      ALPHA=2.0D0
C
      CALL DCWIN3(DR,DI,VR,VI,IFACT,ALPHA,N)
C
```

```
C-----------------------------------------------------------------------
C     WRITE THE WINDOW COEFFICIENTS TO THE OUTPUT DATA FILE.
C-----------------------------------------------------------------------
      OPEN(UNIT=1,FILE='JUNK.DAT')
C
      DO 12 I=1,N
         WRITE(1,200) I-1,DR(I)
  12  CONTINUE
C
      CLOSE(1)
C-----------------------------------------------------------------------
C     COMPUTE THE AMPLITUDE SPECTRUM OF THE WINDOW.
C-----------------------------------------------------------------------
C
      N=1024
C
      DO 15 I=1,N
         DI(I)=0.0D0
         VR(I)=0.0D0
         VI(I)=0.0D0
         IFACT(I)=0
  15  CONTINUE
C
      CALL DFOUR2(DR,DI,N,-1,VR,VI,IFACT)
C
      DO 17 I=1,N
         DR(I)=DSQRT(DR(I)**2+DI(I)**2)
  17  CONTINUE
C
C-----------------------------------------------------------------------
C     NORMALIZE THE AMPLITUDES OF THE SPECTRAL COMPONENTS.
C-----------------------------------------------------------------------
      MAX=DR(1)
C
      DO 18 I=1,N
         IF (DR(I).GT.MAX) THEN
            MAX=DR(I)
         ENDIF
  18  CONTINUE
C
      DO 19 I=1,N
         DR(I)=DR(I)/MAX
C
         IF (DR(I).GT.0.0D0) THEN
            DR(I)=20.0D0*DLOG10(DR(I))
         ELSE
            DR(I)=-100.0D0
         ENDIF
C
  19  CONTINUE
C-----------------------------------------------------------------------
C     WRITE THE AMPLITUDE SPECTRUM TO THE OUTPUT FILE. FREQUENCY
C     SCALING IS ALSO DONE AT THIS TIME.
C-----------------------------------------------------------------------
      OPEN(UNIT=1,FILE='JUNK1.DAT')
C
      DO 20 I=1,N
         FRQ=DBLE(I-1)/DBLE(N/2)
         WRITE(1,100) FRQ,DR(I)
  20  CONTINUE
C
      CLOSE(1)
C
  100 FORMAT(F16.6,5X,F16.6)
  200 FORMAT(I4,5X,F16.6)
C
      END
```

```
C*********************************************************************
      SUBROUTINE DCWIN3(DR,DI,VR,VI,IFACT,ALPHA,N)
C*********************************************** *******************
C
C     FILE NAME: DCWIN3.FOR
C     CREATED: 19940310
C     CREATED BY: KERRY ELLIS
C
C     THIS PROGRAM GENERATES A DOLPH-CHEBYCHEV WINDOW.
C
C     VARIABLES:
C       N       - SEQUENCE LENGTH (SAMPLES).
C       K       - FREQUENCY INDEX VARIABLE.
C       DR      - DATA ARRAY CONTAINING REAL COMPONENT.
C       DI      - DATA ARRAY CONTAINING IMAGINARY COMPONENT.
C       VR      - TEMPORARY ARRAY CONTAINING REAL COMPONENT.
C       VI      - TEMPORARY ARRAY CONTAINING IMAGINARY COMPONENT.
C       IFACT   - TEMPORARY STORAGE ARRAY.
C       ALPHA   - RATIO OF MAIN TO PEAK SIDELOBE LEVEL IN DECADES.
C       BETA    - WINDOW PARAMETER RELATED TO ALPHA AND THE SEQUENCE
C                 LENGTH.
C       MAX     - AMPLITUDE SCALING FACTOR.
C       NUM     - NUMERATOR VALUE.
C       DEN     - DENOMINATOR VALUE.
C       ARG1-3  - INTERMEDIATE VARIABLES FOR FUNCTION ARGUMENTS.
C       PI      - THE VALUE OF PI.
C       TEMP    - TEMPORARY STORAGE VARIABLE.
C
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      INTEGER  N,K,IFACT(N),INDX,INDX1
      DOUBLE PRECISION  DR(N),DI(N),VR(N),VI(N),ALPHA
      DOUBLE PRECISION  Z0,Z,MAX,ARG1,PI,TEMP,TN
C
      PI=DACOS(-1.0D0)
C
C---------------------------------------------------------------------
C      COMPUTE BETA VALUE.
C---------------------------------------------------------------------
C
      TEMP=10.0D0**ALPHA
      ARG1=DLOG(TEMP+DSQRT((TEMP**2)-1.0D0))
      ARG1=ARG1/DBLE(N-1)
      Z0=DCOSH(ARG1)
C
      DO 10 K=1,(N/2)
         Z=Z0*DCOS((PI*DBLE(K-1))/DBLE(N))
C
         IF (K.EQ.1) THEN
            EPS=0.5D0
         ELSE
            EPS=1.0D0
         ENDIF
C
         IF (DABS(Z).LE.1.0D0) THEN
            TEMP=(PI/2.0D0)-DATAN(Z/DSQRT(1.0D0-(Z**2)))
            TEMP=DBLE(N-1)*TEMP
            TN=DCOS(TEMP)
         ELSE
            ARG1=Z+DSQRT((Z**2)-1.0D0)
C
            IF (ARG1.GT.0.0D0) THEN
               TEMP=DLOG(ARG1)
               TEMP=DBLE(N-1)*TEMP
               TN=DCOSH(TEMP)
            ELSE
               TEMP=-1.0D0*DLOG(DABS(ARG1))
```

```
               TEMP=DBLE(N-1)*TEMP
               TN=(DEXP(TEMP)+DEXP(-1.0D0*TEMP))/(-2.0D0)
            ENDIF
         ENDIF
C
         DR(K)=EPS*TN*DCOS(PI*DBLE(K-1)/DBLE(N))/DBLE(N/2)
         DI(K)=EPS*TN*DSIN(PI*DBLE(K-1)/DBLE(N))/DBLE(N/2)
C
  10  CONTINUE
C
C-----------------------------------------------------------------------
C     COMPUTE THE DFT OF THE WINDOW TO OBTAIN THE WINDOW SAMPLES IN
C     THE TIME DOMAIN.
C-----------------------------------------------------------------------
C
      CALL DFOUR2(DR,DI,N/2,-1,VR,VI,IFACT)
C
C-----------------------------------------------------------------------
C     NORMALIZE THE REAL PART OF THE WINDOW SAMPLES BY THE VALUE
C     CONTAINED IN THE FIRST ELEMENT OF THE TRANSFORMED ARRAY  THIS
C     ELEMENT REPRESENTS THE WEIGHTING FACTOR FOR THE CENTER ELEMENT
C     OF THE ARRAY.
C-----------------------------------------------------------------------
C
      MAX=DR(1)
C
      DO 20 K=1,N
         DR(K)=DR(K)/MAX
  20  CONTINUE
C
C-----------------------------------------------------------------------
C     REORGANIZE THE ARRAY ELEMENTS AND PUT THEM IN THE PROPER
C     ORDER FOR THE WINDOW.
C-----------------------------------------------------------------------
C
      DO 30 K=1,(N/2)
C
         IF ((2*K).LE.(N/2)) THEN
            INDX=(N/2)+(2*K)
            DR(INDX)=DR(K+1)
         ENDIF
C
         IF ((2*K-1).LE.(N/2)) THEN
            INDX=(N/2)+(2*K-1)
            IF (((N/2)+1-K).EQ.(N/2)) THEN
               DR(INDX)=DR(1)
            ELSE
               DR(INDX)=DR((N/2)+2-K)
            ENDIF
         ENDIF
C
  30  CONTINUE
C
      DO 40 K=1,(N/2)
         INDX=(N/2) K+1
         INDX1=(N/2)+K
         DR(INDX)=DR(INDX1)
  40  CONTINUE
C
      RETURN
      END
```

Appendix C

Appended Material for Chapter 6

```
/*****************************************************************
/*                                                               */
/*    File Name: rddata4.c                                       */
/*    Created: 19931006                                          */
/*                                                               */
/*    This program is used to collect data from the main receiver */
/*    output channel. Signals on the main channel are sampled at */
/*    a rate of 200 kHz. Note connection is made through channel 0 */
/*    on the terminal board for the A/D converter. Data collected */
/*    using this program is written in binary format to a file   */
/*    whose name has the format yymmdd.xxx, where xxx denotes the */
/*    file number. The length of each file is 20,008 bytes.      */
/*                                                               */
/*    This program was taken from documents supplied by the Data */
/*    Translation company and was adapted for this application.  */
/*                                                               */
/*****************************************************************/

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <dos.h>
#include <math.h>
#include <alloc.h>

#define  TRUE   1
#define  FALSE  0

#define  LO(x)   x & 0xFF
#define  HI(x)   x >> 8
#define  ErrorSet    (((adcsr=inport(ADCSR)) & BIT15) == BIT15)
#define  DmaDone     (((supcsr=inport(SUPCSR)) & BIT15) == BIT15)

#define  BIT0         0x0001
#define  BIT1         0x0002
#define  BIT2         0x0004
#define  BIT3         0x0008
#define  BIT4         0x0010
#define  BIT5         0x0020
#define  BIT6         0x0040
#define  BIT7         0x0080
#define  BIT8         0x0100
#define  BIT9         0x0200
#define  BIT10        0x0400
#define  BIT11        0x0800
#define  BIT12        0x1000
#define  BIT13        0x2000
#define  BIT14        0x4000
#define  BIT15        0x8000

#define  BASE         0x240
#define  ADCSR        BASE+0x0
#define  CHANCSR      BASE+0x2
#define  SUPCSR       BASE+0xC
#define  TMRCTR       BASE+0xE

#define  PAGE_REGS    0x8B
#define  OFFS_REGS    0xC4
#define  COUNT_REGS   0xC6
#define  MASK_REG     0xD4
#define  MODE_REG     0xD6
#define  FF_REG       0xD8

#define  SCALE        0.004882812
#define  OFFSET       10.0

#define  SIZE         13714
```

```
#define  NBUFFS      10
#define  NFILES      10
#define  RATE        0x00EB   /* Sampling Rate= 200 kHz */

typedef unsigned char  BYTE;
typedef unsigned int   WORD;

void GetDMAInfo(WORD bufsize,int far *buffer,WORD *DMAPage,WORD *DMAbase);
void InitDMA5(WORD page,WORD offset,WORD count);
void alocbuff(int far **buffer,WORD buffsize);
void stopall(void);
int InitBoard(void);
int GetData(int far *dataptr,struct date ymd,struct time hms);
WORD supcsr,adcsr;
int numpts;
FILE *OutFile;
int far *dataptr_a[10];
int set_clock();
void file_name(char *fname,struct date ymd,int value);
WORD getDMAlength(int far *buffer);
char fname[11];

void main()
  {
     int i,fcount;
     WORD DMAabase;
     WORD DMAaPage;
     WORD temp;
     struct date ymd[10];
     struct time hms[10];

     /*-------------------------------------------------------------*/
     /*   Initialize variables.                                     */
     /*-------------------------------------------------------------*/

     fcount=0;
     numpts=SIZE;

     /*--------- ----------------------------------------------------*/
     /*   Set up the internal sample clock to digitize the data at */
     /*   a rate of 200 kHz.                                        */
     /*-------------------------------------------------------------*/

     if(set_clock())
       exit(2);

     /*-------------------------------------------------------------*/
     /*   Allocate enough memory for 10 data buffers.               */
     /*-------------------------------------------------------------*/

     for(i=0;i<10;i++)
        alocbuff(&dataptr_a[i],numpts);

     /*-------------------------------------------------------------*/
     /*   Start data collection.                                    */
     /*-------------------------------------------------------------*/

     while(fcount < NFILES)
       {
         i=0;

         while(i < NBUFFS)
           {
             GetDMAInfo(numpts,dataptr_a[i],&DMAaPage,&DMAabase);

             InitDMA5(DMAaPage,DMAabase,numpts);
```

```
            if(InitBoard())
              {
                printf("\n ERROR Initializing BOARD !!! ");
                free(dataptr_a[i]);
                exit(5);
              }

            if((adcsr & BIT15)==BIT15)
              {
                printf("\n Error detected ADCSR:%X
                            SUPCSR:%X",adcsr,supcsr);
                stopall();
                exit(6);
              }

            while(!DmaDone);

            outport(SUPCSR,BIT13 | BIT10);

            outport(ADCSR,BIT0);

            getdate(&ymd[i]);
            gettime(&hms[i]);

            i++;

          }

        for(i=0;i<NBUFFS;i++)
           {
             fcount++;

             file_name(fname,ymd[i],fcount);

             if(GetData(dataptr_a[i],ymd[i],hms[i]))
               {
                 printf("\n Error Collecting Data !!!");
                 free(dataptr_a[i]);
                 exit(7);
               }
           }
         printf("\n fcount= %d",fcount);
       }

    for(i=0;i<10;i++)
       free(dataptr_a[i]);
  }

/*********************************************************************/
/*                                                                   */
/*    Function Name: stopall                                         */
/*                                                                   */
/*    This function halts the operation of the DT2821 board.         */
/*                                                                   */
/*********************************************************************/

void stopall(void)
  {
    printf("\n Error Detected !!");
    outport(SUPCSR,0);
  }

/*********************************************************************/
/*                                                                   */
/*    Function Name: alocbuff                                        */
/*                                                                   */
/*    This function allocates a buffer for storing the data          */
```

```
/*    collected by the DT2821 A/D board.                              */
/*                                                                    */
/*--------------------------------------------------------------------*/

void alocbuff(int far **buffer,WORD buffsize)
   {
      WORD oldsize;
      int *dummyptr;

      if((*buffer=(int far *)malloc((numpts)*sizeof(WORD))) == NULL)
        {
           printf("\n Error Allocating Memory !!!");
           exit(3);
        }

      if((oldsize=getDMAlength(*buffer)) < buffsize)
        {
           free(*buffer);

           /*-------------------------------------------------------*/
           /*  Allocate a dummy buffer to align the next block      */
           /*  with a DMA page. NOTE that the pointer to this       */
           /*  buffer is not maintained by this program thus any    */
           /*  memory allocated is lost to the program.             */
           /*-------------------------------------------------------*/

           dummyptr=malloc(oldsize*sizeof(WORD));

           if((*buffer=(int far *)malloc((numpts)*sizeof(WORD))) == NULL)
             {
                printf("\n Error Allocating Memory !!!");
                exit(3);
             }
        }
   }

/*--------------------------------------------------------------------*/
/*                                                                    */
/*    Function: getDMAlength                                          */
/*                                                                    */
/*    This function calculates the usable length of the DMA buffer    */
/*                                                                    */
/*--------------------------------------------------------------------*/

WORD getDMAlength(int far *buffer)
   {
      long absadr;
      WORD dmastart;

      /*-------------------------------------------------------*/
      /*   Calculate the absolute address of the buffer.       */
      /*-------------------------------------------------------*/

      absadr=((long)FP_SEG(buffer)<<4)+FP_OFF(buffer);

      /*-------------------------------------------------------*/
      /*   Calculate DMA offset address of buffer.             */
      /*-------------------------------------------------------*/

      dmastart=(WORD) (absadr>>1) & 0xFFFF;

      /*-------------------------------------------------------*/
      /*   Return the number of samples before the end of buffer   */
      /*-------------------------------------------------------*/

      return((WORD)(0x10000L-dmastart));
```

```
}

/*****************************************************************/
/*                                                               */
/*    Function Name: GetDMAInfo                                  */
/*                                                               */
/*    This function computes the DMA page number and base address */
/*    of the data buffer for use by the DMA controller.          */
/*                                                               */
/*****************************************************************/

void GetDMAInfo(WORD bufsize,int far *buffer,WORD *DMAPage,WORD
                *DMAbase)
   {
      unsigned long AbsAddr,maxsize;
      unsigned int  OffsAddr,SegAddr;

      /*-------------------------------------------------------*/
      /*   Compute the segment and offset for the data buffer. */
      /*-------------------------------------------------------*/

      SegAddr=FP_SEG(buffer);
      OffsAddr=FP_OFF(buffer);

      /*-------------------------------------------------------*/
      /*   Calculate the absolute address of the buffer. First */
      /*   shift the segment left four and add it to the offset. */
      /*   This is the real absolute address. Now take the result */
      /*   and shift it right 1. This is done to simplify the  */
      /*   address calculations for the DMA controller.        */
      /*-------------------------------------------------------*/

      AbsAddr=(((unsigned long) SegAddr << 4) + OffsAddr) >> 1;

      /*-------------------------------------------------------*/
      /*   Calculate the DMA page. The page is the upper WORD of */
      /*   AbsAddr. Now, shift the page left one. Only even    */
      /*   numbered DMA pages are usable.                      */
      /*-------------------------------------------------------*/

      *DMAPage=((WORD)((AbsAddr & 0xFFFF0000) >> 16) << 1);

      /*-------------------------------------------------------*/
      /*   The 8237A DMA controller needs a base value. The lower */
      /*   WORD of the AbsAddr is the base.                    */
      /*-------------------------------------------------------*/

      *DMAbase=(WORD)(AbsAddr & 0xFFFF);

      /*-------------------------------------------------------*/
      /*   Make sure that the buffer does not cross a 64K WORD */
      /*   DMA page.                                           */
      /*-------------------------------------------------------*/

      maxsize=0x10000L-*DMAbase;

      if(bufsize>maxsize)
        {
          printf("\n ERROR: Buffer Crosses DMA Page Boundary");
          exit(20);
        }
   }

/*****************************************************************/
/*                                                               */
/*    Function Name: InitDMA5                                    */
/*                                                               */
```

```
/*    This function initializes the DMA controller.                */
/*                                                                 */
/*-----------------------------------------------------------------*/

void InitDMA5(WORD page,WORD offset,WORD count)
   {
      outportb(MODE_REG,0x45);          /* Load DMA Mode Register */

      outportb(FF_REG,0x00);           /* Clear Byte Pointer FF */

      outportb(PAGE_REG5,LO(page));      /* Load DMA page number  */

      outportb(OFFS_REG5,LO(offset));    /* Load offset low byte */
      outportb(OFFS_REG5,HI(offset));    /* Load offset high byte */

      outportb(COUNT_REG5,LO(count-1)); /* Load count low byte */
      outportb(COUNT_REG5,HI(count-1)); /* Load count high byte */

      outportb(MASK_REG,0x01);          /* Enable DMA Channel #5 */

   }

/*-----------------------------------------------------------------*/
/*                                                                 */
/*    Function Name: InitBoard                                     */
/*                                                                 */
/*    This function is used to initialize the DT2821 board.        */
/*                                                                 */
/*-----------------------------------------------------------------*/

int InitBoard()
   {
      /*-------------------------------------------------------------*/
      /*   Initialize A/D buffer. Set bits: A/D Initialize, Buffer */
      /*   B and clear DMA Done.                                     */
      /*-------------------------------------------------------------*/

      outport(SUPCSR,BIT6 | BIT9 | BIT13);

      if(ErrorSet)
        return(TRUE);

      /*-------------------------------------------------------------*/
      /*   Enable writes to the channel/gain list and then load      */
      /*   the number of channels to be used.                        */
      /*-------------------------------------------------------------*/

      outport(CHANCSR,BIT15);

      if(ErrorSet)
        return(TRUE);

      /*-------------------------------------------------------------*/
      /*   Load the channel gain list with channel #0.               */
      /*-------------------------------------------------------------*/

      outport(ADCSR,0x0000);

      if(ErrorSet)
        return(TRUE);

      /*-------------------------------------------------------------*/
      /*   Disable writes to the channel/gain list by clearing       */
      /*   Bit 15.                                                   */
      /*-------------------------------------------------------------*/

      outport(CHANCSR,inport(CHANCSR) & 0x7FFF);
```

```
      if(ErrorSet)
        return(TRUE);

      /*-------------------------------------------------------------*
      /*   Enable A/D clock.                                         *
      /*-------------------------------------------------------------*

      outport(ADCSR,BIT9);

      if(ErrorSet)
        return(TRUE);

      /*-------------------------------------------------------------*
      /*   Preload Multiplexer and set A/D Clocked DMA.              */
      /*-------------------------------------------------------------*/

      outport(SUPCSR,BIT4 | BIT10);

      if(ErrorSet)
        return(TRUE);

      /*-------------------------------------------------------------*/
      /*   Wait for multiplexer busy bit to clear indicating that    */
      /*   the data is ready to convert to an analog value.          */
      /*-------------------------------------------------------------*/

      while((inport(ADCSR) & BIT8) == BIT8)

      if(ErrorSet)
        return(TRUE);

      /*-------------------------------------------------------------*/
      /*   Issue a hardware trigger.                                 */
      /*-------------------------------------------------------------*/

      outport(SUPCSR,BIT2 | BIT10);

      if(ErrorSet)
        return(TRUE);

      return(FALSE);

   }

/*********************************************************************/
/*                                                                   */
/*    Function Name:  GetData()                                      */
/*                                                                   */
/*    This function copies the data from the data buffer to disk.    */
/*    Data is stored in binary format.                               */
/*                                                                   */
/*********************************************************************/

int GetData(int far *dataptr,struct date ymd,struct time hms)
  {
     int i,yr;
     char mon,day;
     unsigned char hr,min,sec,hsec;

     yr=ymd.da_year;
     mon=ymd.da_mon;
     day=ymd.da_day;

     hr=hms.ti_hour;
     min=hms.ti_min;
     sec=hms.ti_sec;
```

```
      hsec=hms.ti_hund;

      OutFile=fopen(fname,"wb");

      fwrite(&yr,sizeof(yr),1,OutFile);
      fwrite(&mon,sizeof(mon),1,OutFile);
      fwrite(&day,sizeof(day),1,OutFile);
      fwrite(&hr,sizeof(hr),1,OutFile);
      fwrite(&min,sizeof(min),1,OutFile);
      fwrite(&sec,sizeof(sec),1,OutFile);
      fwrite(&hsec,sizeof(hsec),1,OutFile);

      for(i=3714;i<numpts;i++)
         fwrite(&dataptr[i],sizeof(dataptr[i]),1,OutFile);

      fclose(OutFile);

      return(FALSE);
   }

/*********************************************************************/
/*                                                                   */
/*    Function Name: set_clock                                       */
/*                                                                   */
/*    This function initializes the pacer clock.                     */
/*                                                                   */
/*********************************************************************/

int set_clock()
   {
      outport(SUPCSR,BIT0);

      if(ErrorSet)
        return(TRUE);

      outport(TMRCTR,RATE);

      if(ErrorSet)
        return(TRUE);

      return(FALSE);

   }
/*********************************************************************/
/*                                                                   */
/*    Function Name: file_name                                       */
/*    Created By: Kerry Ellis                                        */
/*                                                                   */
/*    This function creates the file name in which the data will     */
/*    be stored.                                                     */
/*********************************************************************/

void file_name(char *fname,struct date ymd,int value)
   {
      int i,val,yr;
      char ch1,ch2,ch3,str,day,mon;
      static char chtable[]="0123456789";
      unsigned char min,hr,sec,hsec;

      /*---------------------- ----------------------------------*/
      /*   Initialize year, month and day fields.                 */
      /*----------------------------------------------------------*/

      yr=ymd.da_year;
      mon=ymd.da_mon;
      day=ymd.da_day;
```

```
    /*-------------------------------------------------------------*/
    /*   Initialize the Year fields.                               */
    /*-------------------------------------------------------------*/

    val=yr-1900;

    ch1=(char)(val/10);
    ch2=(char)(val-ch1*10);

    fname[0]=chtable[ch1];
    fname[1]=chtable[ch2];

    /*-------------------------------------------------------------*/
    /*   Set up Month fields.                                      */
    /*-------------------------------------------------------------*/

    val=mon;

    ch1=(char)(val/10);
    ch2=(char)(val-ch1*10);

    fname[2]=chtable[ch1];
    fname[3]=chtable[ch2];

    /*-------------------------------------------------------------*/
    /*   Set up Day fields.                                        */
    /*-------------------------------------------------------------*/

    val=day;

    ch1=(char)(val/10);
    ch2=(char)(val-ch1*10);

    fname[4]=chtable[ch1];
    fname[5]=chtable[ch2];

    /*-------------------------------------------------------------*/
    /*   Set up extension indicator "."                            */
    /*-------------------------------------------------------------*/

    fname[6]='.';

    /*-------------------------------------------------------------*/
    /*   Set up file number fields.                                */
    /*-------------------------------------------------------------*/

    val=value;

    ch1=(char)(val/100);
    ch2=(char)((val-ch1*100)/10);
    ch3=(char)(val-ch1*100-ch2*10);

    fname[7]=chtable[ch1];
    fname[8]=chtable[ch2];
    fname[9]=chtable[ch3];
}
```

```
/*****************************************************************/
/*                                                               */
/*    File Name: pcald.c                                         */
/*    Created: 19940323                                          */
/*    Created By: Kerry Ellis                                    */
/*                                                               */
/*    Calibration program using a Dolph-Chebyshev window.        */
/*    This program measures the power of the largest spectral    */
/*    peak in the received signal and writes this power and its  */
/*    corresponding frequency to the specified output file.      */
/*                                                               */
/*    Note: The name of the data file containing the window      */
/*          coeffficients is "hard coded" and must be changed    */
/*          each time a different window is used. The window     */
/*          data file is created by the FORTRAN program DCGEN.FOR. */
/*                                                               */
/*    Variables/Parameters:                                      */
/*      data      - variable into which binary data is read.     */
/*      mth       - contains month.                              */
/*      day       - contains year.                               */
/*      hr        - contains hour.                               */
/*      min       - contains minute.                             */
/*      sec       - contains second.                             */
/*      hsec      - contains hundredths of seconds.              */
/*      yr        - contains year.                               */
/*      i         - index variable.                              */
/*      j         - index variable.                              */
/*      fp1,fp2   - file pointers.                               */
/*      len       - length of file date string.                  */
/*      rec_start- starting record number.                       */
/*      rec_stop  - stopping record number.                      */
/*      nbytes    - number of bytes to be allocated.             */
/*      mem1      - amount of memory free before allocation.     */
/*      mem2      - amount of memory free after allocation.      */
/*      indx      - auxiliary array index variable.              */
/*      value     - floating point value corresponding to the binary */
/*                  value contained in the variable data.        */
/*      pwr       - power level of the largest spectral peak (dBm). */
/*      frq       - frequency of the largest spectral peak (Hz). */
/*      array     - pointer to the array containing the signal data. */
/*      window    - pointer to the array containing the window data. */
/*      name1     - contains the input file name prefix.         */
/*      fname1    - contains the complete input file name.       */
/*      name2     - contains the name of the output file.        */
/*                                                               */
/*****************************************************************/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <alloc.h>
#include <math.h>

#include "four2.c"
#include "pwrspec.c"
#include "f_name.c"

#define  NPNTS       32768L
#define  Data_Size   10000L

#define  FS        200000.0

#define  SCALE   0.004882812
#define  OFFSET  10.0


void main()
```

```
{
  int  data,yr,i,rec_start,rec_stop,len;
  unsigned long  j,nbytes,mem1,mem2,indx;
  char name1[30],fname1[30],name2[30],fname2[30],mth,day;
  unsigned char hr,min,sec,hsec;
  float value,pwr,frq;
  float huge *array;
  float huge *window;
  FILE  *fp1,*fp2;

  void f_name(char *fname,int value,int len);
  void four2(float huge *array,long nn,int isign);
  void pwrspec(float huge *array,long n);

  /*-------------------------------------------------------------*/
  /*  Allocate enough memory for storing the entire array in     */
  /*  memory. Be sure to check that the allocation was successful*/
  /*  before continuing on. It is necessary to check the amount  */
  /*  of heap space allocated instead of only checking to see if */
  /*  the pointer returned by farmalloc is null since the        */
  /*  returned pointer can be ok but the memory hasn't been      */
  /*  allocated.                                                 */
  /*-------------------------------------------------------------*/

  mem1=farcoreleft();

  nbytes=((unsigned long)sizeof(float))*(2*NPNTS);
  array=(float huge *)farmalloc(nbytes);

  mem2=farcoreleft();

  printf("\n %lu %lu",mem1,mem2);

  if ((mem1-mem2) >= nbytes)
     printf("\n Memory Allocation Successful !");
  else
     {
       printf("\n Memory Allocation Unsuccessful !");
       exit(1);
     }

  for(j=0;j<(2*NPNTS);++j)
      array[j]=(float)j;

  for(j=0;j<(2*NPNTS);++j)
     {
       if(array[j]!=((float)j))
         printf("\n Error !");
     }

  printf("\n Memory Test Successful");

  /*-------------------------------------------------------------*/
  /*   Allocate memory for storing the window samples. This      */
  /*   array need only be big enough to store real values ie.    */
  /*   don't allocate space for an imaginary component.          */
  /*-------------------------------------------------------------*/

  mem1=farcoreleft();

  nbytes=((unsigned long)sizeof(float))*Data_Size;
  window=(float huge *)farmalloc(nbytes);

  mem2=farcoreleft();

  printf("\n %lu %lu",mem1,mem2);
```

```
if ((mem1-mem2) >= nbytes)
   printf("\n Window Memory Allocation Successful !");
else
   {
     printf("\n Window Memory Allocation Unsuccessful !");
     exit(1);
   }

for(j=0;j<Data_Size;++j)
    window[j]=(float)j;

for(j=0;j<Data_Size;++j)
   {
     if(window[j]!=((float)j))
       printf("\n Error !");
   }

printf("\n Window Memory Test Successful");

/*-------------------------------------------------------------*/
/*  Initialize the data and window arrays by setting the value */
/*  of all array elements to zero.                             */
/*-------------------------------------------------------------*/

for(j=0;j<(2*NPNTS);++j)
   array[j]=0.0;

for(j=0;j<Data_Size;++j)
   window[j]=0.0;

/*-------------------------------------------------------------*/
/*   Open the file containing the samples for the Dolph -      */
/*   Chebyshev window and read them into the window array.     */
/*-------------------------------------------------------------*/

fp1=fopen("dc40.dat","r");

for(j=0;j<Data_Size;++j)
   {
     fscanf(fp1,"%f \n",&value);
     window[j]=value;
   }

fclose(fp1);

/*-------------------------------------- --------------------*/
/*  Prompt for input file prefix as well as starting and      */
/*  stopping record numbers and output file name.             */
/*-------------------------------------------------------------*/

printf("\n Enter Input File Prefix: ");
scanf("%s",&name1);

printf("\n Enter Starting Record #: ");
scanf("%d",&rec_start);

printf("\n Enter Stopping Record #: ");
scanf("%d",&rec_stop);

printf("\n Enter Output File Name: ");
scanf("%s",&name2);

/*-------------- ----------------------------------------------*/
/*  Open data output file.                                     */
/*-------------------------------------------------------------*/

fp2=fopen(name2,"w");
```

```
/*-------------------------------------------------------------*/
/*  Start reading in the data files and process the data as    */
/*  necessary.                                                 */
/*-------------------------------------------------------------*/

for(i=rec_start;i<=rec_stop;i++)
   {
     for(j=0;j<(2*NPNTS);++j)
        array[j]=0.0;

     len=strlen(name1);
     strcpy(fname1,name1);
     f_name(fname1,i,len);

     printf("\n In-File: %s",fname1);

     fp1=fopen(fname1,"rb");

     if (fp1 != NULL)
        {
          /*-----------------------------------------------*/
          /*  Read date time information from the file.    */
          /*-----------------------------------------------*/

          fread(&yr,sizeof(yr),1,fp1);
          fread(&mth,sizeof(mth),1,fp1);
          fread(&day,sizeof(day),1,fp1);
          fread(&hr,sizeof(hr),1,fp1);
          fread(&min,sizeof(min),1,fp1);
          fread(&sec,sizeof(sec),1,fp1);
          fread(&hsec,sizeof(hsec),1,fp1);

          /*-----------------------------------------------*/
          /*  Read in the actual data in binary format and */
          /*  convert to floating point. The data values are */
          /*  then windowed and stored in array[].         */
          /*-----------------------------------------------*/

          for(j=0;j<Data_Size;++j)
             {
               indx=2*j;
               fread(&data,sizeof(data),1,fp1);
               value=(float)(data*SCALE-OFFSET);
               array[indx]=value*window[j];
               array[indx+1]=0.0;
             }

          fclose(fp1);

          /*-----------------------------------------------*/
          /*  Fourier Transform the data and compute the power */
          /*  spectrum of the signal. The real part of the */
          /*  data array now contains the power spectrum and */
          /*  the imaginary part has been set to zero.     */
          /*-----------------------------------------------*/

          four2(array,NPNTS,1);

          pwrspec(array,NPNTS);

          /*-----------------------------------------------*/
          /*  Scale the power spectrum into dBm.           */
          /*-----------------------------------------------*/

          for(j=0;j<NPNTS;++j)
             {
```

```
                indx=2*j;

                if(array[indx]>0.0)
                  array[indx]=10.0*log10(array[indx]/1.0e-3);
                else
                  array[indx]=-100.0;
              }

            /*-------------------------------------------------*/
            /*  Find the power level of the largest peak.      */
            /*-------------------------------------------------*/

            pwr=array[100];
            frq=100.0*FS/NPNTS;

            for(j=100;j<(NPNTS/2);++j)
               {
                 indx=2*j;

                 if(array[indx]>pwr)
                   {
                     pwr=array[indx];
                     frq=FS*j/NPNTS;
                   }
               }

            /*-------------------------------------------------*/
            /*  Write power and frequency to the output file.  */
            /*-------------------------------------------------*/

            fprintf(fp2,"Power= %f  Frequency= %f \n",pwr,frq);

          }
        else
          printf("\n File Error during Open !!");
      }

  farfree(array);
  farfree(window);
  fclose(fp2);

}
```

```
/*************************************************************************/
/*                                                                       */
/*    Function Name: f_name                                              */
/*    Created: 19940111                                                  */
/*    Created By: Kerry Ellis                                            */
/*                                                                       */
/*    This function creates the file name in which the data will         */
/*    be stored.                                                         */
/*                                                                       */
/*************************************************************************/

void f_name(char *fname,int value,int len)
  {
     int val;
     static char chtable[]="0123456789";
     char ch1,ch2,ch3;

     /*-------------------------------------------- --------------*/
     /*   Set up extension indicator "."                          */
     /*-------------------------------------------- - -----------*/

     fname[len]='.';

     /*-------------------------------------------------------------*/
     /*   Set up file number fields.                                */
     /*-------------------------------------------------------------*/

     val=value;

     ch1=(char)(val/100);
     ch2=(char)((val-ch1*100)/10);
     ch3=(char)(val-ch1*100-ch2*10);

     fname[len+1]=chtable[ch1];
     fname[len+2]=chtable[ch2];
     fname[len+3]=chtable[ch3];
     fname[len+4]='\0';
  }
```

```
/*****************************************************************/
/*                                                               */
/*    File Name: iqbin.c                                         */
/*    Created: 19931115                                          */
/*    Created By: Kerry Ellis                                    */
/*                                                               */
/*    This program reads multiplexed I,Q data from a binary file,*/
/*    demultiplexes it and stores it in real, imaginary form in  */
/*    an ASCII file.                                             */
/*                                                               */
/*    Parameters/Variables:                                      */
/*      fname1    - name of the binary file containing the data. */
/*      fname2    - name of the ASCII file to contain complex data.*/
/*      mth       - contains month.                              */
/*      day       - contains year.                               */
/*      hr        - contains hour.                               */
/*      min       - contains minute.                             */
/*      sec       - contains second.                             */
/*      hsec      - contains hundredths of seconds.              */
/*      idata     - I channel data.                              */
/*      qdata     - Q channel data.                              */
/*      yr        - contains year.                               */
/*      i         - index variable.                              */
/*      ivalue    - I channel value after scaling.               */
/*      qvalue    - Q channel value after scaling.               */
/*      fp1,fp2   - file pointers.                               */
/*                                                               */
/*****************************************************************/

#include <stdio.h>

#define  SCALE   0.004882812
#define  OFFSET  10.0

void main()
  {
     char fname1[30],fname2[30],mth,day;
     unsigned char hr,min,sec,hsec;
     int idata,qdata,yr;
     unsigned int i;
     double  ivalue,qvalue;
     FILE  *fp1,*fp2;

     printf("\n Enter Input File Name: ");
     scanf("%s",&fname1);

     printf("\n Enter Output File Name:");
     scanf("%s",&fname2);

     fp1=fopen(fname1,"rb");

     if (fp1 != NULL)
        {
           fread(&yr,sizeof(yr),1,fp1);
           fread(&mth,sizeof(mth),1,fp1);
           fread(&day,sizeof(day),1,fp1);
           fread(&hr,sizeof(hr),1,fp1);
           fread(&min,sizeof(min),1,fp1);
           fread(&sec,sizeof(sec),1,fp1);
           fread(&hsec,sizeof(hsec),1,fp1);

           printf("\n Year:   %d",yr);
           printf("\n Month:  %d",mth);
           printf("\n Day:    %d",day);
           printf("\n Hour:   %d",hr);
           printf("\n Min:    %d",min);
           printf("\n Sec:    %d",sec);
```

```
        printf("\n Nsec:   %d",nsec);

        fp2=fopen(fname2,"w");

        for(i=0;i<2500;i++)
           {
             fread(&idata,sizeof(idata),1,fp1);
             fread(&qdata,sizeof(qdata),1,fp1);
             ivalue=(double)(idata*SCALE-OFFSET);
             qvalue=(double)(qdata*SCALE-OFFSET);
             fprintf(fp2,"%lf      %lf \n",ivalue,qvalue);
           }

        fclose(fp1);
        fclose(fp2);

     }
   else
     printf("\n File Error during Open !!");
}
```

```
C*****************************************************************
C     RECEIVER POWER MEASUREMENT PROGRAM FOR I/Q CHANNELS
C*****************************************************************
C
C     FILE NAME: IQPWR2.FOR
C     CREATED: 19931213
C     CREATED BY: KERRY ELLIS
C
C     THIS PROGRAM MEASURES THE POWER OF THE LARGEST SPECTRAL PEAK
C     IN THE RECEIVED SIGNAL. THE POWER SPECTRUM IS COMPUTED AFTER
C     WINDOWING THE DATA WITH A DOLP-CHEBYSHEV WINDOW (ALPHA=2). IT
C     SHOULD BE NOTED THAT THE WINDOW SAMPLES ARE READ IN FROM A
C     FILE. THE SAMPLES THEMSELVES ARE CREATED USING THE PROGRAM
C     DCGEN.FOR.
C
C     VARIABLES:
C       N       - LENGTH OF TIME-SERIES (NUMBER OF SAMPLES).
C       J       - INDEX VARIABLE.
C       INDEX   - FREQUENCY INDEX OF THE LARGEST SPECTRAL PEAK.
C       IOS1    - I/O STATUS VARIABLE.
C       PAVG    - POWER OF THE LARGEST SPECTRAL COMPONENT.
C       DR,DI   - REAL AND IMAGINARY PARTS OF THE DATA ARRAY.
C       VR,VI   - REAL AND IMAGINARY PARTS OF THE TEMPORARY STORAGE
C                 ARRAY USED BY THE FFT SUBROUTINE.
C       WIN     - STORAGE ARRAY FOR THE DOLPH-CHEBYSHEV WINDOW
C                 COEFFICIENTS.
C       IFACT   - TEMPORARY ARRAY USED BY THE FFT SUBROUTINE.
C       FNAME1  - NAME OF THE FILE CONTAINING THE NAMES OF THE DATA
C                 FILES TO BE PROCESSED.
C       FNAME2  - NAME OF THE DATA FILE TO BE PROCESSED.
C       FNAME3  - NAME OF DATA OUTPUT FILE.
C
C*****************************************************************
C
      INTEGER  N,J,INDEX,IFACT(2500),IOS1,ICOUNT
      DOUBLE PRECISION  DR(2500),DI(2500),VR(2500),VI(2500)
      DOUBLE PRECISION  WIN(2500),PAVG,F,P
      DOUBLE PRECISION  MAG
      CHARACTER*32  FNAME1,FNAME2,FNAME3
C
C-----------------------------------------------------------------
C     DEFINE PROGRAM CONSTANTS
C-----------------------------------------------------------------
C
      N=2500
      ICOUNT=0
      PAVG=0.0D0
C
C-----------------------------------------------------------------
C     READ IN WINDOW DATA FROM THE FILE DC40IQ.DAT AND STORE IT IN
C     THE WIN ARRAY.
C-----------------------------------------------------------------
C
      OPEN(UNIT=1,FILE='DC40IQ.DAT')
C
      DO 5 J=1,N
         READ(1,300) WIN(J)
  5   CONTINUE
C
      CLOSE(1)
C
C-----------------------------------------------------------------
C     PROMPT USER FOR STORAGE FILE NAMES.
C-----------------------------------------------------------------
C
      PRINT *, 'INPUT FILE NAME ?'
      READ(6,100) FNAME1
```

```
C
      PRINT *, 'NAME OF OUTPUT FILE ?'
      READ(6,100) FNAME3
C
      OPEN(UNIT=1,FILE=FNAME1,STATUS='OLD')
C
      READ(1,100,IOSTAT=IOS1) FNAME2
C
      WHILE(IOS1.EQ.0,
C
        ICOUNT=ICOUNT+1
C
C-----------------------------------------------------------------
C       INITIALIZE ARRAYS.
C-----------------------------------------------------------------
C
        DO 10 J=1,N
           DR(J)=0.0D0
           DI(J)=0.0D0
           VR(J)=0.0D0
           VI(J)=0.0D0
           IFACT(J)=0
  10    CONTINUE

C-----------------------------------------------------------------
C       READ DATA FROM INPUT FILE AND STORE IN ARRAY DR() AND DI().
C-----------------------------------------------------------------
C
        OPEN(UNIT=2,FILE=FNAME2)
C
        DO 20 J=1,N
           READ(2,200) DR(J),DI(J)
  20    CONTINUE
C
        CLOSE(2)
C
C-----------------------------------------------------------------
C       APPLY THE DOLPH-CHEBYSHEV WINDOW TO THE DATA.
C-----------------------------------------------------------------
C
        DO 25 J=1,N
           DR(J)=DR(J)*WIN(J)
           DI(J)=DI(J)*WIN(J)
  25    CONTINUE
C
C-----------------------------------------------------------------
C       TRANSFORM DATA TO OBTAIN THE AMPLITUDE SPECTRUM OF THE SIGNAL.
C       THEN COMPUTE ITS POWER SPECTRUM.
C-----------------------------------------------------------------
C
        CALL DFOUR2(DR,DI,N,-1,VR,VI,IFACT)
C
        DO 30 J=1,N
           DR(J)=MAG(DR(J),DI(J))
           DR(J)=DR(J)**2
  30    CONTINUE
C
C-----------------------------------------------------------------
C       REMOVE FREQUENCY COMPONENTS CLOSE TO DC.
C-----------------------------------------------------------------
C
        DR(1)=0.0D0
        DR(2)=0.0D0
        DR(3)=0.0D0
        DR(4)=0.0D0
        DR(5)=0.0D0
        DR(6)=0.0D0
```

```
        DR(N)=0.0D0
        DR(N-1)=0.0D0
        DR(N-2)=0.0D0
        DR(N-3)=0.0D0
        DR(N-4)=0.0D0
C
C-----------------------------------------------------------------
C       NOW LOCATE THE SPECTRAL PEAK.
C-----------------------------------------------------------------
C
        CALL MAX2(DR,N,INDEX)
C
        P=DR(INDEX)
        P=10.0D0*DLOG10(P/1.0D-3)
        PAVG=PAVG+P
C
        IF (INDEX.LE.(N/2)) THEN
           F=(INDEX-1)*50.0D3/N
        ELSE
           F=(INDEX-N-1)*50.0D3/N
        ENDIF
C
        PRINT *, 'POWER= ',P,'FREQ= ',F
C
        READ(1,100,IOSTAT=IOS1) FNAME2
C
      ENDWHILE
C
C-----------------------------------------------------------------
C     WRITE AVERAGE VALUE TO FILE, CLOSE FILES AND TERMINATE
C     PROGRAM.
C-----------------------------------------------------------------
C
      OPEN(UNIT=3,FILE=FNAME3)
C
      PAVG=PAVG/DBLE(ICOUNT)
      WRITE(3,200) PAVG
C
      CLOSE(3)
C
      CLOSE(1)
C
C-----------------------------------------------------------------
C     PROGRAM COMPLETE !
C-----------------------------------------------------------------
C
  100 FORMAT(A32)
  200 FORMAT(F10.6,F10.6)
  300 FORMAT(F16.6)
C
      END
```

```
C*********************************************************************
      DOUBLE PRECISION FUNCTION MAG(RE,IM)
C*********************************************************************
C
C     FILE NAME: MAG.FOR
C     CREATED: 19911010
C     CREATED BY: KERRY ELLIS
C
C     THIS FUNCTION COMPUTES THE MAGNITUDE OF A COMPLEX NUMBER
C     SPECIFIED BY ITS REAL AND IMAGINARY COMPONENTS.
C
C     VARIABLES:
C       RE    -  REAL PART OF THE COMPLEX NUMBER
C       IM    -  IMAGINARY PART OF THE COMPLEX NUMBER
C
C*********************************************************************
C
      DOUBLE PRECISION   RE,IM
C
      MAG=DSQRT((RE*RE)+(IM*IM))
C
      END
```

```
C*********************************************************************
C     RECEIVER FREQUENCY MEASUREMENT PROGRAM FOR I/Q CHANNELS
C*********************************************************************
C
C     FILE NAME: IQFRQ.FOR
C     CREATED: 19931211
C     CREATED BY: KERRY ELLIS
C
C     THIS PROGRAM MEASURES THE FREQUENCY OF THE LARGEST AMPLITUDE
C     SPECTRAL COMPONENT IN THE SIGNAL.DATA IS WINDOWED USING A
C     DOLPH-CHEBYSHEV WINDOW (ALPHA=2). IT SHOULD BE NOTED THAT THE
C     WINDOW COEFFICIENTS ARE READ IN FROM AN ASCII FILE. THESE
C     VALUES ARE GENERATED BY THE FORTRAN PROGRAM DCGEN.FOR.
C
C     VARIABLES:
C       N       - LENGTH OF TIME-SERIES (NUMBER OF SAMPLES).
C       J       - INDEX VARIABLE.
C       INDEX   - FREQUENCY INDEX OF THE LARGEST SPECTRAL PEAK.
C       IOS1    - I/O STATUS VARIABLE.
C       FS      - SAMPLING RATE USED TO DIGITIZE THE RECEIVER OUTPUT.
C       F       - FREQUENCY OF THE SINUSOID(S) FOUND IN THE DIGITIZED
C                 RECORD.
C       DR,DI   - REAL AND IMAGINARY PARTS OF THE DATA ARRAY.
C       VR,VI   - REAL AND IMAGINARY PARTS OF THE TEMPORARY STORAGE
C                 ARRAY USED BY THE FFT SUBROUTINE.
C       WIN     - STORAGE ARRAY FOR THE DOLPH-CHEBYSHEV WINDOW
C                 COEFFICIENTS.
C       IFACT   - TEMPORARY ARRAY USED BY THE FFT SUBROUTINE.
C       FNAME1  - NAME OF THE FILE CONTAINING THE NAMES OF THE DATA
C                 FILES TO BE PROCESSED.
C       FNAME2  - NAME OF THE DATA FILE TO BE PROCESSED.
C       FNAME3  - NAME OF DATA OUTPUT FILE.
C
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      INTEGER  N,J,INDEX,IFACT(2500),IOS1,ICOUNT
      DOUBLE PRECISION  DR(2500),DI(2500),VR(2500),VI(2500)
      DOUBLE PRECISION  WIN(2500),F,FS,FAVG
      DOUBLE PRECISION  MAG
      CHARACTER*32  FNAME1,FNAME2,FNAME3
C
C---------------------------------------------------------------------
C     DEFINE PROGRAM CONSTANTS
C---------------------------------------------------------------------
C
      N=2500
      FS=50.0D3
      FAVG=0.0D0
      ICOUNT=0
C
C---------------------------------------------------------------------
C     READ IN THE WINDOW SAMPLES FROM THE DATA FILE DC40IQ.DAT.
C---------------------------------------- ----------------------------
C
      OPEN(UNIT=1,FILE='DC40IQ.DAT')
C
      DO 5 J=1,N
         READ(1,300) WIN(J)
  5   CONTINUE
C
      CLOSE(1)
C
C---------------------------------------------------------------------
C     PROMPT USER FOR STORAGE FILE NAMES.
C---------------------------------------------------------------------
C
      PRINT *, 'INPUT FILE NAME ?'
```

```
      READ(6,100) FNAME1
C
      PRINT *, 'NAME OF OUTPUT FILE ?'
      READ(6,100) FNAME3
C
      OPEN(UNIT=1,FILE=FNAME1,STATUS='OLD')
      OPEN(UNIT=3,FILE=FNAME3)
C
      READ(1,100,IOSTAT=IOS1) FNAME2
C
      WHILE(IOS1.EQ.0)
C
        ICOUNT=ICOUNT+1
C
C-----------------------------------------------------------------
C       INITIALIZE ARRAYS.
C-----------------------------------------------------------------
C
        DO 10 J=1,N
           DR(J)=0.0D0
           DI(J)=0.0D0
           VR(J)=0.0D0
           VI(J)=0.0D0
           IFACT(J)=0
  10    CONTINUE

C-----------------------------------------------------------------
C       READ DATA FROM INPUT FILE AND STORE IN ARRAY DR().
C-----------------------------------------------------------------
C
        OPEN(UNIT=2,FILE=FNAME2)
C
        DO 20 J=1,N
           READ(2,200) DR(J),DI(J)
           DI(J)=-1.0D0*DI(J)
  20    CONTINUE
C
        CLOSE(2)
C
C-----------------------------------------------------------------
C       APPLY THE DOLPH-CHEBYSHEV WINDOW TO THE DATA.
C-----------------------------------------------------------------
C
        DO 25 J=1,N
           DR(J)=DR(J)*WIN(J)
           DI(J)=DI(J)*WIN(J)
  25    CONTINUE
C
C-----------------------------------------------------------------
C       TRANSFORM DATA TO OBTAIN THE AMPLITUDE SPECTRUM OF THE SIGNAL.
C       THEN LOCATE THE PEAK AND UPDATE THE STATISTICS ON THE AVERAGE
C       FREQUENCY AND CORRESPONDING PROPAGATION TIME.
C-----------------------------------------------------------------
C
        CALL DFOUR2(DR,DI,N,-1,VR,VI,IFACT)
C
        DO 30 J=1,N
           DR(J)=MAG(DR(J),DI(J))
  30    CONTINUE
C
C-----------------------------------------------------------------
C       REMOVE FREQUENCY COMPONENTS CLOSE TO DC.
C-----------------------------------------------------------------
C
        DR(1)=0.0D0
        DR(2)=0.0D0
        DR(3)=0.0D0
```

```
        DR(4)=0.0D0
        DR(5)=0.0D0
        DR(6)=0.0D0
        DR(N)=0.0D0
        DR(N-1)=0.0D0
        DR(N-2)=0.0D0
        DR(N-3)=0.0D0
        DR(N-4)=0.0D0
C
C-----------------------------------------------------------------------
C       FIND THE SPECTRAL PEAK.
C-----------------------------------------------------------------------
C
        CALL MAX2(DR,N,INDEX)
C
        IF (INDEX.LE (N/2)) THEN
           F=(INDEX-1)*FS/N
        ELSE
           F=(INDEX-N-1)*FS/N
        ENDIF
C
        PRINT *,'INDEX= ',INDEX
        PRINT *,'FREQUENCY= ',F
        FAVG=FAVG+F
C
        IF (ICOUNT.EQ.10) THEN
           FAVG=FAVG/DBLE(ICOUNT)
           WRITE(3,300) FAVG
           FAVG=0.0D0
           ICOUNT=0
        ENDIF
C
        READ(1,100,IOSTAT=IOS1) FNAME2
C
      ENDWHILE
C
      CLOSE(1)
      CLOSE(3)
C
C-----------------------------------------------------------------------
C     PROGRAM COMPLETE !
C-----------------------------------------------------------------------
C
  100 FORMAT(A32)
  200 FORMAT(F10.6,F10.6)
  300 FORMAT(F16.6)
C
      END
```

The Quadrature Recording System

The function of the quadrature recording system is to permit a 40 kHz segment of the receiver output bandwidth to be recorded on two 20 kHz channels. To achieve this result, signals from the receiver output, v(t), are mixed with oscillator signals having the same frequency but differing in phase by 90 degrees as illustrated in Figure E.1. By selecting the oscillator frequency, f_c, to be at the center of the desired 40 kHz segment, the beat frequencies resulting from the mixing process range from 20 kHz, for components at the ends of the spectral range, to 0 Hz as the frequency of v(t) approaches that of the band center (see Figure E.2). If the signals from the I(t) and Q(t) channels are combined according to the equation:

$$g(t) = I(t) - jQ(t) \qquad \text{(E.1)}$$

the spectrum of the resulting complex signal, *g(t)*, will possess spectral characteristics identical to that of the input signal v(t). Since the signal *g(t)* is complex, its spectrum will be single-sided and will indicate the presence of both negative and positive frequency components depending on whether or not the frequency of v(t) is below or above f_c. When the frequency of v(t) is below f_c, negative frequencies are produced, and conversely, when the frequency of v(t) is above f_c, positive frequencies are produced. Given the frequencies contained in the spectrum of the complex signal *g(t)*, the true frequency of the components present in the input signal can be calculated according to the equation:

$$f = f_c + f_M \qquad \text{(E.2)}$$

where; f is the "true" frequency of the input signal, f_M is the frequency measured from the complex signal spectrum and f_c is as previously defined.

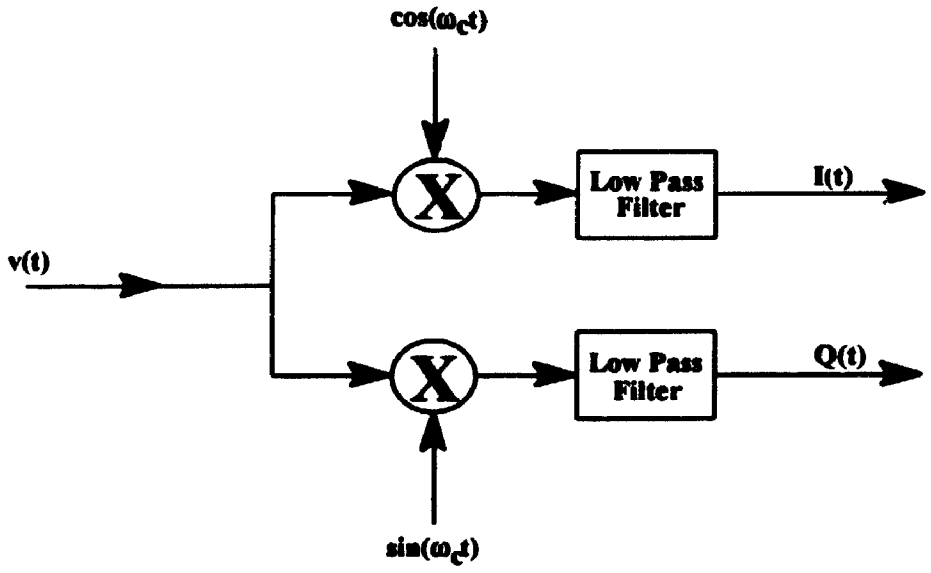


Figure E.1 Simplified Diagram of the Quadrature Detection Process

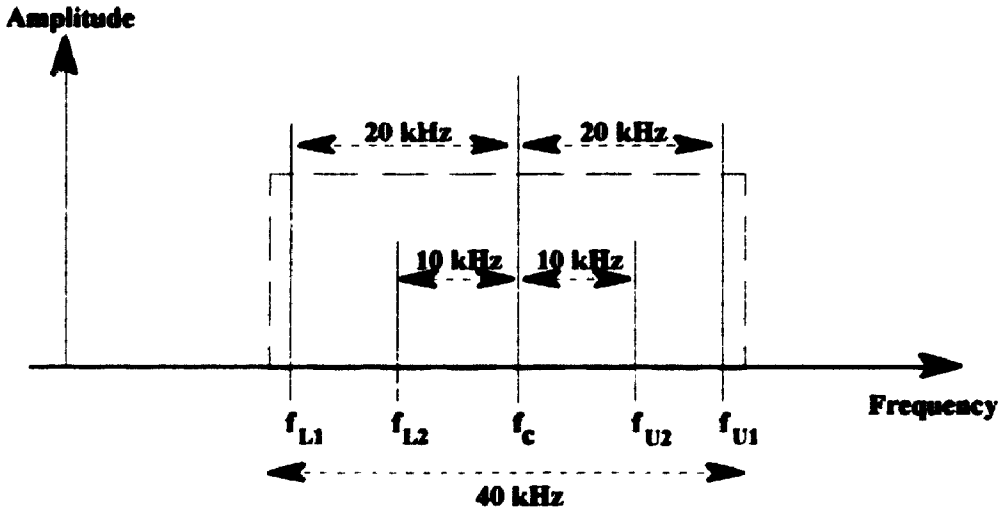


Figure E.2 Signal Spectrum Illustrating Beat Frequency Variations

Appendix D

Appended Material for Chapter 7

```
/*****************************************************************/
/*                                                               */
/*    File Name: plftnp.c                                        */
/*    Created: 19940323                                          */
/*    Created By: Kerry Ellis                                    */
/*                                                               */
/*    This program analyses the data obtained from the chirp radar */
/*    system. The measurements made on this data are the path    */
/*    gain, flight-time, number of paths and average flight-time. */
/*    Useful sections of the path gain/flight-time profiles are  */
/*    written to the output data file specified.                 */
/*                                                               */
/*    Variables/Parameters:                                      */
/*      name1     - input binary file prefix (yymmdd).           */
/*      fname1    - name of the Binary file containing the data. */
/*      name2     - name of the summary output file.             */
/*      name3     - name of the delay output file.               */
/*      name4     - name of the delay spread output file.        */
/*      mth       - contains month.                              */
/*      day       - contains year.                               */
/*      hr        - contains hour.                               */
/*      min       - contains minute.                             */
/*      sec       - contains second.                             */
/*      hsec      - contains hundredths of seconds.              */
/*      yr        - contains year.                               */
/*      i         - index variable.                              */
/*      j         - index variable.                              */
/*      len       - length of file date string (fname1).         */
/*      rec_start- starting record number.                       */
/*      rec_stop - stopping record number.                       */
/*      data      - data value read from the binary file.        */
/*      npaths    - number of resolved paths.                    */
/*      x         - x-coordinate for the output profile. The value */
/*                  of this variable respresents the number of the */
/*                  recording made during the burst.             */
/*      nbytes    - number of bytes that need to be allocated for */
/*                  the storage array.                           */
/*      mem1      - number of free bytes before array memory     */
/*                  allocation.                                  */
/*      mem2      - number of free bytes after array memory      */
/*                  allocation.                                  */
/*      value     - contains the floating point equivalent of the */
/*                  data read from the binary file.              */
/*      pwr       - measure of the power level at a given frequency. */
/*      frq       - signal frequency.                            */
/*      tmp1-3    - temporary storage variables.                 */
/*      tp        - propagation time.                            */
/*      att       - path attenuation                             */
/*      *array    - pointer to the data storage array.           */
/*      *window   - pointer to the window storage array.         */
/*      *fp1      - pointer to input binary data file.           */
/*      *fp2      - pointer to summary output data file.         */
/*      *fp3      - pointer to path gain/flight time profile output */
/*                  data file name.                              */
/*                                                               */
/*****************************************************************/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <alloc.h>
#include <math.h>

#include "four2.c"
#include "pwrspec.c"
#include "f_name.c"
```

```
#define  NPNTS        32768L
#define  Data_Size    10000L

#define  FS         200000.0

#define  SCALE    0.004882812
#define  OFFSET   10.0


void main()
  {
    int  data,yr,i,rec_start,rec_stop,len,npaths,x;
    unsigned long  j,nbytes,mem1,mem2,indx;
    char name1[30],fname1[30],name2[30],fname2[30],mth,day;
    char name3[30],name4[30];
    unsigned char hr,min,sec,hsec;
    float value,pwr,frq,tmp1,tmp2,tmp3,tp,att,t_start,t_stop;
    float huge *array;
    float huge *window;
    FILE  *fp1,*fp2,*fp3;


    void f_name(char *fname,int value,int len);
    void four2(float huge *array,long nn,int isign);
    void pwrspec(float huge *array,long n);

    /*-------------------------------------------------------------*/
    /*  Allocate enough memory for storing the entire array in     */
    /*  memory. Be sure to check that the allocation was successful*/
    /*  before continuing on. It is necessary to check the amount  */
    /*  of heap space allocated instead of only checking to see if */
    /*  the pointer returned by farmalloc is null since the        */
    /*  returned pointer can be ok but the memory hasn't been      */
    /*  allocated.                                                 */
    /*-------------------------------------------------------------*/

    mem1=farcoreleft();

    nbytes=((unsigned long)sizeof(float))*(2*NPNTS);
    array=(float huge *)farmalloc(nbytes);

    mem2=farcoreleft();

    printf("\n %lu %lu",mem1,mem2);

    if ((mem1-mem2) >= nbytes)
       printf("\n Memory Allocation Successful !");
    else
       {
         printf("\n Memory Allocation Unsuccessful !");
         exit(1);
       }

    for(j=0;j<(2*NPNTS);++j)
        array[j]=(float)j;

    for(j=0;j<(2*NPNTS);++j)
       {
         if(array[j]!=((float)j))
           printf("\n Error !");
       }

    printf("\n Memory Test Successful");

    /*-------------------------------------------------------------*/
    /*   Allocate memory for storing the window samples. This      */
    /*   array need only be big enough to store real values ie.    */
```

```
/*   don't allocate space for an imaginary component.          */
/*-------------------------------------------------------------*/

mem1=farcoreleft();

nbytes=((unsigned long)sizeof(float))*Data_Size;
window=(float huge *)farmalloc(nbytes);

mem2=farcoreleft();

printf("\n %lu %lu",mem1,mem2);

if ((mem1-mem2) >= nbytes)
   printf("\n Window Memory Allocation Successful !");
else
   {
     printf("\n Window Memory Allocation Unsuccessful !");
     exit(1);
   }

for(j=0;j<Data_Size;++j)
    window[j]=(float)j;

for(j=0;j<Data_Size;++j)
   {
     if(window[j]!=((float)j))
       printf("\n Error !");
   }

printf("\n Window Memory Test Successful");

/*-------------------------------------------------------------*/
/*  Initialize the data and window arrays by setting the value */
/*  of all array elements to zero.                             */
/*-------------------------------------------------------------*/

for(j=0;j<(2*NPNTS);++j)
   array[j]=0.0;

for(j=0;j<Data_Size;++j)
   window[j]=0.0;

/*-------------------------------------------------------------*/
/*   Open the file containing the samples for the Dolph -      */
/*   Chebyshev window and read them into the window array.     */
/*-------------------------------------------------------------*/

fp1=fopen("dc40.dat","r");

for(j=0;j<Data_Size;++j)
   {
     fscanf(fp1,"%f \n",&value);
     window[j]=value;
   }

fclose(fp1);

/*-------------------------------------------------------------*/
/*  Prompt for input file prefix as well as starting and       */
/*  stopping record numbers and output file name.              */
/*-------------------------------------------------------------*/

printf("\n Enter Input File Prefix: ");
scanf("%s",&name1);

printf("\n Enter Starting Record #: ");
scanf("%d",&rec_start);
```

```
printf("\n Enter Stopping Record #: ");
scanf("%d",&rec_stop);

printf("\n Enter Summary File Name: ");
scanf("%s",&name2);

printf("\n Output Profile File Name: ");
scanf("%s",&name3);

printf("\n Starting Time: ");
scanf("%f",&t_start);

printf("\n Stopping Time: ");
scanf("%f",&t_stop);

/*-------------------------------------------------------------*/
/*  Open files for data output.                                */
/*-------------------------------------------------------------*/

fp2=fopen(name2,"a");
fp3=fopen(name3,"w");

/*-------------------------------------------------------------*/
/*  Start reading in the data files and process the data as    */
/*  necessary.                                                 */
/*-------------------------------------------------------------*/

x=0;

for(i=rec_start;i<=rec_stop;i++)
   {
     for(j=0;j<(2*NPNTS);++j)
        array[j]=0.0;

     x+=1;

     len=strlen(name1);
     strcpy(fname1,name1);
     f_name(fname1,i,len);

     printf("\n In-File: %s",fname1);

     fp1=fopen(fname1,"rb");

     if (fp1 != NULL)
        {
          /*-----------------------------------------------*/
          /*  Read date time information from the file.    */
          /*-----------------------------------------------*/

          fread(&yr,sizeof(yr),1,fp1);
          fread(&mth,sizeof(mth),1,fp1);
          fread(&day,sizeof(day),1,fp1);
          fread(&hr,sizeof(hr),1,fp1);
          fread(&min,sizeof(min),1,fp1);
          fread(&sec,sizeof(sec),1,fp1);
          fread(&hsec,sizeof(hsec),1,fp1);

          /*-----------------------------------------------*/
          /*  Read in the actual data in binary format and */
          /*  convert to floating point. The data values are */
          /*  then windowed and stored in array[].         */
          /*-----------------------------------------------*/

          for(j=0;j<Data_Size;++j)
             {
```

```
      indx=2*j;
      fread(&data,sizeof(data) 1,fp1);
      value=(float)(data*SCALE-OFFSET);
      array[indx]=value*window[j];
      array[indx+1]=0.0;
    }

fclose(fp1);

/*-------------------------------------------------*/
/*  Fourier Transform the data and compute the power */
/*  spectrum of the signal. The real part of the     */
/*  data array now contains the power spectrum and   */
/*  the imaginary part has been set to zero.         */
/*-------------------------------------------------*/

four2(array,NPNTS,1);

pwrspec(array,NPNTS);

/*-------------------------------------------------*/
/*  Scale the power spectrum into dBm and compare    */
/*  the power level with the noise floor power of    */
/*  -14 dBm. Spectral components greater than this   */
/*  level have a SNFR of greater than 3.0 dB and     */
/*  will be used in the calcuation of delay spread   */
/*  and other quantities.                            */
/*-------------------------------------------------*/

for(j=0;j<(NPNTS/2);++j)
   {
     indx=2*j;

     if(array[indx]>0.0)
       {
         array[indx]=10.0*log10(array[indx]/1.0e-3);

         if(array[indx]<-14.0)
           array[indx]=-100.0;
       }
     else
       array[indx]=-100.0;
   }

/*-------------------------------------------------*/
/*  Determine the number of resolvable peaks/paths   */
/*  from the power spectrum and write the path       */
/*  gains, flight-times and number of paths to the   */
/*  output file.                                     */
/*-------------------------------------------------*/

fprintf(fp2,"\n Record Number: %d",i);
fprintf(fp2,"\n Path Gain      Flight-Time");

npaths=0;

for(j=25;j<(NPNTS/2);++j)
   {
     tmp1=array[2*(j-1)];
     tmp2=array[2*j];
     tmp3=array[2*(j+1)];

     if((tmp2>tmp1) && (tmp2>tmp3))
       {
         indx=2*j;
         npaths++;
         frq=FS*j/NPNTS;
```

```
                    tp=2.0711e-3+((frq-12.0e3)/5.0e7);
                    tp/=1.0e-3;
                    pwr=array[indx];
                    att=54.0+7.0+7.0-1.0-3.6-((pwr-111.910)/0.943);
                    att*=-1.0;

                    fprintf(fp2,"\n %f      %f",att,tp);
                  }
               }

            fprintf(fp2,"\n Number of Paths= %d",npaths);

            /*-------------------------------------------------*/
            /*  Write a portion of the path gain/flight-time   */
            /*  profile to the data file specified. Values are */
            /*  written in x,y,z format for 3-d plotting. The  */
            /*  y-coordinate represents the recording number   */
            /*  made during a burst.                           */
            /*-------------------------------------------------*/

            for(j=0;j<(NPNTS/2);++j)
               {
                 indx=2*j;
                 frq=FS*j/NPNTS;
                 tp=2.0711e-3+((frq-12.0e3)/5.0e7);
                 tp/=1.0e-3;
                 pwr=array[indx];
                 att=54.0+7.0+7.0-1.0-3.6-((pwr-111.910)/0.943);
                 att*=-1.0;

                 if((tp>=t_start) && (tp<=t_stop))
                   {
                     if(att<=-200.0)
                       att=-200.0;

                     fprintf(fp3,"%d   %f   %f \n",x,tp,att);
                   }
               }
          }
        else
           printf("\n File Error during Open !!");

      }
  /*-----------------------------------------------------------*/
  /*  Close the output files and release the memory allocated  */
  /*  for array and window storage.                            */
  /*-----------------------------------------------------------*/

  fclose(fp2);
  fclose(fp3);
  farfree(array);
  farfree(window);

}
```

REFERENCES

[1] G.R.Sugar, "Radio Propagation by Reflection from Meteor Trails", Proc. IEEE, Feb. 1964, pp. 116-13o.

[2] A.R. Webster, J. Jones, "A Study of Meteor Scatter Communications", Final Report, D.S.S. Contract # 36001-9-3601/01-SS., April 1991.

[3] D.W. McKinley, "Meteor Science and Engineering", McGraw Hill Book Company, Inc., Toronto, 1961.

[4] J.Z. Scharker, "Meteor Burst Communications", Artech House Inc., Norwood, MA 02062, 1990.

[5] J.A. Weitzen, "A Study of the Ground Illumination Footprint of Meteor Scatter Communication", IEEE Trans. Commun., vol. 38, No. 4, April 1990, pp. 426-430.

[6] W.I. Thompson, III, J.F. Herman, "Coded Speech Transmission on Meteor Burst Communication Systems", IEEE Electo. 88, Boston, May 10-12, 1988.

[7] G.W. Pickard, "A Note on the Relation of Meteor Showers and Radio Reception", Proc. IRE, vol. 19, pp. 1166-1170, July 1931.

[8] V.R. Eshleman, "Theory of Radio Reflections from Electron-Ion Clouds", IRE Trans. Antennas and Propagation., pp. 32-39, Jan. 1955.

[9] L.A. Manning, "Oblique Echoes from Over-Dense Meteor Trails", Journal of Atmospheric and Terrestrial Physics, vol. 14, pp 82-93, Apr 1959

[10] J.A. Weitzen, W.P. Birkemeier, M D. Grossi, "An Estimate of the Capacity of the Meteor Burst Channel", IEEE Trans. on Commun. vol. COM-32, No 8, pp 972-974, Aug. 1984.

[11] A.E. Spezio, "Meteor-Burst Communication Systems: Analysis and Synthesis", NRL Report 8286, Dec. 1978.

[12] J.A. Weitzen, "Predicting the Arrival of Meteors Useful for Meteor Burst Communication", Radio Science, vol. 21, No. 6, pp. 1009-1020, Nov -Dec 1986

[13] A.N. Ince, "Spatial Properties of Meteor-Burst Propagation", IEEE Trans on Commun. vol. COM-28, No. 6, June 1980.

[14] J.D. Oetting, "An Analysis of Meteor Burst Communications for Military Applications", IEEE Trans. on Commun., vol. COM-28, Sept 1980

[15] D.W. Brown, "A Physical Meteor-Burst Propagation Model and Some Significant Results for Communication System Design", IEEE JSAC, vol SAC-3, No 5, pp. 745-755, Sept. 1985.

[16] J.J. Metzner, "Improved Coding Strategies for Meteor Burst Communication", IEEE Trans. on Commun., vol. 38, No. 2, pp. 133-135, Feb 1990

[17] S L. Miller, L.B. Milstein, "Error Correction Coding for a Meteor Burst Channel", IEEE Trans. on Commun., vol. 38, No. 9, pp. 1520-1529, Sept. 1990.

[18] L. B. Milstein, et. al., "Performance of Meteor-Burst Communication Channels", IEEE JSAC, vol. SAC-5, No. 2, pp. 146-153, Feb. 1987.

[19] S.L. Miller, L.B. Milstein, "A Comparison of Protocols for a Meteor-Burst Channel Based on a Time-Varying Channel Model", IEEE Trans. on Commun., vol. 37, No. 1, pp. 18-30, Jan. 1989.

[20] S.L. Chang, D.L. Schilling, "Efficient Communications Using the Meteor-Burst Channel", IEEE Trans. on Commun., vol. 40, No. 1, pp. 119-128, Jan. 1992.

[21] R.M. Braun, G.J. Meyerowitz, "Investigation into Adaptive-Rate Modems for Meteor Burst Communication", Electronics and Communication Engineering Journal, pp. 213-222, Oct. 1991.

[22] F. Akram, N.M. Sheikh, A. Javed, M.D. Grossi, "Impulse Response of a Meteor Burst Communications Channel Determined by Ray-Tracing Techniques", IEEE Trans. on Commun., pp. 467-470, Apr. 1977.

[23] J.A. Weitzen, M.D. Grossi, W.P. Birkemeier, "High-Resolution Multipath Measurements of the Meteor Scatter Channel", Radio Science, vol. 19, no. 1, pp. 375-381, Jan. - Feb. 1984.

[24] J.A. Weitzen, M.J. Sowa, R.A. Scofidio, J. Quinn, "Characterizing the Multipath and Doppler Spreads of the High-Latitude Meteor Burst Communication Channel", IEEE Trans. on Commun., vol. COM-35, No. 10, pp. 1050-1058, Oct. 1987.

[25] G. Eriksson, "An Investigation of the Meteor Channel for Adaptive Communication Systems", IEEE MILCOM 90, pp. 873-877, Vol. 2, 1990.

[26] C.E. Cook, M. Bernfeld, "Radar Signals an Introduction to Theory and Application", Academic Press Inc., New York 10003, 1967.

[27] T. Ohno, Personal Communication.

[28] L.C. Ludeman, "Fundamentals of Digital Signal Processing", Harper and Row, Publishers, Inc., 1986.

[29] F.J. Harris, "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform", IEEE Proceedings, Vol. 66, No. 1, pp. 51-83, Jan. 1978.

[30] B. Edde, "Radar Principles, Technology, Applications", Prentice Hall, Englewood Cliffs, New Jersey 07632, 1993.

[31] A.H. Nuttal, "Generation of Dolph-Chebyshev Weights via a Fast Fourier Transform", IEEE Proceedings, Vol. 62, pg. 1396, 1974.

[32] M.R. Spiegel, "Mathematical Handbook", Schaum's Outline Series in Mathematics, McGraw-Hill Book Company, Toronto, 1968.

[33] D.C. Cox, "Delay Doppler Characteristics of Multipath Propagation at 910 MHz in a Suburban Mobile Radio Environment", IEEE Transactions on Antennas and Propagation, Vol. AP-20, No. 5, pp. 625-635, Sept. 1972.

[34] D.M.J. Devasirvatham, "Multipath Time Delay Spread ir the Digital Portable Radio Environment", IEEE Communications Magazine, Vol. 25, No. 6, pp. 13-20, June 1987.

[35] D.L. Schilling, "Meteor Burst Communications Theory and Practice", John Wiley and Sons, Inc., Toronto, 1993.

[36] E.C. Jordan, K.G. Balmain, "Electromagnetic Waves and Radiating Systems", 2nd Edition, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1968.

[37] C.O. Hines, P.A. Forsyth, "The Forward-Scattering of Radio Waves from Overdense Meteor Trails", Canadian Journal of Physics, Vol. 35, pp. 1033-1041, 1957.

[38] A.A.M. Saleh, A.J. Rustako, Jr., L.J. Cimini, Jr., G.J. Owens and R.S. Roman, "An Experimental TDMA Indoor Radio Communications System Using Slow Frequency Hopping and Coding", IEEE Trans. on Commun., Vol. 39, No. 1, pp. 152-162, Jan. 1991.

[39] A.S. Acampora, J.H. Winters, "System Applications for Wireless Indoor Communications", IEEE Communications Magazine, Vol. 25, No. 8, pp. 11-19, Aug. 1987.

[40] D.M.J. Devasirvatham, "Time Delay Spread and Signal Level Measurements of 850 MHz Radio Waves in Building Environments", IEEE Trans. on Antennas and Propagation, Vol. AP-34, No. 11, pp. 1300-1305, Nov. 1986.