Range Resolved Optical Remote Sensing Using a Continuous Wave, Pseudo-Random Modulated, CO_2 Heterodyne Lidar Backscattered from Aerosols

Badih John Rask
B.S.E.E., Lafayette College, 1988
M.S.E.E., Oregon Graduate Institute, 1991

A dissertation presented to the faculty of the
Oregon Graduate Institute of Science & Technology
in partial fulfillment of the
requirements for the degree
Doctor of Philosophy
in
Electrical Engineering

October, 1995

The dissertation "Range Resolved Optical Remote Sensing Using a Continuous Wave, Pseudo-Random Modulated, CO_2 Heterodyne Lidar Backscattered from Aerosols," written by Badih John Rask has been examined and approved by the following Examination Committee:

J. Fred Holmes, Thesis Adviser
Professor

Reinhart Engelmann
Professor

Rao Gudimetla
Assistant Professor

M. Aslam K. Khalil
Professor

Acknowledgements

May 4th, 1995. I can't believe I'm writing these words, because it means I'm almost done. I've become the gray-beard of the students in EEAP, having been here since the start of the Bush administration. I have persevered. Heck, I've outlasted presidents (in the White House *and* at OGI). Completing this Ph.D. has taught me so much about myself and about life. But perhaps the most important lesson is that no one completes it alone. There have been so many people helping me there is no way I can properly thank them. But here goes a try:

In the Spring of 1990 I was adviserless. Dr. J. Fred Holmes agreed to take me on, even though he had no projects at the time. Soon, this project surfaced, and the work started. There is no way I can adequately thank Dr. Holmes for having the confidence and patience in me throughout the project. He never failed to provide an idea during a difficult period. His suggestions were always clever, and insightful, and if I'm lucky, some of his expertise has rubbed off on me. I will very much miss our almost daily discussions, solving problems and brainstorming. I certainly appreciate his guidance and direction.

I have often wondered where this project would be without the services of John Hunt. There certainly would be a lot less electronics, circuits, and amplifiers. This project would never have gotten off the ground without his contributions. I would also like to thank John for his friendship, interest, and suggestions during the research, and for losing bets.

I would like to thank my colleagues, Feng Chen and Chunyan Zhou, for all their help. Without you guys, I wouldn't even have wanted to be here. You have taught me so much about yourselves (thanks to Lamb for the cooking lessons!). I consider myself grateful just to have your friendship. I will miss our "group meetings." Remember how much fun we had in Seattle. I'm lucky to know you guys. "Correlation.....function."

I would like to thank the five undergraduate students who helped with the research during the summers: Steve, Keith, Casey, Brian, and Ed. Thanks for making the summers fun, guys.

I wish to thank my parents, Mom and Ken, for their love, support, and confidence in me to help me complete this Ph.D. I hope this makes you proud. You have been behind me and have believed in me all my life, and I can't thank you enough. I wish to thank my brother, Pat, and my sisters, Dagny and Molly, for all their encouragement, and love. Thanks to Granny for her prayers, and for teaching me her language, and for passing on a part of herself to me. To my father, Dr. Michael Rask, who recently passed away and did not get the chance to see me graduate, I know he would have been proud. I love you all.

Thanks to Tom, and Chris, for your friendship, and support throughout this Ph.D. Thanks to Tom for all those talks of encouragement from someone who was in my same boat. Thanks to Adam, Ted, Kent, and all my friends at home for your support.

Thanks to the basketball crowd, Feng, Cangsang, the Hammer, and the rest of the Guman. I will miss you guys terribly. (You got lucky in H-O-R-S-E, Cangsang).

Thanks go to Doug Davis for his friendship, and for all his help in the machine shop, even if I didn't have a grant number. Thanks to Jeff Schilling for his friendship, joking, and golf. Thanks to Tim Bernard for all his help with glass cutting and for cutting jokes. Thanks to Phil Di Napoli for all his help in facilities. Sorry about the truck, guys. Thanks to Bev, Cathy, and DJ for your friendship, advice, chocolates, and for making the day go smoother. Thanks to the thesis committee members, professors Holmes, Engelmann, Gudimetla, and Khalil, for their suggestions and comments. Thanks to the library staff, for an outstanding job helping me get resources. (Sorry, Mary, about all the late, overdue books!)

How can I begin to thank my wife, Sharon, for what she's done for me? I can't imagine a more loving, patient, and understanding person on the face of the earth. I thank you, and I love you, Sharon.

Badih John Rask, May, 1995

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

$D_n(\vec{x})$	Index of refraction structure function.
C_n^2	Index of refraction structure constant (strength of turbulence).
$n_1(\vec{x})$	fluctuating (random) part of the index of refraction.
$B_n(\vec{x})$	Covariance of $n_1(\vec{x})$.
$\phi_n(\vec{x})$	Fourier transform of $B_n(\vec{x})$.
λ	Wavelength.
k	Wavenumber ($2\pi/\lambda$).
G_{FS}	Free Space Green's function including random effects of turbulence used in the extended Huygens-Fresnel integral equation.
G_T	Modified Green's function including random effects of turbulence used in the extended Huygens-Fresnel integral equation.
ψ_f	Includes random effects of turbulence on laser electric field from transmitter to target plane.
ψ_b	Includes random effects of turbulence on laser electric field from target to receiver plane.
$\vec{r}$	Transverse coordinate in transmitter plane.
$\vec{\rho}$	Transverse coordinate in target plane.
$\vec{p}$	Transverse coordinate in receiver plane.
$U_T(\vec{r})$	Transmitted laser electric field.
$U(\vec{\rho})$	Laser electric field in target plane.
$U_R(\vec{p})$	Laser electric field in receiver plane.
$T(\vec{\rho})$	Hard target scattering function.

$\Gamma(\vec{\rho}_1, \vec{\rho}_2)$	Mutual intensity function.
$\Gamma(\vec{\rho}_1, \vec{\rho}_2, \tau)$	Time delayed mutual intensity function.
β	Volume backscattering coefficient ($m^{-1}sr^{-1}$).
v_z	Radial wind velocity of an aerosol.
$v_\perp$	Crosswind velocity of an aerosol.
C	Speed of light.
f_D	Doppler shifted frequency.
$S(\vec{\rho})$	Amplitude scattering factor of an aerosol.
$pdf(v_z)$	Probability distribution function of the radial wind velocity of aerosols.
$pdf(v_\perp)$	Probability distribution function of the crosswind velocity of aerosols.
$<v_z>$	Average radial wind speed of aerosols.
$<v_\perp>$	Average crosswind speed of aerosols.
σ_z^2	Variance of the radial wind speed of aerosols.
$\sigma_\perp$	Variance of the crosswind speed of aerosols.
$\phi_S(t)$	Aerosol speckle effect on backscattered laser phase.
$\phi_T(t)$	Turbulence effect on backscattered laser phase.
$\phi_L(t)$	Laser coherence effect on laser phase.
σ_S^2	Variance of laser phase due to aerosols speckle effects.
σ_T^2	Variance of laser phase due to turbulence.
z_c	Distance to center of range bin.
$B_{I_{1,2}}$	Time-delayed crosscorrelation of intensities between two points in receiver plane.

$C_{I_{1,2}}$	Time-delayed crosscovariance of intensities between two points in receiver plane.
H	Spherical wave mutual coherence function, including random effects of turbulence on laser beam electric field.
α_0	Radius of outgoing laser beam.
F_T, F	Focus distance of outgoing laser beam.
F_{LO}	Focus distance of backpropagated local oscillator focus distance.
C_χ	Log amplitude covariance function.
D_ϕ	Phase perturbation due to turbulence.
U_{T_0}	Amplitude of outgoing laser beam electric field.
$C_{I_{lpf}}$	Low-pass filtered crosscovariance of Intensities between two points in the detector plane.
PRC (t)	Pseudo-random code.
$R_{PRC}(\tau)$	Autocorrelation function of pseudo-random code.
L	Range at which backscattering occurs.
D	Range selected for processing.
P_{LO}	Power of local oscillator beam.
P_{BS}	Power of backscattered beam.
e	ELectronic charge.
B	Bandwidth of detectors.
ρ_0	Responsivity of detector (A/W).
SNR	Signal-to-Noise Ratio.
τ_p	Time delay at which maximum (peak) occurs in time-delayed crosscovariance.

U_{LO}	Electric field of local oscillator.
β_0	Radius of local oscillator electric field.
FZS	Fresnel Zone Scale size ($\sqrt{z_c/k}$).
$C_{N_{lpf}}$	Time-delayed crosscovariance normalized by the average intensities.

LIST OF ACRONYMS

LASER	Light amplification through stimulated emission of radiation.
cw	Continuous wave operation of a laser.
PRC	Pseudo-random code
CO_2	Carbon dioxide.
FFT	Fast Fourier transform.
SNR	Signal to noise ratio.
AOM	Acousto-optic modulator.
SZTD	Slope of the time-delayed crosscovariance of intensities at zero time delay.
CZTD	Covariance at 0 time delay.
TDTP	Time delay to peak of the time-delayed crosscovariance of intensities.

Abstract

Range Resolved Optical Remote Sensing Using a Continuous Wave, Pseudo-Random Modulated, CO_2 Heterodyne Lidar Backscattered from Aerosols

Badih John Rask

Supervising Professor: J. Fred Holmes

Optical remote sensing has been a useful technique used to measure the atmospheric wind velocity. In the late 1960's and early 1970's, researchers were using double-ended laser systems to study the effects of the turbulent atmosphere on laser beam propagation, and to determine the crosswinds. While the double-ended systems were useful in verifying theory, their applications were limited to those in which both ends of the propagation path were accessible. Toward the latter part of the 1970's, however, the push for single-ended laser systems became more desirable to locate the laser transmitter and receiver at the same end of the propagation path. The early single-ended systems used a hard target to backscatter the laser radiation to estimate the path averaged crosswind speed between the transmitter/receiver and the hard target. Since the end of the 1980's, interest has shifted to obtaining range resolved wind estimates, where the desired wind measurements can be limited to remote regions, called range bins, instead of an average wind estimate over the entire laser propagation path. Instead of using a metallic hard target as the laser scatterer, air particles, or aerosols, act as a distributed target to backscatter the laser radiation. Since the aerosols move

with the wind, the laser signal will be Doppler shifted by the component of the vector wind along the line of sight from the laser to the scattering volume. Therefore, if the frequency drift from a known modulation frequency due to the Doppler effect can be measured, the wind along the line of sight from laser to aerosols can be determined. This gives the possibility to measure the 3-D wind velocity, where the crosswinds are estimated through the techniques adapted from the path-averaged hard target scenario, and the line-of-sight winds are obtained by detecting the Doppler shift.

Although the range resolved 3-D wind speed laser systems have more flexibility than the path averaged systems, the absence of a hard target introduces several new problems. The first difficulty is that the signal-to-noise ratio (SNR) is very poor (about -20 to -30 dB) because aerosols backscatter much less laser radiation than a hard target. To combat this problem, a processing scheme was developed to detect the amplitude and frequency of a cosine buried in noise specific with the sampling rate and maximum coherent sampling time. Another problem arose because the laser operates cw (continuous wave) rather than pulsed. In the pulsed systems, the receiver can be range gated to look at different parts of the path. But a cw system needs a scheme to discriminate the scattering returns from different parts of the propagation path. This was solved by pseudo-random-code (PRC) modulating the outgoing laser beam phase. The backscattered laser signal can be demodulated with a time delayed version of the PRC to produce an intensity versus range profile.

The theoretical analysis of the time delayed statistics of the received intensity was completed for the case of a cw laser backscattered from aerosols. Additional analysis was devoted to the effect that a finite aperture has on the time delayed statistics. Finally, the system was constructed to be able to propagate over horizontal and vertical paths to study such atmospheric properties as the backscattering coefficient and strength of turbulence in addition to the range resolved wind velocity.

Chapter 1

Introduction

Laser systems, called lidars, are useful tools that can be used to probe information about the atmosphere. By utilizing the effects of the turbulent atmosphere on laser beam propagation, scientists are able to monitor pollution flow near cities and evaluate the hazards of toxic pesticides sprayed over fields. In coal mines, lasers are employed to detect traces of harmful gases to warn workers of otherwise undetectable hazards. In addition, lasers are directed into the atmosphere to determine the strength of backscattering of aerosols. Such information is useful because of the potential to improve weather forecasting. Improving airline safety against the pitfalls of wind shear is another application where the laser can be used to monitor sudden changes in wind speed and direction to give pilots advance warning. Furthermore, for the past 20 years, scientists have been trying to put lidars on satellites to gain better information about global warming. In fact, a recent Shuttle Discovery mission used lidar to study clouds, pollution, and climate changes.[1] In each application, the laser has been used as a remote sensor to gather information about the atmosphere.

Installing lidars on satellites and airplanes requires laser systems to be light, compact, and reliable. In addition, the system must be single-ended, where the laser transmitter and receiver are located at the same end of the propagation path. More than likely, aerosols are used to backscatter the laser radiation. These requirements are not easy to meet. Many different types of lidars offer certain advantages. For example, pulsed laser systems can detect radiation backscattered from the longest

ranges because these systems put out a high peak power pulse. By range gating the receiver, the system can "look" at different parts of the path. However, these advantages come with some significant drawbacks. The range gating of the receiver necessitates some complicated electronics which makes the processing difficult. In addition, the pulsed nature of the laser causes some laser instability as compared to lasers operating continuous wave (cw). The pulsed requirement of the laser also makes these systems more expensive, larger, and less reliable than cw systems.

Laser diodes offer the potential for a very compact and reliable lidar. The small size coupled with the high lasing efficiency make the diode laser a very promising candidate for remote sensing. The setbacks with using diode lasers as transmitters are related to their low output power. This limitation makes it difficult to backscatter off of aerosols due to the small backscattered radiation. The low SNR prevents operation during the day because the background noise swamps the return signal from aerosols (for direct detection). These problems can be overcome by using a diode laser with a narrow line-width (long coherence time) so that a heterodyne diode laser system becomes feasible. To get the range information, the outgoing laser beam can be modulated with a pseudo-random code (PRC). By multiplying the return signal by a delayed version of the PRC, it is possible to discriminate the backscattering from different ranges. Since the laser output can be easily modulated by the diode laser drive current, the PRC can be implemented easily by turning the laser on and off with the PRC. The problem is that diode lasers with linewidths narrow enough for heterodyne lidars are not commercially available. Therefore, most lidars using diode lasers as a source are limited to incoherent, direct detection systems, where detection of line-of-sight winds is not possible.

The most desirable laser system to measure the 3-D winds using aerosols as the backscatterer would be one which adopted the range resolution ability of the

pulsed and modulated diode laser systems, while retaining the simplicity of the cw systems. The lidar developed over the past 3 years from 1991 to 1994 at the Oregon Graduate Institute used in this thesis project is a pseudo-random code, cw, coherent optical heterodyne system operating at $\lambda = 10.6\mu m$. Because the laser operates cw, the laser is simpler, more reliable, and less expensive than a pulsed laser. What separates this cw lidar system from others is the novel implementation of a pseudo-random code on the outgoing laser beam to discriminate between ranges. In fact, the use of a pseudo random code for range resolution has been achieved for the first time using a cw, optical heterodyne system. Optical heterodyne detection is used to increase the sensitivity of the lidar to detect weak backscattered signals on the order of 10^{-15} Watts. The 3-D winds can be detected with this lidar using Doppler techniques to measure the line-of-sight winds, and speckle-turbulence interaction for the crosswinds. The only drawback with this system is that the maximum range detectable is less than that obtainable with the pulsed systems. With only 3 Watts of transmitted power, the backscattered radiation from aerosols is limited to less than 1 km with this system.

The remainder of this thesis is divided into six chapters. Chapter 2 contains background on speckle-turbulence theory. Section 2.1 discusses wave propagation through the turbulent atmosphere, reviewing the important concepts necessary to describe wave propagation. Section 2.2 introduces the Huygens-Fresnel formula and illustrates how to apply it to the propagation of laser beams through the atmosphere. Section 2.3 reviews the important breakthroughs that allowed for single-ended systems using a hard target as backscatterer.

In chapter 3, the concepts of using aerosols to backscatter laser radiation are developed. Section 3.1 introduces the properties of aerosols, including their distribution, size, and concentration in the lower atmosphere (lower 4 km). Section 3.2 shows how the scale size of aerosols relative to the wavelength of the laser effects the scat-

tering profile. In section 3.3 the modifications to the extended Huygens-Fresnel integral are demonstrated along with an example of a typical application. Finally, in section 3.4, phase considerations are mentioned with emphasis on the decorrelation effect of aerosols on the statistics of laser beam propagation.

Chapter 4 contains the bulk of the theory for deriving the time-delayed statistics of a CO_2 laser beam backscattered from aerosols. Section 4.1 derives the time-delayed statistics between two point detectors in the receiving plane. Section 4.2 introduces a time-filter to remove the high frequency speckle part of the time-delayed statistics.

The pseudo-random code, cw, CO_2 heterodyne lidar is introduced in chapter 5. Section 5.1 gives a complete description of the lidar, and how the pseudo random modulation is employed. Section 5.2 discusses the system signal to noise ratio (SNR). Finally, Section 5.3 introduces the processing schemes to get the 3-dimensional wind speed.

Chapter 6 presents the experimental results. Section 6.1 presents the data for the lidar operated as a Doppler wind sensor and range finder. Section 6.2 shows the data for the lidar operated to detect crosswinds.

Finally, chapter 7 summarizes the work of the preceding chapters and gives suggestions for future work in this field.

The electronics of the PRC generator and of the amplifiers for both channels are sketched in Appendix A. The data acquisition and storage programs are listed in Appendix B, while the processing programs are listed in Appendix C. The programs used to compute the time-delayed crosscovariance, the slope at zero time delay crosswind detection scheme, and the aperture averaging effects are listed in Appendix D. Appendix E shows how to simplify the low-pass filtered crosscovariance of intensities to be used in a crosswind detection scheme.

Chapter 2

Wave Propagation Through the Atmosphere

Before developing the time delayed statistics for a laser beam backscattered from aerosols propagating through turbulence, it is useful to review some of the important theoretical developments needed to describe the process. As early as the fifteenth century, scientists were making a transition to studying problems of optical frequencies. One intriguing phenomena which could not be explained with the existing theories of light propagation was diffraction, the appearance of light and dark bands near the boundaries of objects when exposed by light rays. The first reference to diffraction phenomena appears in the work of Leonardo Da Vinci. Later, Christian Huygens, the first proponent of the wave theory of light, introduced what was called the Huygens' construction. Huygens asserted that every point on a wave front was a point source for a spherical wave. The wave front at any later instant may be regarded as the envelope of these spherical waves, called wavelets. In 1818, Augustine-Jean Fresnel showed that diffraction could be explained by applying Huygens' construction along with the principle of interference. Fresnel postulated that the secondary wavelets interfered with each other, giving rise to diffraction. In 1882, Kirchoff developed an integral representation for light wave propagation based on Fresnel's analysis. The Huygens-Fresnel principle combined Huygens' construction of an envelope of spherical wavelets with the idea that these wavelets mutually interfere.[2]

The Huygens-Fresnel principle was used to describe the propagation of light

waves to solve a variety of problems in free space. For example, A. Sommerfeld studied the diffraction of a plane wave by a semi-infinite plane screen. In addition, Fresnel was interested in describing the diffraction of an electric field by a finite aperture. While wave propagation through a homogenous medium covered a vast number of interesting problems, attention turned to more general situations such as the approximate solution to the wave equation in an inhomogeneous medium. While there were many instances where the inhomogeniety was deterministic (not random), most real world problems involved random media, such as light waves propagating through the atmosphere. For example, astronomers have long noticed the twinkling, or scintillation of the stars viewed through telescopes. In fact, by the 1950's scientists were studying the random effect of the earth's atmosphere on star light to aid in the design of telescopes. Any exact description of the atmosphere was hopeless since the effect of the atmosphere on wave propagation was a random process. But in the early 1960's several important advances in describing the turbulent atmosphere in a statistical sense were recognized by Russian scientists. A. N. Kolmogorov was able to find a universal form for one statistical quantity, the mean square velocity difference between two points in the atmosphere.[3] Within the so-called inertial subrange of turbulence, Kolmogorov showed that the mean square velocity difference has a universal form under the assumptions of local homogeneity and isotropy

$$D_v(x) = C_v^2 \, x^{2/3} \tag{2.0}$$

where C_v^2 is related to the total amount of energy in the turbulence and x is the separation between velocity detectors. Since the variations in the refractive index cause random perturbations on light wave propagation, a link was needed between the velocity turbulence model and the refractive index which is a function of temperature and

pressure. This important step was provided by V. I. Tatarskii and his concept of a conservative passive additive.[4] Tatarskii realized that it is difficult to develop the statistics for temperature because it is not a conservative quantity. As a parcel of air changes in height, its temperature changes in order to equalize its pressure. Instead of studying the fluctuations in temperature, Tatarskii studied the fluctuations in potential temperature, the difference between the absolute temperature and the temperature change due to height. This concept lead to a two-thirds power structure function for potential temperature similar to the one for velocity fluctuations. In turn, since the refractive index is a function of temperature, a two-thirds power law followed for the refractive index fluctuations. This step was critical because the wave equation depends on the refractive index. Using an approximation method introduced by S. M. Rytov, Tatarskii developed a perturbative solution to the wave equation to the first order in the fluctuating part of the refractive index. Throughout the 1960's researchers in the Soviet Union and the United States used the Rytov approximation along with the refractive index statistics from Tatarskii and Kolmogorov to describe wave propagation through turbulence.

By the end of the 1960's researchers had most of the tools necessary to describe wave propagation in random media. The integral representation of the Huygens-Fresnel principle was a well established technique to describe wave propagation through free space, the statistics of the turbulent atmosphere were defined and experimentally verified, and the Rytov approximation provided initial insight to the statistics of wave propagation through turbulence. In 1971, a critical, creative method to describe wave propagation through the atmosphere was introduced by R. F. Lutomirski and H. T. Yura.[5] They extended the integral form of the Huygens-Fresnel principle to a medium having a random variation in the refractive index. Their integral form properly described a light wave traveling through turbulence taking into account the

statistics of the fluctuations in the refractive index developed by Kolmogorov and Tatarskii. Although the solution was based on the same approximations as Rytov, this approach became popular because it was easily adaptable to laser beam propagation. Since 1971 there has been an explosion of analytical and experimental research concerning laser beam propagation through turbulence based upon the extended Huygens-Fresnel principle. It should be noted that several Russian researchers had extended the Huygens-Fresnel integral to the turbulent atmosphere prior to 1971 but these works were not translated until later.[6,7,8]

Not all researchers accepted the extended Huygens-Fresnel approach to describing wave propagation through turbulence because of the approximations used in the extension to random media. Another approach, the Path Integral Method, was developed in an attempt to eliminate these approximations.[9,10] However, the only way to provide actual numbers for the statistics of light wave propagation was to use approximations which reduced the result to the extended Huygens-Fresnel integral.

Early work with lasers was done horizontally with a laser at one end and a receiver at the other. These so called double-ended systems were instrumental in studying turbulence and its effect on laser beams.[11,12] In addition to studying the effect of turbulence on laser beam propagation, researchers used the double-ended systems to measure the average crosswind, the wind perpendicular to the outgoing laser beam. By the mid 1970's, there was a push for single-ended systems where the laser and the detector were put at the same end and a metallic hard target at the other end was used to backscatter the laser radiation. The original analysis and design of single-ended systems was carried out by the research team of Richard Kerr, J. Fred Holmes, and Myung Hun Lee, and coworkers, at the Oregon Graduate Center.[13] Their approach was to incorporate the effect of a diffuse scattering metallic hard target on a coherent laser source as it propagated back to the detector. The main idea was to

treat the propagation problem in two steps. The first step was to characterize the coherent laser beam as it traveled through turbulence to a diffuse hard target which totally randomized the scattered beam into a speckle pattern. Then the second step was to characterize the propagation of the speckle pattern as it travelled back through the turbulence to the detector. This idea lead to a number of publications[14-20] as well as several experimental laser systems which measured the average crosswind between the laser and the hard target. Throughout the 1980's these single-ended hard target systems were refined as research work continued using the interaction of speckle and turbulence to measure the average crosswind speed between the laser/receiver and hard target.

The research work carried out in this thesis project takes the problem to the next level. Instead of using a hard target as the diffuse scatterer, the particles in the air called aerosols are used to backscatter the laser beam. The remaining sections of Chapter 2 give a more complete description of turbulence and speckle and the extended Huygens-Fresnel integral equation.

2.1 Wave Propagation Through Turbulence

The effects of turbulence can be noticed on sunny days. As the sun heats the ground, the hood of a car, or a roof top, heat is radiated back to the atmosphere. The appearance of these "heat waves" causes a flickering or wavering of the light visible by the naked eye. It is even possible to notice the wind direction because the heat waves move with the wind. These heat waves are created by the temperature differences between the atmosphere and the objects radiated by the sun. Consequently, the temperature deviations cause random fluctuations in the index of refraction.

The index of refraction can be modeled as

$$n = 1 + n_1(\vec{x}) \tag{2.1}$$

where $n_1(\vec{x})$ is the fluctuating part of the index of refraction due to small temperature differences throughout the atmosphere. These deviations result because of the naturally occurring random fluctuations in wind velocity called turbulence.[21] As a laser beam passes through the atmosphere, it encounters regions of different indexes of refraction called eddies. As shown in figure 2.1a, these turbulent blobs have different scale sizes and move randomly about the atmosphere due to the fluctuations in wind velocity. Although the fluctuations in $n_1(\vec{x})$ are small, to an observer at the end of a path length of several hundred meters, the intensity of a laser beam can fluctuate greatly due to the cumulative effect over the entire path. Because each eddy has a slightly different refractive index, different parts of the wavelets on the originally coherent wavefront are sped up or slowed down depending on the scale size of each eddy. From figure 2.1b, a large scale eddy only changes the direction of propagation because each wavelet sees basically the same index of refraction. Depending on how

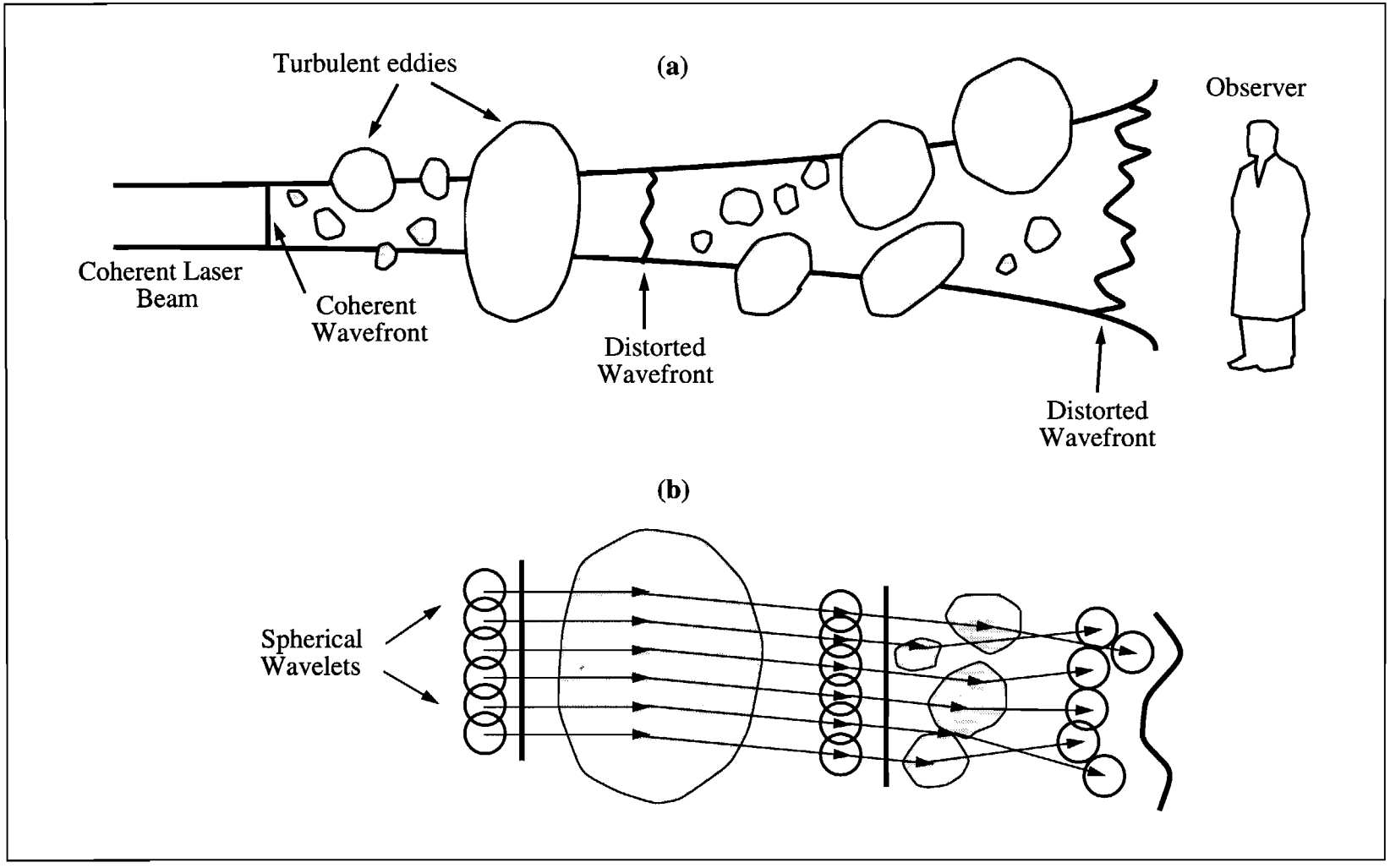


Figure 2.1. Effect of turbulent eddies on laser beam propagation. Large scale eddies change direction while small scale eddies cause diffraction.

the eddy is oriented relative to the wavefront, the beam is slightly steered from its original direction. This is called beam wander. The beam wander angle is always small for one large scale eddy. But as the beam encounters more large scale eddies there is more wander. In section 2.3 it is shown that beam wander actually benefits the analysis of the statistics of laser beam propagation through turbulence.

Another way turbulence can effect a laser beam is the effect of small scale eddies in figure 2.1b. A small scale eddy only acts on a part of the wavefront. Each part of the wavefront is randomly sped up or slowed down. The result is that as the beam continues to travel, the wavefront starts to interfere with itself. This is why the observer sees random variations in the laser beam intensity. Over a reasonably long path length (several hundred meters), the cumulative effects can be quite obvious.

There are several additional concepts to mention. The first is that the atmosphere is not really a discrete system of blobs of constant but different indices of refraction. Rather, the index continuously varies in a random arrangement throughout the atmosphere. Another point is that for isotropic turbulence, the scale size of an eddy lies within the inertial subrange l_0 to L_0. For the lower atmosphere, the inner scale l_0 ~ millimeters and L_0 ~ meters.[21] Finally, the fluctuations in the refractive index $n_1(\vec{\mathbf{r}})$ are assumed to be a 0 mean Gaussian random process.

The structure function is given by Kolmogorov

$$D_n(\vec{\mathbf{x}}_1, \vec{\mathbf{x}}_1 + \vec{\mathbf{x}}) = \langle [n_1(\vec{\mathbf{x}}_1 + \vec{\mathbf{x}}) - n_1(\vec{\mathbf{x}}_1)]^2 \rangle = C_n^2 x^{2/3} \tag{2.2}$$

where C_n^2 is related to the strength of turbulence. There is no dependence on the absolute location in the atmosphere, only the distance between two points. The covariance of the index fluctuations can be derived from $D_n(\vec{\mathbf{x}})$ and from using Fourier analysis. Let $B_n(\vec{\mathbf{x}})$ be the covariance of $n_1(\vec{\mathbf{x}})$,

$$B_n(\vec{x}) = \langle n_1(\vec{x}_1)\, n_1(\vec{x}_1 + \vec{x}) \rangle. \tag{2.3}$$

Then the Fourier transform of the covariance is

$$\phi_n(\vec{k}) = \frac{1}{(2\pi)^3} \int d^3\vec{r}\, B_n(\vec{x})\, e^{-i\vec{k}\cdot\vec{x}} \tag{2.4}$$

where $\vec{k}$ is the wave number. Under the assumption of isotropy and using Kolmogorov's structure function,[21]

$$\phi_n(\vec{k}) = 0.033\, C_n^2 k^{-11/3} \tag{2.5}$$

which is called the Kolmogorov spectrum of the index of refraction. This is a very important result and is used in the extended Huygens-Fresnel integral in the next section.

2.2 The Extended Huygens-Fresnel Principle

Throughout the 1960's, the Rytov approximation was the most commonly used method for developing the statistics of laser propagation through turbulence. But in 1971, Lutomirski and Yura proposed a more convenient way to analyze laser beam propagation. In order to gain insight into the extended Huygens-Fresnel method, the basic steps of the Huygens-Fresnel integral are presented. The scalar wave equation in an atmosphere of isotropic random refractive index is[22]

$$\nabla^2 U'(\vec{\rho}, z) + k^2 n^2(\vec{\rho}, z)\, U'(\vec{\rho}, z) = 0 \tag{2.6}$$

where $U'(\vec{\rho}, z)$ is the electric field, k is the wave number of a monochromatic wave, and $n(\vec{\rho}, z)$ is the index of refraction

$$n(\vec{\rho}, z) = 1 + n_1(\vec{\rho}, z)\,. \tag{2.7}$$

Consider a laser beam propagating in the +z-direction as shown in figure 2.2.

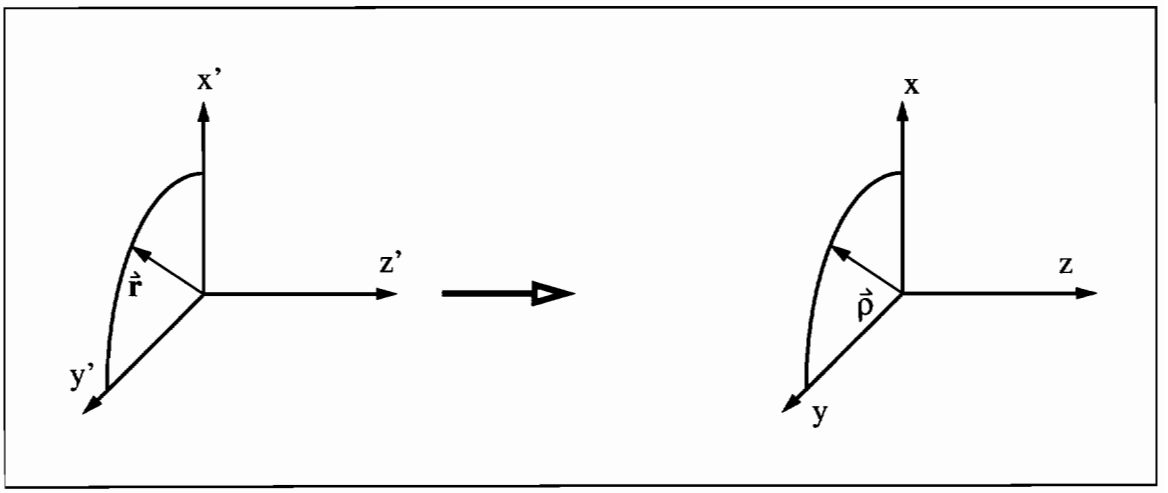


Figure 2.2. Geometry for the coordinate systems for a laser beam propagating in the +z-direction.

For a monochromatic wave propagating in the +z-direction, $U'(\vec{\rho}, z)$ can be represented as

$$U'(\vec{\rho}, z) = U(\vec{\rho}, z)\, e^{ikz} \tag{2.8}$$

lumping any index effects into $U(\vec{\rho}, z)$. At this point, $U(\vec{\rho}, z)$ is a slowly varying function of z. Substituting equation (2.8) into the original wave equation produces

$$\frac{\partial^2}{\partial z^2}[U(\vec{\rho}, z)\, e^{ikz}] + \left[\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right][U(\vec{\rho}, z)\, e^{ikz}] + k^2[1 + n_1]^2 U(\vec{\rho}, z)\, e^{ikz} = 0. \tag{2.9}$$

The next step is to evaluate $\frac{\partial^2}{\partial z^2}[U(\vec{\rho}, z)\, e^{ikz}]$:

$$\frac{\partial^2}{\partial z^2}[U(\vec{\rho}, z)\, e^{ikz}] = \left[\frac{\partial^2}{\partial z^2}U(\vec{\rho}, z) + 2ik\frac{\partial}{\partial z}U(\vec{\rho}, z) - k^2 U(\vec{\rho}, z)\right] e^{ikz}. \tag{2.10}$$

Using the paraxial approximation,[23]

$$\frac{\partial^2}{\partial z^2}U(\vec{\rho}, z) \ll k\left|\frac{\partial}{\partial z}U(\vec{\rho}, z)\right| \tag{2.11}$$

which means that the variations of $U(\vec{\rho}, z)$ in the z-direction are negligible over a propagation path of a few wavelengths. Inserting equation (2.10) back into equation (2.9) results in the paraxial wave equation

$$\nabla_T^2 U(\vec{\rho}, z) + 2ik\frac{\partial}{\partial z}U(\vec{\rho}, z) + k^2[2n_1]\, U(\vec{\rho}, z) = 0 \tag{2.12}$$

where $n_1 = n_1(\vec{\rho}, z)$ and where $\nabla_T^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$. The next step is to expand the electric field into a series of terms of order n_1

$$U = U_0 + U_1 + \ldots \tag{2.13}$$

where each term U_i is on the order of $n_1^{(i)}$. The term U_0 is the free space solution and U_1 is the first order term, and so on. Substituting the series for U into the paraxial wave equation gives

$$\nabla_T^2 [U_0 + U_1] + 2ik\left[\frac{\partial U_0}{\partial z} + \frac{\partial U_1}{\partial z}\right] + k^2 [2n_1] [U_0 + U_1] = 0 \tag{2.14}$$

where only the 0^{th} and 1^{st} order terms are kept. Regrouping by the order of $n_1^{(i)}$

$$n_1^{(0)}: \qquad \nabla_T^2 U_0(\vec{\rho}, z) + 2ik\frac{\partial}{\partial z} U_0(\vec{\rho}, z) = 0 \tag{2.15}$$

$$n_1^{(1)}: \quad \nabla_T^2 U_1(\vec{\rho}, z) + 2ik\frac{\partial}{\partial z} U_1(\vec{\rho}, z) = -2k^2 n_1 U_0(\vec{\rho}, z) \tag{2.16}$$

Equation (2.16) says that the first order term satisfies a paraxial wave equation with the source term on the right hand side. Using the Rytov approximation, equation (2.16) is solved by convolving the source term with the Green's function for the paraxial wave equation. Its solution is

$$U_1(\vec{\rho}, z) = \int_V d^2\hat{\mathbf{r}}\, dz'\ G_{FS}(\vec{\rho}, \hat{\mathbf{r}}, z, z')\ [2k^2 n_1(\vec{\rho}, z)\, U_0(\vec{\rho}, z)]\,. \tag{2.17}$$

where $G_{FS}(\vec{\rho}, \hat{\mathbf{r}}, z, z')$ is the free space Green's function for the paraxial wave equation. This integral is defined over the volume enclosing the electric field sources.

Throughout the 1960's equation (2.17) was used to derive the higher order statistics for a wave propagating through turbulence. While it did give initial insight into the effect of turbulence, the extended Huygens-Fresnel method became popular for

several reasons. One reason was that the integral solution for $U_1(\vec{\rho}, z)$ could be reduced to a surface integral over the aperture of the original wave using Green's theorem, instead of working with the three dimensional volume integral of equation (2.17). Another advantage occurred when the Huygens-Fresnel integral was applied to laser beam propagation because the geometry allowed for some additional approximations which further simplified the solution. Finally, Lutomirski and Yura showed that the extended Huygens-Fresnel method easily separated the properties of the random medium from the geometry of the aperture and propagation distance which made it more straight forward to calculate the higher order statistics.

The regular Huygens-Fresnel principle is presented first before applying the extension to a random medium. From Born and Wolf[2], the regular Huygens-Fresnel integral for free space is

$$U(\vec{\rho}, z) = \frac{1}{4\pi}\int_S \{ U(\vec{r}, z') \frac{\partial}{\partial n} G_{FS}(\vec{\rho}, \vec{r}, z, z') - G_{FS}(\vec{\rho}, \vec{r}, z, z') \frac{\partial}{\partial n} U(\vec{r}, z') \} \, d^2\vec{r}. \tag{2.18}$$

In this integral equation for homogeneous media, $G_{FS}(\vec{\rho}, \vec{r}, z, z')$ is the free space Green's function and $U(\vec{r}, z')$ is the original electric field distribution. $U(\vec{\rho}, z)$ is the resultant electric field. The surface S is the surface that surrounds the original volume, V, from equation (2.17). Already the solution is a two dimensional integral over the exit aperture instead of a three dimensional volume integral. This equation is useful to describe electric field propagation because the only variables that need to be specified are the original electric field, the aperture size from which it originates, and the free space Green's function. The Huygens-Fresnel integral is applied to a laser beam traveling in the +z direction without turbulence shown in Figure 2.3a. Specifically, equation (2.18) is devoted to the case where the aperture is oriented normal to the z-direction and the initial electric field is propagating along the z-axis. Under

these conditions, $\frac{\partial}{\partial n} \rightarrow \frac{\partial}{\partial z}$ and equation (2.18) can be approximated to be

$$U(\vec{\rho}, z) = \frac{k e^{ikz}}{2\pi i} \int_S d^2\hat{\mathbf{r}}\ G_{FS}(\vec{\rho}, \hat{\mathbf{r}}, z, z')\, U(\hat{\mathbf{r}}, z') . \tag{2.19}$$

In deriving equation (2.19), the approximation was used that the wavefront over aperture $\hat{\mathbf{r}}$ is slowly varying compared to a wavelength. The free space Green's function for the paraxial wave equation is

$$G_{FS}(\vec{\rho}, \hat{\mathbf{r}}, z, z') = \frac{\exp\left[\frac{ik(\vec{\rho} - \hat{\mathbf{r}})^2}{2(z - z')}\right]}{z - z'} . \tag{2.20}$$

The physical interpretation of this integral solution is that $U(\vec{\rho}, z)$ is the result of super-imposing a spherical wave traveling through space integrated over the original exit aperture given an initial electric field distribution.

Figure 2.3b shows the geometry when the laser beam travels through turbulence. The main idea of the extended Huygens-Fresnel principle is that the free space Green's function given by equation (2.20) is modified to include the effect of the turbulence on a spherical wave propagating through turbulence. Using Lutomirski's and Yura's notation, the Green's function for the turbulent atmosphere can be written as

$$G_T(\vec{\rho}, \hat{\mathbf{r}}, z, z') = \frac{\exp\left[\frac{ik(\vec{\rho} - \hat{\mathbf{r}})^2}{2(z - z')} + \psi_f(\hat{\mathbf{r}}, \vec{\rho}, z, z')\right]}{z - z'} \tag{2.21}$$

where ψ_f includes the random effect of turbulence on the *forward* path from aperture $\hat{\mathbf{r}}$ to the field point $\vec{\rho}$. Turbulence effects both the amplitude and phase. Therefore, ψ_f can be written as $\psi_f = \chi + i\phi$ where χ is called the log-amplitude effect and $i\phi$ is called the phase effect of the turbulence. The original analytical derivation for ψ_f was obtained using the Rytov approximation. Consequently, the extended Huygens-

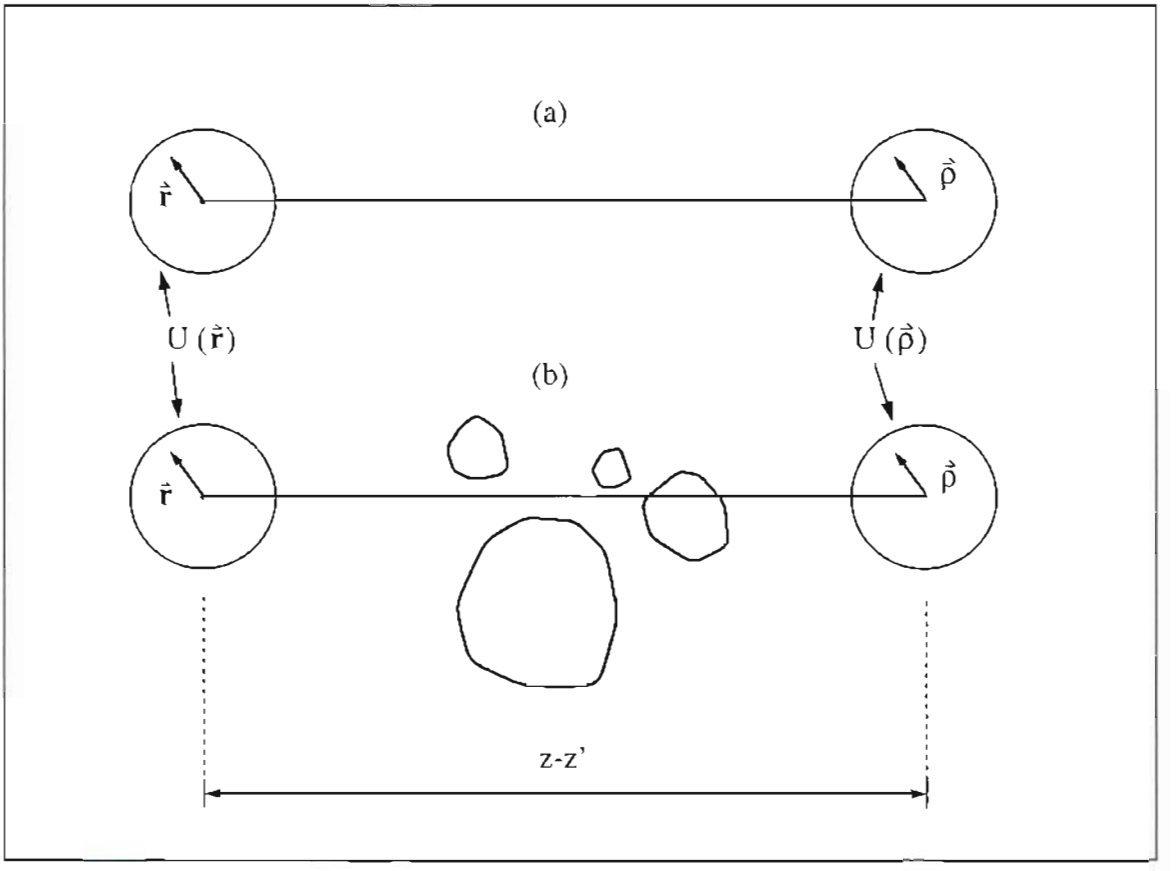


Figure 2.3. Geometry for the Huygens-Fresnel integral solution. (a) No turbulence (b) With turbulence.

Fresnel principle is a modification of the free space Huygens-Fresnel principle to include the turbulence effect adapted from the Rytov approximation.

The connection between ψ and the random index of refraction will provide more insight into why the Green's function for turbulence is valid. Following Tatarskii, the electric field can be written as a combination of an amplitude and phase term

$$U = Ae^{is} \tag{2.22}$$

So, as in equation (2.13) where the electric field is expanded into a series of terms of

order n_1, $U = U_0 + U_1$ keeping the first two terms. Using the notation of equation (2.22),

$$U_0 = A_0 e^{is_0}$$

$$U_1 = A_1 e^{is_1}.$$

$$U = A_0 e^{is_0} + A_1 e^{is_1}.$$

The function ψ is a measure of the amplitude and phase fluctuations relative to the free space values. So ψ is defined as

$$\psi = \ln\left[\frac{U}{U_0}\right] = \ln\left[\frac{A}{A_0} e^{i(s-s_0)}\right] = \ln\left[\frac{A}{A_0}\right] + i(s - s_0) \tag{2.23}$$

It can be approximated that $\ln\left[\frac{A}{A_0}\right] = \ln\left[1 + \frac{A_1}{A_0}\right] \approx \frac{A_1}{A_0}$ since $A_1 \ll A_0$. Also, $s - s_0 = s_1$. Therefore,

$$\psi = \frac{A_1}{A_0} + is_1 = \chi + i\phi$$

$$= \mathrm{Re}\left[\frac{U_1}{U_0}\right] + i\mathrm{Im}\left[\frac{U_1}{U_0}\right] \tag{2.24}$$

By assuming $\psi = \frac{U_1}{u_0}$ and substituting this into the wave equation (2.16), an expression for ψ can be achieved. The solution for ψ using Green's theorem is

$$\psi = \frac{1}{U_0(\vec{\rho}, z)} \int_V n_1(\vec{\rho}, z)\, U_0(\vec{r}, z')\, G_{FS}(\vec{\rho}, \vec{r}, z, z')\, d^2\vec{r}\, dz' \tag{2.25}$$

where, again, the volume integral, V, encloses any electric field sources. Since n_1 is a

zero-mean Gaussian random variable, ψ is a zero-mean Gaussian random variable because $\psi \sim \int_V n_1 dV$. The next step is to let $U_0(\vec{\rho}, z)$ be the electric field due to a point source in equation (2.25). The statistics can now be developed, keeping in mind that the extended Huygens-Fresnel integral contains a factor e^{ψ} which is a log normal random variable. Yura outlines the procedure in developing $<e^{\psi}>$ and $<e^{\psi+\psi^*}>$, factors that occur commonly in laser beam propagation through turbulence, in reference [5]. To get the basic idea, Yura shows that

$$<\exp[\psi(\vec{\rho}_1, \vec{r}_1) + \psi^*(\vec{\rho}_2, \vec{r}_2)]> = \exp[-<|\psi|^2> - <\psi(\vec{\rho}_1, \vec{r}_1)\psi^*(\vec{\rho}_2, \vec{r}_2)>] \tag{2.26}$$

where $\vec{\rho}_1$ and $\vec{\rho}_2$ are two different points in the scattering plane, $\vec{r}_1$ and $\vec{r}_2$ are two different points in the transmitting aperture, and, from equation (2.25),

$$<|\psi|^2> = 2\pi k^2 z \int \phi_n(\vec{k})\, d^3\vec{k} \tag{2.27}$$

and $\phi_n(\vec{k})$ is the spectrum of the index of refraction fluctuations introduced in equation (2.5). Yura only derived the statistics through the second order. Later, Lee, Holmes, and Kerr derived the generalized n^{th} order spherical wave mutual coherence function, using the statistics of log normal random variables.[24]

An example illustrates the technique. The mutual intensity function is computed between two points in the $\vec{\rho}$ plane in Figure 2.3b

$$\Gamma(\vec{\rho}_1, \vec{\rho}_2, z, z') = <U(\vec{\rho}_1, z)\, U^*(\vec{\rho}_2, z)>. \tag{2.28}$$

Using the extended Huygens-Fresnel integral the electric field in the $\vec{\rho}$ plane is

$$U(\vec{\rho}, z) = \frac{ke^{ikz}}{2\pi iz}\int d^2\vec{r}\, U_T(\vec{r}, z') \exp\left[\frac{ik}{2z}(\vec{\rho} - \vec{r})^2 + \psi_f(\vec{\rho}, \vec{r}, z, z')\right] \qquad (2.29)$$

where $U_T(\vec{r}, z')$ is the transmitted electric field of a laser beam in the $\vec{r}$ aperture S and $\psi_f(\vec{\rho}, \vec{r}, z, z')$ includes the effect of turbulence on a spherical wave on the forward path from the $\vec{r}$ plane to the $\vec{\rho}$ plane. The idea is to form the product $U(\vec{\rho}_1, z)\, U^*(\vec{\rho}_2, z)$ and then take the ensemble average

$$\Gamma(\vec{\rho}_1, \vec{\rho}_2, z, z') = \frac{k^2}{(2\pi)^2 z^2}\iint d^2\vec{r}_1 d^2\vec{r}_2\, U_T(\vec{r}_1, z')\, U_T^*(\vec{r}_2, z')$$

$$\times \exp\left[\frac{ik}{2z}(\vec{\rho}_1 - \vec{r}_1)^2 - \frac{ik}{2z}(\vec{\rho}_2 - \vec{r}_2)^2\right]$$

$$\times \langle \exp[\psi_f(\vec{\rho}_1, \vec{r}_1, z, z') + \psi_f^*(\vec{\rho}_2, \vec{r}_2, z, z')] \rangle . \qquad (2.30)$$

The ensemble average only applies to the random effect of the turbulence since all other terms are deterministic. The term in angle brackets is the second order spherical wave mutual coherence function. The remaining steps are to substitute for the mutual coherence function and for the electric field in the $\vec{r}$ plane and to perform the integrations.

Through the early 1970's use of the extended Huygens-Fresnel integral gradually increased. However, it was only used to analyze double-ended laser systems. The next section shows how to use the interaction between turbulence and speckle created by the scattering of a laser beam from a diffuse surface to allow for single-ended systems.

2.3 Single-Ended Lidars Using Speckle-Turbulence Interaction

Double-ended systems were useful to study the effect of the atmosphere on laser beam propagation and to estimate the crosswind speed. The setback, however, was that the laser was at one end and the detector was at the other. A more practical setup would put the laser and detector at the same location, as shown in figure 2.4. This problem was attacked by Lee, Holmes, and Kerr in the mid 1970's. Their approach was to use a diffuse scattering hard target to transform the laser signal into a speckle pattern which propagated back to the receiver. This allowed the entire laser system to be located at the same place but it also required some new analysis on how to characterize a speckle pattern as it propagated through the turbulent atmosphere back to the receiver.

Speckle results when a source of coherent light is backscattered from a rough surface. The general idea is shown in figure 2.5. From figure 2.5a, a source of coherent light which reflects from a mirror results in no change to the coherent wavefront. The reason the wavefront does not change is because the angle of incidence and reflection are the same for every wavelet and thus they do not interfere.

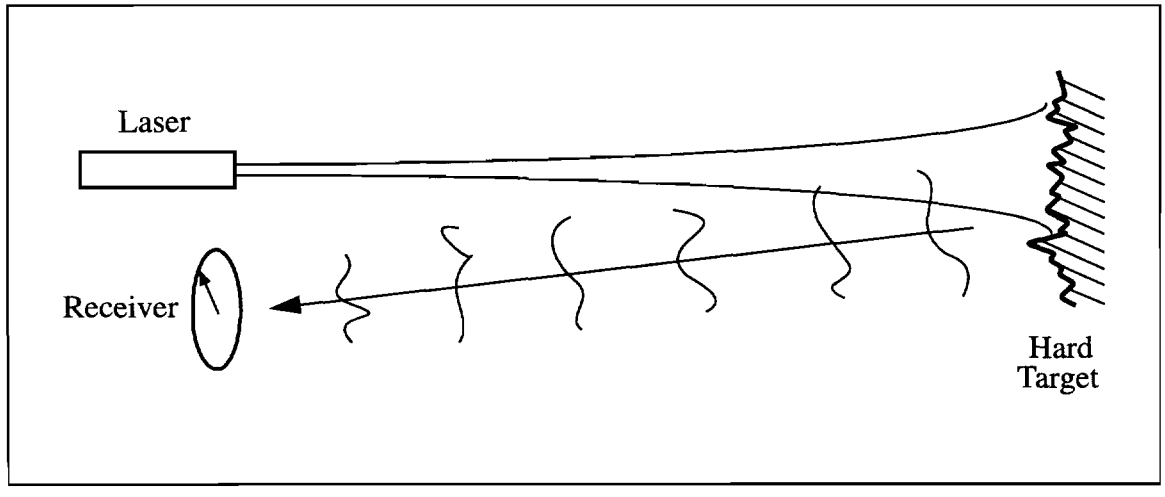


Figure 2.4 Schematic of a single-ended laser system.

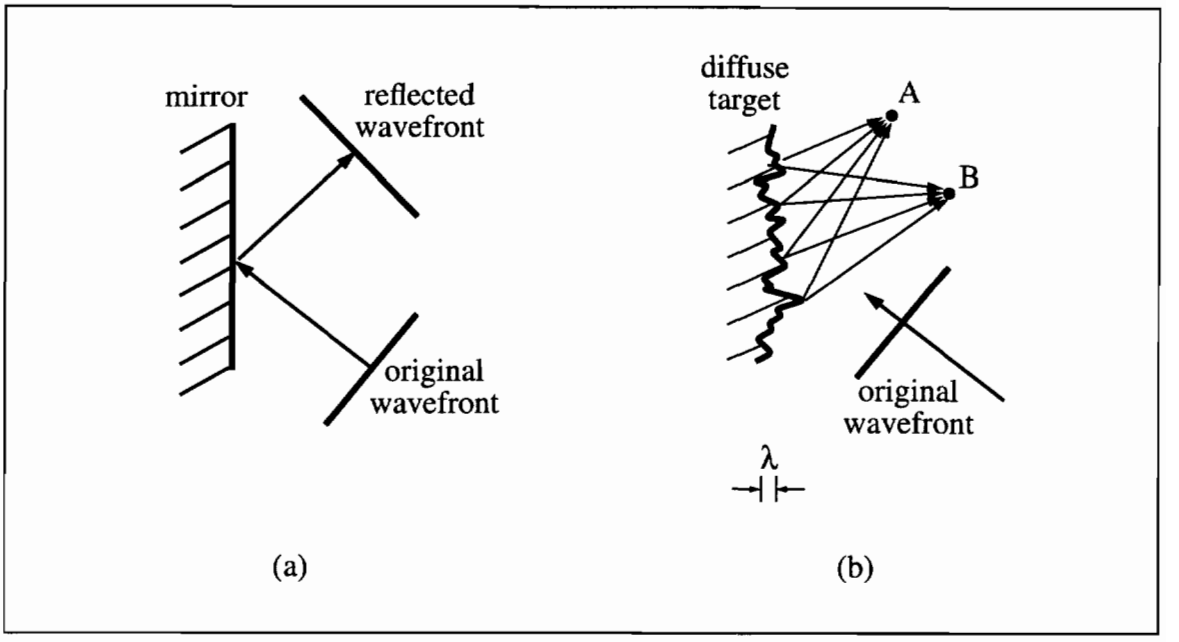


Figure 2.5 Generation of a speckle pattern. (a) a mirror only reflects wavefront because angle of incidence equals angle of reflection. (b) a diffuse target scatterers the incident wavefront in all directions. The scattered wavelets mutually interfere creating a speckle pattern.

However, a rough surface scatters the incident laser beam in all directions causing the wavelets to mutually interfere as shown in figure 2.5b. The electric fields at points A and B are the summations of wavelets that have traveled different path lengths. If the surface heights of the individual scatterers differ by at least a wavelength, then each contribution has a different phase. The result is that the scattered intensity consists of a multitude of bright spots, where the interference is highly constructive, dark spots, where the interference is highly destructive, and intensity levels in between these limits. Such an arrangement of random light and dark spots is called a speckle pattern.[25]

The resultant speckle field at points A and B can be written as the sum of the contributions from the rough surface

$$U(\vec{\rho}, z) = \sum_{i=1}^{N} \frac{1}{\sqrt{N}} U_i(\vec{\rho}, z) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} |U_i| e^{i\phi_i}. \tag{2.31}$$

In this equation, N is the number of contributors and ϕ_i is the phase of each contributor. The surface heights of the individual scatterers are random and uncorrelated with each other. This fact leads to several assumptions that provides the statistics of the resultant speckle field. The first is that the amplitudes and phases of the speckle field immediately after scattering are statistically independent for any two points on the speckle field wave front. The phase of each U_i is produced only by its path length while the amplitude is determined by the strength of scattering of each section of the rough surface. In addition, for a fully developed speckle pattern, the phases are uniformly distributed between $-\pi$ and π because the surface heights are at least as rough as one wavelength. Therefore, the path length differences result in all possible phase shifts between $-\pi$ and π. Using these two assumptions, the ensemble average of the speckle field can be written as

$$<U(\vec{\rho}, z)> = \frac{1}{\sqrt{N}} \sum_{l=1}^{N} <|U_i|><\cos(\phi_l)> + i\frac{1}{\sqrt{N}} \sum_{l=1}^{N} <|U_i|><\sin(\phi_l)>$$

$$= 0 \qquad (2.32)$$

because $<\cos(\phi_l)>$ and $<\sin(\phi_l)> = 0$. Therefore, the resultant speckle field is a zero-mean random process. Because the surface heights of the individual scatterers are random and uncorrelated, the electric fields at points A and B are statistically independent close to the surface. The analytical way to express statistical independence is that the electric fields at points A and B are δ-function correlated

$$<U_A(\vec{\rho}_1, z)\, U_B^*(\vec{\rho}_2, z)> = <U_A(\vec{\rho}_1, z)\, U_B^*(\vec{\rho}_2, z)>\delta(\vec{\rho}_1 - \vec{\rho}_2)\,. \qquad (2.33)$$

As the number of contributors increases, the central limit theorem takes over,

and the real and imaginary parts of the speckle field become zero-mean Gaussian random variables.[26] D. Draper, Holmes, and J. Peacock[27] in their study of the laser speckle phase found that the number of independent scatterers for a typical diffuse hard target is on the order of 10^5. This high number for N certainly indicates that the speckle field is Gaussian.

The schematic of the speckle-turbulence propagation problem for a single-ended system using a diffuse hard target is shown in figure 2.6. The transmitted laser beam at position I travels through turbulence to position II just before scattering as shown in figure 2.6a. The turbulence modulates the amplitude and phase of the laser beam as it propagates on the forward path to the target. The system would be double-ended if a detector were placed at the end of the path instead of the hard target. The diffuse hard target randomizes the beam into a speckle pattern just after scattering at position III as shown in figure 2.6b. The wave front of the speckle field is shown by random light and dark spots which represent the areas of coherent and incoherent interference. These areas can be thought of as wavelets of the speckle field. These wavelets are zero-mean, Gaussian, and δ-function correlated. The speckle field propagates back to the detector at position IV in figure 2.6c after being modulated again by the turbulence on the return path. Because the turbulence causes additional interference and mixing of the speckle wavelets on the return path, different points on the wave front at the receiver are no longer Gaussian and independent.

The statistics for the electric fields can now be derived for single-ended systems using the extended Huygens-Fresnel principle and speckle-turbulence interaction. The first step is to use the extended Huygens-Fresnel integral to describe the electric field after propagating through turbulence just before scattering given the initial transmitted beam. The method for doing this was discussed in section 2.2 and resulted in equation (2.29). The next step is to express the field immediately after scattering as a function of the field just before the target multiplied by the target

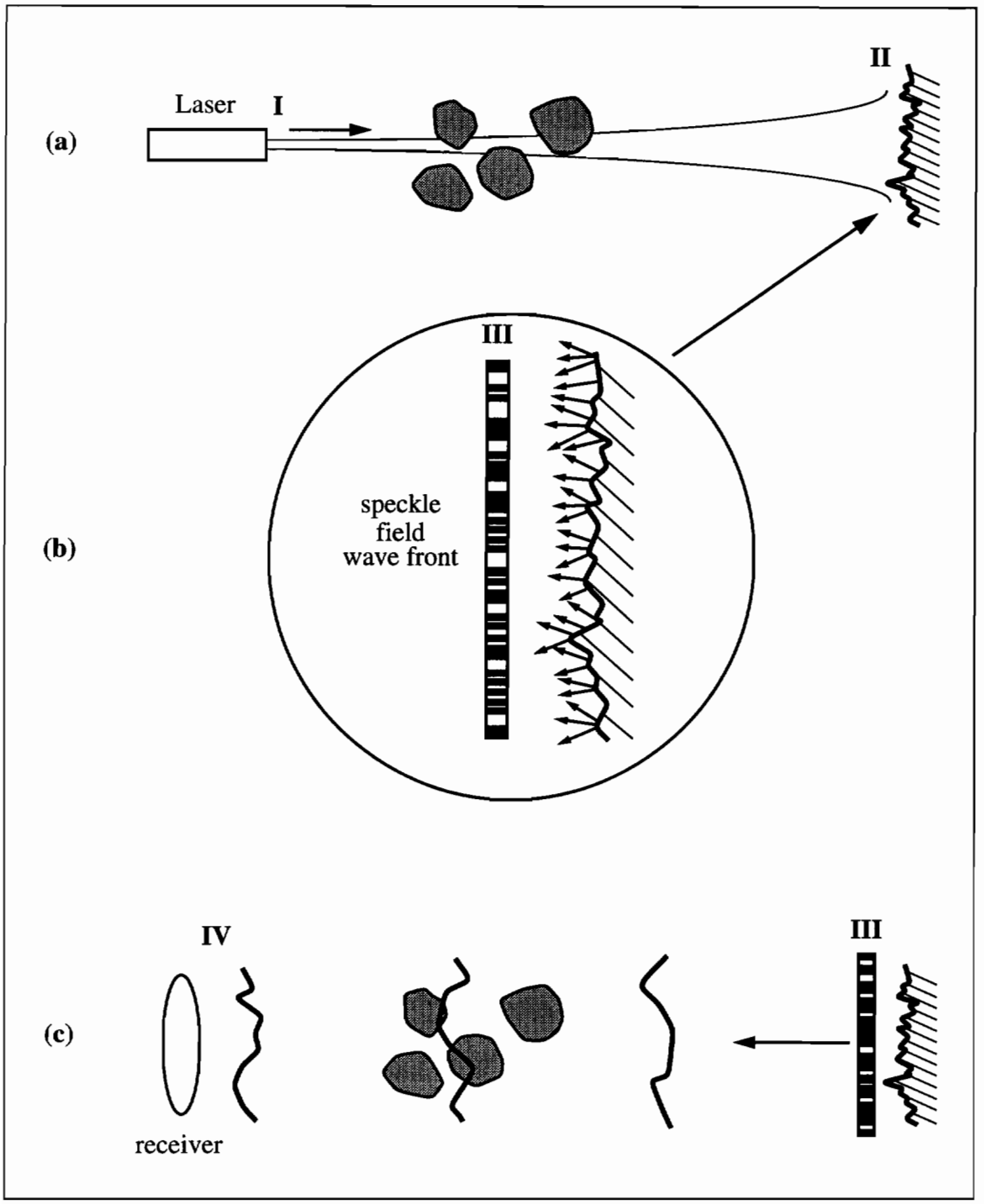


Figure 2.6 Characterization of the electric fields in a single-ended laser system using a diffuse hard target as scatterer. (a) Transmitted beam at I travels through turbulence to II just before scattering. (b) Scattered beam at III results in a speckle field. (c) Speckle field travels back through the turbulence to the detector.

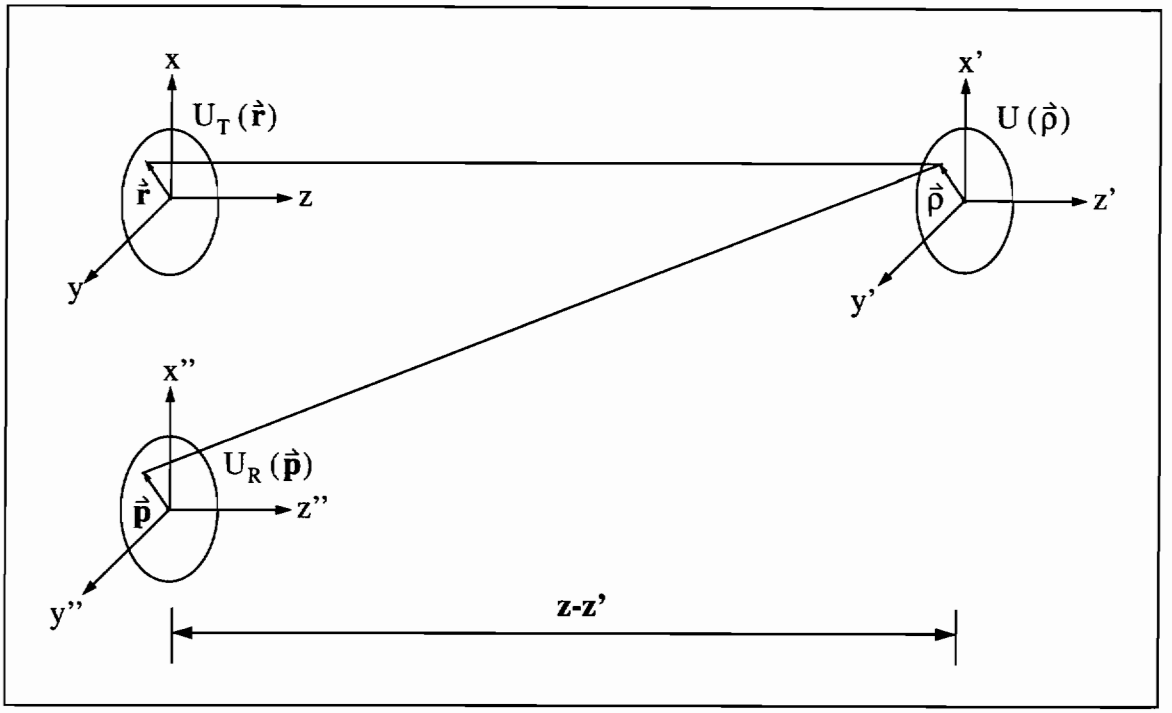


Figure 2.7 Geometry for the coordinate systems for the single-ended laser system.

scattering factor. This factor takes into account the effects of the diffuse target on the incident fields. Finally, the fields at the receiver are formulated by reapplying the extended Huygens-Fresnel principle for the return path given the scattered fields.

To illustrate the method, the mutual intensity function is derived for the electric fields at the detector. The geometry for the problem is shown in figure 2.7. Using the extended Huygens-Fresnel integral, the field just before scattering is represented by

$$U(\vec{\rho}, z) = \frac{ke^{ikz}}{2\pi iz}\int d^2\hat{\mathbf{r}}\, U_T(\hat{\mathbf{r}}, z')\exp\left[\frac{ik}{2z}(\vec{\rho}-\hat{\mathbf{r}})^2 + \psi_f(\vec{\rho}, \hat{\mathbf{r}}, z, z')\right] \qquad (2.34)$$

where $U_T(\hat{\mathbf{r}}, z')$ is the transmitted laser field and ψ_f describes the effect of the turbulence on the laser beam propagating on the forward path from the transmitter to the diffuse target. The field immediately after scattering can be written as

$$U_S(\vec{\rho}, z) = U(\vec{\rho}, z)\, T(\vec{\rho}) \tag{2.35}$$

where $T(\vec{\rho})$ is the target scattering factor which includes the effect of the diffuse target on the incident field. The field at the receiver can be obtained by reapplying the extended Huygens-Fresnel principle

$$U_R(\vec{\mathbf{p}}) = \frac{ke^{ikz}}{2\pi iz}\int d^2\vec{\rho}\, T(\vec{\rho})\, U(\vec{\rho}, z) \exp\left[\frac{ik}{2z}(\vec{\mathbf{p}} - \vec{\rho})^2 + \psi_b(\vec{\mathbf{p}}, \vec{\rho}, z, z')\right] \tag{2.36}$$

where ψ_b describes the effect of the turbulence on the *backward* path from the target to the receiver. The mutual intensity function at the receiver is

$$\Gamma(\vec{\mathbf{p}}_1, \vec{\mathbf{p}}_2) = \langle U_R(\vec{\mathbf{p}}_1)\, U_R^*(\vec{\mathbf{p}}_2)\rangle$$

$$\Gamma(\vec{\mathbf{p}}_1, \vec{\mathbf{p}}_2) = \frac{k^2}{(2\pi)^2 z^2}\iint d^2\vec{\rho}_1 d^2\vec{\rho}_2 \exp\left[\frac{ik}{2z}(\vec{\mathbf{p}}_1 - \vec{\rho}_1)^2 - \frac{ik}{2z}(\vec{\mathbf{p}}_2 - \vec{\rho}_2)^2\right]$$

$$\times \langle T(\vec{\rho}_1)\, T^*(\vec{\rho}_2)\, U(\vec{\rho}_1, z)\, U^*(\vec{\rho}_2, z) \exp\left[\psi_b(\vec{\mathbf{p}}_1, \vec{\rho}_1, z, z') + \psi_b^*(\vec{\mathbf{p}}_2, \vec{\rho}_2, z, z')\right]\rangle$$

(2.37)

Since the target scattering function and the turbulence are independent, the term in angle brackets can be rewritten as

$$\langle T(\vec{\rho}_1)\, T^*(\vec{\rho}_2)\rangle\langle U(\vec{\rho}_1, z)\, U^*(\vec{\rho}_2, z) \exp\left[\psi_b(\vec{\mathbf{p}}_1, \vec{\rho}_1, z, z') + \psi_b^*(\vec{\mathbf{p}}_2, \vec{\rho}_2, z, z')\right]\rangle \tag{2.38}$$

The first term in angle brackets is the mean value product of the target scattering factor which implies that the averaging must be performed over an ensemble of diffuse

targets. But there is only one target. This is where the beam wander phenomenon benefits the analysis. Since the large scale eddies cause the beam to be slightly but perpetually altered from its original direction, the beam constantly sees a different section of the diffuse target. As the beam continues to meander over the surface, a different speckle field is generated, called an evolving speckle pattern. The beam wander causes the same effect as if there were an ensemble of different targets. Not only are two different wavelets of the speckle field uncorrelated, but each wavelet for one instant in time is also independent from the speckle wavelet at the same location but for a different instant in time because the speckle field is evolving.

The fields after scattering are also δ-function correlated. Therefore, the mean value product of the target scattering factor can be written as

$$<T(\vec{\rho}_1)\,T^*(\vec{\rho}_2)> = T_0^2\,\delta(\vec{\rho}_1 - \vec{\rho}_2) \tag{2.39}$$

where T_0 is related to the strength of backscattering of the target. One of the benefits of a diffuse scattering surface becomes apparent since the δ-function reduces the number of integrations. Applying the δ-function, the mutual intensity function becomes

$$\Gamma(\vec{\rho}_1, \vec{\rho}_2) = \frac{k^2 T_0^2}{(2\pi)^2 z^2}\int d^2\vec{\rho}_1 \exp\left[\frac{ik}{2z}(\vec{\rho}_1 - \vec{\rho}_1)^2 - \frac{ik}{2z}(\vec{\rho}_2 - \vec{\rho}_1)^2\right]$$

$$\times <U(\vec{\rho}_1, z)\,U^*(\vec{\rho}_1, z)\exp[\psi_b(\vec{\rho}_1, \vec{\rho}_1, z, z') + \psi_b^*(\vec{\rho}_2, \vec{\rho}_1, z, z')]>. \tag{2.40}$$

Substituting the expression for the field before scattering, the mutual intensity function can be written as

$$\Gamma(\vec{\mathbf{p}}_1, \vec{\mathbf{p}}_2) = \frac{k^4 T_0^2}{(2\pi)^4 z^4} \iint d^2\vec{r}_1 d^2\vec{r}_2 \int d^2\vec{\rho}_1 \exp\left[\frac{ik}{2z}(\vec{\rho}_1 - \vec{r}_1)^2 - \frac{ik}{2z}(\vec{\rho}_1 - \vec{r}_2)^2\right]$$

$$\times U_T(\vec{r}_1, z') U_T(\vec{r}_2, z') \exp\left[\frac{ik}{2z}(\vec{\mathbf{p}}_1 - \vec{\rho}_1)^2 - \frac{ik}{2z}(\vec{\mathbf{p}}_2 - \vec{\rho}_1)^2\right]$$

$$\times \langle \exp[\psi_f(\vec{\rho}_1, \vec{r}_1) + \psi_f^*(\vec{\rho}_1, \vec{r}_2) + \psi_b(\vec{\mathbf{p}}_1, \vec{\rho}_1) + \psi_b^*(\vec{\mathbf{p}}_2, \vec{\rho}_1)] \rangle . \qquad (2.41)$$

The term in angle brackets is the fourth order mutual coherence function. The remaining steps are to substitute the expression for the fourth order mutual coherence function and to perform the integrations. Setting $p_1 = p_2$ in equation (2.41) reduces the mutual intensity function to the expression for the average intensity at one point in the receiver aperture.

So far the only type of single-ended system discussed has been the type with a diffuse hard target as a scatterer. In Chapter 3, aerosols replace hard targets as the backscatterer.

This chapter reviewed some of the important theoretical developments to statistically describe laser propagation through turbulence. Speckle was introduced into the process when a metallic rough surface was used to backscatter the laser beam in order to locate the laser and receiver at the same end of the propagation path. The time delayed mutual coherence function was computed between two points in the receiver plane to illustrate the method of applying the extended Huygens-Fresnel principle to laser beam propagation through turbulence.

Chapter 3

The Effect of Aerosols on Single-Ended Lidars

Since the advent of the laser in the 1960's, researchers have noticed the potential to monitor the environment. An understanding of the properties of aerosols became important because of the environmental push to measure and control pollutants in the atmosphere. For example, lasers have been used to evaluate the hazards of certain toxic substances, such as pesticides sprayed over fields. Lasers have also been utilized to determine the mean flow of pollution created from car exhaust near cities. In coal mines, lasers have been employed to detect traces of harmful gases to warn workers of otherwise undetectable hazards. In addition, lasers have been directed into the atmosphere to determine the strength of backscattering of aerosols. Such information has been useful because of the potential to improve weather forecasting. In each application, the laser has been used as a remote sensor to gather information about the properties, behavior, and physical principles of aerosols.

In this thesis project, aerosols are used to backscatter a laser beam to estimate the atmospheric wind speed and to study the effect of aerosols on laser beam propagation. The use of aerosols to backscatter laser beams creates some extra problems not present with the hard target systems. Aerosols backscatter much less laser radiation than hard targets making the detection process difficult. This leads to a poor signal-to-noise ratio at the detector necessitating some additional signal processing. In contrast to diffuse metallic surfaces, aerosols are moving targets which must be included in the analysis of the statistics of laser beam propagation.

Section 3.1 introduces the properties of aerosols, including their distribution, size, and concentration in the lower atmosphere (lower 4 km). Section 3.2 shows how the scale size of aerosols relative to the wavelength of the laser effects the scattering profile. This presentation leads to the volume backscattering coefficient. In section 3.3 the modifications to the extended Huygens-Fresnel integral are demonstrated along with an example of a typical application. Finally, in section 3.4, phase considerations are mentioned with emphasis on the decorrelation effect of aerosols on the statistics of laser beam propagation.

3.1 Properties of Aerosols

When Mt. St. Helens erupted in May, 1980 (figure 3.1) the entire Pacific Northwest was covered by a giant volcanic dust cloud. Millions of tons of ash were spouted into the air, scattering eastward as far as several hundred kilometers. The fall-out blanketed the streets of Northwest cities such as Portland, Oregon. Cars passing along residential streets kicked up ash plumes causing people to wear breathing masks. This natural source of pollution is one example of aerosols, clouds of dispersed matter in the atmosphere. Aerosols, then, are defined as particles suspended in a gaseous medium, in this case, air.[28] Particles that are too large succumb to the forces of gravity and fall to the earth, while particles that are too small rapidly attach to the larger

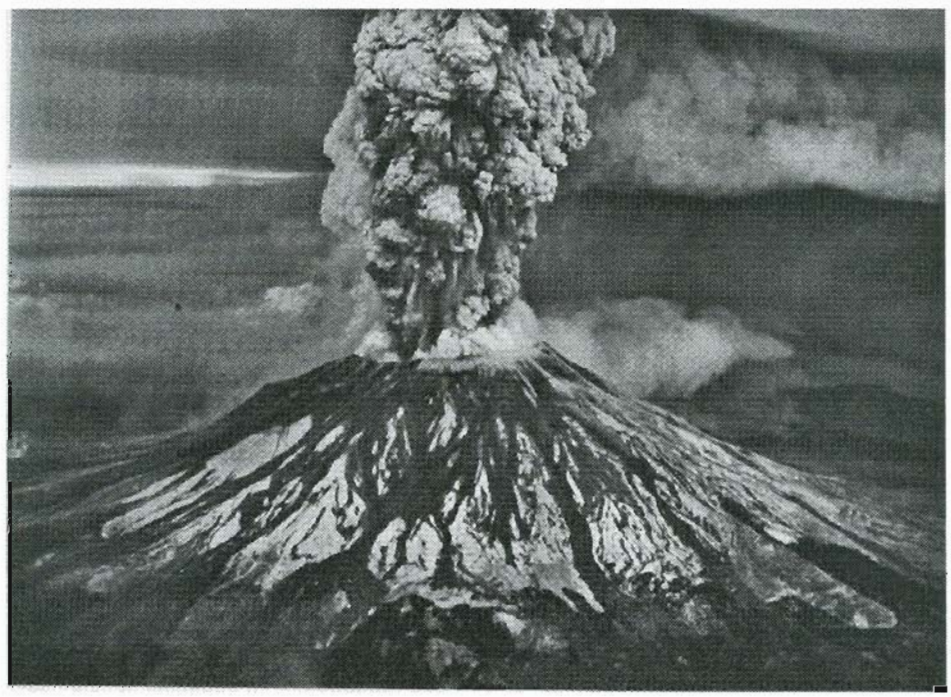


Figure 3.1 Mt. St. Helens was a giant source of aerosols, erupting millions of tons of volcanic dust into the atmosphere in May, 1980. [From Hidy[29]]

aerosol particles. Therefore, most aerosols fall within a certain range of scale sizes, from about $0.01\mu m$ to $20\mu m$, a range of over 3 orders of magnitude.

Volcanic dust is one example of aerosol particles. Other types are Aitken or condensation nuclei, smog and haze, car exhaust, pollens, dust, dirt particles, smokes, and sea salt nuclei. Figure 3.2 shows a chart of different particles along with their typical scale sizes. Rain, drizzle, snow, and hail are not examples of aerosols because they do not stay suspended in air, but are included in figure 3.2 for reference. The smallest sized particles are called Aitken particles. These particles have scale sizes less than $0.1\mu m$. The next range of particles occur in the range between $0.1\mu m$ and $1\mu m$. These are called large aerosols. Particles which have scale sizes greater than $1\mu m$ are classified in the giant category.

Some electron micrographs of actual aerosol particles are shown in figure 3.3. Figure 3.3a shows a typical concentration of large continental aerosols in Central Europe collected by Junge.[28] Other pictures of aerosols appear in figures 3.3b and c, showing the difference in concentration between rural and urban locations. Since cities have more pollution than the countryside, the aerosol concentration is greater in cities. From these figures, it can be seen that some aerosols have near spherical shapes, but most are irregular.

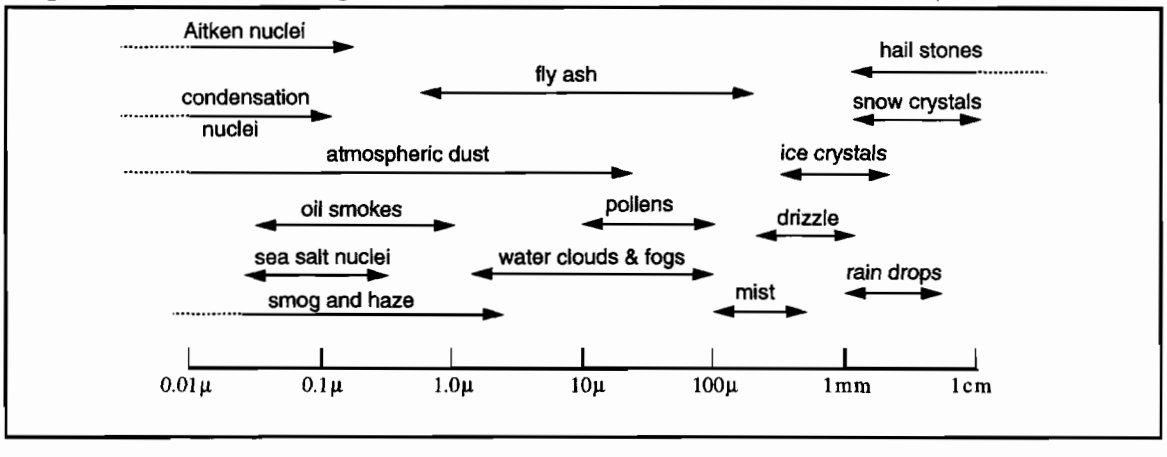


Figure 3.2 Scale sizes of some common aerosol particles. [from Measures[30]]

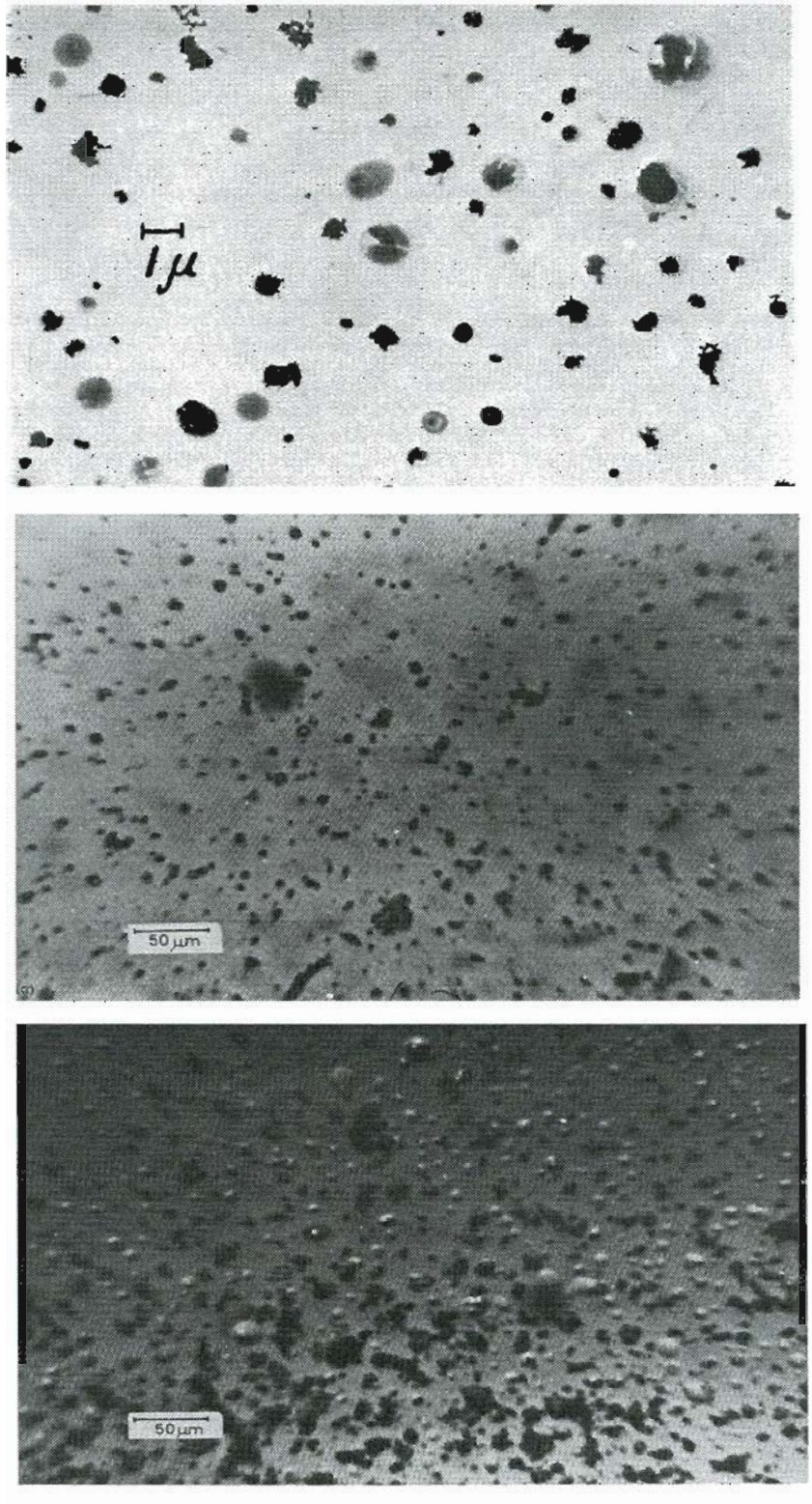


Figure 3.3 Electron micrographs of typical aerosols. (a) Continental aerosols in Central Europe [from Junge[28]]. (b) aerosols at a rural site. (c) aerosols at an urban site [b and c from Twomey[31]].

Therefore, irregularly shaped aerosols are given an equivalent radius of spherical particles with the same properties. This technique is used in the next section to study the effect of particle size on backscattering.

There are at least two processes which effect the number concentration of aerosols in the lower atmosphere as shown in figure 3.4. The first factor is due to thermal updrafts as shown in figure 3.4a. As the sun heats the ground, heat radiates back to the sky. This rising current of air helps to keep more aerosols suspended. Another factor which effects the concentration of aerosols is the washout due to rainfall as shown in figure 3.4b. Rain has a cleansing effect. As drops fall to the ground, they collide with the aerosols and bring them down, cleaning the air. This process is of great interest to scientists studying acid rain. Although the air is cleansed, the rain brings the pollution from factories back to the earth, effecting fish habitats and

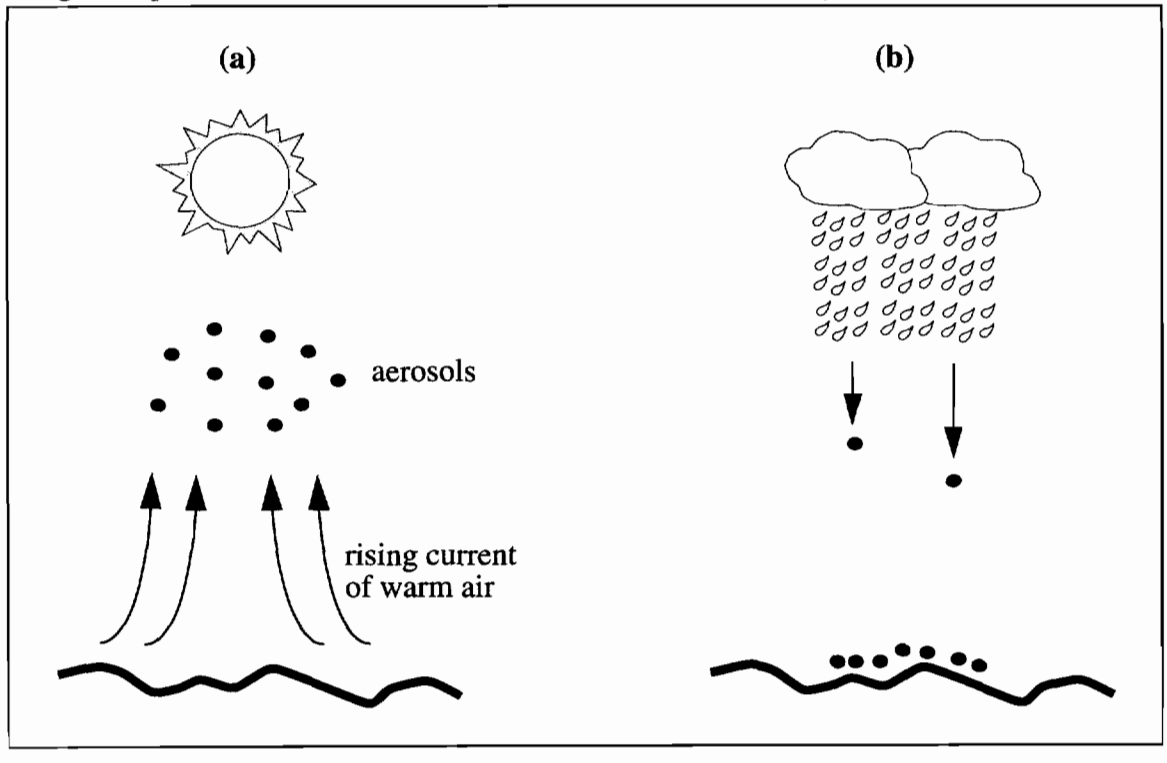


Figure 3.4 Factors effecting aerosol number concentration. (a) the sun warms the ground creating a thermal updraft which keeps aerosols suspended in the air. (b) rain washes out the aerosols and cleanses the air.

drinking water. Sometimes weather fronts can carry polluted aeroticulates for several hundred kilometers. This phenomena is common on the East Coast of the United States as exhaust from factories in the Midwest falls as acid rain. After the rain stops, Aerosols can begin to "build up" their number concentration again with each non-rainy day.

The number and size distributions of aerosols vary for each part of the world and change from day to day. It is generally impossible to give an exact mathematical model for the distributions, although scientists have tried to fit proposed curves to measured data. The best way to determine the distributions is to rely on measured data. In the field of laser remote sensing, it is important to know something about the number and size distributions because the backscattering coefficient depends on these quantities. The backscattering coefficient gives the fraction of power of the original laser beam which is backscattered by a volume of aerosol particles with a certain size and number distribution. Fortunately, most of the empirical data measured at sea level at different locations around the world show many similarities. The only real way to know the exact particle distribution is to measure it each time data is taken.

A review of the number and size distributions of many researchers is found in Junge.[28] The general shape of most continental distributions is presented by figure 3.5. From figure 3.5, the scale size of aerosols covers about 4 orders of magnitude. Most of the aerosols over continents lie within the large range, $0.1\mu m$ to $1\mu m$. Other researchers have reported size distributions showing most of the scale sizes are located around $0.1\mu m$ or slightly below. At higher humidity levels, there are more aerosols in the large range.[28]

The dynamics of aerosol motion can be described by the random walk, or Brownian motion.[29] Aerosol particles experience an agitation from colliding with mol-

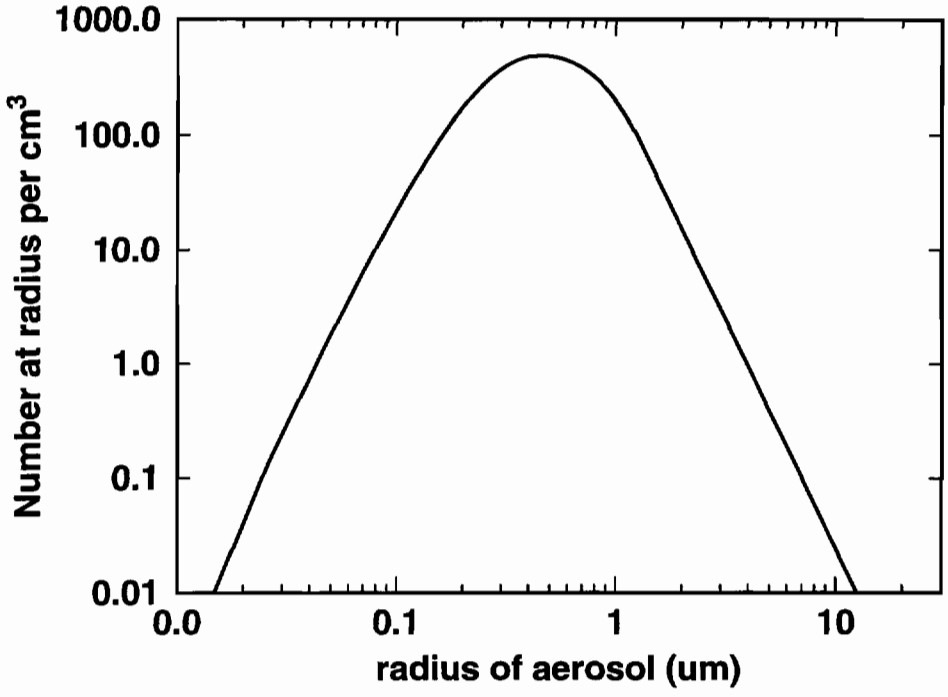


Figure 3.5 Size distribution for continental aerosols, summary adapted from Junge.[28]

ecules creating a thermal motion analogous to that of the surrounding gas molecules. Introduced by the work of Robert Brown in the early nineteenth century, Brownian motion of aerosols creates a situation where the aerosols are moving in relative, random motion due to thermal energy, even if there is no wind present. When the wind blows, the aerosols are assumed to move with the wind with an average speed equal to the wind speed with fluctuations about the mean due to the fluctuations in the wind speed as well as Brownian motion. Section 4 of this chapter examines more closely the statistics of aerosol motion.

The next section shows how the particle size distribution relative to the wavelength effects the scattering profile of a laser beam. This development leads to the volume backscattering coefficient.

3.2 Effect of Aerosol Size on Backscattering

The aerosol scale size is the most important parameter for characterizing the behavior of aerosols.[32] In fact, some properties depend very strongly on the scale size. This section shows how the scattering profile of a laser beam is determined by the particle size relative to the wavelength of the laser. This discussion leads to the volume backscattering coefficient.

There are several different types of scattering that concern laser beam propagation. Molecular, or Rayleigh scattering is the scattering of an incident electric field by atoms or molecules. The particle scale size is much less than the wavelength of the electric field. Rayleigh scattering is perfectly elastic so that there is no change of frequency of the scattered field. Another type of scattering is called Raman scattering. In this case, laser radiation is inelastically scattered causing a frequency shift of the scattered field. The most significant type of scattering is due to the type studied by G. Mie in the early 1900's. Mie scattering is concerned with the situations when the scale size of the aerosols is comparable to the wavelength of the electric field. Mie scattering is perfectly elastic resulting in no change in frequency to the scattered field. Figure 3.6 shows the relevance of each type of scattering to laser remote sensing. From this chart it can be seen that Mie scattering is the most dominant scattering mechanism effecting laser beam propagation. All other types of scattering are on the order of 7 to 23 orders of magnitude less than Mie scattering. Therefore, the other types of scattering are ignored in the analysis of laser remote sensing.

In 1908 Mie published the first analytical treatment of the scattering of an electric field by dielectric spheres, or aerosols. Reviews of Mie's work are found in Born and Wolf[2] and Van De Hulst.[33] The diagram of the scattering problem is shown in figure 3.7. The initial electric field U_T considered in the analysis is a plane wave

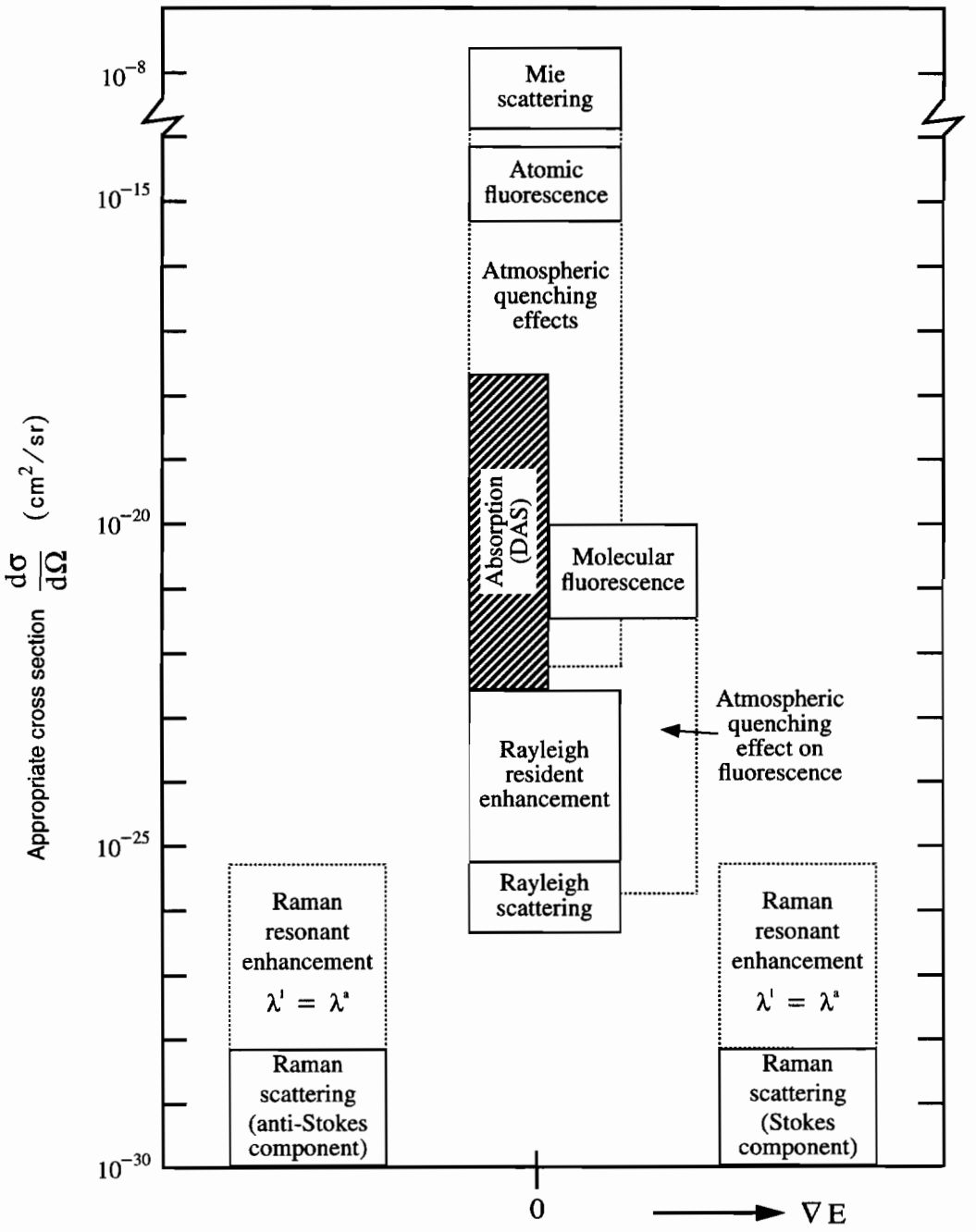


Figure 3.6 Different types of scattering and their relevance to laser remote sensing (from Measures[30]).

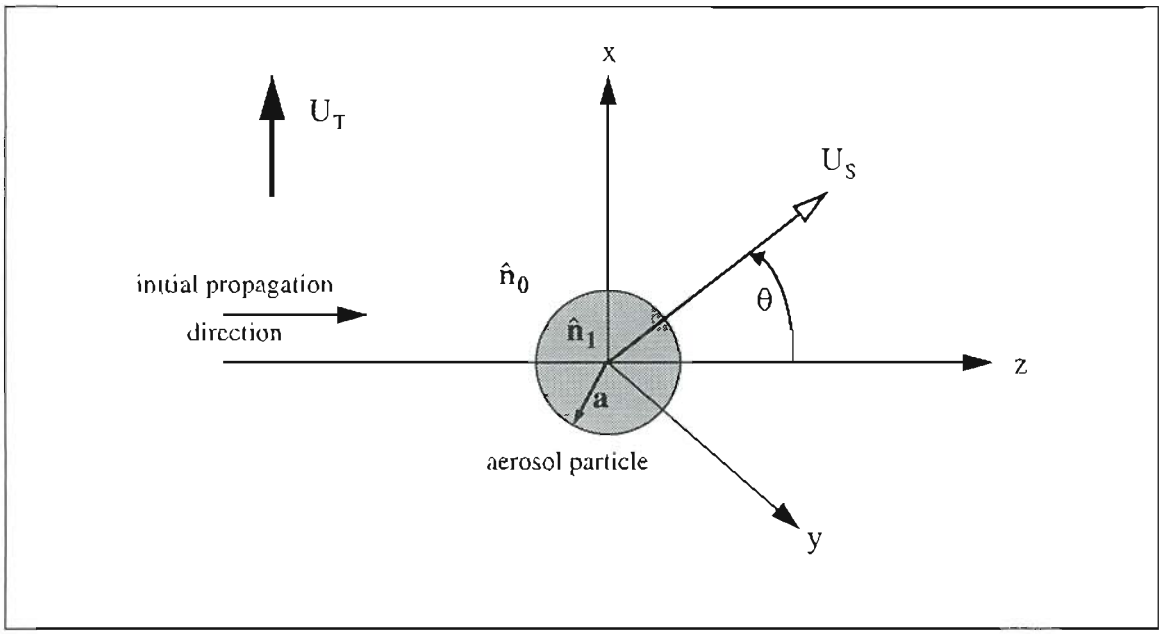


Figure 3.7 Geometry for an electric field propagating in the +z direction scattered by a perfectly spherical aerosol particle.

with electric field propagating in the +z direction, linearly polarized in the +*x* direction. A laser beam is usually Gaussian distributed, but to a tiny aerosol, a plane wave is an adequate approximation. The particle is assumed to be perfectly spherical with an index of refraction $\hat{n}_1$ different from that of the surrounding medium (air) $\hat{n}_0$. The aerosol particle has a radius equal to a. The expression for the scattered electric field U_S is obtained by solving Maxwell's equations subject to certain boundary conditions across the air-aerosol interface. The expression for U_S is complicated, being an infinite series solution of Legendre polynomials, spherical Bessel functions, and Hankel functions. Some useful insight into the effect of particle size on U_S is obtained by looking at the far-field scattered expressions for the scattered intensity. The far-field expressions are obtained by substituting the asymptotic expressions for the special functions. A computer program was written to plot the far-field intensity distribution for studying the effect of aerosol radius on the scattering profile.

Figure 3.8a-e shows the scattering diagrams of an incident laser beam scattered from aerosol particles of different scale sizes. Each figure shows the cross section of the scattering profile sliced along the plane of the initial polarization direction of the electric field. When the aerosol scale size is very small compared to the wavelength, as shown in figure 3.8a, the scattering profile is perfectly symmetric. Half of the incident radiation is scattered into both the forward and backward directions. This result is identical to the case discovered by Lord Rayleigh in the study of scattering by molecules. As the aerosol scale size increases, the scattering profile exhibits the forward dominant scattering behavior, as shown in figure 3.8b. In this figure, the aerosol scale size is about 10% of the wavelength, yet the tendency to scatter more radiation in the forward direction is already noticeable. Figure 3.8c shows the scattering profile when the aerosol scale size is 25% of the wavelength. Even more of the incident laser beam is forward scattered. Figure 3.8d shows the "whale" scattering profile when the aerosol scale size is half of the wavelength. A scale size beyond half of the wavelength is exhibited in figure 3.8e. The scattering profile becomes complicated with many angular side lobes, yet the tendency to forward-scatter the laser radiation dominates.

Scattering of a laser beam in the atmosphere is more complex than figures 3.8a-e indicate. Atmospheric scattering involves aerosols that vary in composition, size, and shape. The subsequent scattering properties, therefore, are extremely complex. Instead of studying the individual properties of each aerosol, general information about the atmosphere can be obtained by analyzing the volume backscattering coefficient. The backscattering coefficient can be expressed as

$$\beta \sim \frac{1}{\lambda^3} \int_{a_1}^{a_2} i\,(a, n, \theta = 180^\circ, \lambda)\; N\,(a)\, da \qquad (3.1)$$

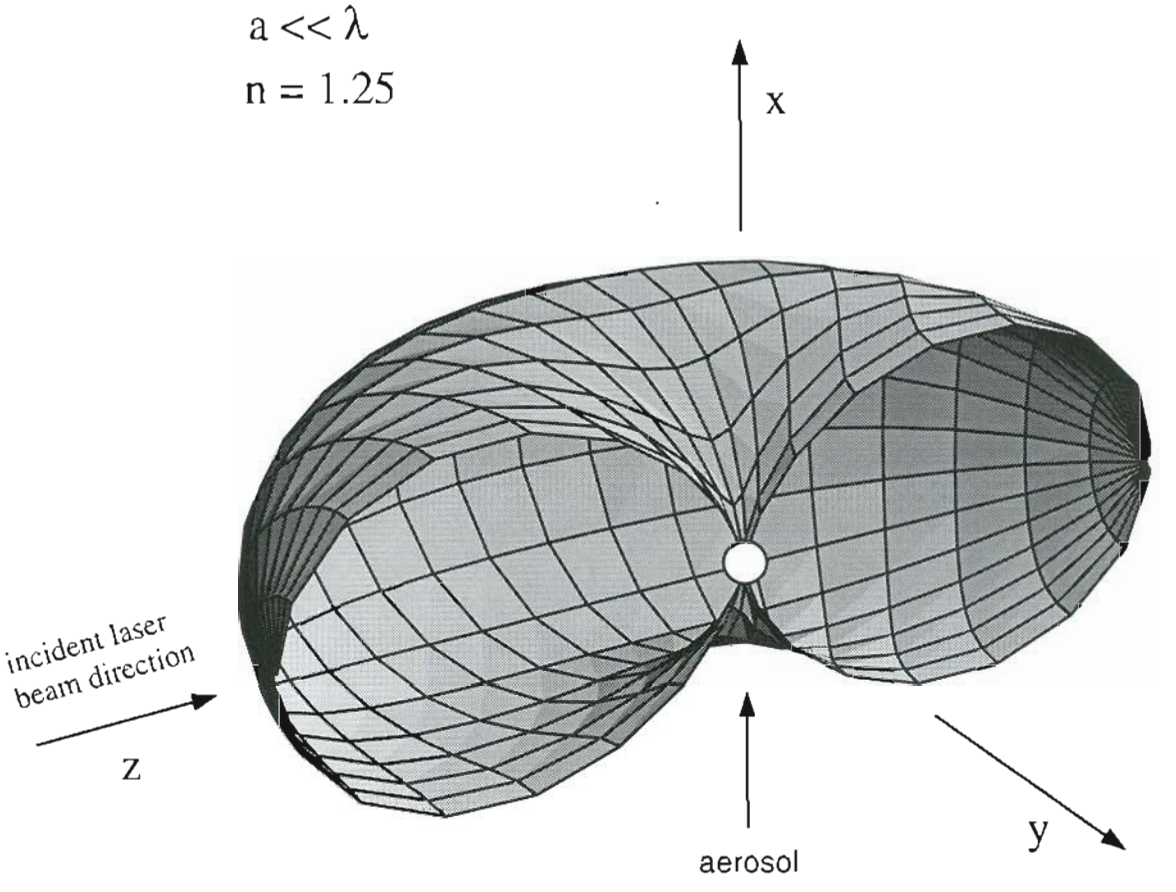


Figure 3.8a Cross section of the scattering profile of a laser beam propagating in the +z direction. Aerosol radius a<<λ (Rayleigh limit).

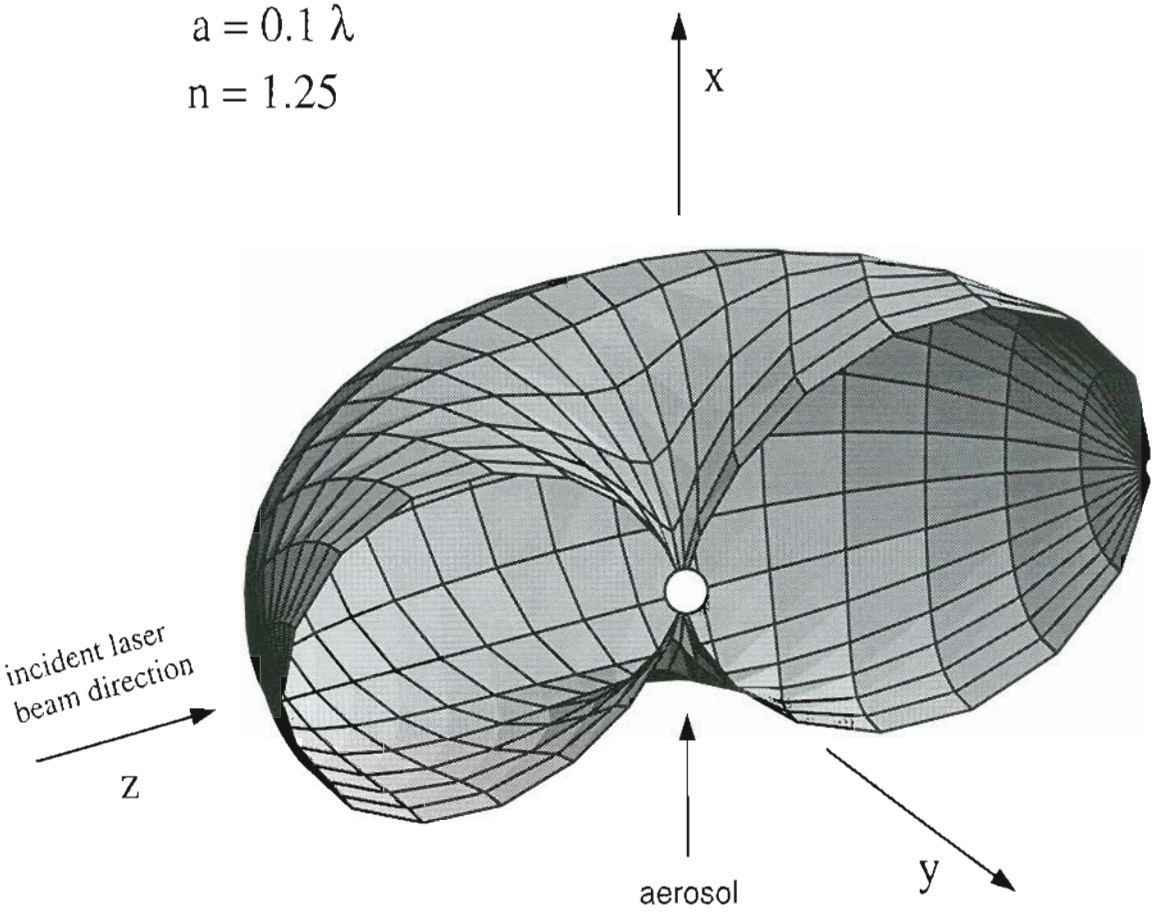


Figure 3.8b Cross section of the scattering profile of a laser beam propagating in the +z direction. Aerosol radius a = 0.1λ.

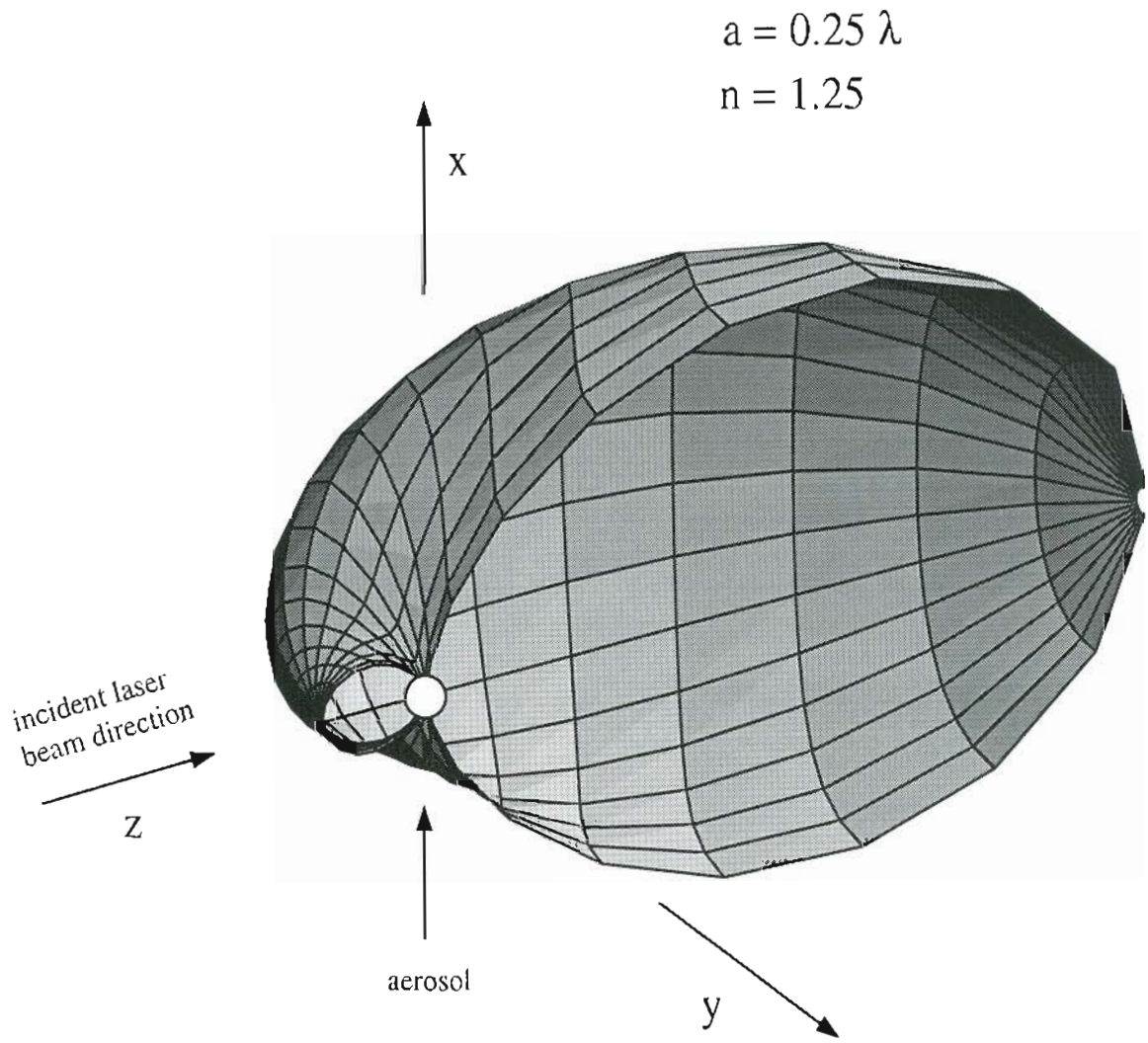


Figure 3.8c Cross section of the scattering profile of a laser beam propagating in the +z direction. Aerosol radius a = 0.25λ.

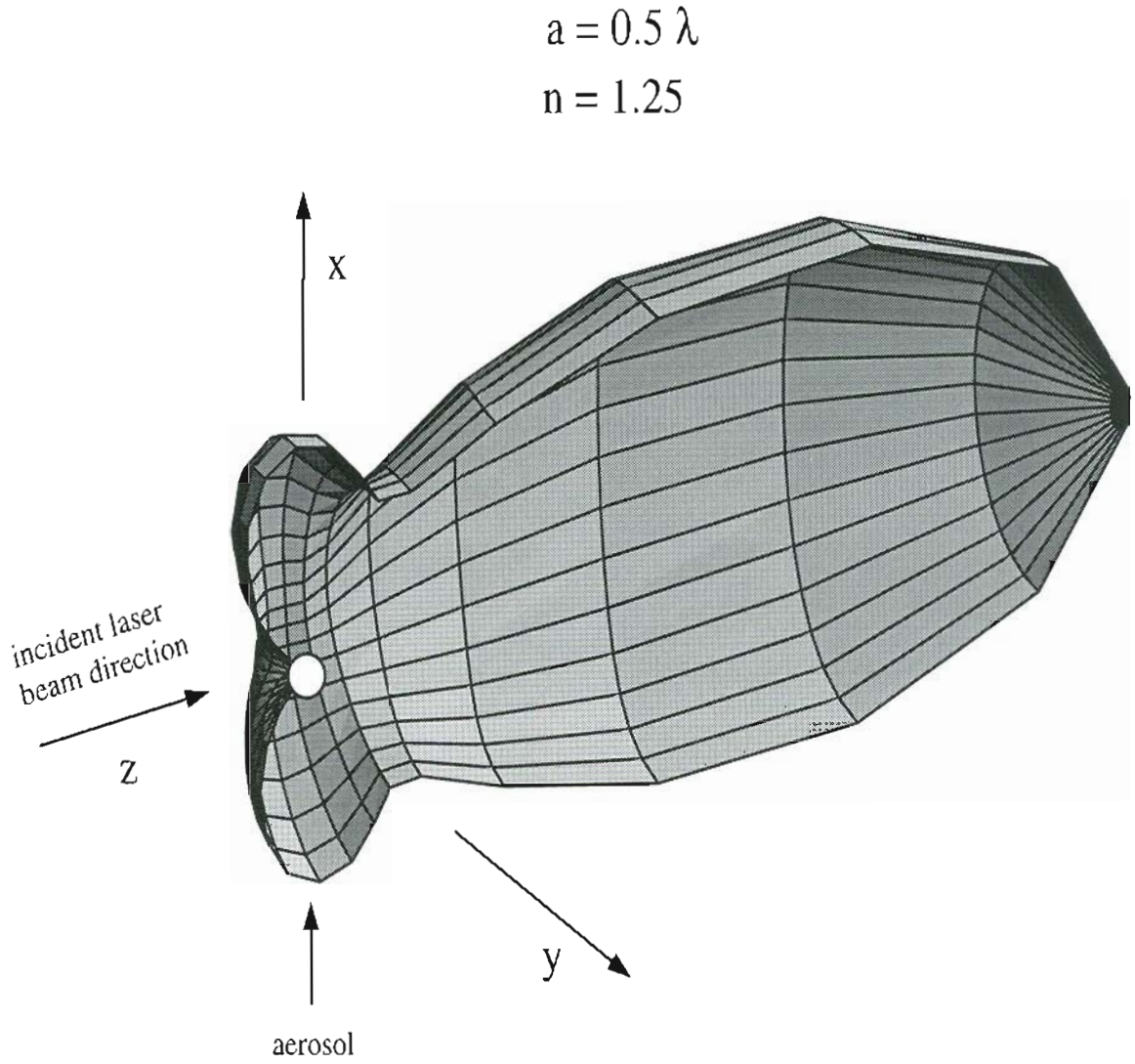


Figure 3.8d Cross section of the scattering profile of a laser beam propagating in the +z direction. Aerosols radius a = 0.5λ.

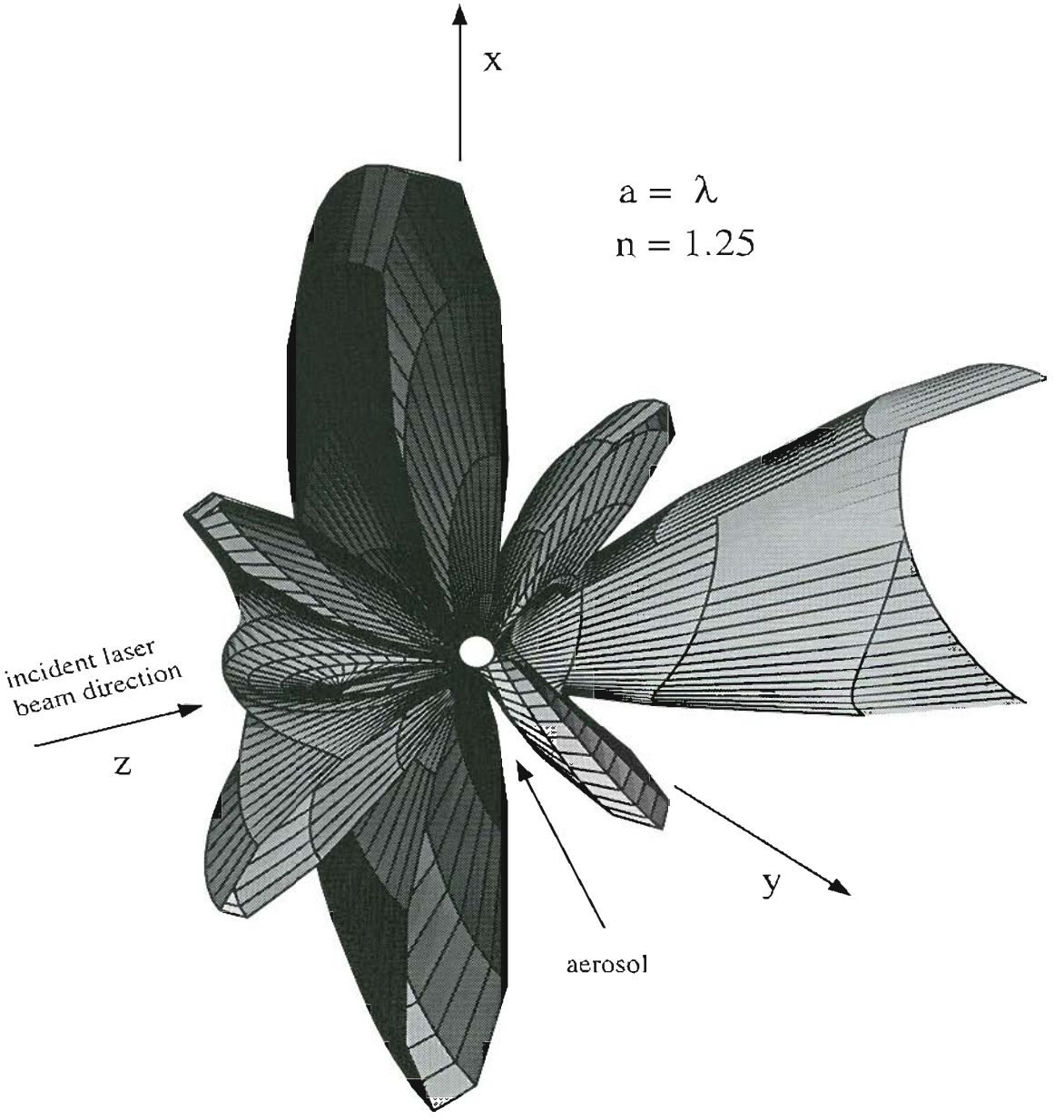


Figure 3.8e Cross section of the scattering profile of a laser beam propagating in the +z direction. Aerosols radius a = λ.

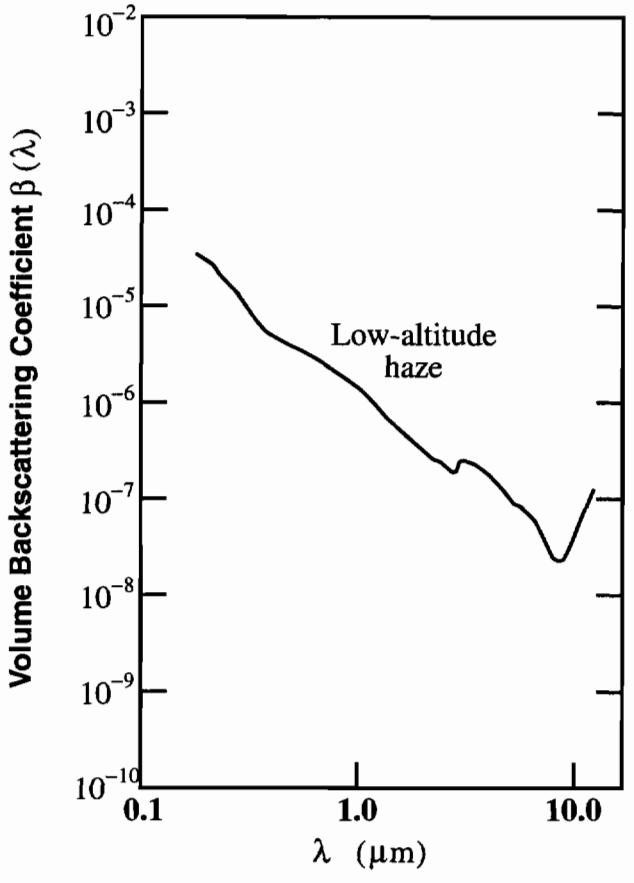


Figure 3.9 Aerosol volume backscattering coefficient as a function of wavelength in the lower atmosphere. (From Wright, et. al.[35])

where i() is the scattered intensity, n is the index of refraction of the particle, λ is the wavelength of the laser, a is the aerosol scale size, and N(a) is the number distribution of the aerosols, with $N_{total} \approx 1000 cm^{-3}$ in the lower atmosphere. The units of β are $m^{-1}sr^{-1}$. This equation predicts a 3rd power inverse dependence on wavelength.

There have been extensive measurements of β over the last 20 years at many wavelengths. Figure 3.9 shows the aerosol volume backscattering coefficient as a function of wavelength measured by Wright, et. al.[35] This figure shows the dependency of the backscattering coefficient on the laser wavelength. The backscattering

coefficient decreases several orders of magnitude as the wavelength increases from 0.1μm to 10μm. Since the rate of reduction is less than λ^{-3}, some researchers have proposed that β is inversely proportional to λ to some power between 2 and 3.

Relevant to this thesis project are the measurements of β at CO_2 wavelengths ($\lambda = 10.6\mu m$). Figures 3.10a and b present representations of the volume backscattering coefficient taken in California and Colorado at $\lambda = 10.6\mu m$. In each figure, β is plotted as a function of height. Figure 3.10a shows the lidar return above Pasadena, California, from Tratt and Menzies[36] while figure 3.10b contains the model profiles for β used by Wave Propagation Laboratories taken in Boulder, Colorado from Post, et. al.[37] The lower atmospheric measurements place β somewhere around $10^{-8} m^{-1} sr^{-1}$. Figure 3.10b shows that β depends on the season as well, indicating that measurements made in spring and summer can be 1 or 2 orders of magnitude greater than those made in the winter and fall. This is due to the factors effecting the number concentration discussed in figure 3.4. In the winter, β can drop at least an order of magnitude. In chapter 5 it is shown that the Signal-to-Noise ratio (SNR) is proportional to β. Therefore, the higher the β, the better the SNR. Figure 3.10b suggests that the highest SNR for aerosol scattering occurs in the summer months.

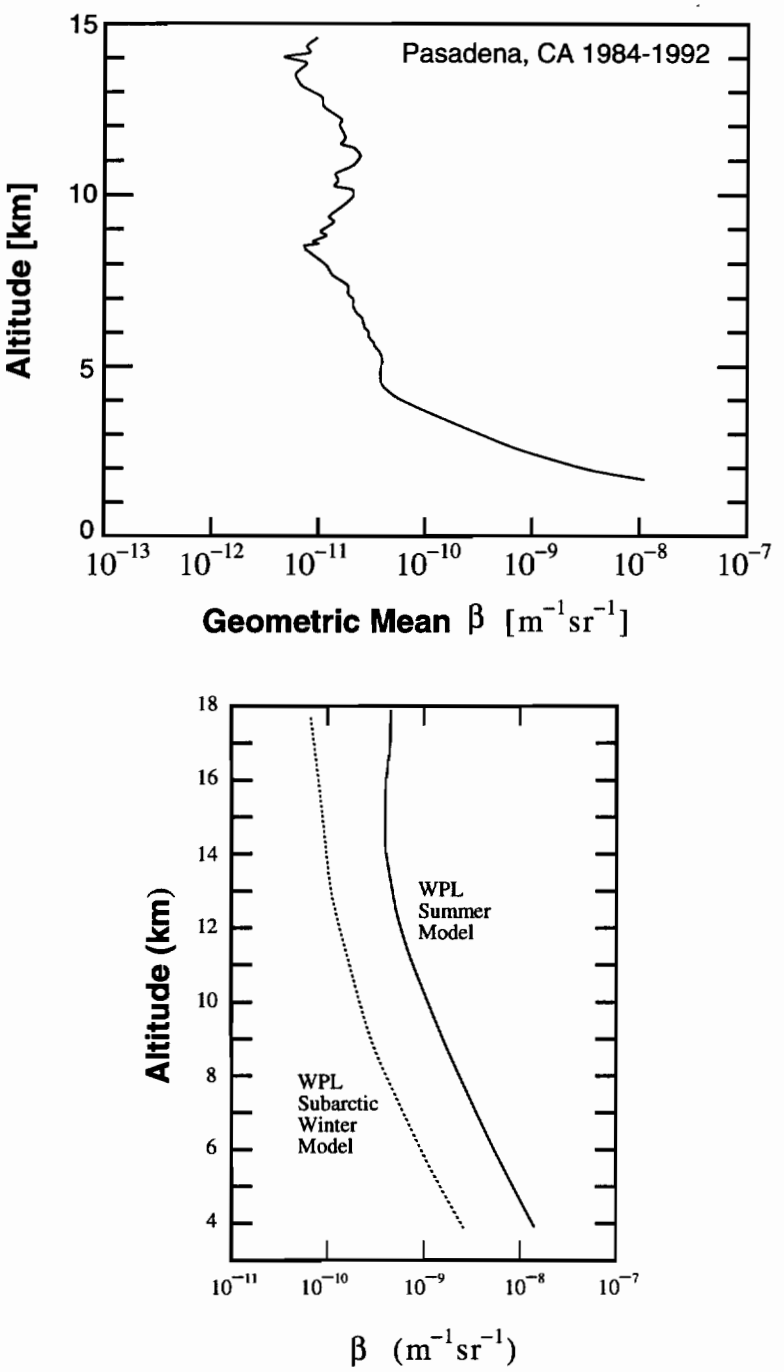


Figure 3.10 Representations of the volume backscattering coefficient. (a) Measured over Pasadena, California (from Tratt and Menzies[36]) (b) Model Profiles used by Wave Propagation Laboratories Boulder, Colorado (from Post, et. al.[37])

3.3 Modifying the Extended Huygens-Fresnel Integral to Include Aerosols

There are two main differences between backscattering from aerosols and a metallic hard target. The first is that aerosols backscatter up to 5 orders of magnitude less radiation, decreasing the SNR considerably. The second is that aerosols form a moving target which must be included in the analysis of the statistics. The schematic of the aerosol problem is shown in figure 3.11 below. A laser beam is aimed into the atmosphere and is backscattered off of aerosol particles moving with the wind. The backscattered laser signal is collected by a detector at the same location as the transmitter, forming a single ended system introduced in Chapter 2. Figure 3.12a shows the geometry of the coordinate systems. Because the aerosols move with the wind, it is necessary to include time in the analysis. Including time in the analysis introduces the Doppler shift to the laser frequency, as shown in figure 3.12b. As the vector wind blows the aerosol particles around the atmosphere, a Doppler shift results due to the component of the wind along the z axis. A Taylor expansion of the position of the

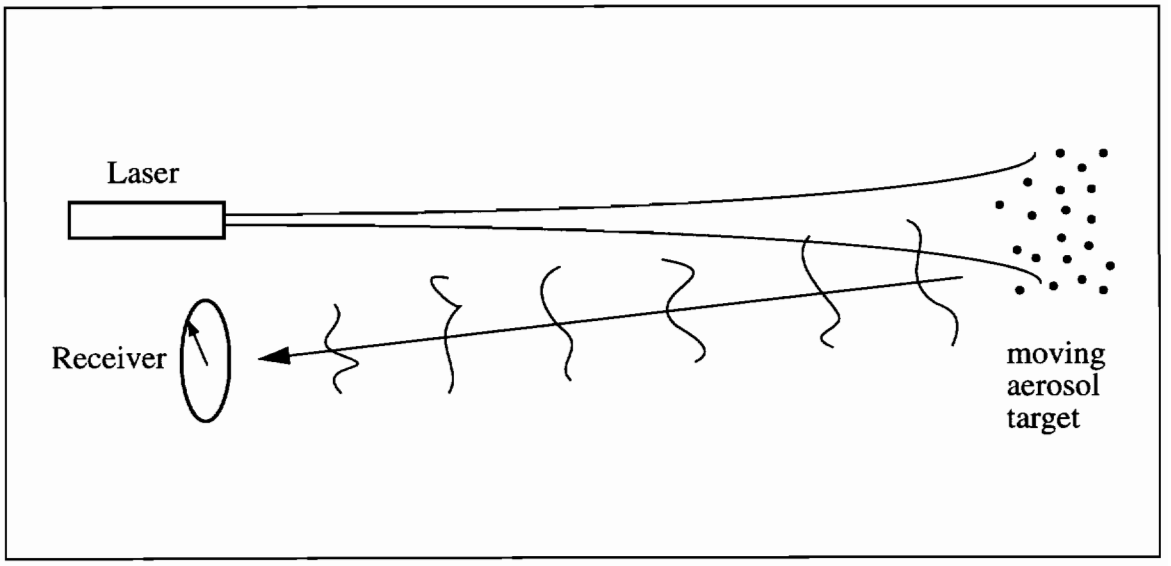


Figure 3.11 Schematic of the single-ended laser system backscattered from aerosols.

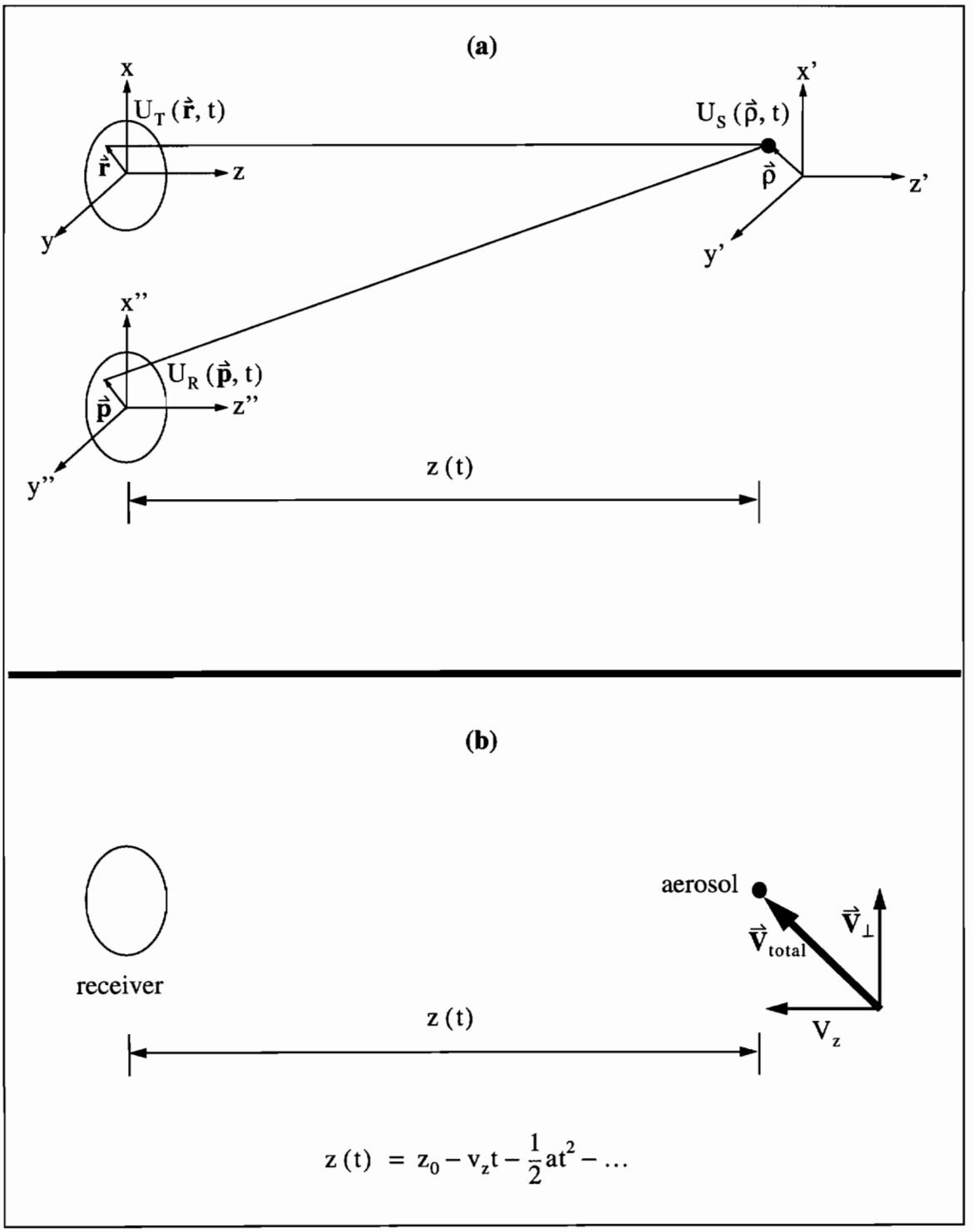


Figure 3.12 (a) Coordinate system of the aerosol backscattering target single ended lidar system. (b) Aerosol moves with the wind. A Taylor expansion of the position of the particle in the z direction results in the Doppler shift of the laser signal.

aerosol along the z axis leads to a representation of how the z coordinate is changing in time

$$z(t) = z_0 - v_z t - \frac{1}{2}at^2 - \ldots$$

$$\approx z_0 - v_z t \tag{3.2}$$

In equation (3.2) the acceleration has been dropped because it is negligible over typical sampling times. The minus sign indicates that the z distance is decreasing. A similar equation can be written for the transverse velocity of the aerosol particle

$$\vec{\rho}(t) \approx \vec{\rho}_0 - \vec{v}_\perp t \tag{3.3}$$

where $\vec{v}_\perp$ is the transverse component of the velocity of the aerosol. These equations can be used in the analysis of the electric field of a laser beam propagating through turbulence, backscattered from aerosol particles. The use of aerosols as a backscatterer allows for the detection of the z-directed wind speed as well as the crosswinds. The detection of the z-directed wind speed is not possible using a hard target, since a hard target does not move with the wind.

The problem can be separated into two steps. The first step is concerned with the z-directed winds, and the second step is concentrated on the crosswinds. The electric field at the transmitter in figure 12a can be represented as

$$u_T(\vec{r}, t) = U_T(\vec{r}) \cos(2\pi f_0 t) \tag{3.4}$$

where the monochromatic part of the beam is represented by the cosine. In equation (3.4), $U_T(\vec{r})$ is the initial electric field distribution in the transmitter plane, $f_0 = C/\lambda$

is the frequency of the laser beam, and C is the speed of light. The electric field just after scattering off of 1 aerosol particle can be written as

$$u_S(\vec{\rho}, t) = U_S(\vec{\rho}, t)\cos\left[2\pi f_0\left(t - \frac{z(t)}{C}\right)\right]. \tag{3.5}$$

Because of turbulence, the electric field amplitude u_S is also a function of time. The field at the receiver becomes

$$u_R(\vec{\mathbf{p}}, t) = U_R(\vec{\mathbf{p}}, t)\cos\left[2\pi f_0\left(t - \frac{2z(t)}{C}\right)\right]. \tag{3.6}$$

Inserting equation (3.2), the Taylor expansion for the position of the aerosol, into equation (3.6) produces

$$u_R(\vec{\mathbf{p}}, t) = U_R(\vec{\mathbf{p}}, t)\cos\left[2\pi t\left(f_0 + \frac{2v_z}{C}f_0\right) - \frac{4\pi f_0 z_0}{C}\right]. \tag{3.7}$$

Again, time is included in U_R because turbulence causes the amplitude to fluctuate in time. The Doppler shift is recognized to be

$$f_D = \frac{2v_z f_0}{C} = \frac{2v_z}{\lambda}. \tag{3.8}$$

Solving for v_z gives the z-directed wind speed for a single aerosol particle

$$v_z = \frac{f_D \lambda}{2}. \tag{3.9}$$

For a CO_2 laser, the wavelength is 10.6μm. Therefore, for a 1 MHz Doppler shift, the velocity of the aerosol is 5.3 m/s according to equation (3.9). Estimating the z-directed wind speed is as simple as determining the Doppler shift and using equation

(3.9).

In order to estimate the crosswinds, the extended Huygens-Fresnel integral must be modified to include the effect of the aerosols. The analytical expression for the crosswinds is derived in chapter 4. The best way to examine the modifications is to illustrate them with a slight change to the example in chapter 2.3. The *time delayed* mutual intensity function is now derived for the case of aerosol scattering. The geometry is the coordinate systems shown in figure 3.12a. The analysis with aerosols is based on two references, Churnside and Yura[38] and Murty.[39] The basic theoretical formulations were developed by Churnside and Yura but only for speckle and no turbulence. Murty included the effect of turbulence and developed an expression for the time-delayed mutual intensity function, but used several approximations to derive a closed form solution.

The electric field just before scattering is

$$U(\vec{\rho}, z) = \frac{ke^{ikz}}{2\pi iz}\int d^2\hat{\mathbf{r}}\, U_T(\hat{\mathbf{r}}, z')\exp\left[\frac{ik}{2z}(\vec{\rho}-\hat{\mathbf{r}})^2 + \psi_f(\vec{\rho}, \hat{\mathbf{r}}, z, z')\right] \tag{3.10}$$

where $U_T(\hat{\mathbf{r}}, z')$ is the transmitted laser field, z is the path length coordinate, and ψ_f describes the effect of the turbulence on a spherical wave propagating on the forward path from the transmitter to the aerosols. The field immediately after scattering due to a single aerosol can be written as

$$u_S(\vec{\rho}, z) = S(\vec{\rho})\, U(\vec{\rho}, z) \tag{3.11}$$

where $S(\vec{\rho})$ is called the amplitude scattering factor of an aerosol particle. This field propagates back to the detector through turbulence. At the receiver, the electric field due to single particle scattering is

$$u_R(\vec{p}) = \frac{S(\vec{\rho})}{z} U(\vec{\rho}, z) e^{ikz} \exp\left[\frac{ik}{2z}(\vec{p}-\vec{\rho})^2 + \psi_b(\vec{p}, \vec{\rho}, z, z')\right] \tag{3.12}$$

where ψ_b describes the effect of the turbulence on the backward path from the target to the receiver. There is no integration over the target area because equation (3.12) is the electric field at the receiver due to the backscattering of only one particle. The time delayed representations for the electric fields can be obtained by replacing $\vec{\rho}$ with $\vec{\rho} - \vec{v}_\perp \tau$ and z by $z - v_z \tau$ in equations (3.10) and (3.12). This means that the aerosol has moved from $\vec{\rho}$ to $\vec{\rho} - \vec{v}_\perp \tau$ and from z to $z - v_z \tau$ in time delay τ. Equations (3.10) and (3.12) become

$$U(\vec{\rho}, \tau) = \frac{k \exp[ik(z - v_z\tau)]}{2\pi i z} \times \int d^2\vec{r}\; U_T(\vec{r}) \exp\left[\frac{ik}{2z}(\vec{\rho} - \vec{v}_\perp\tau - \vec{r})^2 + \psi_f(\vec{\rho} - \vec{v}_\perp\tau, \vec{r}, \tau)\right] \tag{3.13}$$

and

$$u_R(\vec{p}, \tau) = \frac{S(\vec{\rho})}{z} U(\vec{\rho}, \tau) \exp[ik(z - v_z\tau)] \times \exp\left[\frac{ik}{2z}[\vec{p} - (\vec{\rho} - \vec{v}_\perp\tau)]^2 + \psi_b(\vec{p}, \vec{\rho} - \vec{v}_\perp\tau, \tau)\right]. \tag{3.14}$$

In these equations, $z - v_z\tau \approx z$ is approximated outside of exponentials. Inside the exponentials, $z - v_z\tau$ is kept because of phase considerations. In addition, $\vec{\rho} - \vec{v}_\perp\tau \approx \vec{\rho}$ in $S(\vec{\rho})$. Equation (3.14) is still due to single particle scattering.

To obtain the total electric field $U_R(\vec{p}, \tau)$ at the receiver, equation (3.14) must be integrated over the entire volume of aerosols. In addition, the velocities $\vec{v}_\tau$ and v_z are random variables which must be accounted for in the ensemble average over the scattering volume. In order to simplify the integrations, the particle velocities are

assumed to be Gaussian distributed. This assumption in not quite true. In fact, Kolmogorov[3] found that the mean square velocity difference depends on the distance between velocity probes to the 5/3 power within some inertial subrange of distances. However, Dutton and Hojstrup[40] show that the observed statistical structure of the velocity turbulence is nearly Gaussian. The Gaussian assumption is only used to describe the velocity distributions of the aerosols, but is not used in the spherical wave mutual coherence function. The approximate Gaussian aerosol velocity probability distributions are

$$\mathrm{pdf}(v_z) = \frac{1}{\sqrt{2\pi}\sigma_z}\exp\left[\frac{-(v_z - \langle v_z\rangle)^2}{2\sigma_z^2}\right] \tag{3.15}$$

$$\mathrm{pdf}(\vec{v}_\perp) = \frac{1}{2\pi\sigma_\perp^2}\exp\left[\frac{-(\vec{v}_\perp - \langle\vec{v}_\perp\rangle)^2}{\sigma_\perp^2}\right] \tag{3.16}$$

where $\langle v_z\rangle$ and σ_z^2 are the mean and variance of the z-directed velocity, and $\langle\vec{v}_\perp\rangle$ and $\sigma_\perp^2$ are the corresponding quantities for the transverse velocity. Typically, $\sigma_{z,\perp}$ is about 10% of the mean wind speed.[38] There is also a probability distribution for the position of each aerosol. Under the conditions of local homogeneity and isotropy,

$$\mathrm{pdf}(\vec{\rho}, z) = \frac{1}{\mathrm{Vol}} \tag{3.17}$$

which implies that the aerosols are uniformly distributed throughout the volume, Vol.

The time delayed mutual intensity is now calculated, first for single particle scattering. The time delayed mutual intensity function due to single particle scattering is given by

$$\gamma(\vec{\rho}_1, \vec{\rho}_2, \tau) = \langle u_R(\vec{\rho}_1, 0)\, u_R^*(\vec{\rho}_2, \tau)\rangle \tag{3.18}$$

$$\gamma(\vec{\rho}_1, \vec{\rho}_2, \tau) = \frac{1}{z_1 z_2} \exp\left[ikv_{z_2}\tau + \frac{ik}{2z_1}(\vec{\rho}_1 - \vec{\rho}_1)^2 + \frac{ik}{2z_2}[\vec{\rho}_2 - (\vec{\rho}_2 - \vec{v}_{\perp_2}\tau)]^2 \right]$$

$$\times < U(\vec{\rho}_1, 0)\, U^*(\vec{\rho}_2, \tau)\, S(\vec{\rho}_1)\, S^*(\vec{\rho}_2)$$

$$\times \exp[\Psi_b(\vec{\rho}_1, \vec{\rho}_1, 0) + \psi_b^*(\vec{\rho}_2, \vec{\rho}_2 - \vec{v}_{\perp_2}\tau, \tau)] > . \qquad (3.19)$$

Inserting the expression for $U(\vec{\rho}_1, 0)$ and $U^*(\vec{\rho}_2, \tau)$ into equation (3.19) leads to the following result

$$\gamma(\vec{\rho}_1, \vec{\rho}_2, \tau) =$$

$$\frac{k^2 \exp[2ik(z_1 - z_2) + 2ikv_{z_2}\tau]}{(2\pi)^2 z_1^2 z_2^2} \exp\left[\frac{ik}{2z_1}(\vec{\rho}_1 - \vec{\rho}_1)^2 - \frac{ik}{2z_2}[\vec{\rho}_2 - (\vec{\rho}_2 - \vec{v}_{\perp_2}\tau)]^2 \right]$$

$$\times \iint d^2\vec{r}_1 d^2\vec{r}_2 \; U_T(\vec{r}_1)\, U_T^*(\vec{r}_2) \exp\left[\frac{ik}{2z_1}(\vec{\rho}_1 - \vec{r}_1)^2 - \frac{ik}{2z_2}[\vec{\rho}_2 - \vec{v}_{\perp_2}\tau - \vec{r}_2]^2 \right]$$

$$\times < S(\vec{\rho}_1)\, S^*(\vec{\rho}_2)$$

$$\times \exp[\psi_f(\vec{\rho}_1, \vec{r}_1) + \psi_f^*(\vec{\rho}_2 - \vec{v}_{\perp_2}\tau, \vec{r}_2, \tau) + \Psi_b(\vec{\rho}_1, \vec{\rho}_1, 0) + \psi_b^*(\vec{\rho}_2, \vec{\rho}_2 - \vec{v}_{\perp_2}\tau, \tau)] >$$

(3.20)

Since the aerosol amplitude scattering factor and the turbulence are statistically independent, the term in angle brackets in equation (3.19) can be rewritten as

$$< S(\vec{\rho}_1)\, S^*(\vec{\rho}_2) >$$

$$\times < \exp[\psi_f(\vec{\rho}_1, \vec{r}_1) + \psi_f^*(\vec{\rho}_2 - \vec{v}_{\perp_2}\tau, \vec{r}_2, \tau) + \Psi_b(\vec{\rho}_1, \vec{\rho}_1, 0) + \psi_b^*(\vec{\rho}_2, \vec{\rho}_2 - \vec{v}_{\perp_2}\tau, \tau)] >$$

(3.21)

The aerosols are located randomly throughout the scattering volume with a separation of more than a wavelength of the laser. Therefore, the scattered electric field just after scattering is Gaussian distributed and δ-function correlated. The aerosols form a perfectly diffuse scattering medium just like the rough hard target introduced in chapter 2. Using these facts, the ensemble average of the aerosol amplitude scattering factors can be written as

$$<S(\vec{\rho}_1)\,S^*(\vec{\rho}_2)> = S^2 N \delta(\vec{\rho}_1 - \vec{\rho}_2)\,\delta(z_1 - z_2)\,\delta(\vec{v}_{\perp_1} - \vec{v}_{\perp_2})\,\delta(v_{z_1} - v_{z_2}) \qquad (3.22)$$

where N is the number of particles in the volume. The second factor in angle brackets in equation (3.17) is $H(\vec{\rho}_1, \vec{\rho}_2, \vec{\rho}_1, \vec{\rho}_2, \vec{r}_1, \vec{r}_2, \vec{v}_{\perp_2}, \tau)$, the fourth dimensional spherical wave mutual coherence function.[24]

The total time delayed mutual intensity function is obtained by integrating equation (3.16) over the scattering volume and velocity statistics of the aerosols

$$\Gamma(\vec{\rho}_1, \vec{\rho}_2, \tau) = \iint d^2\vec{\rho}_1 d^2\vec{\rho}_2 \iint dz_1 dz_2 \iint d^2\vec{v}_{\perp_1} d^2\vec{v}_{\perp_2} pdf(\vec{v}_{\perp_1})\,pdf(\vec{v}_{\perp_2})$$

$$\times \iint dv_{z_1} dv_{z_2} pdf(v_{z_1})\,pdf(v_{z_2})\,\frac{\gamma(\vec{\rho}_1, \vec{\rho}_2, \tau)}{(Vol)^2}. \qquad (3.23)$$

Using the δ-functions of equation (3.18) causes six of the integrations in equation (3.19) to be trivial. For example,

$$\iint d^2\vec{\rho}_2 dz_2 = Vol$$

and

$$\int d^2\vec{v}_{\perp_2} pdf(\vec{v}_{\perp_2}) = \int dv_{z_2} pdf(v_{z_2}) = 1.$$

Using these simplifications, equation (3.19) reduces to

$$\Gamma(\vec{\rho}_1, \vec{\rho}_2, \tau) = \frac{k^2 S^2 N}{Vol} \iint d^2\vec{\rho}_1 dz_1 \int d^2\vec{v}_{\perp_1} pdf(\vec{v}_{\perp_1}) \int dv_{z_1} pdf(v_{z_1}) \iint d^2\vec{r}_1 d^2\vec{r}_2$$

$$\times \frac{U_T(\vec{r}_1) U_T^*(\vec{r}_2)}{z_1^4} \exp\left[2ikv_{z_1}\tau + \frac{ik}{2z_1}(\vec{\rho}_1 - \vec{\rho}_1)^2 - \frac{ik}{2z_2}[\vec{\rho}_2 - (\vec{\rho}_1 - \vec{v}_{\perp_1}\tau)]^2 \right]$$

$$\times \exp\left[\frac{ik}{2z_1}(\vec{\rho}_1 - \vec{r}_1)^2 - \frac{ik}{2z_1}(\vec{\rho}_1 - \vec{v}_{\perp_1}\tau - \vec{r}_2)^2 \right] H(\vec{\rho}_1, \vec{\rho}_2, \vec{\rho}_1, \vec{r}_1, \vec{r}_2, \vec{v}_{\perp_1}, \tau)$$

(3.24)

Recognizing that the backscattering coefficient, $\beta = S^2N/Vol$, equation (3.20) becomes

$$\Gamma(\vec{\rho}_1, \vec{\rho}_2, \tau) = k^2\beta \iint d^2\vec{\rho}_1 dz_1 \int d^2\vec{v}_{\perp_1} pdf(\vec{v}_{\perp_1}) \int dv_{z_1} pdf(v_{z_1}) \iint d^2\vec{r}_1 d^2\vec{r}_2$$

$$\times \frac{U_T(\vec{r}_1) U_T^*(\vec{r}_2)}{z_1^4} \exp\left[2ikv_{z_1}\tau + \frac{ik}{2z_1}(\vec{\rho}_1 - \vec{\rho}_1)^2 - \frac{ik}{2z_2}[\vec{\rho}_2 - (\vec{\rho}_1 - \vec{v}_{\perp_1}\tau)]^2 \right]$$

$$\times \exp\left[\frac{ik}{2z_1}(\vec{\rho}_1 - \vec{r}_1)^2 - \frac{ik}{2z_1}(\vec{\rho}_1 - \vec{v}_{\perp_1}\tau - \vec{r}_2)^2 \right] H(\vec{\rho}_1, \vec{\rho}_2, \vec{\rho}_1, \vec{r}_1, \vec{r}_2, \vec{v}_{\perp_1}, \tau)$$

(3.25)

The remaining steps are to substitute the expressions for the transmitted electric fields, the spherical wave mutual coherence function, and the pdf's of the aerosol velocities and to perform the integrations. Setting $\tau = 0$ and $p_1 = p_2$ gives the expression for the average received power for a point detector.

3.4 Phase Considerations, Decorrelation Effect

This section discusses the decorrelation effects of aerosols on the backscattered laser signals. The use of aerosols as the backscatterer creates some additional concerns not present with the hard target systems. One of those concerns is the decorrelation time of the backscattered laser signal due to speckle. In contrast to hard targets, aerosols move randomly about the atmosphere blown by the wind. The approximate Gaussian statistics of their motion predict that the aerosols move with a mean wind speed, but have fluctuations about the mean value. These fluctuations indicate that aerosols move relative to one another. It is speculated that this relative movement causes a decorrelation effect that reduces the coherence time. If the amplitude and intensity of the returned laser signals are obtained from the peak frequency of a FFT of the heterodyne signal, then the decorrelation effect reduces the length of time which can be used to compute the FFT. However, this section demonstrates that the decorrelation time applies only to the intensity, and the only factor which limits the length of time used to calculate a FFT is the coherence time of the laser used in this thesis project.

To demonstrate the decorrelation of the intensity, the integral over v_{z_1}, the axial aerosol velocity, in equation (3.21) of the previous section is performed:

$$\int dv_{z_1} pdf(v_{z_1}) \exp[2ikv_{z_1}\tau] =$$

$$\int_0^\infty v_{z_1} dv_{z_1} \exp[2ikv_{z_1}\tau] \frac{1}{\sqrt{2\pi}\sigma_{z_1}} \exp\left[\frac{-(v_{z_1} - \langle v_{z_1} \rangle)^2}{2\sigma_{z_1}^2}\right].$$

From Fourier transform theory, this integral is equal to[41]

$$\exp\left[i\tau k\tau<v_z>\right]\exp\left[-2\sigma_z^2(k\tau)^2\right]$$

Rewriting equation (3.21),

$$\Gamma(\vec{\rho}_1,\vec{\rho}_2,\tau) = \exp\left[i\tau k\tau<v_z>\right]\exp\left[-2\sigma_z^2(k\tau)^2\right]\ F(\vec{\rho},\vec{r},\vec{v}_\perp,\vec{p},\tau)\,. \tag{3.26}$$

where

$$F(\vec{\rho},\vec{r},\vec{v}_\perp,\vec{p},\tau) = k\beta^2\iint d^2\vec{\rho}_1 dz_1 \int d^2\vec{v}_{\perp_1} pdf(\vec{v}_{\perp_1}) \iint d^2\vec{r}_1 d^2\vec{r}_2$$

$$\times\frac{U_T(\vec{r}_1)\,U_T^*(\vec{r}_2)}{z_1^4}\exp\left[\frac{ik}{2z_1}(\vec{p}_1-\vec{\rho}_1)^2-\frac{ik}{2z_2}\left[\vec{p}_2-(\vec{\rho}_1-\vec{v}_{\perp_1}\tau)\right]^2\right]$$

$$\times\exp\left[\frac{ik}{2z_1}(\vec{\rho}_1-\vec{r}_1)^2-\frac{ik}{2z_1}(\vec{\rho}_1-\vec{v}_{\perp_1}\tau-\vec{r}_2)^2\right]H(\vec{p}_1,\vec{p}_2,\vec{\rho}_1,\vec{r}_1,\vec{r}_2,\vec{v}_{\perp_1},\tau) \tag{3.27}$$

The second exponential factor in equation (3.26) describes the intensity decorrelation. The e^{-2} point occurs when

$$\tau = \frac{1}{\sigma_z k}. \tag{3.28}$$

For a CO_2 laser, $k = 5.93\times10^5 m^{-1}$ and for a 2m/s wind speed, $\sigma_z^2 = 0.04$. The decorrelation time from equation (3.23) is $\tau \approx 8\mu s$. This rapid decorrelation of the intensity is due to the relative, random z-directed motion of aerosols along the propagation path.

Intensity decorrelation times in the microsecond range have been experimentally measured by several researchers. For example, Hardesty et. al.,[42] using a cw CO_2 laser focussed at 30, 100, and 500 meters, measured a decorrelation time of $2\mu s$. Vaugn et. al.,[43] using a cw CO_2 laser found the decorrelation time to be around

10μs. Schotland et. al.[44] reported a value on the order of tens of microseconds. Ancellet and Menzies,[45] using a pulsed TEA CO_2 laser, measured a decorrelation time between 1.5 and 5 microseconds. The common misconception, however, was that the decorrelation time of the intensity determined the maximum temporal window in which to compute a FFT of the return signal.

It is shown in this section that the maximum FFT window is determined by the decorrelation time of the specific laser used in this thesis project, and is not limited by the aerosol decorrelation or the effects of turbulence. If the coherence time of the laser were longer, eventually the effects of turbulence would start to limit the decorrelation time. The electric field at the detector backscattered from 1 aerosol particle is derived in the previous section, equation (3.7),

$$u_R(\vec{p}, t) = U_R(\vec{p}, t)\cos\left[2\pi t(f_0 + f_D) - \frac{4\pi f_0 z_0}{C}\right] \tag{3.29}$$

where f_D represents the Doppler shift imparted by the z component of the aerosol velocity. Equation (3.24) would be correct if the laser had an infinite coherence time and there were no turbulence. Since all lasers cannot lase on one single frequency line but have a finite bandwidth, the different frequencies present in a laser beam can eventually get out of phase with each other. The time required for two different frequencies to get out of phase by a full cycle is called the coherence time. The corresponding distance travelled is the coherence length.[46] The coherence time due to the finite bandwidth of the laser can be represented by a phase angle, $\phi_L(t)$, within the cosine in equation (3.29). An additional term within the phase of the cosine can be introduced which represents the effects of turbulence. In chapter 2 the effects of turbulence on the intensity are discussed. However, there is an additional effect on the phase. As the laser propagates through turbulence, the turbulent eddies cause the laser

wavefront to interfere with itself. This diffraction not only causes random variations in the laser beam intensity, but also disrupts the laser phase as well. Therefore, a term representing the turbulence effects on the phase, $\phi_T(t)$, must also be included within the cosine. Introducing the laser and turbulence phase functions, and rewriting the terms of equation (3.24) results in

$$u_R(\vec{\rho}, t) = U_R(\vec{\rho}, t) \cos\left[2\pi(f_0 + f_D)\left(t - \frac{2z_0}{C}\right) + \frac{4\pi f_D z_0}{C} + \phi_L(t) + \phi_T(t)\right]. \quad (3.30)$$

This equation is the expression for the returned signal backscattered from 1 aerosol. The total electric field is the sum of the contributions from N aerosols within the scattering volume

$$U_T(\vec{\rho}, t) = \sum_{i=1}^{N} u_{R_i}(\vec{\rho}, t) \cos\left[2\pi(f_0 + f_{D_i})\left(t - \frac{2z_i}{C}\right) + \frac{4\pi f_{D_i} z_i}{C} + \phi_L(t) + \phi_T(t)\right] \quad (3.31)$$

The first term in the argument of the cosine considers the round trip delay time, and the Doppler shift imparted by each aerosol. Since the z-directed velocities are approximately Gaussian with a standard deviation of about 10% of the mean velocity, the Doppler frequencies f_{D_i} are Gaussian distributed with a 10% standard deviation. The second term in the argument of the cosine gives a random phase shift to each contribution due to the random distribution of aerosols throughout the atmosphere. This term creates the speckle field which propagates back to the receiver through turbulence because the scatterers are separated by more than a wavelength of the laser beam. Representing the speckle effect by $\phi_S(t)$, equation (3.26) can be rewritten as

$$U_T(\vec{p}, t) = \sum_{i=1}^{N} u_{R_i}(\vec{p}, t) \cos\left[2\pi (f_0 + f_{D_i})\left(t - \frac{2z_i}{C}\right) + \phi_S(t) + \phi_L(t) + \phi_T(t)\right]. \quad (3.32)$$

If the aerosols do not move, the speckle field generated by aerosol scattering would be identical to the hard target case. However, since the aerosols are in constant random motion, and move relative to one another, the speckle field decorrelates. But equation (3.22) already takes into account the relative motion of the aerosols which leads to the decorrelation of the *intensity*. In order to determine the maximum FFT time, or temporal coherence, the decorrelation of the *phase* must be examined.

The only way the temporal coherence will be reduced is if one of the ϕ terms in equation (3.27) has a large variance *and* fluctuates rapidly. If, for example, one ϕ term has a large variance, but is slowly varying, the phase will be nearly constant for a long time, allowing for a long FFT window. On the other hand, if the ϕ term is rapidly varying but has a small variance, the effect on the phase is negligible because the phase is approximately constant, with very small fluctuations.

The statistics of the laser phase has been studied by several researchers. Goodman[26] was the first to analytically study the statistics of the intensity and phase of a laser speckle pattern. Later, Uozumi and Asakura[47] generalized Goodman's results of the first order probability density function of the laser speckle phase. Later, Draper, Holmes, and Peacock[48] derived the phase distribution model for both speckle and turbulence for the unwrapped phase, the phase values beyond the principle values of 0 to 2π. Their work was done for a diffuse metallic scattering surface, but can be adapted to aerosols. As introduced in chapter 2, turbulence causes the beam to wander over a diffuse hard target or a volume of aerosol scatterers. Therefore, the laser beam constantly sees a different portion of the scattering surface. This meandering of the beam creates a constantly changing (evolving) speckle pattern, each with its own phase. Since the speckle patterns produced are uncorrelated with each other, the phases are

uncorrelated as well. Consequently, the FFT window might be limited to the length of time it takes to generate a new speckle pattern. Examining the statistics and the spectrum of the phase fluctuations of the backscattered laser signal reveals why the FFT window is limited only by the coherence time of the laser used in this project.

Draper, et. al., shows that the probability density function of the phase is Gaussian, and the phase perturbations due to speckle and turbulence are additive and Gaussian. Therefore, the phase variance due to speckle and turbulence can be studied separately. The equivalent optical phase variance due to speckle *per aerosol*, σ_a^2, can be defined in terms of the wavelength of the laser and the z-directed movement of each aerosol[49]

$$\sigma_a^2 = \left(\frac{4\pi}{\lambda}\right)^2 (v_z\tau)^2. \tag{3.33}$$

This equation means that an aerosol particle which moves a distance $v_z\tau$ creates an optical phase variance *for that one particle* given by equation (3.28). A new expression for σ_a^2 can be derived by using the value of τ which causes the intensity to decorrelate to the e^{-2} point specified by equation (3.23). Substituting this expression for τ into equation (3.28) gives

$$\begin{aligned}\sigma_a^2 &= \left(\frac{4\pi}{\lambda}\right)^2 \left(\frac{v_z}{\sigma_z k}\right)^2 \\ &= \left(\frac{2v_z}{\sigma_z}\right)^2 .\end{aligned} \tag{3.34}$$

Equation (3.29) gives the optical phase variance from speckle due to one aerosol parti-

cle. The speckle variance due to N particles located within the scattering volume is given by

$$\sigma_s^2 = \frac{\sigma_a^2}{N}. \tag{3.35}$$

As the number of contributors increases, the variance of the phase due to speckle decreases. Inserting equation (3.34) into (3.35) gives the speckle phase variance due to aerosol scattering

$$\sigma_s^2 = \frac{1}{N}\left(\frac{2v_z}{\sigma_z}\right)^2 \tag{3.36}$$

indicating a result independent of wavelength.

Under certain atmospheric conditions, the optical phase variance due to turbulence reduces to[48]

$$\sigma_T^2 = 0.079\pi^2 k^2 L C_n^2 L_0^{5/3} \tag{3.37}$$

where L is the path length, C_n^2 is the strength of turbulence, and L_0 is the outer scale of turbulence described in Chapter 2. For equation (3.37) to be valid, L_0 must be many orders of magnitude larger than l_0, the inner scale of turbulence, and $L\lambda$ needs to be small compared to L_0^2. For a $C0_2$ laser, a path length on the order of 1 km, and typical inner and outer turbulence scale sizes of 1 m and 1 mm, respectively, the requirements are met and equation (3.32) can be used to determine the optical phase variance due to turbulence. For typical numbers ($\sigma_z \approx 0.1 v_z$, $N \approx 10^9 m^{-3} \times 1 m^3$ scattering volume, L = 500 meters, $\lambda = 10.6 \mu m$, and $C_n^2 = 10^{-14}$),

$$\sigma_s^2 \approx 4 \times 10^{-7}$$

and

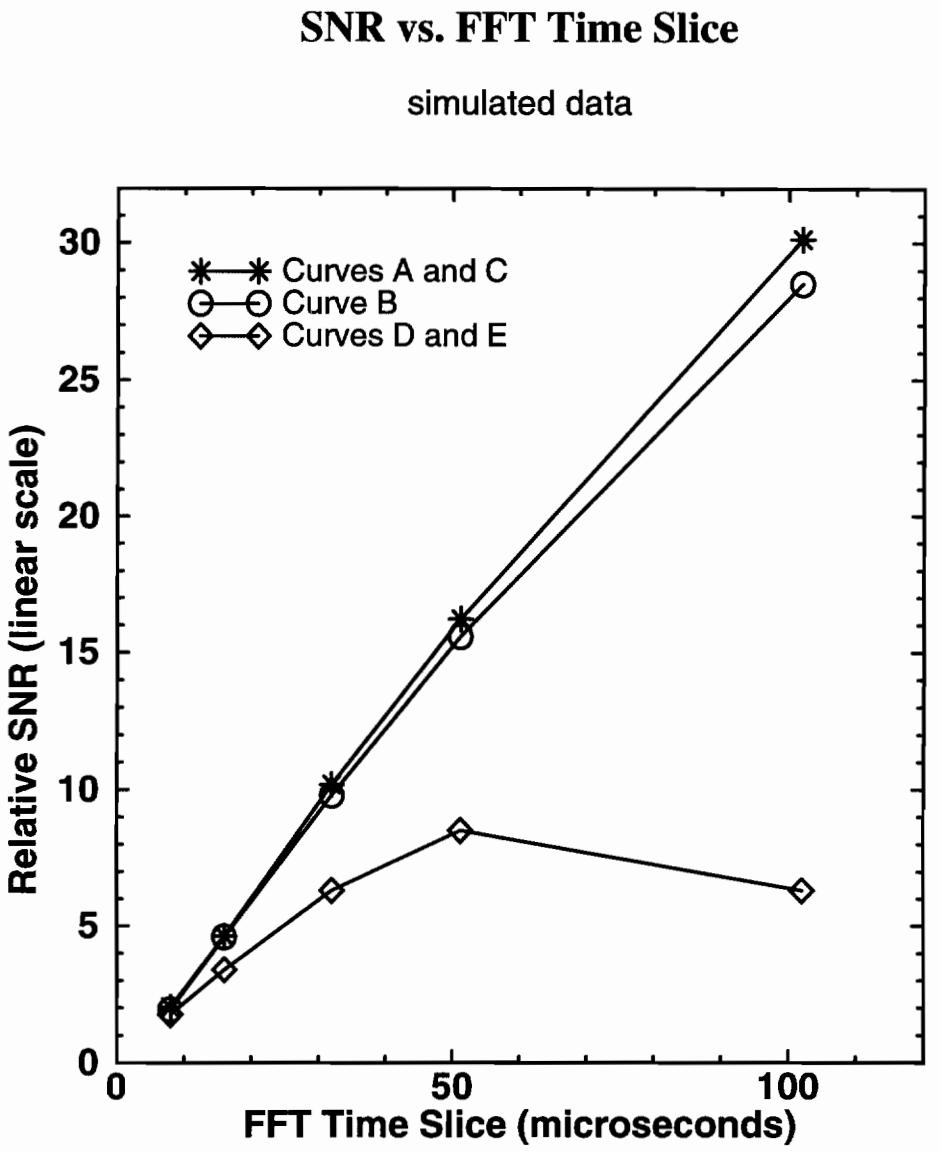
$$\sigma_T^2 \approx 1.5$$

indicating that turbulence dominates the phase variance. The temporal power spectrum of the phase fluctuations due to turbulence is reviewed by Lawrence and Strohbehn,[50] who report the spectrum of phase fluctuations to be around 15 Hz. Although the phase variance due to turbulence is much higher than that due to speckle, turbulence does not cause the phase to change very rapidly.

Not only do each of the ϕ terms in equation (3.27) have their own variance, they also fluctuate with their own frequency spectrum. The frequency spectrum of the laser has already been discussed, having a coherence time of about 50 μs. Even if the variance of the optical phase due to the laser decorrelation is large, since $\phi_L(t)$ remains nearly constant over a time slice of 50 microseconds, a 50-microsecond FFT time can be used. Therefore, the only way that one of the ϕ terms can limit the FFT time is to have a large variance *and* fluctuate rapidly. The phase variance due to speckle fluctuates rapidly, changing value in about 10 microseconds, but its variance is so small, the effect on the total optical phase is negligible. The phase variance due to turbulence has a much higher variance, but fluctuates so slowly, the effect on the total optical phase is again negligible. Consequently, the only ϕ term limiting the FFT time is the decorrelation term of the laser used in this project.

Some insight into the way each random phase term effects the processing can be obtained by writing a computer program which computes the signal to noise ratio of a sine wave

$$A\cos\left[2\pi f_0 t + \phi_S(t) + \phi_L(t) + \phi_T(t)\right]$$



Curve A: $A\cos[2\pi ft] + n(t)$

Curve B: $A\cos[2\pi ft + \phi_S(t)] + n(t)$

Curve C: $A\cos[2\pi ft + \phi_T(t)] + n(t)$

Curve D: $A\cos[2\pi ft + \phi_L(t)] + n(t)$

Curve E: $A\cos[2\pi ft + \phi_S(t) + \phi_T(t) + \phi_L(t)] + n(t)$

Figure 3.13. Processed, simulated data to investigate the effect of each random phase term on the SNR.

buried in white Gaussian noise, n(t), for different FFT time slices. The speckle and turbulence phase terms are modeled by Gaussian random processes with the respective variances computed above. The speckle phase term changes every ten microseconds (rapidly varying) while the turbulence phase term changes every 500 microseconds (slowly varying). The phase term due to the laser decorrelation is modeled by a Gaussian random process with a variance of 3 (fairly large) and changes every 50 microseconds to match the current decorrelation time of the CO_2 laser used in this experiment.

Figure 3.13 shows the result of the processing. The same number of data points (100K) were used for each FFT time slice for accurate comparison. The effective sampling rate is 20 MHz and f_0 is 6 MHz. Therefore, a 1024-point FFT represents 51.2 microseconds of time and each consecutive data point occurs every 0.05 microseconds. These parameters are chosen to match the actual experimental values. Curve A in the graph confirms that the SNR increases linearly with an increasing FFT time slice when there are no random phase terms in the argument of the cosine. Curve B shows the SNR vs. time slice when only the speckle phase term, $\phi_S(t)$, is present. Although the curve starts to deviate from curve A, the SNR still increases with each time slice. Curve C is generated by processing data with only the turbulence phase term, $\phi_T(t)$, in the argument of the cosine, implying that the turbulence phase term has no detrimental effect on the processing. Curve D is processed with only the laser phase term, $\phi_L(t)$, included within the argument of the cosine, indicating that the laser decorrelation phase term has the most detrimental effect on the processed SNR. In fact, curve E, which includes all three random phase terms, is no different than curve D, showing that the temporal coherence is limited only by the laser decorrelation time. Because the laser decorrelation phase term changes every 50 microseconds and has a high variance, this term dominates the effect on the SNR.

The turbulence phase term also has a high variance, but it varies so slowly, it has no effect. Meanwhile, the speckle phase term is rapidly varying, but because there are so many aerosols contributing to the backscattered signal, the variance is too small to effect the SNR. In chapter 5, the SNR vs. FFT time slice is presented of actual data (see figure 5.18) which resembles curve E, indicating that the only random phase term that effects the processing is the decorrelation time of the laser.

That the decorrelaton of our laser is the primary cause for the reduction in temporal coherence has major implications on the way data can be processed. Since the laser is the primary source for limiting the maximum FFT time, using a laser with a longer coherence time could greatly enhance the SNR, and consequently, improve the ability to remotely sense information about the atmosphere.

This chapter gave a description of aerosols in terms of factors that influence number concentration, and typical size distribution in the lower atmosphere (lower 4km). The effect of aerosol size was studied with relation to the volume backscattering coefficient. The modifications to the extended Huygens-Fresnel principle for a laser beam backscattering off of aerosols were introduced which showed how a z-directed wind speed produced a Doppler shift in the laser frequency. A distribution in the z-directed velocity of the aerosols leads to a rapid (~10 microsecond) decorrelation of the backscattered intensity at the receiver. The phase considerations were examined that predicted that the effect of aerosol speckle should not limit the coherence time as compared to the phase effects due to turbulence and that due to the finite coherence time of the CO_2 laser used in this project.

Chapter 4

Time-Delayed Statistics of a Coherent, cw, CO_2 Heterodyne Laser Beam Backscattered from Aerosols

The background for deriving the time-delayed statistics of a CO_2 laser beam backscattered from aerosols has been discussed in chapters 2 and 3. Section 4.1 extends the application of the theory presented in the two previous chapters to the derivation of the time-delayed (or time-lagged) crosscovariance of the intensities between two points in the receiving plane for a CO_2 laser beam backscattered from aerosols. Although the lidar used in this project is a coherent, heterodyne system, applying a low-pass type of filter introduced in section 4.2 can successfully remove the high frequency aerosol speckle part of the time-delayed crosscovariance, leaving only the turbulence contribution. Since the aerosol speckle can be removed, the solutions for the statistics look as if the laser source were incoherent with direct detection. The expressions for the time delayed statistics become simpler, making it easier to develop a crosswind detection scheme.

4.1 Time-Delayed Statistics for Two Point Detectors

Some of the effects of backscattering from aerosols have already been discussed in chapters 2 and 3. In chapter 2, the combination of speckle and turbulence was discussed for single ended lidar systems using a diffuse metallic hard target to backscatter laser radiation. In chapter 3, atmospheric aerosols replaced hard targets as the backscatter. The mutual intensity functions were derived to illustrate the technique of applying the extended Huygens-Fresnel integral. From the derivation in chapter 3 it was shown that aerosols cause the intensity to decorrelate rapidly but have negligible effect on the phase. Additional insight into the way the atmosphere effects laser beam propagation can be obtained by deriving the time-delayed crosscovariance of the intensities between two points in the receiver plane for a laser beam backscattered from aerosols.

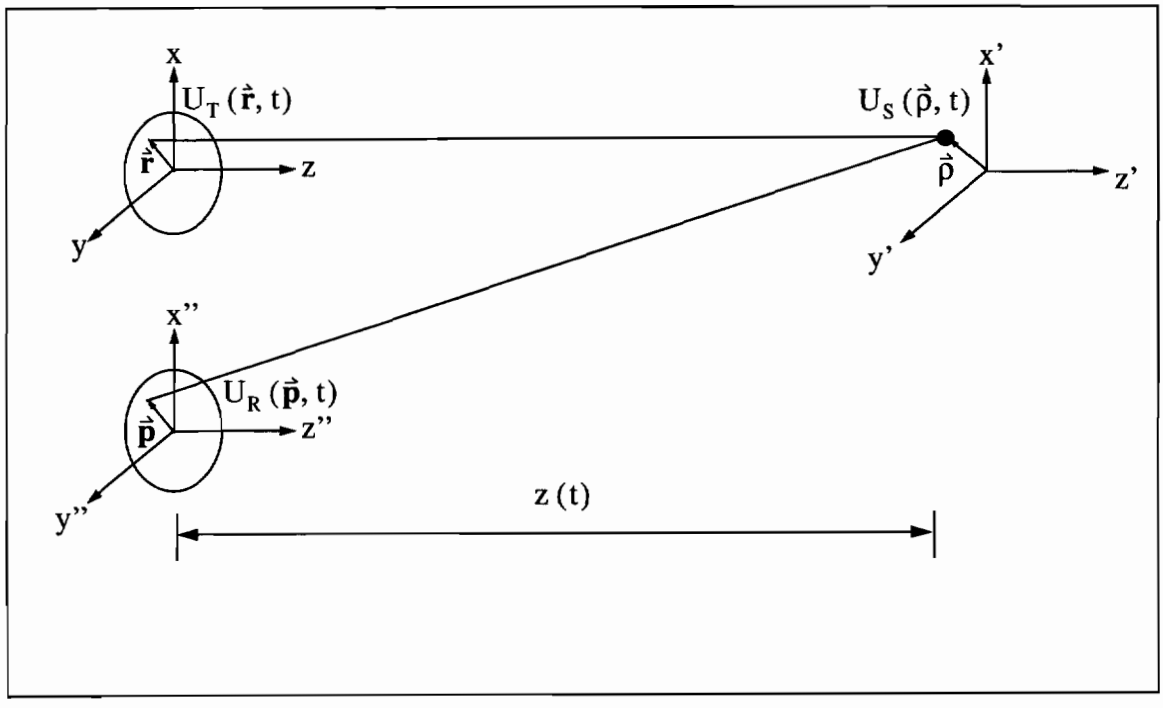


Figure 4.1 Coordinate systems for the single-ended lidar system backscattered from aerosols.

The geometry for a laser beam backscattered from aerosols propagating through turbulence is presented in figure 4.1. The transmitter coordinate is $\vec{\mathbf{r}}$, the coordinate of the aerosols is $\vec{\rho}$, and the detector coordinate is $\vec{\mathbf{p}}$. The distance between transmitter/receiver and the aerosol particle is $z(t)$. In section 3.3 the modifications to the Huygens-Fresnel integral were introduced and the method of applying the integral was demonstrated for the example of computing the time-delayed mutual intensity function, $\Gamma(\vec{\mathbf{p}}_1, \vec{\mathbf{p}}_2, \tau)$, equation (3.20), between two points in the detector plane. The derivation of the time-delayed (or time-lagged) crosscovariance of the intensities between two points in the detector plane will be accomplished in three steps. First, the time-lagged crosscorrelation of the intensities, $b_{I_{1,2}}(\vec{\mathbf{p}}_1, \vec{\mathbf{p}}_2, \tau)$, for the single aerosol scattering case will be formulated. Second, the total time-lagged crosscorrelation of the intensities, $B_{I_{1,2}}(\vec{\mathbf{p}}_1, \vec{\mathbf{p}}_2, \tau)$, will be obtained by integrating the single scattering solution over the aerosol scattering volume for the total crosscorrelation. Finally, the total time-lagged crosscovariance will be derived from the relation

$$C_{I_{1,2}}(\vec{\mathbf{p}}_1, \vec{\mathbf{p}}_2, \tau) = B_{I_{1,2}}(\vec{\mathbf{p}}_1, \vec{\mathbf{p}}_2, \tau) - \langle I(\vec{\mathbf{p}}_1)\rangle\langle I(\vec{\mathbf{p}}_2, \tau)\rangle. \tag{4.1}$$

The derivation of equation (4.1) follows Churnside and Yura,[38] who derived expressions for the time-delayed mutual intensity and correlation functions for speckle fluctuations, but did not include turbulence in the analysis. Murty[39] extended the work to include turbulence and derived an expression for the time-delayed mutual intensity function, but used a quadratic form for the wave structure function instead of the five-thirds law (which is correct), negligible transverse velocity, and considered points only along the beam axis.

The time-lagged crosscorrelation function of two intensities in the detector

plane for single aerosol scattering is given by

$$b_{I_{1,2}}(\vec{\rho}_1, \vec{\rho}_2, \tau) = \langle u_R(\vec{\rho}_1, 0)\, u_R^*(\vec{\rho}_1, 0)\, u_R(\vec{\rho}_2, \tau)\, u_R^*(\vec{\rho}_2, \tau) \rangle . \tag{4.2}$$

Using the extended Huygens-Fresnel integral modified for aerosols, the received electric fields can be expressed as

$$u_R(\vec{\rho}_1, 0) = \frac{S(\vec{\rho}_1)}{z_1} U(\vec{\rho}_1, 0) \exp\left[ikz_1 + \frac{ik}{2z_1} (\vec{\rho}_1 - \vec{\rho}_1)^2 + \psi_b(\vec{\rho}_1, \vec{\rho}_1, 0) \right]$$

$$u^*_R(\vec{\rho}_1, 0) = \frac{S^*(\vec{\rho}_2)}{z_2} U^*(\vec{\rho}_2, 0) \exp\left[-ikz_2 - \frac{ik}{2z_2} (\vec{\rho}_1 - \vec{\rho}_2)^2 + \psi_b^*(\vec{\rho}_1, \vec{\rho}_2, 0) \right]$$

$$u_R(\vec{\rho}_2, \tau) = \frac{S(\vec{\rho}_3)}{z_3} U(\vec{\rho}_3, \tau) \exp\left[ik(z_3 - v_{z_3}\tau) + \frac{ik}{2z_3} [\vec{\rho}_2 - (\vec{\rho}_3 - \vec{v}_{\perp_3}\tau)]^2 \right]$$

$$\times \exp[\psi_b(\vec{\rho}_2, \vec{\rho}_3 - \vec{v}_{\perp_3}\tau, \tau)]$$

and

$$u_R^*(\vec{\rho}_2, \tau) = \frac{S^*(\vec{\rho}_4)}{z_4} U^*(\vec{\rho}_4, \tau) \exp\left[-ik(z_4 - v_{z_4}\tau) - \frac{ik}{2z_4} [\vec{\rho}_2 - (\vec{\rho}_4 - \vec{v}_{\perp_4}\tau)]^2 \right] \cdot$$

$$\times \exp\left[\psi_b^*(\vec{\rho}_2, \vec{\rho}_4 - \vec{v}_{\perp_4}\tau, \tau) \right] \tag{4.3}$$

In these equations, $S(\vec{\rho})$ is the single aerosol amplitude scattering factor, ψ_b includes the effect of turbulence on the backscattered laser electric field from aerosol to detector plane, and $U(\vec{\rho}, \tau)$ is the electric field in the aerosol plane just before scattering. Inserting the expressions for the electric fields back into equation (4.2) gives the expression for the correlation function for single aerosol scattering

$$b_{I_{1,2}}(\vec{\rho}_1, \vec{\rho}_2, \tau) = \frac{1}{z_1 z_2 z_3 z_4} \exp\left[ikz_1 - ikz_2 + ik(z_3 - v_{z_3}\tau) - ik(z_4 - v_{z_4}\tau)\right]$$

$$\times \exp\left[\frac{ik}{2z_1}(\vec{\rho}_1 - \vec{\rho}_1)^2 - \frac{ik}{2z_2}(\vec{\rho}_1 - \vec{\rho}_2)^2\right]$$

$$\times \exp\left[\frac{ik}{2z_3}[\vec{\rho}_2 - (\vec{\rho}_3 - \vec{v}_{\perp_3}\tau)]^2 - \frac{ik}{2z_4}[\vec{\rho}_2 - (\vec{\rho}_4 - \vec{v}_{\perp_4}\tau)]^2\right]$$

$$\times \langle S(\vec{\rho}_1) S^*(\vec{\rho}_2) S(\vec{\rho}_3) S^*(\vec{\rho}_4) U(\vec{\rho}_1, 0) U^*(\vec{\rho}_2, 0) U(\vec{\rho}_3, \tau) U^*(\vec{\rho}_4, \tau)$$

$$\times \exp\left[\psi_b(\vec{\rho}_1, \vec{\rho}_1, 0) + \psi_b^*(\vec{\rho}_1, \vec{\rho}_2, 0) + \psi_b(\vec{\rho}_2, \vec{\rho}_3 - \vec{v}_{\perp_3}\tau, \tau) + \psi_b^*(\vec{\rho}_2, \vec{\rho}_4 - \vec{v}_{\perp_4}\tau, \tau)\right] \rangle \quad (4.4)$$

This expression for the correlation function is complete, because all of the random terms are included within the ensemble average. Since the electric fields just before scattering contain terms which include the effect of turbulence on the electric fields on the forward path from transmitter to aerosols, the random part of equation (4.4) would become an eighth order spherical wave mutual coherence function. The evaluation of the eighth order mutual coherence function generates 40 terms, which makes the correlation function too complicated to manage. Keeping the forward and backward turbulence terms within one ensemble average is equivalent to the dependent paths case, because the forward and backward turbulence terms are treated as dependent, and therefore not separated into two ensemble averages. The simplification is to use the independent paths assumption to treat the forward and backward propagation paths as independent in order to create two, more simple, fourth order turbulence coherence functions. This assumption is more valid for bistatic lidars, where the transmitted and backscattered laser fields are biaxial. However, the lidar used in this project is coaxial; the outgoing and backscattered laser radiation share the same path. Consequently, both fields observe the same turbulence. Therefore, the independent paths assumption

is not strictly valid for the case of a coaxial lidar, but is used to make the derivation of the time-lagged cross covariance more manageable. The independent paths assumption is the only approximation used in the derivation of the time-delayed cross-covariance.

Using the independent paths assumption, the term within the ensemble average in equation (4.4) can be separated into two ensemble averages

$$<S(\vec{\rho}_1)\,S^*(\vec{\rho}_2)\,S(\vec{\rho}_3)\,S^*(\vec{\rho}_4)\,U(\vec{\rho}_1,0)\,U^*(\vec{\rho}_2,0)\,U(\vec{\rho}_3,\tau)\,U^*(\vec{\rho}_4,\tau)>\times$$

$$<\exp\left[\psi_b(\vec{\mathbf{p}}_1,\vec{\rho}_1,0)+\psi_b^*(\vec{\mathbf{p}}_1,\vec{\rho}_2,0)+\psi_b(\vec{\mathbf{p}}_2,\vec{\rho}_3-\vec{\mathbf{v}}_{\perp_3}\tau,\tau)+\psi_b^*(\vec{\mathbf{p}}_2,\vec{\rho}_4-\vec{\mathbf{v}}_{\perp_4}\tau,\tau)\right]>$$

(4.5)

The second ensemble average is the fourth order spherical wave mutual coherence function, $H(\vec{\mathbf{p}}_1,\vec{\mathbf{p}}_2,\vec{\rho}_1,\vec{\rho}_2,\vec{\rho}_3,\vec{\rho}_4,\vec{\mathbf{v}}_{\perp_3},\vec{\mathbf{v}}_{\perp_4},\tau)$, describing the effects of turbulence on the backscattered laser beam over the backward path from the aerosols to the detector. The first ensemble average in equation (4.5) can be rewritten using the principles reviewed in chapters 2 and 3. Specifically, since the fields just after scattering and the amplitude scattering factors are Gaussian and independent,

$$<S(\vec{\rho}_1)\,S^*(\vec{\rho}_2)\,S(\vec{\rho}_3)\,S^*(\vec{\rho}_4)\,U(\vec{\rho}_1,0)\,U^*(\vec{\rho}_2,0)\,U(\vec{\rho}_3,\tau)\,U^*(\vec{\rho}_4,\tau)>\;=$$

$$<S(\vec{\rho}_1)\,S^*(\vec{\rho}_2)><U(\vec{\rho}_1,0)\,U^*(\vec{\rho}_2,0)><S(\vec{\rho}_3)\,S^*(\vec{\rho}_4)><U(\vec{\rho}_3,\tau)\,U^*(\vec{\rho}_4,\tau)>+$$

$$<S(\vec{\rho}_1)\,S^*(\vec{\rho}_4)><U(\vec{\rho}_1,0)\,U^*(\vec{\rho}_4,\tau)><S(\vec{\rho}_3)\,S^*(\vec{\rho}_2)><U(\vec{\rho}_3,\tau)\,U^*(\vec{\rho}_2,0)>.$$

(4.6)

The ensemble average for the cross product between two scattering factors was derived in chapter 3 and resulted in equation (3.18). Using this result in equation (4.6) yields

$$<S(\vec{\rho}_1)\,S^*(\vec{\rho}_2)\,S(\vec{\rho}_3)\,S^*(\vec{\rho}_4)\,U(\vec{\rho}_1,0)\,U^*(\vec{\rho}_2,0)\,U(\vec{\rho}_3,\tau)\,U^*(\vec{\rho}_4,\tau)> =$$

$$<U(\vec{\rho}_1,0)\,U^*(\vec{\rho}_2,0)>S^2N\delta(\vec{\rho}_1-\vec{\rho}_2)\,\delta(z_1-z_2)\,\delta(\vec{v}_{\perp_1}-\vec{v}_{\perp_2})\,\delta(v_{z_1}-v_{z_2})$$

$$\times<U(\vec{\rho}_3,\tau)\,U^*(\vec{\rho}_4,\tau)>S^2N\delta(\vec{\rho}_3-\vec{\rho}_4)\,\delta(z_3-z_4)\,\delta(\vec{v}_{\perp_3}-\vec{v}_{\perp_4})\,\delta(v_{z_3}-v_{z_4})$$

+

$$<U(\vec{\rho}_1,0)\,U^*(\vec{\rho}_4,0)>S^2N\delta(\vec{\rho}_1-\vec{\rho}_4)\,\delta(z_1-z_4)\,\delta(\vec{v}_{\perp_1}-\vec{v}_{\perp_4})\,\delta(v_{z_1}-v_{z_4})$$

$$\times<U(\vec{\rho}_3,\tau)\,U^*(\vec{\rho}_2,\tau)>S^2N\delta(\vec{\rho}_3-\vec{\rho}_2)\,\delta(z_3-z_2)\,\delta(\vec{v}_{\perp_3}-\vec{v}_{\perp_2})\,\delta(v_{z_3}-v_{z_2}) \quad (4.7)$$

where S is the single aerosol amplitude scattering factor and N is the number of particles in the volume. The correlation function for single aerosol scattering results in the summation of two terms

$$b_{1_{1,2}}(\vec{\mathbf{p}}_1,\vec{\mathbf{p}}_2,\tau) = b_1(\vec{\mathbf{p}}_1,\vec{\mathbf{p}}_2,\tau) + b_2(\vec{\mathbf{p}}_1,\vec{\mathbf{p}}_2,\tau)\,. \quad (4.8)$$

The first term in the summation corresponds to substituting the first term on the right hand side of equation (4.7) back into equation (4.4). Concentrating on this term,

$$b_1(\vec{\mathbf{p}}_1, \vec{\mathbf{p}}_2, \tau) = \frac{1}{z_1 z_2 z_3 z_4} \exp\left[ikz_1 - ikz_2 + ik(z_3 - v_{z_3}\tau) - ik(z_4 - v_{z_4}\tau)\right]$$

$$\times \exp\left[\frac{ik}{2z_1}(\vec{\mathbf{p}}_1 - \vec{\rho}_1)^2 - \frac{ik}{2z_2}(\vec{\mathbf{p}}_1 - \vec{\rho}_2)^2\right]$$

$$\times \exp\left[\frac{ik}{2z_3}[\vec{\mathbf{p}}_2 - (\vec{\rho}_3 - \vec{\mathbf{v}}_{\perp_3}\tau)]^2 - \frac{ik}{2z_4}[\vec{\mathbf{p}}_2 - (\vec{\rho}_4 - \vec{\mathbf{v}}_{\perp_4}\tau)]^2\right]$$

$$\times \langle U(\vec{\rho}_1, 0)\, U^*(\vec{\rho}_2, 0)\rangle S^2 N \delta(\vec{\rho}_1 - \vec{\rho}_2)\, \delta(z_1 - z_2)\, \delta(\vec{\mathbf{v}}_{\perp_1} - \vec{\mathbf{v}}_{\perp_2})\, \delta(v_{z_1} - v_{z_2})$$

$$\times \langle U(\vec{\rho}_3, \tau)\, U^*(\vec{\rho}_4, \tau)\rangle S^2 N \delta(\vec{\rho}_3 - \vec{\rho}_4)\, \delta(z_3 - z_4)\, \delta(\vec{\mathbf{v}}_{\perp_3} - \vec{\mathbf{v}}_{\perp_4})\, \delta(v_{z_3} - v_{z_4}) \,. \tag{4.9}$$

The fact that the fields immediately after scattering are Gaussian and independent reduces the fourth order mean value product to the product of two, second order, mutual intensity functions. The total correlation function for the first term is obtained by integrating equation (4.9) over the scattering volume and statistics of the velocity distributions and aerosol position. After applying the delta functions, the total correlation function for term 1 of equation (4.8) becomes,

$$B_1(\vec{\mathbf{p}}_1, \vec{\mathbf{p}}_2, \tau) = \iint d^2\vec{\rho}_1 d^2\vec{\rho}_2 \iint dz_1 dz_2 \iint d^2\vec{\mathbf{v}}_{\perp_1} d^2\vec{\mathbf{v}}_{\perp_2} \mathrm{pdf}(\vec{\mathbf{v}}_{\perp_1})\, \mathrm{pdf}(\vec{\mathbf{v}}_{\perp_2})$$

$$\times \iint dv_{z_1} dv_{z_2} \mathrm{pdf}(v_{z_1})\, \mathrm{pdf}(v_{z_2}) \frac{\beta^2}{z_1^2 z_2^2} \langle U(\vec{\rho}_1, 0)\, U^*(\vec{\rho}_1, 0)\rangle \langle U(\vec{\rho}_2, \tau)\, U^*(\vec{\rho}_2, \tau)\rangle$$

$$\times H(\vec{\mathbf{p}}_1, \vec{\mathbf{p}}_2, \vec{\rho}_1, \vec{\rho}_2, \vec{\mathbf{v}}_{\perp_4}, \tau) \quad . \tag{4.10}$$

In this equation, the backscattering coefficient, β, has replaced $\frac{S^2 N}{\mathrm{Vol}}$. The first and second terms in ensemble brackets are mutual intensity functions which can be evalu-

ated by inserting the expression for the electric field just before scattering, equation (3.13) with $U_T(\vec{r}) = U_{T_0}\exp\left[-\left(\frac{1}{2\alpha_0^2}+\frac{ik}{2F}\right)r^2\right]$ defined for the transmitted electric field in the exit aperture sent to the atmosphere,

$$U(\vec{\rho},\tau) = \frac{kU_{T_0}\exp[ik(z-v_z\tau)]}{i2\pi z}\int d^2\vec{r}\ \exp\left[-\left(\frac{1}{2\alpha_0^2}+\frac{ik}{2F}\right)r^2\right]$$

$$\times\exp\left[\frac{ik}{2z}(\vec{\rho}-\vec{v}_\perp\tau-\vec{r})^2\right]\exp[\psi_f(\vec{r},\vec{\rho}-\vec{v}_\perp\tau,\tau)] \qquad (4.11)$$

into the ensemble averages in equation (4.10). In equation (4.11), F and α_0 are the focus distance and the exp(-2) intensity radius of the outgoing laser beam, respectively. The coordinate r represents the transmitter plane, ψ_f describes the effect of the turbulent atmosphere on a spherical wave propagating from the transmitter to the aerosol particle, and U_{T_0} is the amplitude of the outgoing Gaussian laser beam.

The first mutual intensity function in equation (4.10) is

$$<U(\vec{\rho}_1,0)U^*(\vec{\rho}_1,0)> = \frac{k^2U_{T_0}^2}{(2\pi z_1)^2}\iint d^2\vec{r}_1 d^2\vec{r}_2\exp\left[-\frac{1}{2\alpha_0^2}(r_1^2+r_2^2)\right]$$

$$\times\exp\left[-\frac{ik}{2F}(r_1^2-r_2^2)\right]\exp\left[\frac{ik}{2z_1}(\vec{\rho}_1-\vec{r}_1)^2-\frac{ik}{2z_1}(\vec{\rho}_1-\vec{r}_2)^2\right]$$

$$\times<\exp[\psi_f(\vec{r}_1,\vec{\rho}_1,0)+\psi_f^*(\vec{r}_2,\vec{\rho}_1,0)]>. \qquad (4.12)$$

The term in ensemble brackets in equation (4.12) can be evaluated to be[24]

$$<> = \exp\left[-\frac{1}{2}D_\phi(\vec{r}_1-\vec{r}_2,0,0)\right]$$

$$= \exp\left[-\frac{2.91}{2}z_1k^2\int_0^1 d\omega C_n^2(\omega)|\omega(\vec{r}_1-\vec{r}_2)|^{5/3}\right]$$

where k is the wave number and $C_n^2(\omega)$ is the strength of turbulence. The integration variable, ω, is the normalized path length from transmitter to aerosol particle. With this definition, equation (4.12) becomes

$$<U(\vec{\rho}_1, 0)\, U^*(\vec{\rho}_1, 0)> = \frac{k^2 U_{T_0}^2}{(2\pi z_1)^2}\iint d^2\vec{r}_1 d^2\vec{r}_2 \exp\left[-\frac{1}{2\alpha_0^2}(r_1^2 + r_2^2)\right]$$

$$\times \exp\left[-\frac{ik}{2F}(r_1^2 - r_2^2)\right]\exp\left[\frac{ik}{2z_1}(\vec{\rho}_1 - \vec{r}_1)^2 - \frac{ik}{2z_1}(\vec{\rho}_1 - \vec{r}_2)^2\right]$$

$$\times \exp\left[-\frac{1}{2}D_\phi(\vec{r}_1 - \vec{r}_2, 0, 0)\right] . \qquad (4.13)$$

The **r** integrations can be more easily evaluated by making the change of variables

$$\vec{r}_1 - \vec{r}_2 = \vec{r} \qquad \vec{r}_1 + \vec{r}_2 = 2\vec{R}$$

$$r_1^2 + r_2^2 = 2R^2 + \frac{1}{2}r^2 \qquad \vec{r}_1 = \vec{R} + \frac{1}{2}\vec{r} \qquad \text{and}$$

$$r_1^2 - r_2^2 = 2\vec{R}\cdot\vec{r} \qquad \vec{r}_2 = \vec{R} - \frac{1}{2}\vec{r}.$$

With this change of variables, equation (4.12) becomes

$$<U(\vec{\rho}_1, 0)\, U^*(\vec{\rho}_1, 0)> = \frac{k^2 U_{T_0}^2}{(2\pi z_1)^2}\iint d^2\vec{r} d^2\vec{R}\ \exp\left[-\frac{1}{2\alpha_0^2}\left(2R^2 + \frac{1}{2}r^2\right)\right]$$

$$\times \exp\left[-\frac{ik}{2F}(2\vec{R}\cdot\vec{r})\right]\exp\left[\frac{ik}{2z_1}(-2\vec{\rho}_1\cdot\vec{r} + 2\vec{R}\cdot\vec{r})\right]$$

$$\times \exp\left[-\frac{1}{2}D_\phi(\vec{r}, 0, 0)\right] . \qquad (4.14)$$

The $\vec{R}$ terms of equation (4.14) can be grouped and integrated[51]

$$\int d^2\vec{R} = \int d^2\vec{R}\exp\left[-\frac{R^2}{2\alpha_0^2}\right]\exp\left[ik\vec{R}\cdot\vec{r}\left(\frac{1}{z_1} - \frac{1}{F}\right)\right]$$

$$= \int_0^\infty R dR \exp\left[-\frac{R^2}{2\alpha_0^2}\right] \int_0^{2\pi} d\theta \exp\left[ikRr\cos(\theta_{Rr})\left(\frac{1}{z_1} - \frac{1}{F}\right)\right]$$

$$= \pi\alpha_0^2 \exp\left[-\frac{k^2\alpha_0^2}{4}\left(\frac{1}{z_1} - \frac{1}{F}\right)^2 r^2\right] .$$

With this integration, equation (4.14) simplifies to

$$<U(\vec{\rho}_1, 0)\, U^*(\vec{\rho}_1, 0)> = \frac{\pi\alpha_0^2 k^2 U_{T_0}^2}{(2\pi z_1)^2} \int d^2\hat{r} \exp\left[-\frac{r^2}{4\alpha_0^2}\right] \exp\left[-\frac{k^2\alpha_0^2}{4}\left(\frac{1}{z_1} - \frac{1}{F}\right)^2 r^2\right]$$

$$\times \exp\left[-\frac{ik}{z_1}\vec{\rho}_1 \cdot \hat{r}\right] \exp\left[-\frac{1}{2} D_\phi(\hat{r}, 0, 0)\right] . \qquad (4.15)$$

This equation is taken as far as it can go without approximations. Following the same steps as the first mutual intensity function, the second mutual intensity function reduces to

$$<U(\vec{\rho}_2, \tau)\, U^*(\vec{\rho}_2, \tau)> = \frac{\pi\alpha_0^2 k^2 U_{T_0}^2}{(2\pi z_2)^2} \int d^2\hat{r} \exp\left[-\frac{r^2}{4\alpha_0^2}\right] \exp\left[-\frac{k^2\alpha_0^2}{4}\left(\frac{1}{z_1} - \frac{1}{F}\right)^2 r^2\right]$$

$$\times \exp\left[-\frac{ik}{z_1}\vec{\rho}_2 \cdot \hat{r} + \frac{ik}{z_2}\hat{r} \cdot \vec{v}_2\tau\right] \exp\left[-\frac{1}{2} D_\phi(\hat{r}, 0, 0)\right] \qquad (4.16)$$

which is as far as it can be taken without approximations.

The last random term in equation (4.10) is the fourth order spherical wave mutual coherence function, $H(\vec{p}_1, \vec{p}_2, \vec{\rho}_1, \vec{\rho}_2, \vec{v}_{\perp_4}, \tau)$, which evaluates to[24]

$$H(\vec{p}_1, \vec{p}_2, \vec{\rho}_1, \vec{\rho}_2, \vec{v}_{\perp_4}, \tau) = \exp\left[4C_\chi(\vec{\rho}_1 - \vec{\rho}_2 + \vec{v}_{\perp_2}\tau, \vec{p}_1 - \vec{p}_2, -\tau)\right] \qquad (4.17)$$

where $C_\chi(\vec{\rho}, \vec{p}, \tau)$ is the log amplitude covariance given by

$$C_\chi(\vec{\rho},\vec{p},\tau) = 0.132\pi^2k^2C_n^2z\int d\omega\int du\ u^{-8/3}\sin^2\left[\frac{\omega(1-\omega)u^2z}{2k}\right]$$

$$\times J_0\left[u\left|\omega\vec{p}+(1-\omega)\vec{\rho}-\langle\vec{v}_\perp\rangle\tau\right|\right]. \qquad (4.18)$$

Finally, the $\int d^2\vec{v}_{\perp_1}$, $\int dv_{z_1}$, and $\int dv_{z_2}$ integrals of equation (4.10) are trivial:

$$\int d^2\vec{v}_{\perp_1}pdf(\vec{v}_{\perp_1}) = 1$$

$$\int dv_{z_1}pdf(v_{z_1}) = 1$$

$$\int dv_{z_2}pdf(v_{z_2}) = 1.$$

Substituting both mutual intensity functions and the fourth order spherical wave mutual coherence function back into equation (4.10) gives the expression for the correlation function for the first term in the summation of the total correlation function

$$B_1(\vec{p}_1,\vec{p}_2,\tau) = \frac{\beta^2}{\pi^2}\left(\frac{kU_{T_0}\alpha_0}{2}\right)^4\int\int d^2\vec{\rho}_1d^2\vec{\rho}_2\int\int dz_1dz_2\int d^2\vec{v}_{\perp_2}pdf(\vec{v}_{\perp_2})\int\int d^2\vec{r}_1d^2\vec{r}_2$$

$$\times\frac{1}{(z_1z_2)^4}\exp\left[-\frac{1}{4\alpha_0^2}(r_1^2+r_2^2)-\left(\frac{k\alpha_0}{2}\right)^2\left(\frac{1}{z_1}-\frac{1}{F}\right)^2r_1^2\right]$$

$$\times\exp\left[-\left(\frac{k\alpha_0}{2}\right)^2\left(\frac{1}{z_2}-\frac{1}{F}\right)^2r_2^2\right]\exp\left[-\frac{ik}{z_1}\vec{\rho}_1\cdot\vec{r}_1-\frac{ik}{z_2}\vec{\rho}_2\cdot\vec{r}_2+\frac{ik}{z_2}\vec{r}_2\cdot\vec{v}_{\perp_2}\tau\right]$$

$$\times\exp\left[-\frac{1}{2}D_\phi(\vec{r}_1,0,0)-\frac{1}{2}D_\phi(\vec{r}_2,0,0)\right]\exp\left[4C_\chi(\vec{\rho}_1-\vec{\rho}_2+\vec{v}_{\perp_2}\tau,\vec{p}_1-\vec{p}_2,-\tau)\right]. \qquad (4.19)$$

The ρ integrations can be more easily accomplished with another transformation of

variables similar to the one on p. 82

$$\vec{\rho}_1 - \vec{\rho}_2 = \vec{\rho} \qquad\qquad \vec{\rho}_1 + \vec{\rho}_2 = 2\vec{\mathbf{R}}$$

making the first term in the summation of the total correlation function

$$B_1(\vec{\mathbf{p}}_1, \vec{\mathbf{p}}_2, \tau) = \frac{\beta^2}{\pi^2}\left(\frac{kU_{T_0}\alpha_0}{2}\right)^4 \iint d^2\vec{\rho}\, d^2\vec{\mathbf{R}} \iint dz_1 dz_2 \int d^2\vec{\mathbf{v}}_{\perp_2} pdf(\vec{\mathbf{v}}_{\perp_2}) \iint d^2\vec{\mathbf{r}}_1 d^2\vec{\mathbf{r}}_2$$

$$\times \frac{1}{(z_1 z_2)^4} \exp\left[-\frac{1}{4\alpha_0^2}(r_1^2 + r_2^2) - \left(\frac{k\alpha_0}{2}\right)^2 \left(\frac{1}{z_1} - \frac{1}{F}\right)^2 r_1^2\right]$$

$$\times \exp\left[-\left(\frac{k\alpha_0}{2}\right)^2 \left(\frac{1}{z_2} - \frac{1}{F}\right)^2 r_2^2\right] \exp\left[-\frac{ik}{z_1}\vec{\mathbf{r}}_1 \cdot \left(\vec{\mathbf{R}} + \frac{1}{2}\vec{\rho}\right) - \frac{ik}{z_2}\vec{\mathbf{r}}_2 \cdot \left(\vec{\mathbf{R}} - \frac{1}{2}\vec{\rho}\right)\right]$$

$$\times \exp\left[\frac{ik}{z_2}\vec{\mathbf{r}}_2 \cdot \vec{\mathbf{v}}_{\perp_2}\tau\right] \exp\left[-\frac{1}{2}D_\phi(\vec{\mathbf{r}}_1, 0, 0) - \frac{1}{2}D_\phi(\vec{\mathbf{r}}_2, 0, 0)\right] \exp\left[4C_\chi(\vec{\rho} + \vec{\mathbf{v}}_{\perp_2}\tau, \vec{\mathbf{p}}_1 - \vec{\mathbf{p}}_2, -\tau)\right] \tag{4.20}$$

The $\vec{\mathbf{R}}$ integration can be performed[51]

$$\int d^2\vec{\mathbf{R}} = 2\pi\delta\left[k\left|\frac{\vec{\mathbf{r}}_1}{z_1} + \frac{\vec{\mathbf{r}}_2}{z_2}\right|\right]$$

and with this delta function, the $\vec{\mathbf{r}}_2$ integration is easy[52], reducing equation (4.20) to

$$B_1(\vec{\mathbf{p}}_1, \vec{\mathbf{p}}_2, \tau) = \frac{2\beta^2}{k\pi}\left(\frac{kU_{T_0}\alpha_0}{2}\right)^4 \int d^2\vec{\rho} \iint dz_1 dz_2 \int d^2\vec{\mathbf{v}}_{\perp_2} pdf(\vec{\mathbf{v}}_{\perp_2}) \int d^2\vec{\mathbf{r}}_1 \frac{z_2}{(z_1 z_2)^4}$$

$$\times \exp\left[-\frac{r_1^2}{4\alpha_0^2}\left(1 + \frac{z_2^2}{z_1^2}\right)\right] \exp\left[-\left(\frac{k\alpha_0}{2}\right)^2 r_1^2 \left(\frac{1}{z_1} - \frac{1}{F}\right)^2\right] \exp\left[-\left(\frac{k\alpha_0}{2}\right)^2 r_1^2 \left(\frac{1}{z_2} - \frac{1}{F}\right)^2 \left(\frac{z_2}{z_1}\right)^2\right]$$

$$\times \exp\left[-\frac{ik}{z_1}\vec{\mathbf{r}}_1 \cdot (\vec{\rho} + \vec{\mathbf{v}}_{\perp_2}\tau)\right] \exp\left[-\frac{1}{2}D_\phi(\vec{\mathbf{r}}_1, 0, 0) - \frac{1}{2}D_\phi\left(\frac{z_2}{z_1}\vec{\mathbf{r}}_1, 0, 0\right)\right]$$

$$\times \exp\left[4C_\chi(\vec{\rho} + \vec{\mathbf{v}}_{\perp_2}\tau, \vec{\mathbf{p}}_1 - \vec{\mathbf{p}}_2, -\tau)\right] \quad . \tag{4.21}$$

This equation is as far as the first term in the total correlation function of equation (4.10) can be taken without any further approximations.

The second term in the expression for the total correlation function can be derived by substituting the second term in the summation of equation (4.7) into equation (4.4), applying the delta functions, and integrating over the scattering volume and velocity statistics

$$B_2(\vec{\rho}_1, \vec{\rho}_2, \tau) = \beta^2 \iint d^2\vec{\rho}_1 d^2\vec{\rho}_2 \iint dz_1 dz_2 \iint d\vec{v}_{\perp_1} d\vec{v}_{\perp_2} \iint dv_{z_1} dv_{z_2} pdf(\vec{v}_{\perp_1})\, pdf(\vec{v}_{\perp_2})$$

$$\times \frac{1}{(z_1 z_2)^2} pdf(v_{z_1})\, pdf(v_{z_2}) < U(\vec{\rho}_2, 0)\, U^*(\vec{\rho}_2, \tau) > \exp[ikv_{z_2}\tau - ikv_{z_1}\tau]$$

$$\times < U(\vec{\rho}_1, \tau)\, U^*(\vec{\rho}_1, 0) > \exp\left[\frac{ik}{2z_2}(\vec{\rho}_1 - \vec{\rho}_2)^2 - \frac{ik}{2z_1}(\vec{\rho}_1 - \vec{\rho}_1)^2 + \frac{ik}{2z_1}(\vec{\rho}_2 - \vec{\rho}_1 + \vec{v}_{\perp_1}\tau)^2\right]$$

$$\times \exp\left[-\frac{ik}{2z_2}(\vec{\rho}_2 - \vec{\rho}_2 + \vec{v}_{\perp_2}\tau)^2\right] H_2(\vec{\rho}_2 - \vec{\rho}_1, \vec{v}_{\perp_1}, \vec{v}_{\perp_2}, \vec{\rho}_1 - \vec{\rho}_2, \tau) \qquad (4.22)$$

where

$$H_2(\vec{\rho}_2 - \vec{\rho}_1, \vec{v}_{\perp_1}, \vec{v}_{\perp_2}, \vec{\rho}_1 - \vec{\rho}_2, \tau) =$$

$$\exp\left[\psi_b(\vec{\rho}_2, \vec{\rho}_1, 0) + \psi_b^*(\vec{\rho}_1, \vec{\rho}_1, 0) + \psi_b(\vec{\rho}_1 - \vec{v}_{\perp_1}\tau, \vec{\rho}_2, \tau) + \psi_b^*(\vec{\rho}_2 - \vec{v}_{\perp_2}\tau, \vec{\rho}_2, \tau)\right].$$

(4.23)

As with the first term in the total correlation function, the second term has three random terms consisting of two mutual intensity functions and one fourth order spherical wave mutual coherence function. The main difference between the first and second terms in the correlation function is that the second term depends on the z directed velocity, v_z. As was demonstrated in chapter 3, the z-directed wind speed causes a

rapid decorrelation of the intensity. Similarly, the second term of the correlation function will be shown to decorrelate rapidly in time because of the dependence on the z-directed wind. In section 4.2, it will be shown that the second term can be filtered and removed using the fact the second term decorrelates more rapidly in time compared with the first term.

The evaluation of B_2 begins by evaluating each of the mutual intensity functions. The first mutual intensity function is, using equation (4.11),

$$<U(\vec{\rho}_2, 0)\, U^*(\vec{\rho}_2, \tau)> = \frac{k^2 U_{T_0}^2 \exp[ikv_{z_2}\tau]}{(2\pi)^2 z_2^2} \iint d^2\hat{r}_1 d^2\hat{r}_2 \exp\left[-\frac{1}{2\alpha_0^2}(r_1^2 + r_2^2)\right]$$

$$\times \exp\left[-\frac{ik}{2F}(r_1^2 - r_2^2)\right] \exp\left[\frac{ik}{2z_2}(r_1^2 - r_2^2 - 2\vec{\rho}_2 \cdot (\hat{r}_1 - \hat{r}_2 - \hat{v}_{\perp_2}\tau) - 2\hat{r}_2 \cdot \hat{v}_{\perp_2}\tau - v_{\perp_2}^2\tau^2)\right]$$

$$\times \exp\left[-\frac{1}{2} D_\phi(\hat{r}_1 - \hat{r}_2, \hat{v}_{\perp_2}\tau, -\tau)\right]. \qquad (4.24)$$

By the change of variables

$$\hat{r}_1 - \hat{r}_2 = \hat{r} \qquad\qquad \hat{r}_1 + \hat{r}_2 = 2\vec{R}$$

the $\vec{R}$ integration can be performed in closed form[51]

$$\int d^2\vec{R} = \iint R\,dR\,d\theta\, \exp\left[-\frac{R^2}{\alpha_0^2}\right] \exp\left[-ikR\left|\frac{\hat{r}}{F} - \frac{\hat{r}}{z_2} + \frac{\hat{v}_{\perp_2}\tau}{z_2}\right| \cos\theta\right]$$

$$= \pi\alpha_0^2 \exp\left[-\left(\frac{\alpha_0}{2}\right)^2 k^2 \left(\frac{\hat{r}}{F} - \frac{\hat{r}}{z_2} + \frac{\hat{v}_{\perp_2}\tau}{z_2}\right)^2\right].$$

With this integration, equation (4.24) becomes

$$<U(\vec{\rho}_2, 0)\, U^*(\vec{\rho}_2, \tau)> = \frac{\alpha_0^2 k^2 U_{T_0}^2 \exp[ikv_{z_2}\tau]}{\pi(2z_2)^2} \int d^2\hat{r}\, \exp\left[-\frac{r^2}{4\alpha_0^2}\right]$$

$$\times \exp\left[-\left(\frac{k\alpha_0}{2}\right)^2\left(\frac{\vec{r}}{F}-\frac{\vec{r}}{z_2}+\frac{\vec{v}_{\perp_2}\tau}{z_2}\right)^2\right]\exp\left[\frac{ik}{2z_2}\left(-2\vec{\rho}_2\cdot\left(\vec{r}-\vec{v}_{\perp_2}\tau\right)+\vec{r}\cdot\vec{v}_{\perp_2}\tau-v_{\perp_2}^2\tau^2\right)\right]$$

$$\times \exp\left[-\frac{1}{2}D_\phi\left(\vec{r},\vec{v}_{\perp_2}\tau,-\tau\right)\right] \tag{4.25}$$

which is as far as it can be taken without approximations.

Using similar steps to derive the first mutual intensity function, the second mutual intensity function can be reduced to

$$<U(\vec{\rho}_1,\tau)\,U^*(\vec{\rho}_1,0)> = \frac{(\alpha_0 k)^2 U_{T_0}^2 \exp\left[-ikv_{z_1}\tau\right]}{\pi(2z_1)^2}\int d^2\vec{r}\ \exp\left[-\frac{r^2}{4\alpha_0^2}\right]$$

$$\times \exp\left[-\left(\frac{k\alpha_0}{2}\right)^2\left(\frac{\vec{r}}{F}-\frac{\vec{r}}{z_1}+\frac{\vec{v}_{\perp_1}\tau}{z_1}\right)\right]\exp\left[-\frac{ik}{2z_1}\left(-2\vec{\rho}_1\cdot\left(\vec{r}-\vec{v}_{\perp_1}\tau\right)+\vec{r}\cdot\vec{v}_{\perp_1}\tau-v_{\perp_1}^2\tau^2\right)\right]$$

$$\times \exp\left[-\frac{1}{2}D_\phi\left(\vec{r},-\vec{v}_{\perp_1}\tau,\tau\right)\right] \tag{4.26}$$

The second term in the summation of the total correlation function becomes

$$B_2(\vec{\rho}_1,\vec{\rho}_2,\tau) = \frac{\beta^2(\alpha_0 k U_{T_0})^4}{16\pi^2}\iint d^2\vec{\rho}_1 d^2\vec{\rho}_2\iint dz_1 dz_2\iint d\vec{v}_{\perp_1} d\vec{v}_{\perp_2}\iint dv_{z_1} dv_{z_2}$$

$$\times\iint d^2\vec{r}_1 d^2\vec{r}_2\frac{pdf(\vec{v}_{\perp_1})\,pdf(\vec{v}_{\perp_2})\,pdf(v_{z_1})\,pdf(v_{z_2})}{(z_1 z_2)^4}\exp\left[2ikv_{z_2}\tau-2ikv_{z_1}\tau\right]$$

$$\times\exp\left[-\frac{1}{4\alpha_0^2}\left(r_1^2+r_2^2\right)\right]\exp\left[\frac{ik}{2z_2}\left(\vec{\rho}_1-\vec{\rho}_2\right)^2-\frac{ik}{2z_1}\left(\vec{\rho}_1-\vec{\rho}_1\right)^2+\frac{ik}{2z_1}\left(\vec{\rho}_2-\vec{\rho}_1+\vec{v}_{\perp_1}\tau\right)^2\right]$$

$$\times\exp\left[-\left(\frac{k\alpha_0}{2}\right)^2\left(\frac{\vec{r}_1}{F}-\frac{\vec{r}_1}{z_2}+\frac{\vec{v}_{\perp_1}\tau}{z_2}\right)\right]\exp\left[-\left(\frac{k\alpha_0}{2}\right)^2\left(\frac{\vec{r}_2}{F}-\frac{\vec{r}_2}{z_1}+\frac{\vec{v}_{\perp_2}\tau}{z_1}\right)^2\right]$$

$$\times\exp\left[-\frac{ik}{2z_1}\left(-2\vec{\rho}_1\cdot\left(\vec{r}_1-\vec{v}_{\perp_1}\tau\right)+\vec{r}_1\cdot\vec{v}_{\perp_1}\tau-v_{\perp_1}^2\tau^2\right)\right]\exp\left[-\frac{1}{2}D_\phi\left(\vec{r}_1,-\vec{v}_{\perp_1}\tau,\tau\right)\right]$$

$$\times \exp\left[\frac{ik}{2z_2}\left(-2\vec{\rho}_2\cdot\left(\vec{r}_2-\vec{v}_{\perp_2}\tau\right)+\vec{r}_2\cdot\vec{v}_{\perp_2}\tau-v_{\perp_2}^2\tau^2\right)\right]\exp\left[-\frac{1}{2}D_\phi\left(\vec{r}_2,\vec{v}_{\perp_2}\tau,-\tau\right)\right]$$

$$\times \exp\left[-\frac{ik}{2z_2}\left(\vec{p}_2-\vec{\rho}_2+\vec{v}_{\perp_2}\tau\right)^2\right]H_2\left(\vec{\rho}_2-\vec{\rho}_1,\vec{v}_{\perp_1},\vec{v}_{\perp_2},\vec{p}_1-\vec{p}_2,\tau\right). \qquad (4.27)$$

There are still two integrals which can be done in closed form, the integrals over the z-directed velocity components. Using the following definition for the probability distribution function of the z-directed velocity

$$\mathrm{pdf}\left(v_{z_1}\right) = \frac{1}{\sqrt{2\pi}\sigma_z}\exp\left[\frac{-\left(v_z-\langle v_z\rangle\right)^2}{2\sigma^2}\right]$$

the $\int dv_{z_1}$ and $\int dv_{z_2}$ integrations are[41]

$$\int dv_{z_1}\frac{\exp\left[-2ikv_{z_1}\tau\right]}{\sqrt{2\pi}\sigma_{z_1}}\exp\left[\frac{-\left(v_{z_1}-\langle v_{z_1}\rangle\right)^2}{2\sigma_{z_1}^2}\right] = \exp\left[-2ik\langle v_{z_1}\rangle\tau\right]\exp\left[-2\left(\sigma_{z_1}k\tau\right)^2\right]$$

$$\int dv_{z_2}\frac{\exp\left[2ikv_{z_2}\tau\right]}{\sqrt{2\pi}\sigma_{z_2}}\exp\left[\frac{-\left(v_{z_2}-\langle v_{z_2}\rangle\right)^2}{2\sigma_{z_2}^2}\right] = \exp\left[2ik\langle v_{z_2}\rangle\tau\right]\exp\left[-2\left(\sigma_{z_2}k\tau\right)^2\right]$$

Since $\langle v_{z_1}\rangle = \langle v_{z_2}\rangle$ and $\sigma_{z_1} = \sigma_{z_2}$, the second term of the total correlation function becomes

$$B_2\left(\vec{p}_1,\vec{p}_2,\tau\right) = \frac{\beta^2\left(\alpha_0 kU_{T_0}\right)^4\exp\left[-4\left(\sigma_z k\tau\right)^2\right]}{16\pi^2}\iint d^2\vec{\rho}_1 d^2\vec{\rho}_2\iint dz_1 dz_2\iint d\vec{v}_{\perp_1}d\vec{v}_{\perp_2}$$

$$\times\iint d^2\vec{r}_1 d^2\vec{r}_2\frac{\mathrm{pdf}\left(\vec{v}_{\perp_1}\right)\mathrm{pdf}\left(\vec{v}_{\perp_2}\right)}{\left(z_1z_2\right)^4}\exp\left[-\frac{1}{4\alpha_0^2}\left(r_1^2+r_2^2\right)\right]\exp\left[-\left(\frac{k\alpha_0}{2}\right)^2\left(\frac{\vec{r}_1}{F}-\frac{\vec{r}_1}{z_2}+\frac{\vec{v}_{\perp_1}\tau}{z_2}\right)\right]$$

$$\times\exp\left[\frac{ik}{2z_2}\left(\vec{p}_1-\vec{\rho}_2\right)^2-\frac{ik}{2z_1}\left(\vec{p}_1-\vec{\rho}_1\right)^2+\frac{ik}{2z_1}\left(\vec{p}_2-\vec{\rho}_1+\vec{v}_{\perp_1}\tau\right)^2-\frac{ik}{2z_2}\left(\vec{p}_2-\vec{\rho}_2+\vec{v}_{\perp_2}\tau\right)^2\right]$$

$$\times \exp\left[-\frac{ik}{2z_1}\left(-2\vec{\rho}_1 \cdot \left(\hat{\vec{r}}_1 - \hat{\vec{v}}_{\perp_1}\tau\right) + \hat{\vec{r}}_1 \cdot \hat{\vec{v}}_{\perp_1}\tau - v_{\perp_1}^2\tau^2\right)\right] \exp\left[-\frac{1}{2}D_\phi\left(\hat{\vec{r}}_1, -\hat{\vec{v}}_{\perp_1}\tau, \tau\right)\right]$$

$$\times \exp\left[\frac{ik}{2z_2}\left(-2\vec{\rho}_2 \cdot \left(\hat{\vec{r}}_2 - \hat{\vec{v}}_{\perp_2}\tau\right) + \hat{\vec{r}}_2 \cdot \hat{\vec{v}}_{\perp_2}\tau - v_{\perp_2}^2\tau^2\right)\right] \exp\left[-\frac{1}{2}D_\phi\left(\hat{\vec{r}}_2, \hat{\vec{v}}_{\perp_2}\tau, -\tau\right)\right]$$

$$\times \exp\left[-\left(\frac{k\alpha_0}{2}\right)^2\left(\frac{\hat{\vec{r}}_2}{F} - \frac{\hat{\vec{r}}_2}{z_1} + \frac{\hat{\vec{v}}_{\perp_2}\tau}{z_1}\right)^2\right] H_2\left(\vec{\rho}_2 - \vec{\rho}_1, \hat{\vec{v}}_{\perp_1}, \hat{\vec{v}}_{\perp_2}, \vec{p}_1 - \vec{p}_2, \tau\right). \qquad (4.28)$$

The presence of the term $\exp\left[-4\left(\sigma_z k\tau\right)^2\right]$ causes the second term in the total correlation function to approach zero rapidly as τ increases. Consequently, the second term decorrelates much more rapidly in time than the first term in the total correlation function.

The next step in developing the time-delayed crosscovariance of the intensities is to form

$$C_{I_{1,2}}\left(\vec{p}_1, \vec{p}_2, \tau\right) = B_{I_{1,2}}\left(\vec{p}_1, \vec{p}_2, \tau\right) - \langle I\left(\vec{p}_1\right)\rangle\langle I\left(\vec{p}_2, \tau\right)\rangle$$

where $B_{I_{1,2}}\left(\vec{p}_1, \vec{p}_2, \tau\right) = B_1\left(\vec{p}_1, \vec{p}_2, \tau\right) + B_2\left(\vec{p}_1, \vec{p}_2, \tau\right)$ and $\langle I\rangle$ is the average intensity at one point in the receiver plane. Without going through the derivation of $\langle I\left(\vec{p}_1\right)\rangle\langle I\left(\vec{p}_2, \tau\right)\rangle$ in detail, it can be shown that this mean value product is exactly $B_1\left(\vec{p}_1, \vec{p}_2, \tau\right)$, equation (4.21), but without the $\exp\left[4C_\chi\left(\vec{\rho} + \hat{\vec{v}}_{\perp_2}\tau, \vec{p}_1 - \vec{p}_2, -\tau\right)\right]$ term. Therefore, the total time-delayed crosscovariance of two intensities in the receiver aperture for the case of aerosol backscattering is, under the assumption of independent paths,

$$C_{I_{1,2}}\left(\vec{p}_1, \vec{p}_2, \tau\right) = \frac{2\beta^2}{k\pi}\left(\frac{kU_{T_0}\alpha_0}{2}\right)^4 \int d^2\vec{\rho} \iint dz_1 dz_2 \int d^2\hat{\vec{v}}_{\perp_2} pdf\left(\hat{\vec{v}}_{\perp_2}\right) \int d^2\hat{\vec{r}}_1 \frac{z_2}{\left(z_1 z_2\right)^4}$$

$$\times \exp\left[-\frac{r_1^2}{4\alpha_0^2}\left(1 + \frac{z_2^2}{z_1^2}\right)\right] \exp\left[-\left(\frac{k\alpha_0}{2}\right)^2 r_1^2\left(\frac{1}{z_1} - \frac{1}{F}\right)^2\right] \exp\left[-\left(\frac{k\alpha_0}{2}\right)^2 r_1^2\left(\frac{1}{z_2} - \frac{1}{F}\right)^2\left(\frac{z_2}{z_1}\right)^2\right]$$

$$\times \exp\left[-\frac{ik}{z_1}\vec{r}_1 \cdot (\vec{\rho} + \vec{v}_{\perp_2}\tau)\right] \exp\left[-\frac{1}{2}D_\phi(\vec{r}_1, 0, 0) - \frac{1}{2}D_\phi(\frac{z_2}{z_1}\vec{r}_1, 0, 0)\right]$$

$$\times \left[\exp\left[4C_\chi(\vec{\rho} + \vec{v}_{\perp_2}\tau, \vec{p}_1 - \vec{p}_2, -\tau)\right] - 1\right]$$

+

$$\frac{\beta^2(\alpha_0 k U_{T_0})^4 \exp\left[-4(\sigma_z k\tau)^2\right]}{16\pi^2} \iint d^2\vec{\rho}_1 d^2\vec{\rho}_2 \iint dz_1 dz_2 \iint d\vec{v}_{\perp_1} d\vec{v}_{\perp_2}$$

$$\times \iint d^2\vec{r}_1 d^2\vec{r}_2 \frac{pdf(\vec{v}_{\perp_1})\, pdf(\vec{v}_{\perp_2})}{(z_1 z_2)^4} \exp\left[-\frac{1}{4\alpha_0^2}(r_1^2 + r_2^2)\right] \exp\left[-\left(\frac{k\alpha_0}{2}\right)^2 \left(\frac{\vec{r}_1}{F} - \frac{\vec{r}_1}{z_1} + \frac{\vec{v}_{\perp_1}\tau}{z_1}\right)^2\right]$$

$$\times \exp\left[\frac{ik}{2z_2}(\vec{p}_1 - \vec{\rho}_2)^2 - \frac{ik}{2z_1}(\vec{p}_1 - \vec{\rho}_1)^2 + \frac{ik}{2z_1}(\vec{p}_2 - \vec{\rho}_1 + \vec{v}_{\perp_1}\tau)^2 - \frac{ik}{2z_2}(\vec{p}_2 - \vec{\rho}_2 + \vec{v}_{\perp_2}\tau)^2\right]$$

$$\times \exp\left[-\frac{ik}{2z_1}(-2\vec{\rho}_1 \cdot (\vec{r}_1 - \vec{v}_{\perp_1}\tau) + \vec{r}_1 \cdot \vec{v}_{\perp_1}\tau - v_{\perp_1}^2\tau^2)\right] \exp\left[-\frac{1}{2}D_\phi(\vec{r}_1, -\vec{v}_{\perp_1}\tau, \tau)\right]$$

$$\times \exp\left[\frac{ik}{2z_2}(-2\vec{\rho}_2 \cdot (\vec{r}_2 - \vec{v}_{\perp_2}\tau) + \vec{r}_2 \cdot \vec{v}_{\perp_2}\tau - v_{\perp_2}^2\tau^2)\right] \exp\left[-\frac{1}{2}D_\phi(\vec{r}_2, \vec{v}_{\perp_2}\tau, -\tau)\right]$$

$$\times \exp\left[-\left(\frac{k\alpha_0}{2}\right)^2 \left(\frac{\vec{r}_2}{F} - \frac{\vec{r}_2}{z_2} + \frac{\vec{v}_{\perp_2}\tau}{z_2}\right)^2\right] H_2(\vec{\rho}_2 - \vec{\rho}_1, \vec{v}_{\perp_1}, \vec{v}_{\perp_2}, \vec{p}_1 - \vec{p}_2, \tau) \qquad (4.29)$$

Equation (4.29) is a very complicated expression for the time-delayed covariance function between two points in the receiver aperture. Each term cannot be reduced to closed form expressions because the integration variables appear in the turbulence terms, making the integrations impossible to complete in closed form.

However, the first term in equation (4.29) can be simplified by using an approach[16] to expand the r-dependent terms in a Fourier-Bessel series given by

$$f(r) = \sum_m b_m J_0\left(\frac{P_m}{A} r\right) \tag{4.30}$$

where P_m are the zeroes of J_0, b_m are the coefficients given by

$$b_m = \frac{2}{A^2 J_1(P_m)} \int_0^A r f(r) J_0\left(\frac{P_m}{A} r\right) dr \tag{4.31}$$

where

$$f(r) = \int d^2\vec{r}_1 \left(\exp\left[-\frac{r_1^2}{4\alpha_0^2}\left(1 + \frac{z_2^2}{z_1^2}\right)\right]\right) \exp\left[-\left(\frac{k\alpha_0}{2}\right)^2 r_1^2 \left(\frac{1}{z_1} - \frac{1}{F}\right)^2\right]$$

$$\times \exp\left[-\left(\frac{k\alpha_0}{2}\right)^2 r_1^2 \left(\frac{1}{z_2} - \frac{1}{F}\right)^2 \left(\frac{z_2}{z_1}\right)^2\right]$$

$$\times \exp\left[-\frac{1}{2} D_\phi(\vec{r}_1, 0, 0) - \frac{1}{2} D_\phi\left(\frac{z_2}{z_1}\vec{r}_1, 0, 0\right)\right]$$

and A is a suitable upper bound where $f(r)$ has decayed to a negligible value. The expansion can be done because $f(r)$ decays rapidly as r increases so that only a few terms in the series need to be kept. In addition, no other integration variables appear in the expansion. To illustrate the method of using the expansion, the first term in the total crosscovariance $C_{I_{1,2}}(\vec{p}_1, \vec{p}_2, \tau) = C_{I_1}(\vec{p}_1, \vec{p}_2, \tau) + C_{I_2}(\vec{p}_1, \vec{p}_2, \tau)$ becomes

$$C_{I_1}(\vec{p}_1, \vec{p}_2, \tau) = \frac{2\beta^2}{k\pi}\left(\frac{kU_{T_0}\alpha_0}{2}\right)^4 \int d^2\vec{\rho} \iint dz_1 dz_2 \int d^2\vec{v}_{\perp_2} pdf(\vec{v}_{\perp_2}) \int d^2\vec{r}_1 \frac{z_2}{(z_1 z_2)^4}$$

$$\times f(r) \exp\left[-\frac{ik}{z_1}\vec{r}_1 \cdot (\vec{\rho} + \vec{v}_{\perp_2}\tau)\right]\left[\exp\left[4C_\chi(\vec{\rho} + \vec{v}_{\perp_2}\tau, \vec{p}_1 - \vec{p}_2, -\tau)\right] - 1\right] \tag{4.32}$$

Substituting equation (4.30) for $f(r)$, the r-dependent terms become

$$\int d^2\vec{r}_1 = \iint r_1 dr_1 d\theta \sum_m b_m J_0\left(\frac{P_m}{A} r_1\right) \exp\left[-\frac{ik}{z_1}\vec{r}_1 \cdot (\vec{\rho} + \vec{v}_{\perp_2}\tau)\right] \tag{4.33}$$

The θ integration can be performed to yield

$$\int d^2\vec{r}_1 = 2\pi \int r dr \sum_m b_m J_0\left(\frac{P_m}{A} r_1\right) J_0\left(\frac{k}{z_1} r_1 \left|\vec{\rho} + \vec{v}_{\perp_2}\tau\right|\right) . \tag{4.34}$$

Using[51]

$$\int_0^\infty x dx J_0(\alpha x) J_0(\beta x) = \frac{2\delta(\alpha - \beta)}{\alpha + \beta}$$

equation (4.34) becomes

$$\int d^2\vec{r}_1 = 4\pi \sum_m b_m \frac{\delta\left(\frac{k}{z_1}\left|\vec{\rho} + \vec{v}_{\perp_2}\tau\right| - \frac{P_m}{A}\right)}{\frac{k}{z_1}\left|\vec{\rho} + \vec{v}_{\perp_2}\tau\right| + \frac{P_m}{A}} \tag{4.35}$$

Substituting equation (4.35) back into equation (4.32),

$$C_{I_1}(\vec{\rho}_1, \vec{\rho}_2, -\tau) = \frac{8\beta^2}{k}\left(\frac{kU_{T_0}\alpha_0}{2}\right)^4 \iint dz_1 dz_2 \int d^2\vec{\rho} \frac{z_2}{(z_1 z_2)^4} \int d^2\vec{v}_{\perp_2} pdf(\vec{v}_{\perp_2})$$

$$\times 4\pi \sum_m b_m \left(\frac{\delta\left(\frac{k}{z_1}\left|\vec{\rho} - \vec{v}_{\perp_2}\tau\right| - \frac{P_m}{A}\right)}{\frac{k}{z_1}\left|\vec{\rho} - \vec{v}_{\perp_2}\tau\right| + \frac{P_m}{A}} \right) \left[\exp\left[4C_\chi(\vec{\rho} - \vec{v}_{\perp_2}\tau, \vec{\rho}_1 - \vec{\rho}_2, \tau)\right] - 1\right] \tag{4.36}$$

where τ has been replaced by $-\tau$. Another change of variables

$$\vec{\rho} - \vec{v}_{\perp_2}\tau = \vec{R} \qquad \vec{\rho} + \vec{v}_{\perp_2}\tau = 2\vec{Q}$$

transforms equation (4.36) to

$$C_{I_1}(\vec{p}, -\tau) = \frac{8\beta^2}{k}\left(\frac{kU_{\tau_0}\alpha_0}{2}\right)^4 \iint dz_1 dz_2 \frac{z_2}{(z_1 z_2)^4} \int d^2\vec{Q} \int d^2\vec{R}$$

$$\times \sum_m b_m \left(\frac{\delta\left(\frac{k}{z_1}R - \frac{P_m}{A}\right)}{\frac{k}{z_1}R + \frac{P_m}{A}} \right) \frac{1}{2\pi(\tau\sigma_\perp)^2} \exp\left[-\left(\frac{\vec{R} - \langle \vec{v}_\perp \rangle \tau - \frac{1}{2}\vec{Q}}{\tau\sigma_\perp} \right)^2 \right]$$

$$\times \left[\exp\left[4C_\chi(\vec{R}, \vec{p}, \tau) \right] - 1 \right] \tag{4.37}$$

where $\vec{p}$ has replaced $\vec{p}_1 - \vec{p}_2$. The $\int d^2\vec{Q}$ integrations can be performed[51] to reduce equation (4.38) to

$$C_{I_1}(\vec{p}, -\tau) = \frac{\beta^2}{2k}(kU_{\tau_0}\alpha_0)^4 \iint dz_1 dz_2 \frac{z_2}{(z_1 z_2)^4} \int d^2\vec{R} \sum_m b_m \frac{\delta\left(\frac{k}{z_1}R - \frac{P_m}{A}\right)}{\frac{k}{z_1}R + \frac{P_m}{A}}$$

$$\times \left[\exp\left[4C_\chi(\vec{R}, \vec{p}, \tau) \right] - 1 \right] \quad . \tag{4.38}$$

The $\vec{R}$ integration can be performed with the aid of the delta function to yield

$$C_{I_1}(\vec{\rho},\tau) = \frac{\beta^2 k}{2}(U_{T_0}\alpha_0)^4 \int d\theta_R \iint dz_1 dz_2 \frac{z_1}{(z_1 z_2)^3} \sum_m b_m$$

$$\times \left[\exp\left(4C_\chi \left(\frac{P_m z_1}{Ak}, \cos\theta_R, \vec{\rho}, -\tau \right) \right) - 1 \right] \tag{4.39}$$

where $-\tau$ has replaced τ. This equation is as far as the first term can be reduced without using additional approximations.

There are two interesting results concerning C_{I_1}. The first is that if there were no turbulence, i.e. $C_n^2 = 0$, C_{I_1} would be 0. This is demonstrated by setting $C_n^2 = 0$ into the expression for C_χ, equation (4.18),

$$C_\chi \left(\frac{P_m z_1}{Ak}, \cos\theta_R, \vec{\rho}_1 - \vec{\rho}_2, -\tau \right)_{C_n^2 = 0} = 0$$

and, consequently,

$$C_{I_1}(\vec{\rho},\tau) \sim \exp[4C_\chi] - 1$$

$$\sim \exp[0] - 1$$

$$\sim 1 - 1 = 0$$

The second interesting result is that there are no speckle terms in equation (4.39) which cause the first term in the total time-delayed crosscovariance function to decay. This implies that all of the speckle terms are contained in C_{I_2}, the second term in the summation of the total time-delayed crosscovariance function. It has already been shown in chapter 3 that aerosol scattering causes the speckle fluctuations to be rapidly

varying compared to the more slowly varying turbulence effects. Consequently, C_{I_2} contains all of the high frequency speckle.

The second term in the total crosscovariance can be reduced further by realizing that the aerosol speckle term

$$\exp\left[-4\left(\sigma_z k\tau\right)^2\right]$$

causes a rapid decay with respect to τ. This term is the only speckle or turbulence term that decays on the order of microseconds. One way to reduce the complexity of C_{I_2} is to realize that the τ terms within the turbulence and other speckle terms are constant over the practical range of τ compared to the aerosol speckle term, and can be simplified by setting $\tau = 0$ in all other τ terms except for the terms which contain phase information. This approximation is valid because the relative rate of decay of the aerosol speckle term is several orders of magnitude greater than that of the other speckle and turbulence terms with respect to τ. Using this approximation,

$$C_{I_2}(\vec{\rho}_1, \vec{\rho}_2, \tau) = \frac{\beta^2 (\alpha_0 k U_{T_0})^4 \exp\left[-4(\sigma_z k\tau)^2\right]}{16\pi^2} \iint d^2\vec{\rho}_1 d^2\vec{\rho}_2 \iint dz_1 dz_2 \iint d\vec{v}_{\perp_1} d\vec{v}_{\perp_2}$$

$$\times \iint d^2\vec{r}_1 d^2\vec{r}_2 \frac{pdf(\vec{v}_{\perp_1})\, pdf(\vec{v}_{\perp_2})}{(z_1 z_2)^4} \exp\left[-\frac{1}{4\alpha_0^2}(r_1^2 + r_2^2)\right] \exp\left[-\left(\frac{k\alpha_0}{2}\right)^2 \left(\frac{\vec{r}_1}{F} - \frac{\vec{r}_1}{z_1}\right)^2\right]$$

$$\times \exp\left[\frac{ik}{2z_2}(\vec{\rho}_1 - \vec{\rho}_2)^2 - \frac{ik}{2z_1}(\vec{\rho}_1 - \vec{\rho}_1)^2 + \frac{ik}{2z_1}(\vec{\rho}_2 - \vec{\rho}_1 + \vec{v}_{\perp_1}\tau)^2 - \frac{ik}{2z_2}(\vec{\rho}_2 - \vec{\rho}_2 + \vec{v}_{\perp_2}\tau)^2\right]$$

$$\times \exp\left[-\frac{ik}{2z_1}(-2\vec{\rho}_1 \cdot (\vec{r}_1 - \vec{v}_{\perp_1}\tau) + \vec{r}_1 \cdot \vec{v}_{\perp_1}\tau - v_{\perp_1}^2\tau^2)\right] \exp\left[-\frac{1}{2}D_\phi(\vec{r}_1)\right]$$

$$\times \exp\left[\frac{ik}{2z_2}(-2\vec{\rho}_2 \cdot (\vec{r}_2 - \vec{v}_{\perp_2}\tau) + \vec{r}_2 \cdot \vec{v}_{\perp_2}\tau - v_{\perp_2}^2\tau^2)\right] \exp\left[-\frac{1}{2}D_\phi(\vec{r}_2)\right]$$

$$\times \exp\left[-\left(\frac{k\alpha_0}{2}\right)^2\left(\frac{\vec{r}_2}{F}-\frac{\vec{r}_2}{z_2}\right)^2\right] H_2(\vec{\rho}_2-\vec{\rho}_1, \vec{p}_1-\vec{p}_2) \qquad (4.40)$$

The outline in completing the integrations follows the same procedure as to complete those of C_{I_1}. By making use of a Fourier-Bessel expansion on the terms containing r_1 and r_2, most of the integrations can be carried out in closed form. The result is

$$C_{I_2}(\vec{p}_1, \vec{p}_2, \tau) = \frac{\beta^2(\alpha_0 U_{T_0})^4}{8}\exp\left[-4(\sigma_z k\tau)^2\right]\iint d\theta_{R_1} d\theta_{R_2}\iint dz_1 dz_2\left(\frac{1}{z_1 z_2}\right)^2$$

$$\times \sum_l b_l \sum_n b_n \exp\left[\frac{ik}{2z_z}(p_1^2-p_2^2-2p\frac{P_l}{A}\frac{z_2}{k}\cos(\theta_{R_1}))\right]$$

$$\times \exp\left[-\frac{ik}{2z_1}(p_1^2-p_2^2-2p(\frac{P_n}{A}\frac{z_1}{k})\cos(\theta_{R_2}))\right] H_2(P_l, P_n, \vec{p}, \theta_{R_1}, \theta_{R_2}) \qquad (4.41)$$

where

$$b_l = \frac{2}{A^2 J_1^2(P_l)}\int_0^A r_1 f(r_1) J_0(\frac{P_l}{A} r_1)\, dr_1 \ ,$$

$$f(r_1) = \exp\left[-\frac{r_1^2}{4\alpha_0^2}\right]\exp\left[-\left(\frac{k\alpha_0}{2}\right)^2\left(\frac{1}{F}-\frac{1}{z_1}\right)^2 r_1^2\right]\exp\left[-\frac{1}{2}D_\phi(r_1)\right] \ ,$$

$$b_n = \frac{2}{A^2 J_1^2(P_n)}\int_0^A r_2 f(r_2) J_0(\frac{P_n}{A} r_2)\, dr_2 \text{, and}$$

$$f(r_2) = \exp\left[-\frac{r_2^2}{4\alpha_0^2}\right]\exp\left[-\left(\frac{k\alpha_0}{2}\right)^2\left(\frac{1}{F}-\frac{1}{z_2}\right)^2 r_2^2\right]\exp\left[-\frac{1}{2}D_\phi(r_2)\right] .$$

Summing equations (4.41) and (4.39) gives the total time-delayed crosscovariance of intensities between two points in the receiver aperture

$$C_{I_{1,2}}(\vec{\rho},\tau) = \frac{\beta^2 k}{2}(U_{T_0}\alpha_0)^4 \int d\theta_R \iint dz_1 dz_2 \frac{z_1}{(z_1 z_2)^3} \sum_m b_m$$

$$\times \left[\exp\left(4C_\chi \left(\frac{P_m z_1}{Ak}, \cos\theta_R, \vec{\rho}, -\tau \right) \right) - 1 \right]$$

$$+$$

$$\frac{\beta^2 (\alpha_0 U_{T_0})^4}{8} \exp\left[-4(\sigma_z k\tau)^2\right] \iint d\theta_{R_1} d\theta_{R_2} \iint dz_1 dz_2 \left(\frac{1}{z_1 z_2} \right)^2$$

$$\times \sum_l b_l \sum_n b_n \exp\left[\frac{ik}{2z_z} \left(p_1^2 - p_2^2 - 2p \frac{P_1}{A} \frac{z_2}{k} \cos(\theta_{R_1}) \right) \right]$$

$$\times \exp\left[-\frac{ik}{2z_1} \left(p_1^2 - p_2^2 - 2p \left(\frac{P_n}{A} \frac{z_1}{k} \right) \cos(\theta_{R_2}) \right) \right] H_2(P_l, P_n, \vec{\rho}, \theta_{R_1}, \theta_{R_2}) \quad . \quad (4.42)$$

Unfortunately, equation (4.42) is dominated by the aerosol speckle term $\exp\left[-4(\sigma_z k\tau)^2\right]$ making it difficult to extract an expression for the crosswind velocity. However, the next section introduces a low-pass filtering process which removes C_{I_2}, making it easier to develop a crosswind detection scheme.

4.2 Low-Pass Filtered, Time-Delayed Statistics

Figure 4.3 illustrates the differences between the two terms in the time-delayed crosscovariance. Each term is plotted to show how each term decorrelates with time. Both terms are normalized by the average intensities to produce a comparison on the same relative scale. The radial wind speed, $<v_z>$ is chosen to be 2m/s, the crosswind speed, $<v_\perp>$, is chosen to be 2m/s, the turbulence strength C_n^2 is 10^{-14}, and the distance to the center of the range bin is 500 meters. The first term, C_{I_1}, contains only effects from turbulence and decorrelates on the scale of milliseconds. The second term, however, decorrelates rapidly within a few microseconds. Another difference is that the peak of the first term is shifted from the origin depending on the crosswind speed but the second term is not shifted because the speckle term

$$\exp\left[-4\left(k\sigma_z\tau\right)^2\right]$$

decorrelates so rapidly that the shift is never seen.

Since the two terms in the time-delayed crosscovariance occupy such different spectra, perhaps the application of some kind of time filter would remove the high frequency speckle term, C_{I_2}, from equation (4.37) and simplify the time-delayed crosscovariance. The filter must be chosen carefully so that the high frequency speckle is removed but the lower frequency turbulence is left intact. In other words, the cutoff frequency should be higher than the turbulence fluctuations but lower than the aerosol speckle fluctuations. The choice is easy since the spectra do not overlap. In the

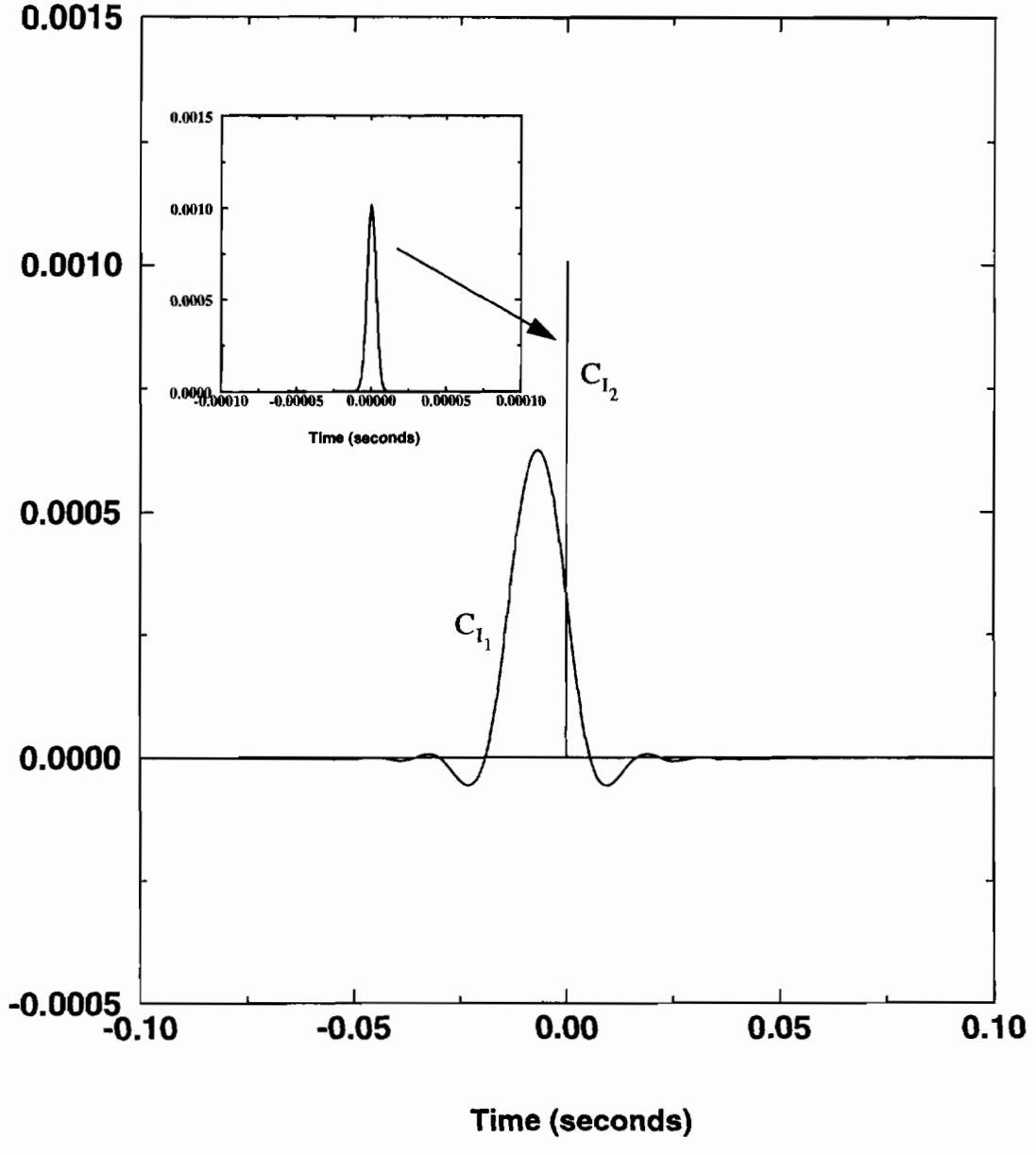
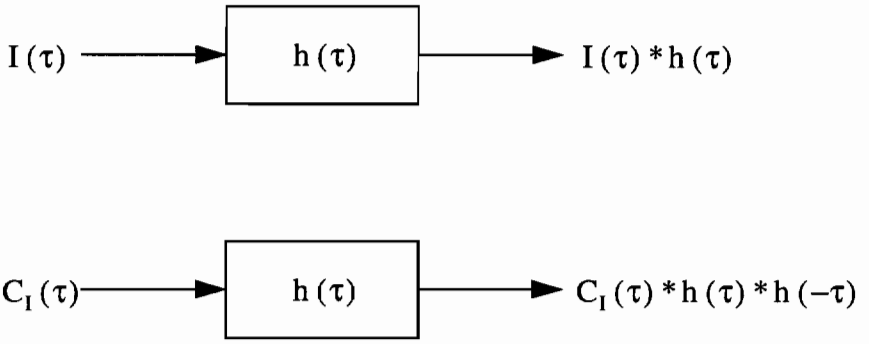
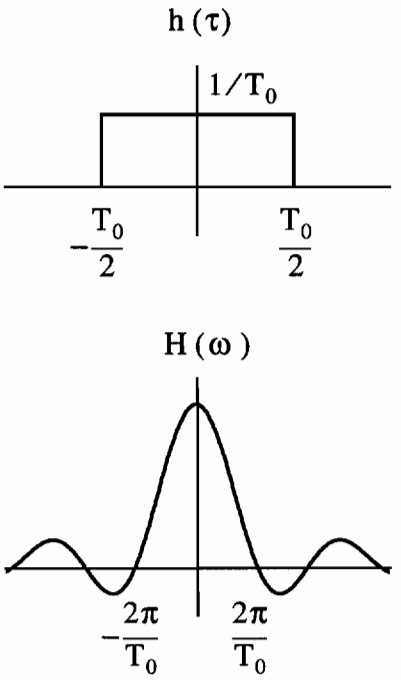


Figure 4.2 Theoretical plots of the two terms in the time-delayed crosscovariance of intensities. The first term, which contains only turbulence terms, decorrelates on the order of milliseconds. The second term, which contains a combination of turbulence and high frequency speckle, decorrelates on the order of microseconds.

actual processing scheme of the experimental data, the filtering will be performed on the intensities before computing the crosscovariance. Because the filtering process is linear, linear theory can be used to relate the filtering process to the crosscovariance itself. If $I(\tau)$ represents the intensity, $h(\tau)$ the filter response, and $C_I(\tau)$ the cross-covariance,[54]



The filter response function is a low pass type of filter.



The choice of T_0 defines the cutoff frequency. Applying the filter to the time-delayed crosscovariance,

$$C_{I_{1,2}}(\vec{p}, \vec{Q}, \tau) * h(\tau) * h(-\tau) = \left[C_{I_1}(\vec{p}, \tau) + C_{I_2}(\vec{p}, \vec{Q}, \tau) \right] * h(\tau) * h(-\tau) \quad .$$

Concentrating on C_{I_1},

$$C_{I_1}(\vec{p}, \tau) * h(\tau) = \int_{-\infty}^{\infty} h(\gamma)\, C_{I_1}(\vec{p}, \tau - \gamma)\, d\gamma \quad .$$

$$= \int_{-T_0/2}^{T_0/2} h(\gamma)\, C_{I_1}(\vec{p}, \tau - \gamma)\, d\gamma$$

The integration time, T_0, is less than the decorrelation time of C_{I_1}. Therefore, C_{I_1} is a constant over the integration range and can be taken out of the integral at $\gamma = 0$. The integrals become

$$C_{I_1}(\vec{p}, \tau) * h(\tau) * h(-\tau) = C_{I_1}(\vec{p}, \tau)$$

which implies that the first term passes through the filter unaffected because its spectrum lies below the cutoff frequency of the filter.

The filter cannot be applied to the second term in the form of equation (4.41) because τ must be kept in the turbulence terms before time filtering. However, the integrals in the complete expression for C_{I_2}, equation (4.29), are impossible to complete. Since the goal is to show that the second term becomes negligible as compared to first term after time filtering, the filter will be applied to the second term with $C_n^2 = 0$. With no turbulence, only the speckle terms remain which serves as an upper

bound to C_{I_2}. In other words, showing that the time filter causes C_{I_2} to be negligible with $C_n^2 = 0$ is more strict than if turbulence were included because turbulence only causes C_{I_2} to decay faster. Setting $C_n^2 = 0$ in the second term of equation (4.29) yields

$$C_{I_2}(\vec{\rho}_1, \vec{\rho}_2, \tau)\Big|_{C_n^2 = 0} = C_{IS_2}(\vec{\rho}_1, \vec{\rho}_2, \tau) =$$

$$\frac{\beta^2 (\alpha_0 k U_{T_0})^4 \exp\left[-4(\sigma_z k\tau)^2\right]}{16\pi^2} \iint d^2\vec{\rho}_1 d^2\vec{\rho}_2 \iint dz_1 dz_2 \iint d\vec{v}_{\perp_1} d\vec{v}_{\perp_2}$$

$$\times \iint d^2\vec{r}_1 d^2\vec{r}_2 \frac{pdf(\vec{v}_{\perp_1})\, pdf(\vec{v}_{\perp_2})}{(z_1 z_2)^4} \exp\left[-\frac{1}{4\alpha_0^2}(r_1^2 + r_2^2)\right] \exp\left[-\left(\frac{k\alpha_0}{2}\right)^2 \left(\frac{\vec{r}_1}{F} - \frac{\vec{r}_1}{z_1} + \frac{\vec{v}_{\perp_1}\tau}{z_1}\right)^2\right]$$

$$\times \exp\left[\frac{ik}{2z_2}(\vec{\rho}_1 - \vec{\rho}_2)^2 - \frac{ik}{2z_1}(\vec{\rho}_1 - \vec{\rho}_1)^2 + \frac{ik}{2z_1}(\vec{\rho}_2 - \vec{\rho}_1 + \vec{v}_{\perp_1}\tau)^2 - \frac{ik}{2z_2}(\vec{\rho}_2 - \vec{\rho}_2 + \vec{v}_{\perp_2}\tau)^2\right]$$

$$\times \exp\left[-\frac{ik}{2z_1}(-2\vec{\rho}_1 \cdot (\vec{r}_1 - \vec{v}_{\perp_1}\tau) + \vec{r}_1 \cdot \vec{v}_{\perp_1}\tau - v_{\perp_1}^2\tau^2)\right]$$

$$\times \exp\left[\frac{ik}{2z_2}(-2\vec{\rho}_2 \cdot (\vec{r}_2 - \vec{v}_{\perp_2}\tau) + \vec{r}_2 \cdot \vec{v}_{\perp_2}\tau - v_{\perp_2}^2\tau^2)\right]$$

$$\times \exp\left[-\left(\frac{k\alpha_0}{2}\right)^2 \left(\frac{\vec{r}_2}{F} - \frac{\vec{r}_2}{z_2} + \frac{\vec{v}_{\perp_2}\tau}{z_2}\right)^2\right] \tag{4.43}$$

which contains all of the speckle terms but no turbulence. The $\vec{r}$, $\vec{\rho}$, and $\vec{v}_\perp$ integrals can be completed to give

$$C_{IS_2}(\vec{\rho}, \vec{Q}, \tau) = \frac{\beta^2\pi^2(\alpha_0 U_{T_0})^4 \exp[-4(\sigma_z k\tau)^2]}{4} \iint dz_1 dz_2 \frac{1}{(z_1 z_2)^2}$$

$$\times \exp\left[-\frac{1}{4\alpha_0^2}(\vec{\rho} + 2<\vec{v}_{\perp_1}>\tau)^2\right] \exp\left[-\frac{1}{4\alpha_0^2}(\vec{\rho} + 2<\vec{v}_{\perp_2}>\tau)^2\right]$$

$$\times \exp\left[-\frac{1}{4}\left(\frac{k\alpha_0}{z_1}\right)^2 \left[(1-\frac{z_1}{F})\vec{\rho} + (1-\frac{2z_1}{F})<\vec{v}_{\perp_1}>\tau\right]\right] \exp\left[-\frac{ik}{z_1}\vec{Q}\cdot(\vec{\rho} + <\vec{v}_{\perp_1}>\tau)\right]$$

$$\times \exp\left[-\frac{1}{4}\left(\frac{k\alpha_0}{z_2}\right)^2 \left[(1-\frac{z_2}{F})\vec{\rho} + (1-\frac{2z_2}{F})<\vec{v}_{\perp_2}>\tau\right]\right] \exp\left[-\frac{ik}{z_2}\vec{Q}\cdot(\vec{\rho} + <\vec{v}_{\perp_2}>\tau)\right]. \quad (4.44)$$

Applying the filter to C_{IS_2}, all factors containing τ are nearly constant over the integration range except the aerosol speckle term. Therefore,

$$C_{I_2}(\vec{\rho}, \vec{Q}, \tau) * h(\tau) = f(\tau) \int_{-T_0/2}^{T_0/2} h(\gamma)\, e^{-4(k\sigma_z)^2(\tau-\gamma)^2} d\gamma$$

$$\approx \frac{f(\tau)}{k\sigma_z}$$

where $f(\tau)$ represents the factors of C_{I_2} that are constant or slowly varying functions of τ. The remaining integral is trivial

$$C_{I_2}(\vec{\rho}, \vec{Q}, \tau) * h(\tau) * h(-\tau) = \frac{f(\tau)}{k\sigma_z}.$$

The effect of the low pass filtering is to introduce the scale factor $\frac{1}{k\sigma_z}$ on the second term of the crosscovariance. For typical z-directed wind speeds of around 2m/s, this scale factor causes the second term to be negligible compared to the first term. Therefore, applying the proper low-pass filter can successfully remove the high frequency aerosol speckle, simplifying the expression for the time-delayed crosscovariance of the intensities between two point detectors to

$$C_{I_{lpf}}(\vec{\rho},\tau) \approx C_{I_1}(\vec{\rho},\tau) = \frac{\beta^2 k}{2}(U_{T_0}\alpha_0)^4 \int d\theta_R \iint dz_1 dz_2 \frac{z_1}{(z_1 z_2)^3} \sum_m b_m$$

$$\times \left[\exp\left(4C_\chi\left(\frac{P_m z_1}{Ak}, \cos\theta_R, \vec{\rho}, -\tau \right) \right) - 1 \right] \qquad (4.45)$$

where the subscript lpf stands for "low pass filtered." The only remaining integrals are those concerning θ_R, and z_1 and z_2. Pulsed lidars confine the range to a certain region by range gating the receiver to confine the backscattered radiation to a certain region. In this project, the outgoing cw laser beam is pseudo-random code modulated so that the backscattered laser intensity is limited to a certain region. The details of recovering the backscattered laser intensity only from one region and rejecting other regions are discussed in chapter 5. Because of the range discrimination of the lidar, the limits of integration on z are z_a to z_b, where $z_a - z_b$ is the z-dimension of the scattering volume. The distance $z_a - z_b$ is called a range bin. Figure 4.3 shows several plots of equation (4.45) for several different path lengths. The time-filtered, time-delayed crosscovariance of intensities backscattered from aerosols only depends on the log amplitude turbulence term. No phase perturbation turbulence terms exist in the time-filtered result, nor are there any speckle terms. This implies that the major effect of time-filtering the statistics of a coherent laser source is that the laser source acts as

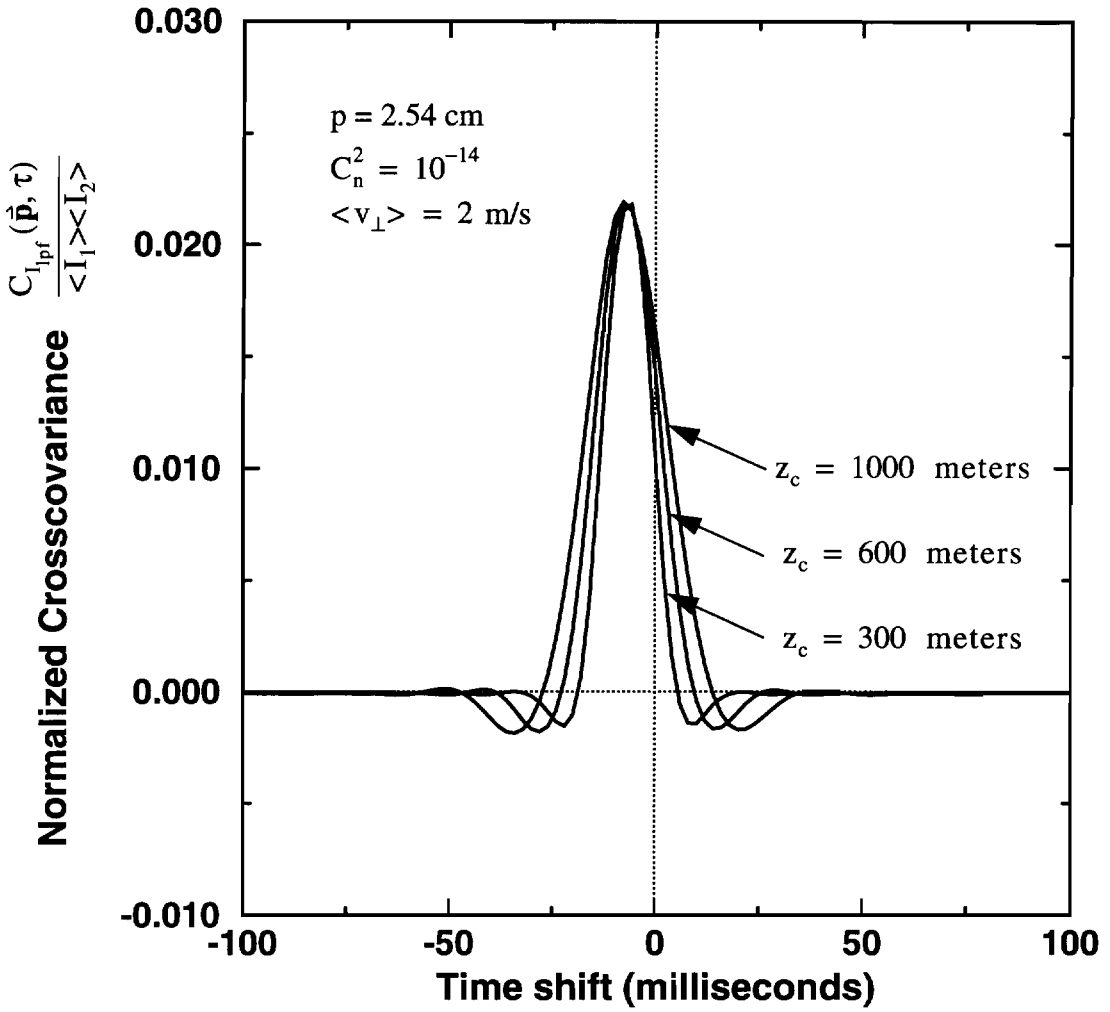


Figure 4.3 Low-pass filtered, time-delayed crosscovariance of intensities backscattered from aerosols. The time shift depends on the crosswind speed and the detector spacing. The width of the crosscovariance widens as the distance to the center of the aerosol backscatterers increases.

if it were incoherent. In fact, equation (4.39) is similar to the solution to the time-delayed crosscovariance of intensities for an incoherent laser source and direct detection.[55,56,57] This result is only achievable using the appropriate processing scheme, one which successfully removes the high frequency speckle term but leaves the lower frequency turbulence term intact.

This chapter developed the time-delayed crosscovariance of the intensities of a coherent, CO_2 heterodyne lidar backscattered from aerosols which becomes the sum of two terms with drastically different decorrelation times. The first term is a function of turbulence induced fluctuations dominated by the log-amplitude covariance function with a decorrelation time on the order of milliseconds. The seconds term is dominated by a high-frequency aerosol speckle term which causes the second term to decorrelate on the order of microseconds. The total time-delayed crosscovariance, therefore, is a complicated function of fluctuations induced by speckle and turbulence. However, the application of a low-pass type of filter can remove the second high-frequency term from the analysis, reducing the crosscovariance to a result which only depends on turbulence. In fact, the low-pass filtered crosscovariance for a coherent, cw, CO_2 heterodyne lidar backscattered from aerosols resembles the one for an incoherent lidar with direct detection, simplifying the analysis.

Chapter 5

The Experimental Lidar System

The most common lidars used to probe information about the atmosphere at $\lambda = 10.6\mu m$ are pulsed laser systems. A high powered pulse is transmitted for a very short time (on the scale of micro seconds) to the atmosphere and the pulse backscatters off of aerosol particles and travels back to the receiver. To control the distance to the desired scattering region, the receiver is range gated. The width of the scattering region, or range bin, is determined by the amount of time the receiver gate is open. Although the laser pulse is on for a short time, the peak power is so high that the maximum range detectable is longer than that detected by continuous wave (cw) systems. For example, the Royal Signals and Radar Establishment (RSRE) lidar, a 1.5 Watt, cw, 10.6 micron system used to measure backscattering coefficient profiles (β) in the early 1980's, had a range of 100 meters to the focus. However, the Wave Propagation Laboratory (WPL) lidar, a 2-microsecond duration, .1 Joule, pulsed, 10.6 micron system used to compare β values with the RSRE system, had a maximum range of over 10 km.[58] Because the receiver is range gated, no crosstalk from other range bins can enter the receiver as opposed to cw systems. However, these advantages over cw lidars are not without significant drawbacks. The range gating of the receiver necessitates some complicated electronics which makes the processing difficult. In addition, the pulsed nature of the laser causes some laser instability as compared to the cw laser systems. The pulsed requirement of the laser also makes these systems more expensive, larger, and less reliable than cw systems. The most desirable

lidar system would be one which adopted the range resolution ability of the pulsed systems while retaining the simplicity of the cw systems.

The lidar developed over the past 3 years from 1991 to 1994 at the Oregon Graduate Institute used in this thesis project is a pseudo-random code, cw, coherent optical heterodyne system operating at $\lambda = 10.6\mu m$. Because the laser operates cw, the laser is simpler, more reliable, and less expensive than a pulsed laser. What separates this cw lidar system apart from others is the novel implementation of a pseudo-random code on the outgoing laser beam to discriminate between ranges. In fact, the use of a pseudo random code for range resolution has been achieved for the first time using a cw, optical heterodyne system. Optical heterodyne detection is used to increase the sensitivity of the lidar to detect weak backscattered signals on the order of 10^{-15} Watts. The maximum output power transmitted to the atmosphere is only about 3 Watts. Section 5.1 gives a complete description of the lidar, and how the pseudo random modulation is employed. Section 5.2 discusses the system signal to noise ratio (SNR). Finally, Section 5.3 introduces the processing schemes to get the 3-dimensional wind speed.

5.1 Pseudo-Random Code, CW, CO_2 Lidar

Three representations of the CO_2 heterodyne system are presented in figures 5.1a, 5.1b, and 5.1c. The first figure, figure 5.1a, is a general schematic which shows the basic functionality of the system. Figure 5.1b illustrates a more detailed top view of the layout of the system, introducing each optical component and its relative position on the optical table. Each element is labeled and briefly described in table 5.1. The system has been designed so that the laser beam has a height of 2" above the optical table. However, the use of an Odyssey 8" diameter, 36" focal length telescope as the transmitting mirror places part of the laser beam 8" above the optical table. Figure 5.1c is a 3-D representation of this critical part of the system.

The lidar has been designed, constructed, and tested over a period of three years from 1991 to 1994. The main idea has been to modify the existing heterodyne system,[18] designed to measure the path averaged crosswind speed between the laser/ receiver and a hard target, to measure the path resolved 3-D wind speed using aerosols as the backscatterer. The first design change was to construct a coaxial system. The old laser system, a biaxial lidar which aligned the receiving mirrors at one fixed distance to the hard target, was not designed to detect backscattering from different parts of the path. Since backscattering from aerosols could occur anywhere, a coaxial system was better suited for aerosol backscattering. Another design change was to diphase modulate the outgoing laser beam with a pseudo-random code. Without this step, range resolution was impossible. Because aerosols backscatter much less laser radiation than hard targets, the old 5 Watt CO_2 laser was replaced with a 10 Watt updated model to provide twice the radiation to the atmosphere as the old system. Since theoretical as well as experimental evidence suggested that the majority of the backscattering occurred near the range where the laser was focussed, the transmitter

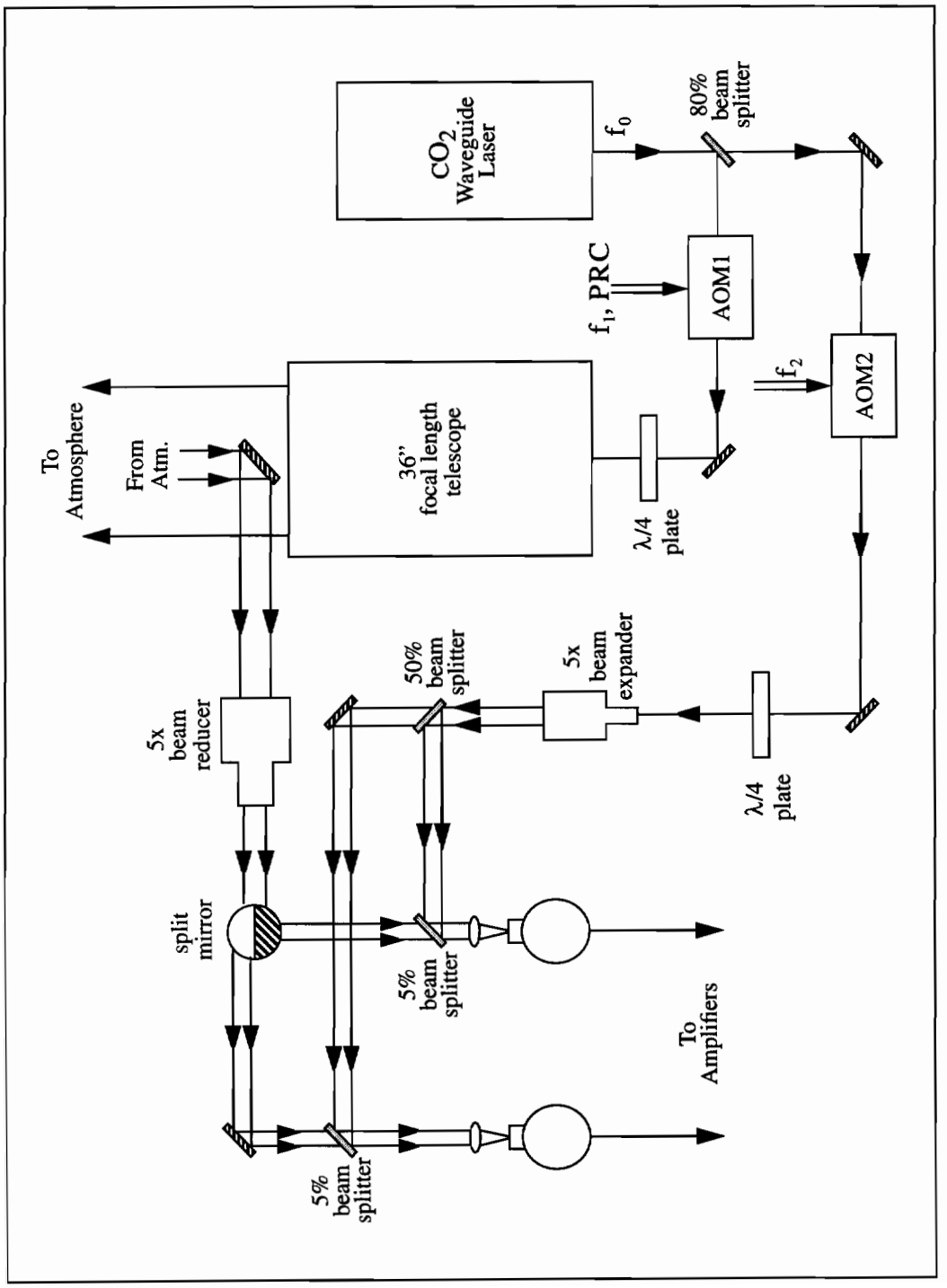


Figure 5.1a Schematic of CO_2 heterodyne system.

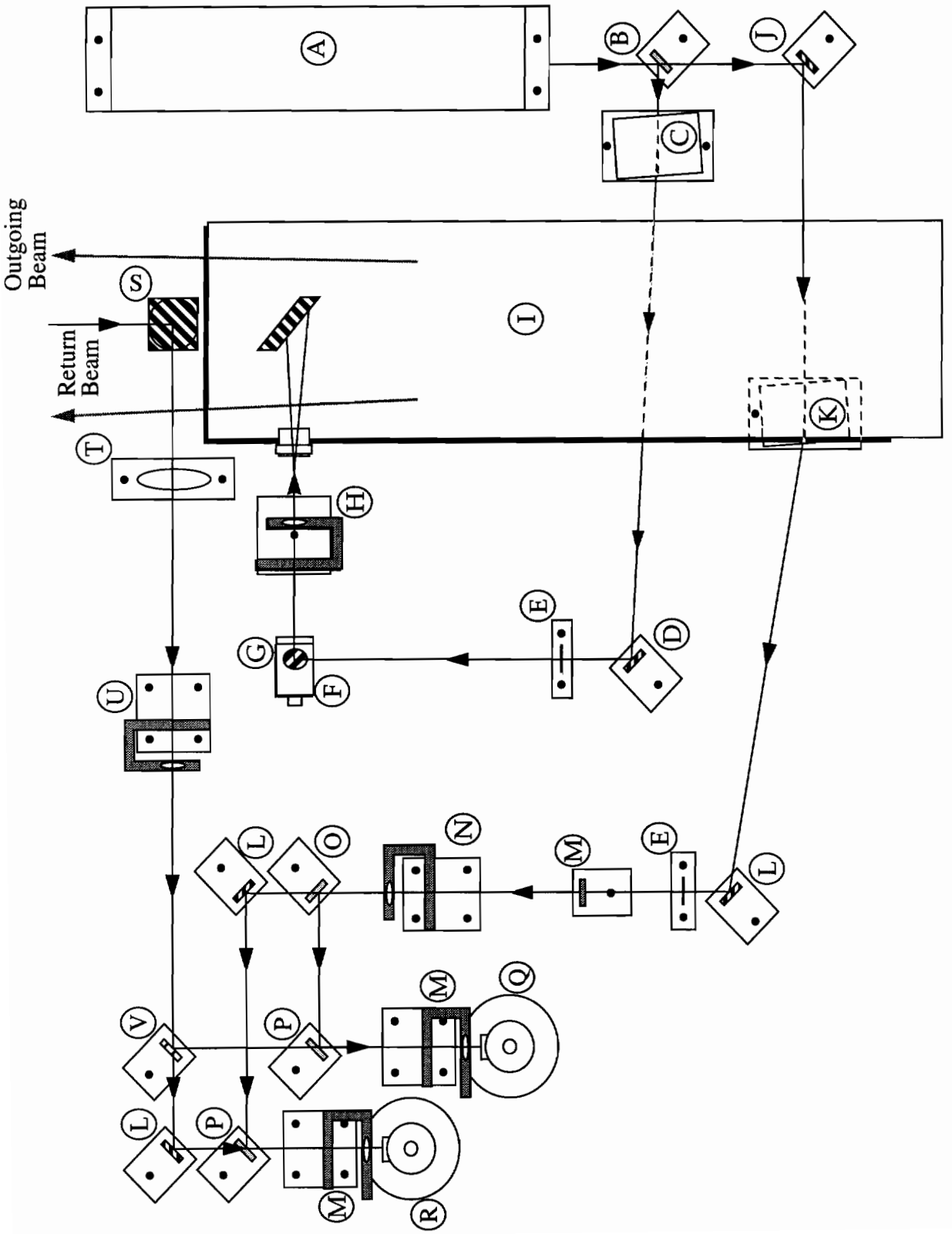


Figure 5.1b Top view of the layout of the CO_2 optical heterodyne system.

Table 5.1: Description of the Optical Components of the CO_2 Heterodyne Lidar

Optical Element	Description/Purpose
A	Synrad 48-1-28V CO_2 Laser, P_{out} = 10 Watts, $f_0 = c/10.6e\text{-}06$
B	Beam Splitter, ~80% to Atmosphere, ~20% to LO
C	AOM 1 => $f_0 + f_1$ (f_1 = 37.5 MHz)
D	Mirror
E	Quarter Wave Plate
F	Mirror (Horizontal to 6" Vertical)
G	Mirror (6" Vertical to Horizontal)
H	1.5" focal length lens F/3
I	Odyssey 8" diameter, 36" focal length telescope
J	Mirror
K	AOM 2 => $f_0 + f_2$ (f_2 = 43.5 MHz)
L	Mirror
M	1" focal length lens F/1
N	5" focal length lens F/5
O	Beam Splitter, ~50% deflected ~50% passed through
P	Beam Splitter, ~5% deflected ~95% passed through
Q	Fermionics #PV-K100 detector, 200x200 micron square surface
R	NERC #MPV11-.2-B50 detector, 200X200 micron square
S	2 Flat Receiving Mirrors - 3" major x 2" minor eliptical mirror on axis with telescope and 3" square mirror underneath
T	3" diameter, 15" focal length lens F/5
U	-3" focal lenght lens F/-3
V	Horizontally-split Lower-half Silvered Mirror (Half Moon)

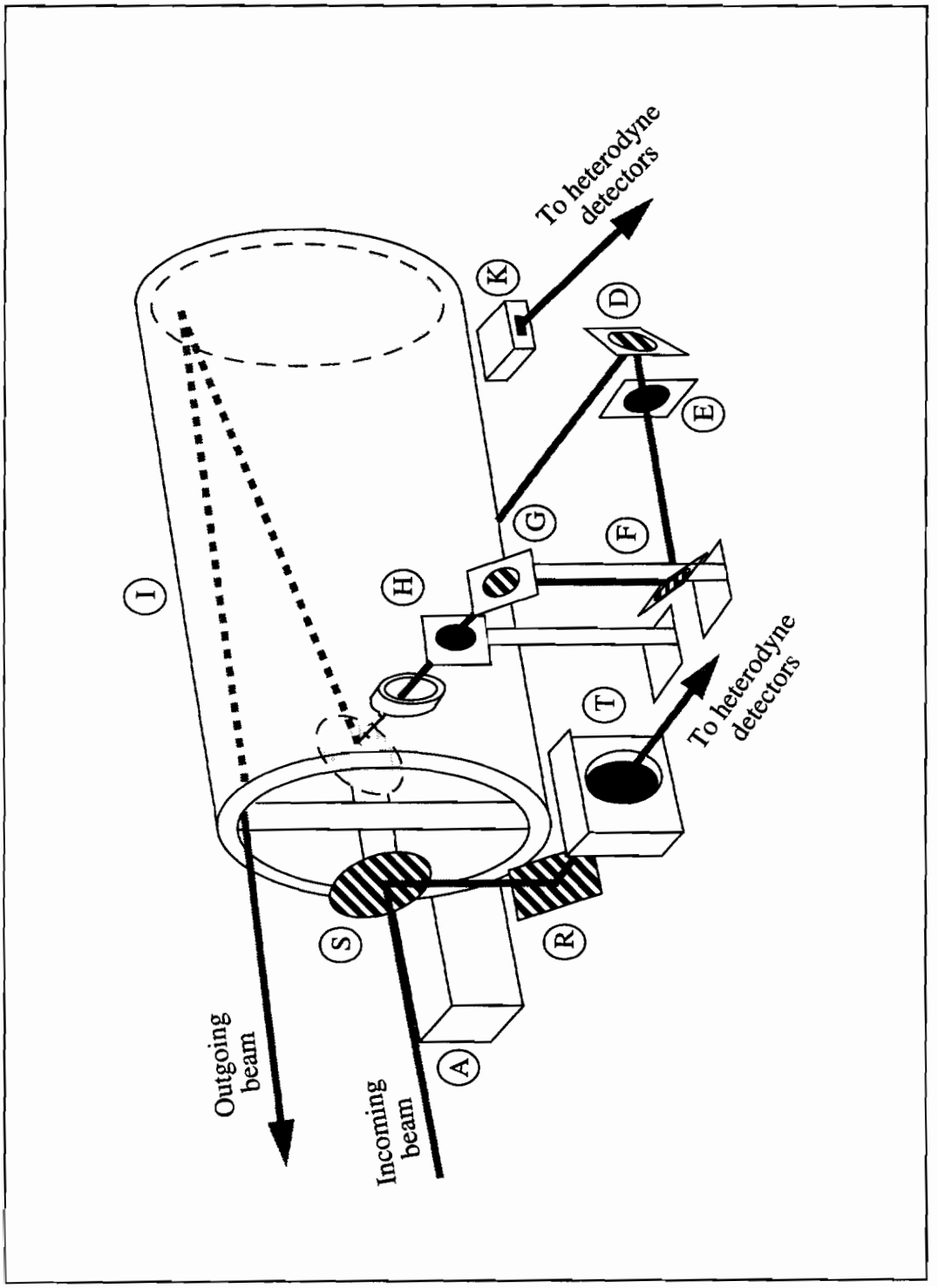


Figure 5.1c 3-D view of the transmitter/receiver part of the heterodyne system.

optics were completely redesigned to extend the maximum focus distance by a factor of 4 over the old system. The next design change in detecting weak backscattering from aerosols was to increase the receiving mirror size to maximize the system SNR. The effective receiving area needed to be chosen carefully because a heterodyne system detects the electric field of the backscattered laser beam, not the intensity. Therefore, a receiving area which was too large would have caused aperture averaging, reducing the SNR. Finally, the amplifier/receiver part of the system was completely redesigned and constructed from the amplifiers to the signal processing and storage equipment.

The schematic of the CO_2 heterodyne system, figure 5.1a, illustrates the basic characteristics of the lidar. The heart of the lidar is the Synrad CO_2 laser which has a beam radius of 1.85mm at the exit aperture and a cw output power of 10 Watts. Most of the power from the laser is directed to the transmitted path by an 80% germanium beam splitter. The beam passes through the first acoustooptic modulator, AOM1, which performs two functions. Not only does AOM1 upshift the frequency of the laser by 37.5 MHz, it also diphase modulates the beam as it continues along the transmitter path. The beam passes through a $\lambda/4$ phase retarder which circularly polarizes the outgoing beam. This step is necessary because irregularly shaped aerosols (not perfectly spherical) induce a polarization change to a linearly polarized beam[33], causing fluctuations in intensity and phase. However, irregularly shaped aerosols do not cause a polarization change for a circularly polarized beam in the backscattered direction.[34]

The beam is then directed to an Odyssey 8" diameter, 36" focal length telescope which expands the beam radius to 71.7mm (2.82 inches). The existence of the internal turning mirror shown in figures 5.1b and 5.1c blocks part of the outgoing beam. A computer program was written to investigate whether the internal turning mirror (or spider) blocks too much transmitted power. Figure 5.2 shows the ratio of

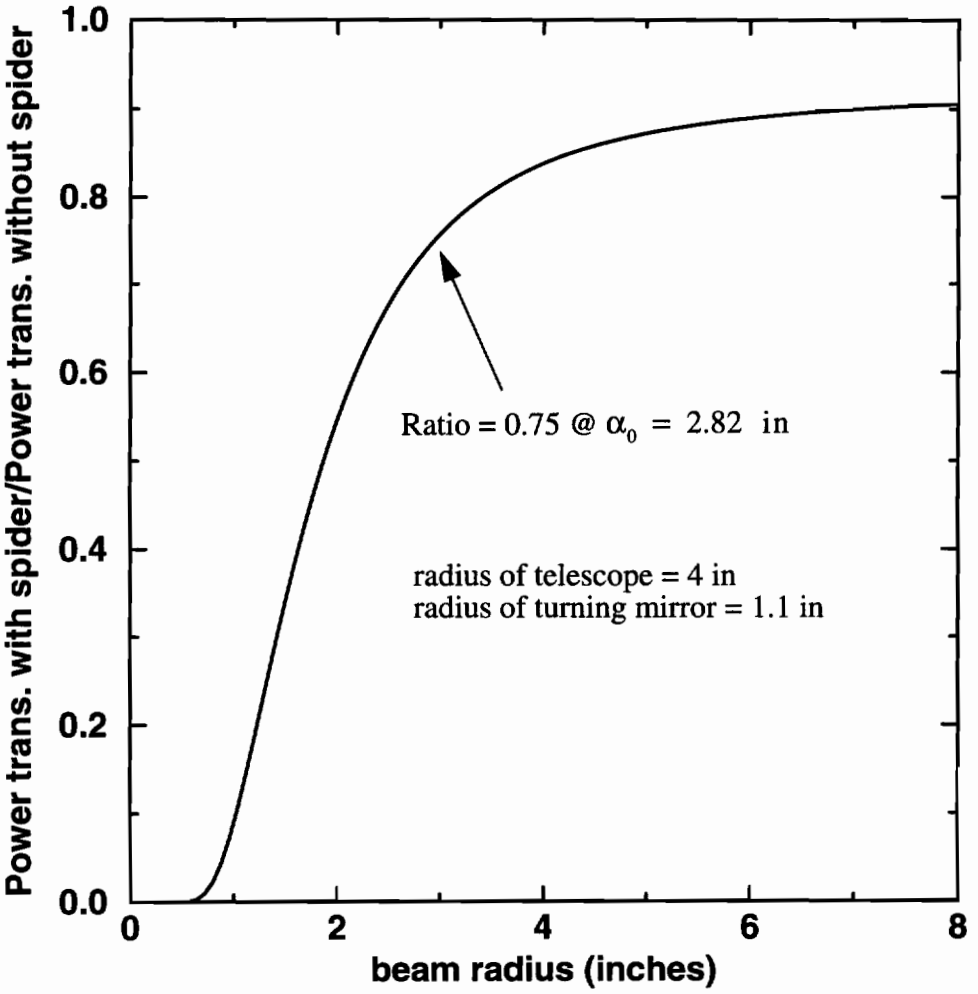


Figure 5.2 Ratio of power transmitted with internal turning mirror to the power transmitted without turning mirror for different beam radii. For a beam radius of 2.82 inches, about 75% passes through.

transmitted power with the spider blocking the center of the telescope to the power transmitted without the spider for a Gaussian laser beam with different beam radii. For a beam radius of 2.82 inches used in the actual system, about 75% of the power is transmitted. Therefore, enough power is transmitted without too much sacrifice.

Part of the beam that is not directed to the transmitter part of the optics is reserved for the local oscillator (LO). The LO beam is directed through the second acoustooptic modulator, AOM2, and is upshifted by a frequency of 43.5 MHz. The purpose of the second modulator is to provide better optical isolation between the transmitter and LO paths,[59] and to create a difference frequency between the transmitter beam and the LO beam of 6 MHz. After the second AOM, the LO beam is directed through another $\lambda/4$ circular polarizer to match that of the backscattered beam. The local oscillator beam is then expanded by a 5X beam expander for two reasons. The first is to prevent the beam from rapidly diverging, and the second is to overfill the 1 inch diameter lenses to match the wavefront of the backscattered beam. After propagating several hundred meters from the scattering location, the backscattered beam is nearly a plane wave. Therefore, the LO beam radius is expanded to match the incoming beam wavefront as closely as possible. Although the LO beam intensity distribution is Gaussian, overfilling the 1 inch lenses makes the beam shape more plane.

The backscattered laser beam is collected by a 2 inch diameter mirror mounted at the center of the transmitting telescope making the lidar coaxial. Because only one receiving mirror is used, some method of splitting the incoming beam into two separate fields is essential to be able to compute the transverse spatial and temporal statistics. Figure 5.3 shows how the beam is separated into two equal fields by a split mirror. The equivalent aperture area for each detector becomes a half moon, with a radius of 1/2 the diameter of the receiving mirror, or 1". After the split mirror, each

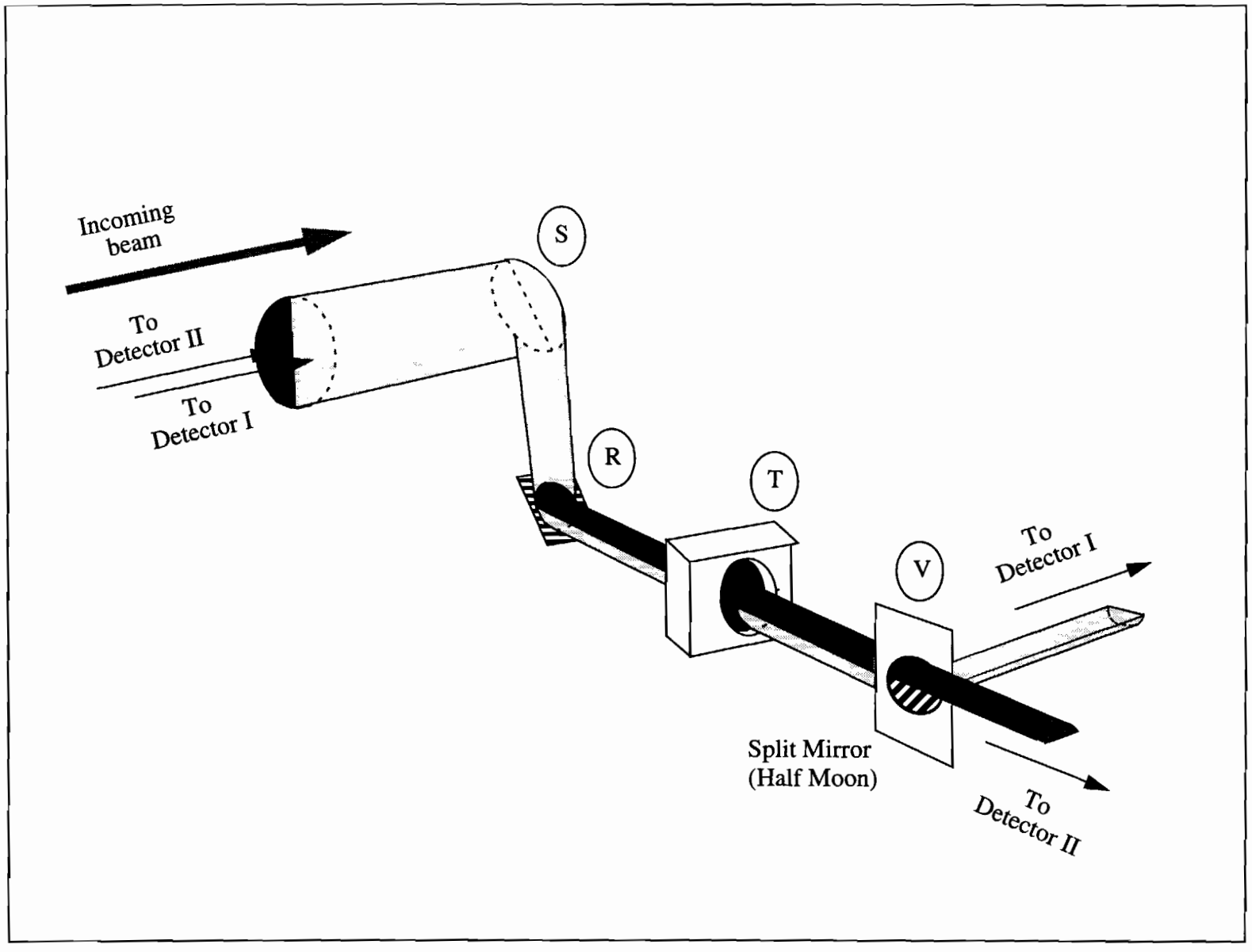


Figure 5.3 The split mirror (half moon) separates the incoming beam into two equal portions.

backscattered field is combined with the local oscillator electric field by a 5% beam splitter and the resultant field is focussed onto a .2mm X .2mm square HgDcTe detector by a 1" focal length lens. The detectors output a diode current proportional to the applied optical power on the detector surface area. The detectors represent the last stage of the optical part of the lidar.

The signal detection, recording and storage equipment are shown schematically in figure 5.4. After the current is generated by the photodiodes, the diode current is collected by a transimpedance amplifier that converts the current to a voltage. The voltage signal is then amplified further to match the full scale digitization range of the A/D board. Each channel is digitized at a sampling rate of 20MHz by a CS220 two-channel digitizing board which is fully software controlled. The CS220 resides in two of the expansion slots of a Gateway 2000 4DX 486 personal computer. The CS220 can store up to 4 Megabytes of continuous data samples in its own memory before samples are overwritten. The data is transferred to the hard drive of the 486 PC before being stored permanently on the Exabyte 8500 tape drive. The Exabyte 8500 uses 8mm cam-corder magnetic tapes that can store up to 5 Gigabytes of data. The data can be transferred from the CS220 to the PC's hard drive at 1 Mbyte per second, but the transfer rate from hard drive to Exabyte tape drive is only 250 Kbytes per second. The data acquisition and storage programs are listed in Appendix B.

5.1.1 Review of Pseudo-Random Codes

To understand how the pseudo-random code modulation can be used to determine the range at which the backscattering occurs, and therefore the intensity vs. range, a review of the most important properties of pseudo-random sequences is helpful. A more complete listing of the properties of pseudo-random sequences and arrays

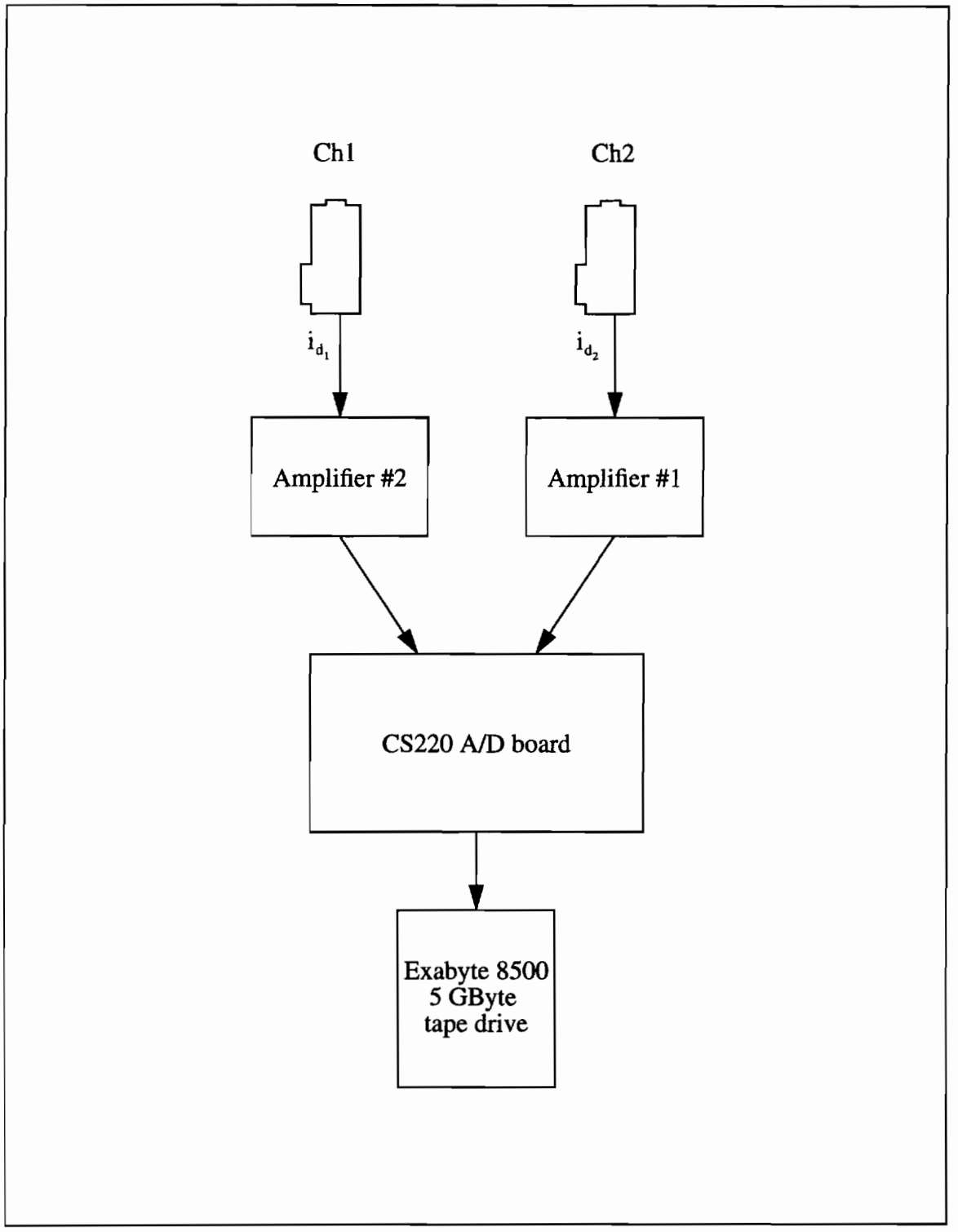


Figure 5.4 Component layout diagram of the signal amplification, detection and storage equipment.

is given by MacWilliams and Sloane.[60] Pseudo-random sequences are periodic binary sequences of length $M = 2^n - 1$, where n is any positive integer. The most important property of such a sequence is its periodic autocorrelation function. Figure 5.5 shows a PRC of length 7 (n=3) and its autocorrelation function. The PRC takes on the values of 1 or -1 according to the characteristic polynomial for the specific sequence. Therefore, the PRC can be generated by feedback shift registers for any $M = 2^n - 1$ provided the polynomial is known. Reference [60] contains a complete list of polynomials for degrees n up to 40. The properties of the PRC sequences can be demonstrated by converting the sequence to a time function shown in figure 5.5. With a minimum pulse width of T_0, called the gate width, the autocorrelation function becomes

$$R_{PRC}(\tau) = 1 - \frac{|\tau|}{T_0} \tag{5.1}$$

for $|\tau| \leq T_0$ and $\frac{1}{M}$ elsewhere. A longer length code may be used to reduce the contribution to the autocorrelation function away from $|\tau| = T_0$.

5.1.2 PRC Modulation of Direct Detection Systems

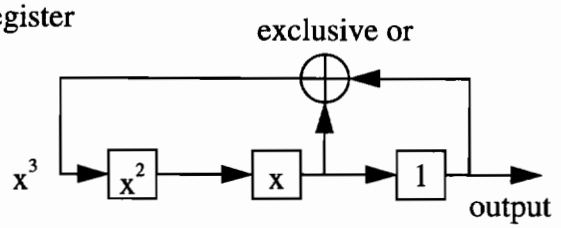
The easiest way to incorporate the PRC with a lidar system is to use an on/off modulation scheme for a direct detection system.[61,62,63] For direct detection systems, the laser is simply turned on and off according to the PRC sequence. To demonstrate the method of using a PRC sequence with a direct detection cw lidar backscattered from aerosols, the outgoing laser intensity is

$$I(t) = I_0 PRC(t) \tag{5.2}$$

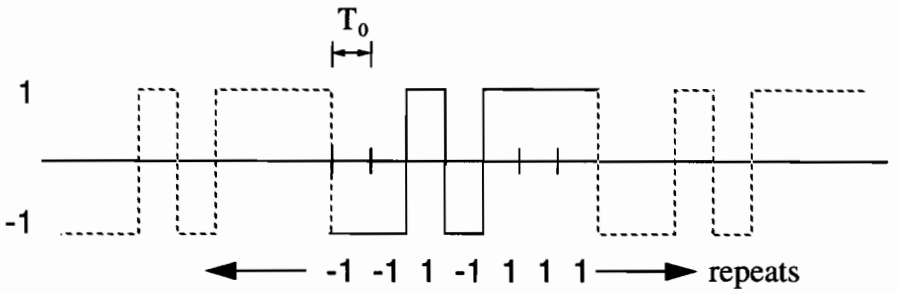
- 7-state PRC: 0 0 1 0 1 1 1

- Characteristic polynomial: $h(x) = x^3 + x + 1$

- Feedback shift register for M = 7:



- 7-State PRC converted to time function:



- Autocorrelation $\langle PRC(t)\, PRC(t+\tau) \rangle$:

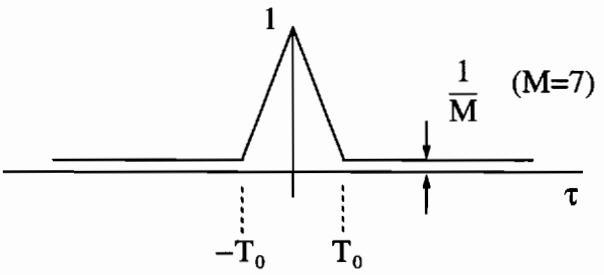


Figure 5.5 Properties of the PRC for M = 7.

where I_0 is the average cw output power. The laser backscatters off of aerosols located at any range within the field of view of the receiving optics. The returned intensity from the backscattering at some range L becomes

$$I_R(t-\frac{2L}{c}) = I_{BS}PRC(t-\frac{2L}{c}) \tag{5.3}$$

where c is the speed of light. The received signal is pseudo-random code modulated, turned on and off with the PRC, delayed by the round trip time to the aerosol particle. For range processing, the received signal is multiplied by a delayed version of the original PRC sequence

$$\left[I_{BS}PRC(t-\frac{2L}{c})\right] \times PRC(t-\frac{2D}{c}) \tag{5.4}$$

where D is the delay distance for the desired range bin. Taking the ensemble average of equation (5.4), the processed signal becomes

$$I_P = I_{BS}<PRC(t-\frac{2L}{c})\,PRC(t-\frac{2D}{c})>$$

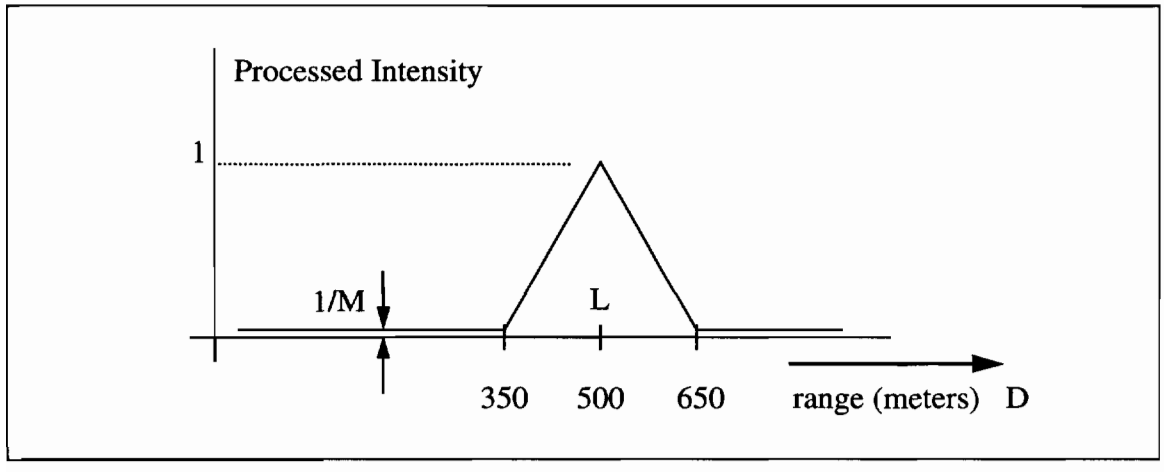


Figure 5.6 Path weight function for L = 500 meters for a direct detection lidar.

$$= I_{BS} R_{PRC}\left[\frac{2}{c}(L-D)\right] \tag{5.5}$$

Because of the autocorrelation property of the PRC, the main contribution to the processed signal occurs when

$$|L-D| \leq \frac{cT_0}{2} .$$

If T_0, the gate width of the PRC, is 1 microsecond, and D is chosen to be 500 meters, the only contributions occur for L = 350 to 650 meters (300 meters full-width, or 150 meters full-width, half-maximum) with the greatest contribution at 500 meters. Figure 5.6 shows the path weight function for L = 500 meters. The autocorrelation property of the PRC is the key to obtaining the intensity vs. range profiles. By iterating the value of D and therefore the time delay of the PRC which multiplies the returned signal, the intensity vs. range profile can be generated. It is important to remember that for each range, the contribution to the profile is most dominant at the center of the range bin L=D but includes contributions up to $\frac{cT_0}{2}$ on both sides, as shown in figure 5.6. Outside of the range bin, the contribution is not 0 but is 1/M. Therefore, 1/M represents the level of crosstalk between range bins for a direct detection system.

5.1.3 Implementing PRC Modulation to CW, Heterodyne Lidar

Because the direct detection systems must use on-off PRC modulation, half of the signal is lost. The only way to recover the other half of the signal is to use a coherent, heterodyne system. Instead of using on-off PRC modulation, the outgoing laser beam can be diphase (+1,-1 PRC) modulated because the amplitude *and* phase of

the backscattered signal are detected. Therefore, the laser is always on implying a potential 3dB improvement in the signal to noise ratio. So far, the major stumbling block has been how to implement the diphase modulation on the outgoing laser beam of a coherent, heterodyne system.

The lidar developed in this project is the first PRC diphase modulated, heterodyne system. The key to implementing the diphase modulation lies in understanding how an acoustooptic modulator works. Acoustooptic modulators (AOMs) use the interaction between an acoustic (sound) wave and a light wave to create a frequency change in the light wave based on the conservation of energy principle. Figure 5.7a shows a schematic of how AOMs work. An electrical sine wave is sent to a transducer which converts the electric signal to a sound wave at the same frequency. The transducer is located at one end of the AOM and an acoustic absorber is placed at the other end to prevent reflections. The transducer sends a traveling sound wave across the AOM to the absorber. When a laser beam is oriented to the Bragg angle, θ, from the normal to the propagation direction of the acoustic wave, a deflected beam is created with a frequency that is upshifted by an amount equal to the frequency of the sound wave. Figure 5.7b explains the reason for the change in frequency and direction based on the conservation of energy principle. In order to conserve energy and momentum, the incident laser beam's direction is changed (Bragg reflection) and its frequency is upshifted. This is called the acoustooptic effect.

AOMs have been used in heterodyne lidars mainly as frequency shifters to create a frequency difference between the transmitted and local oscillator beams that lies within the pass band of the detector. AOMs have also been used as on-off modulators by simply turning the electrical drive signal to the transducer on and off according to a pseudo-random sequence. But again, in the on-off scheme, half of the signal is lost.

Instead of turning the drive signal on and off, it is possible to diphase modu-

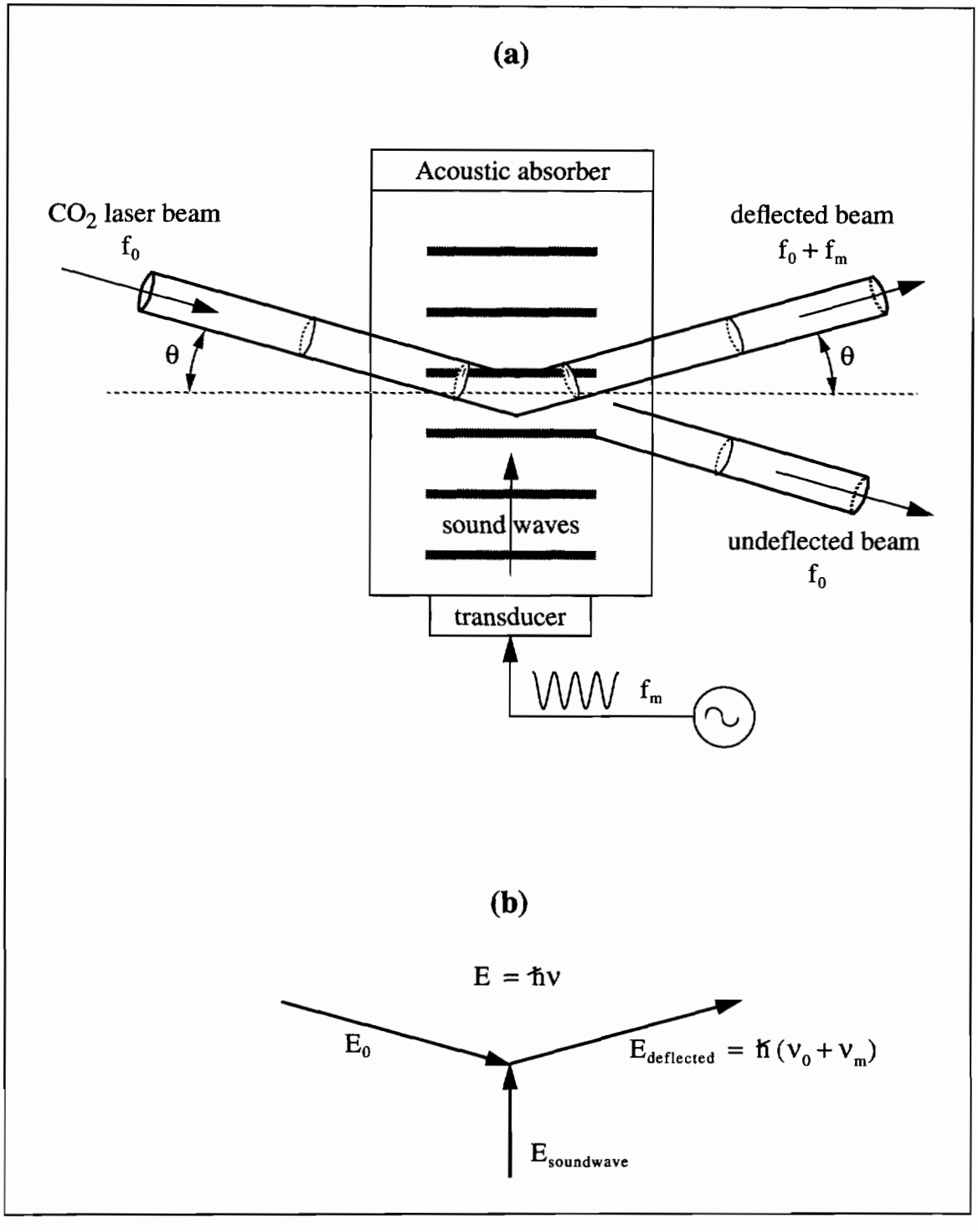


Figure 5.7 Schematic of AOM explaining the change in direction and frequency. (a) Acoustic waves travel across AOM deflecting laser beam and upshifting its frequency. (b) Change in direction and frequency occurs in order to conserve energy.

late a CO_2 laser beam with an AOM. This type of operation with an AOM is not suggested by the manufacturer. In fact, diphase modulating the laser with an AOM has not previously been documented. Figure 5.8 presents the basic idea. The electrical drive signal is a diphase modulated sine wave

$$PRC(t)\cos(2\pi f_m t) \tag{5.6}$$

where $PRC(t)$ is a pseudo-random signal taking on the values of 1 and -1. Equation (5.6) can be rewritten as

$$PRC(t)\cos(2\pi f_m t) = \cos[2\pi f_m t + PRC'(t)\,\pi] \tag{5.7}$$

where $PRC'(t)$ is a pseudo-random sequence taking on the values of +1 and 0. When $PRC'(t)$ equals 0, equation (5.7) becomes $\cos[2\pi f_m t]$. But when $PRC'(t)$ equals 1, equation (5.7) becomes $\cos[2\pi f_m t + \pi] = -\cos[2\pi f_m t]$. Therefore, the sine wave is diphase modulated by the PRC.

In order to achieve diphase modulation of the cw laser source, the diphase modulated electrical signal is sent to the AOM transducer as shown in figure 5.8a. The transducer converts the electrical diphase signal to a sound wave which travels across the AOM crystal. The sound wave traveling across the AOM crystal in figure 5.8a is shown at one of the transitions from one phase to the other. The solid lines of the acoustic wave represent the diphase. The dotted lines are drawn as a reference if there were no diphase. The shortest distance between a dotted line and a solid line depict a phase delay of 180 degrees. Figure 5.8b shows the expected waveforms. The deflected beam will have some finite transition time to change from one phase to

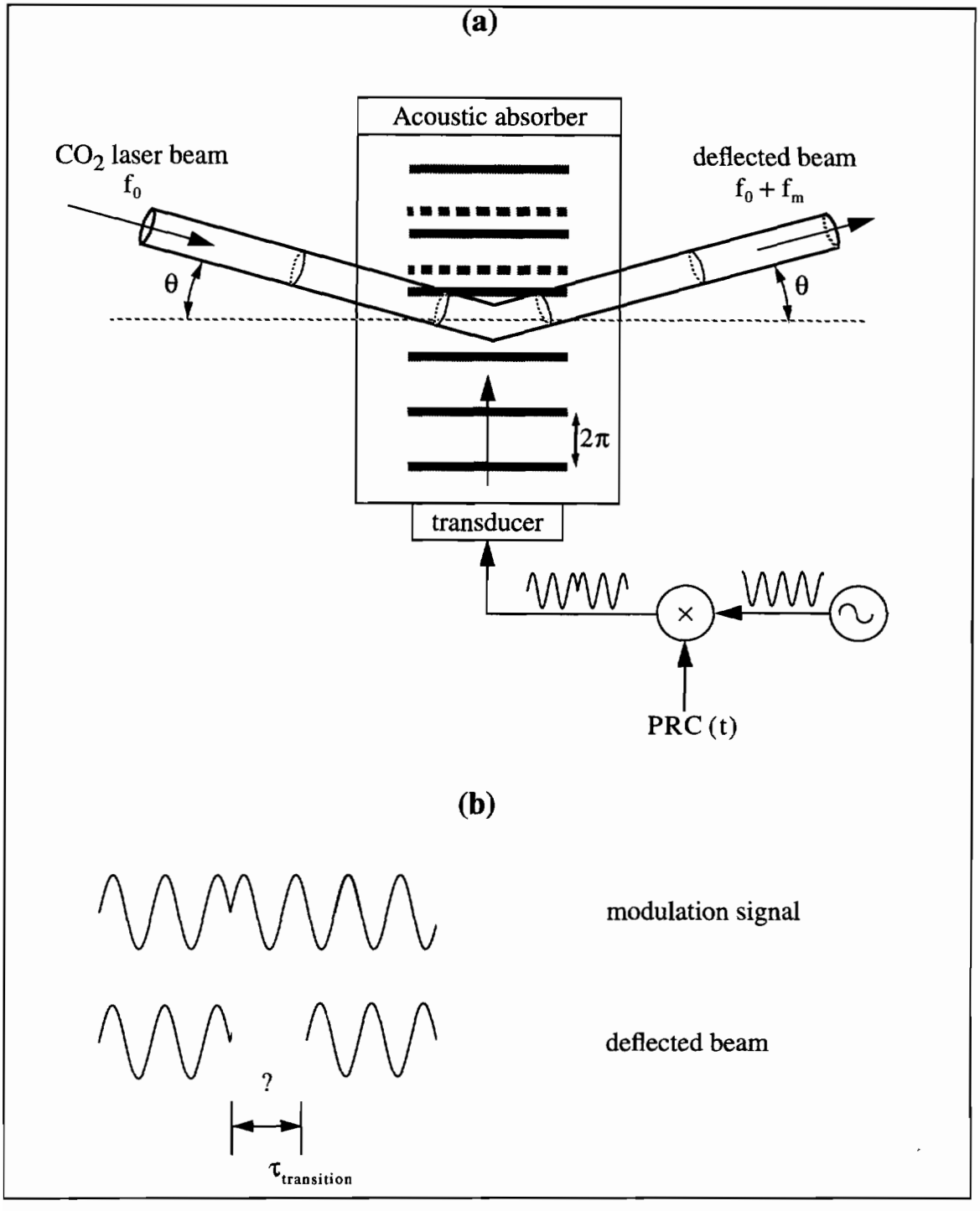


Figure 5.8 Schematic of AOM used to diphase modulate the laser beam. (a) The diphase drive signal is converted to acoustic waves. Dotted lines are a reference if there were no diphase. (b) When the acoustic wave changes phase, a finite transition time results because the phase change must propagate across the entire diameter of the laser beam.

another because the phase change in the acoustic wave takes time to propagate across the diameter of the laser beam.

The schematic of the optical set-up to test the first di-phase operation of a cw laser source is shown in figure 5.9, while figure 5.10 presents the results of the first experiment to test the diphase operation of the AOM, built by Isomet, model #1207B-3. The bottom curve in figure 5.10 represents data from the first diphase operation of the AOM. For comparison, the top curve is without diphase. Before the phase change, both curves are in phase, with the valleys of each sine wave aligned and labeled in the plot. During the phase change, there is a transition time of about 300 nanoseconds when the phase is not defined. After the transition period, both curves are out of phase by 180 degrees, with the peak of the top curve aligned with the valley of the bottom curve. Therefore, the CO_2 laser beam is diphase modulated by the AOM. Because the phase is not defined during the transition period, there could be a drop in received signal power as compared to the power received without diphase operation. For that reason, the reduction in received power during diphase operation was measured and compared to the received power without diphase operation. It was found that the received power during diphase operation with $T_0 = 1\mu s$ was 91% of that without the diphase. Therefore, although the transition time is 300ns, there is only a 9% reduction in received optical power due to the diphase modulation of the laser beam.

5.0.1 Range Resolution of PRC Modulation CW Heterodyne Lidar

Computing the Intensity vs. Range profiles for the PRC heterodyne systems is a slightly different process than for direct detection systems because heterodyne systems detect the amplitude and phase of the backscattered laser beam rather than just the intensity. Figure 5.11 shows how the photomixing of the LO and backscattered

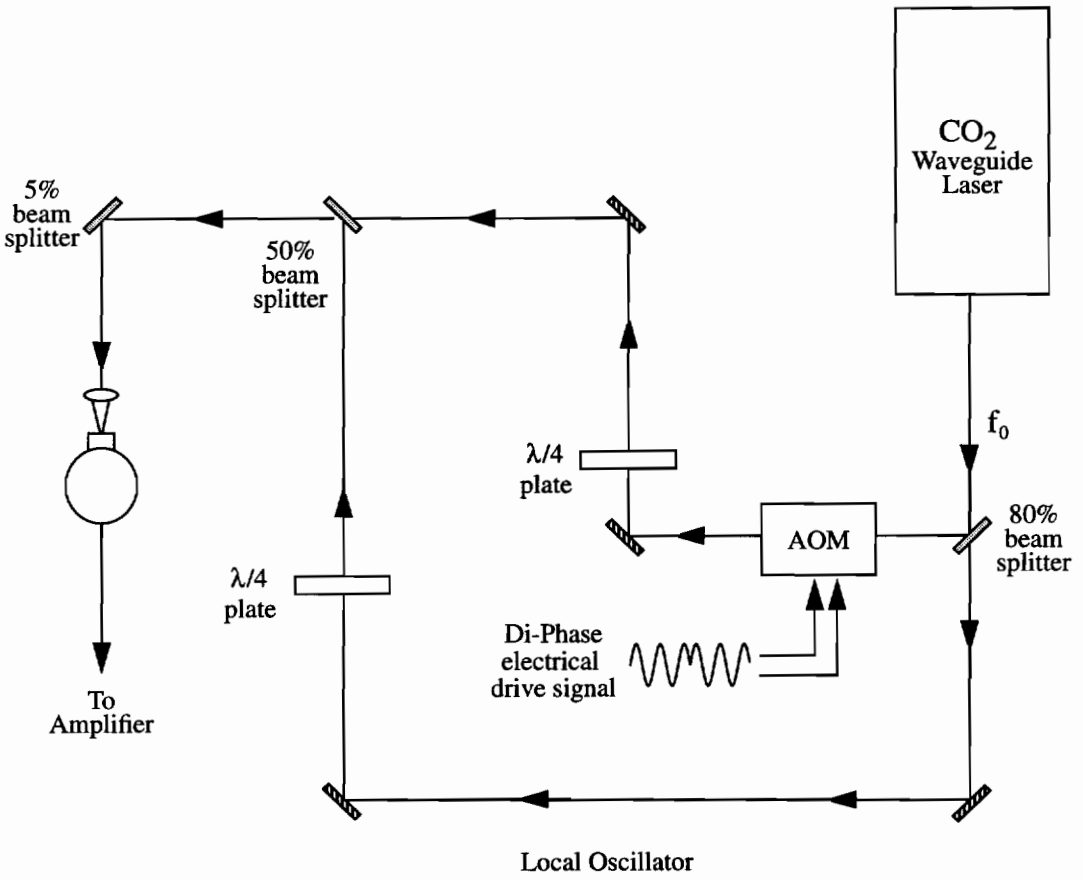


Figure 5.9 Schematic of the experimental lidar to test di-phase operation of AOM.

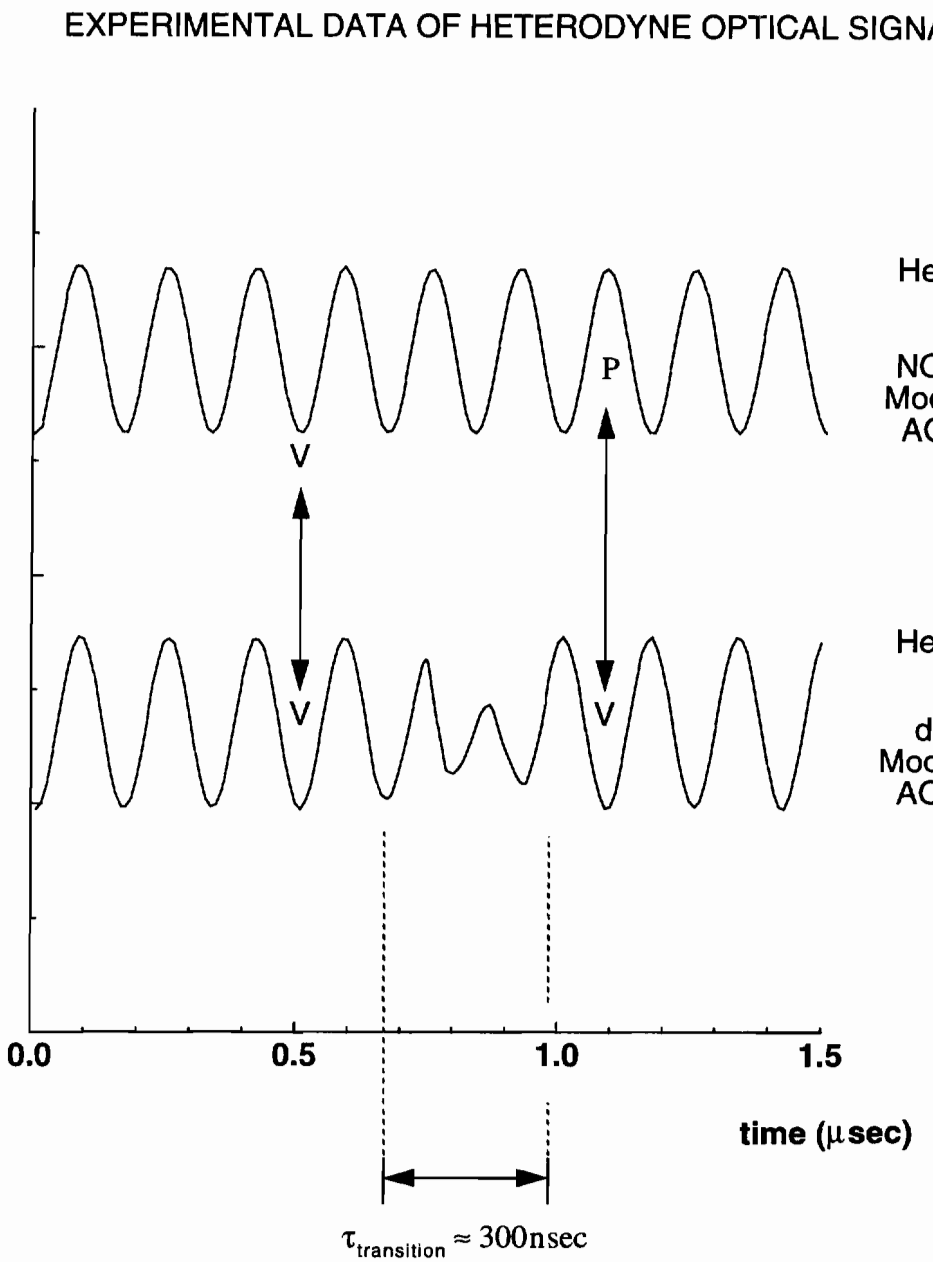


Figure 5.10 First experiment to measure diphase modulation of a CO_2 laser beam. Bottom curve is diphase. Top curve is no diphase for reference. Before phase change, both curves are in phase. After phase change, curves are out of phase by 180 degrees after a transition time of about 300 ns.

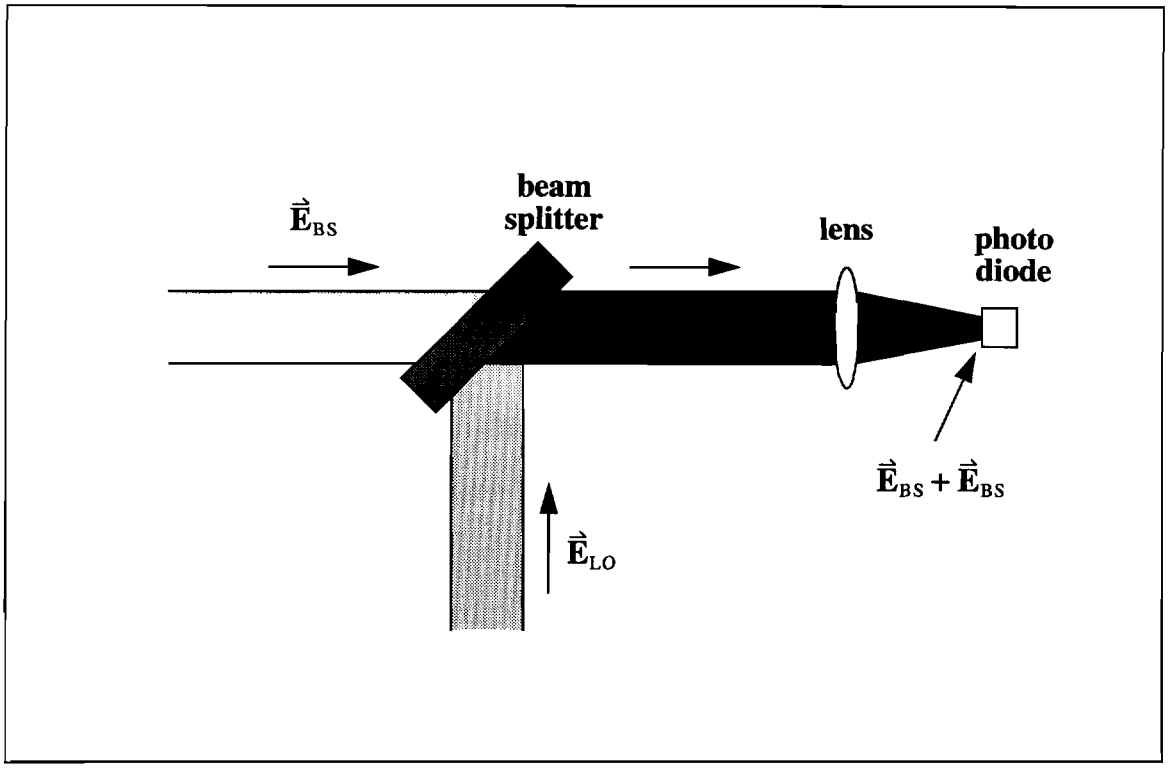


Figure 5.11 The backscattered electric field, $\vec{E}_{BS}$, and the local oscillator electric field, $\vec{E}_{LO}$, are combined and focussed onto the photo diode.

electric field occurs. Both electric field components are combined by a 5% beam splitter and focussed onto the photodiode surface by a 1" focal length lens. The electric field incident on the photodiode surface can be written as

$$\vec{E}_{diode} = \vec{E}_{BS} + \vec{E}_{LO} \tag{5.8}$$

where

$$\vec{E}_{LO} = A_{LO}\cos[2\pi f_{LO}t] \tag{5.9}$$

and

$$\vec{E}_{BS} = A_{BS}(t)\,PRC\left(t - \frac{2L}{c}\right)\cos\left[2\pi\left(f_T + f_D\right)t + \phi_{total}(t)\right] . \qquad (5.10)$$

In these equations, f_{LO} is the frequency of the LO beam, $f_T + f_D$ is the frequency of the backscattered beam which includes the Doppler shift, f_D, and $\phi_{total}(t)$ represents the phase decorrelation due to turbulence, aerosol speckle, and laser decorrelation discussed in detail at the end of chapter 3. The detector puts out a current proportional to the intensity

$$i_d(t) \approx \left|\vec{E}_{BS} + \vec{E}_{LO}\right|^2 . \qquad (5.11)$$

The pass band of the detectors is DC to about 20 MHz. Therefore, not all of the terms generated by equation (5.11) are seen at the output. The only terms that survive are

$$i_{DC} \sim \frac{A_{LO}^2 + A_{BS}^2(t)}{2} \qquad \text{DC term, generates shot noise}$$

$$i_{het}(t) \sim \frac{A_{LO}A_{BS}(t)}{2} PRC\left(t - \frac{2L}{c}\right)\cos\left[2\pi\left(f_{het} + f_D\right)t + \phi_{total}(t)\right]$$

heterodyne signal current

where $f_{het} = f_{LO} - f_T = 6\text{MHz}$. The DC term is responsible for generating the signal shot noise important for the SNR calculations in the next section. The heterodyne term is the term used to calculate the intensity vs. range, and the time delayed statistics. Since A_{LO} is a constant, the heterodyne signal current, $i_{het}(t)$, is proportional to

the backscattered amplitude, diphase modulated by the PRC delayed to the scattering location, L, and centered at a frequency of 6MHz + the Doppler shifted frequency. Since the heterodyne signal current is diphase modulated, the average received signal is roughly twice that of the on-off scheme.

The next step in recovering the intensity vs. range is to multiply $i_{het}(t)$ by

$$PRC\left(t-\frac{2D}{c}\right)$$

where D is the distance to the center of the desired range bin

$$i_{proc}(t) \approx PRC\left(t-\frac{2D}{c}\right) \times \left[A_{BS}(t)\, PRC\left(t-\frac{2L}{c}\right) \cos\left[2\pi\left(f_{het}+f_D\right)t+\phi_{total}(t)\right] \right] . \tag{5.12}$$

Averaging the recovered signal as with the direct detection processing scheme is not possible since a time average of a sine wave is 0. However, the intensity may be found by taking the FFT of the recovered signal and finding the peak amplitude from the spectrum. The level of crosstalk in the direct detection systems is 1/M, the code length. But the level of crosstalk for a heterodyne system is different because the intensity is taken from the peak of a FFT. The crosstalk for a heterodyne lidar is the ratio of the peak intensity of the FFT when the backscattered signal is recovered exactly (when D = L) to the peak intensity when the backscattered signal is recovered at the next range bin (when D is greater than $L+cT_0/2$ or less than $L-cT_0/2$). Figure 5.13 shows the spectra of two simulated data sets backscattered from the same range, but recovered by the PRC delayed to two different ranges. When D = L, the

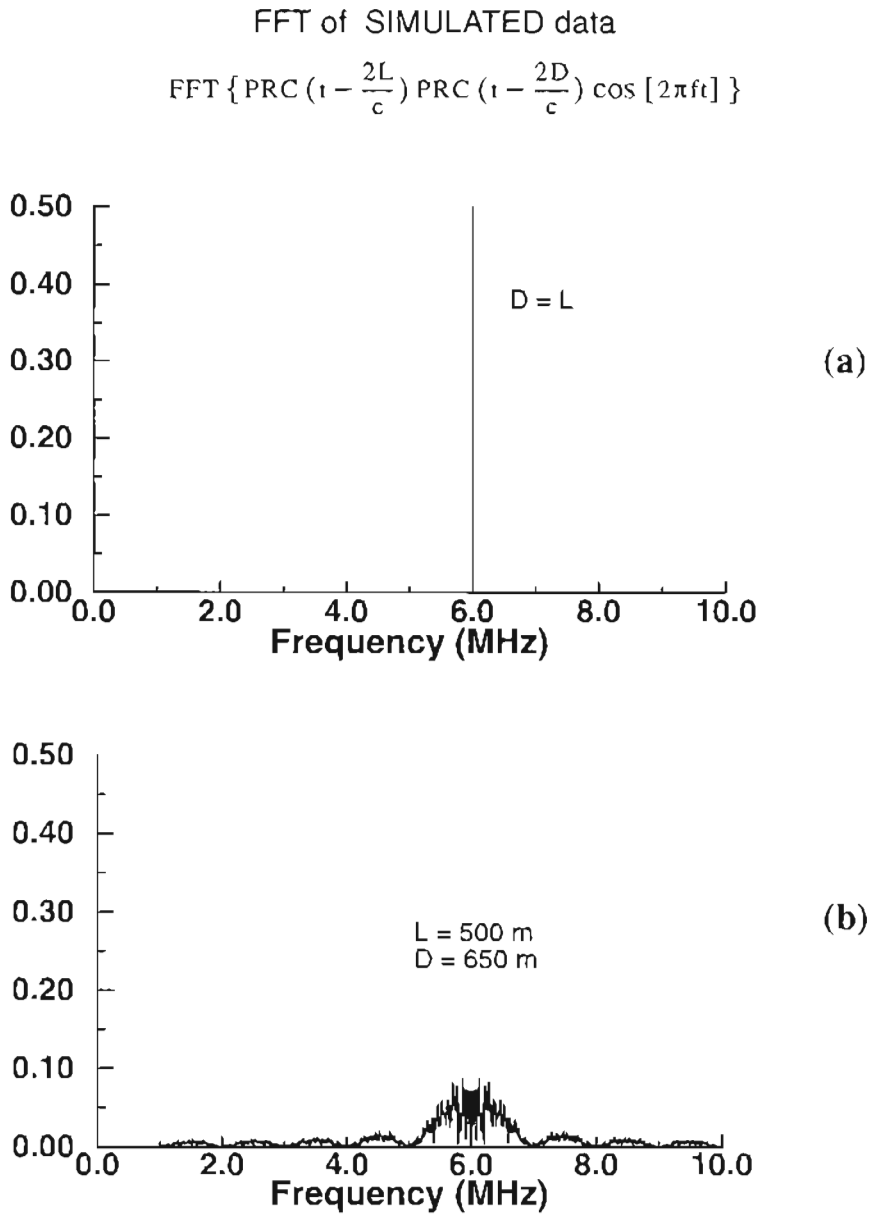


Figure 5.12 FFT of simulated data, recovered at two different ranges. (a) D=L (b) L=500m, D=650m. Since information is taken from FFT, crosstalk is the ratio of peak when D=L to peak when D is outside range bin where scattering occurs.

FFT of equation (5.12) is just that of a pure cosine as shown in figure 5.12a. But when D is greater than $L+cT_0/2$ or less than $L-cT_0/2$, the FFT of equation (5.12) is that of a cosine multiplied by

$$PRC\left(t-\frac{2L}{c}\right)PRC\left(t-\frac{2D}{c}\right)$$

which is just another pseudo-random sequence of 1's and -1's. Therefore, the FFT of the unrecovered simulated data is the spread spectrum shown in figure 5.12b. For a 63-state PRC used in this project, the crosstalk of the intensity is $(0.5)^2/(0.07)^2 = 50$ and the path weight function is shown in figure 5.13 below. From the FFT comes all of the information about the backscattered signal. The peak value gives an estimate of the backscattered intensity, and the frequency where the maximum intensity occurs provides the Doppler shift.

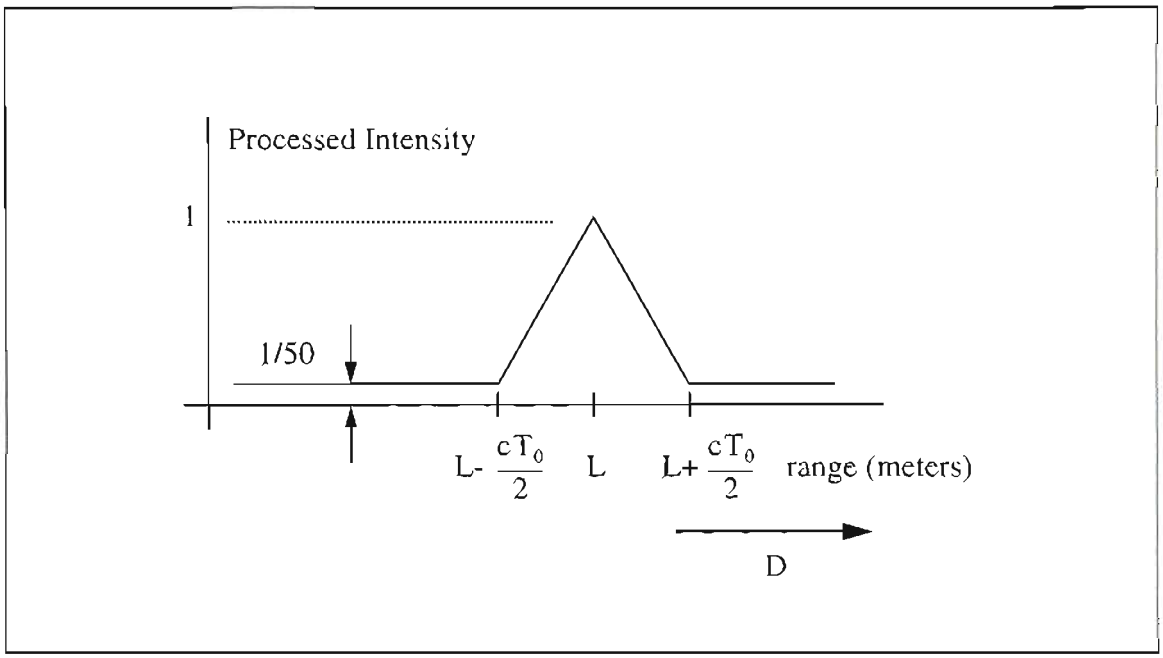


Figure 5.13 Theoretical path weight function for a heterodyne system, 63-state PRC.

Figure 5.14 presents the processed experimental intensity vs. range for the lidar backscattered from a hard target located at 830 meters. The profile has the triangle shape as it should. The crosstalk is about 1/50, which matches the predicted level. The peak is about 300 meters wide corresponding to the width of the range bin, cT_0.

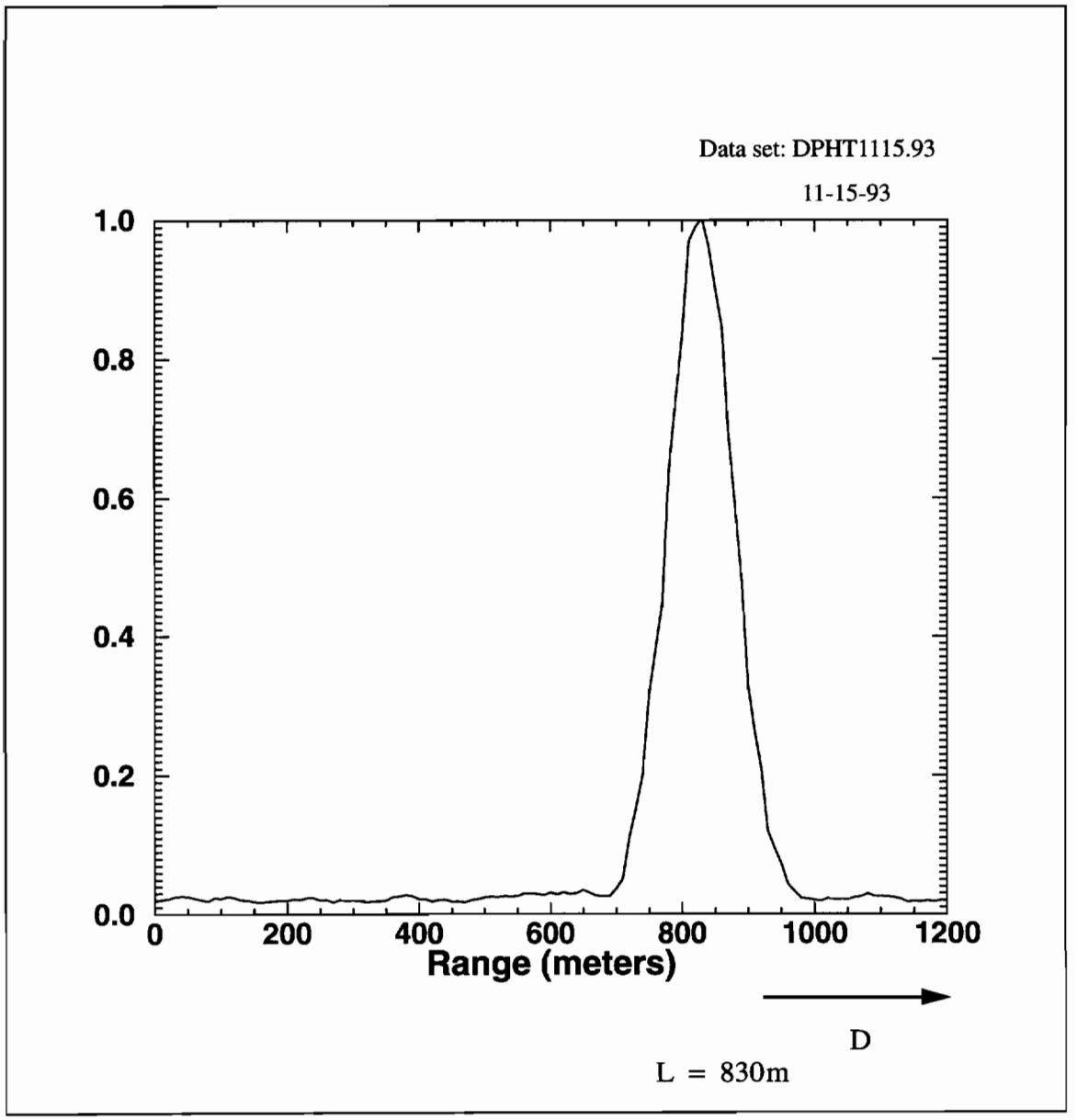


Figure 5.14 Processed intensity vs. range for a hard target located at 830 meters.

5.2 Signal to Noise Ratio (SNR)

The ability of any optical heterodyne lidar to remotely sense information about the atmosphere depends on the Signal to Noise Ratio (SNR) of the system. Each optical adjustment, alignment, or design decision is performed to try to operate the system at the optimum SNR. After transmitting as much power as possible to the atmosphere and optimally matching the optical LO electric field wavefront to the backscattered electric field wavefront, the other aspects in maximizing the SNR are concerned with the detector and its amplification electronics, and selecting the appropriate size of the receiver aperture area so that aperture averaging is not a significant problem. Of all the tasks necessary to optimize the SNR, the most difficult task is operating the detector properly so that the SNR is maximum.

5.2.1 SNR Using Photodiodes for Coherent Heterodyne Detection

The main part of the receiving apparatus of an optical heterodyne lidar is the detector, which may be a photoconductive type of detector or a photodiode, operated in either the photoconductive (reverse biased) or photovoltaic mode. The lidar developed in this project uses photodiodes operated in the reverse bias mode in order to minimize the device capacitance and increase the frequency response. A photodiode may be modeled as a current generator in parallel with a diode resistance and capacitance as shown in figure 5.15. When the LO and backscattered electric fields are focussed onto the photodiode surface, a diode current, i_d, is generated proportional to the applied optical power. Optimum operation occurs when the input impedance of the electronics is much less than the detector resistance to allow virtually all the diode current to be collected by the electronics. Using a transimpedance amplifier with a

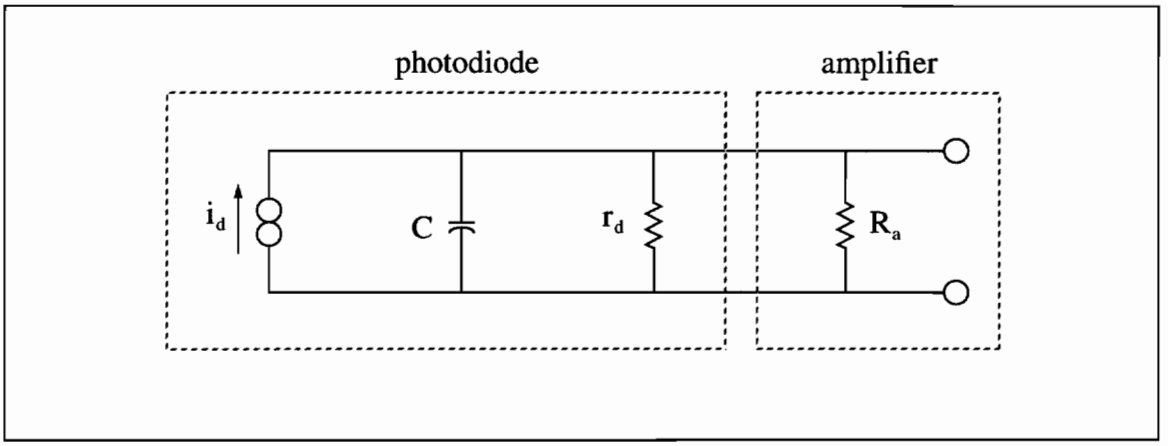


Figure 5.15 Photodiode equivalent circuit connected to amplifier.

low input resistance is one way to collect virtually all of the diode current. Since the transimpedance amplifiers used in the project have an input resistance of about 75 Ohms and the equivalent diode resistance is on the order of 10^3 Ohms, this requirement is satisfied.

For ideal photodiodes, the current generated is linearly proportional to the applied optical power for any power level. Therefore, the shot noise generated by the quantum fluctuations in the LO electric field dominate over the amplifier Johnson noise, dark current, and background radiation noise. This is called LO, shot-noise limited operation. For ideal photodiodes it is always possible to operate in the signal shot-noise limit. For real photodiodes, however, the detector responsivity becomes nonlinear due to saturation effects, and as more LO power is applied, the SNR peaks at a finite LO power level. Therefore, the LO power level that maximizes the SNR may not be enough to cause optimal signal shot-noise limited operation.

Equation (5.13) gives the current output by the detectors proportional to the intensity. Two terms result: the DC term, which is responsible for the shot noise gen-

erated by the LO electric field, and the heterodyne signal term, which is the processed signal. After multiplying by the delayed PRC, the photodiode current can be written

$$i_d(t) \approx \rho_0 [A_{LO}^2 + A_{BS}^2(t)] + 2\rho_0 A_{LO} A_{BS}(t) \cos [2\pi (f_{het} + f_D) t + \phi_{total}(t)] \quad (5.13)$$

where ρ_0 is the responsivity of the detector measured in A/W. Equation (5.13) can be rewritten as

$$i_d(t) = \rho_0 (P_{LO} + P_{BS}) + 2\rho_0 \sqrt{P_{LO} P_{BS}} \cos [2\pi (f_{het} + f_D) t + \phi_{total}(t)] \quad (5.14)$$

where P_{LO} and P_{BS} are the LO and backscattered signal powers incident on the detector. The first two terms of equation (5.14) comprise the DC term

$$i_{DC} = \rho_0 (P_{LO} + P_{BS}) \approx \rho_0 P_{LO} \quad (5.15)$$

since $P_{LO} \gg P_{BS}$, and the last term is the heterodyne signal term

$$i_{het}(t) = 2\rho_0 \sqrt{P_{LO} P_{BS}} \cos [2\pi (f_{het} + f_D) t + \phi_{total}(t)] \ . \quad (5.16)$$

If the detectors do not saturate as more LO optical power is applied, then ρ_0 is a constant. The SNR will now be formulated with reference to figure 5.15 for the ideal case, assuming that the amplifier following the detector limits the bandwidth to B Hertz, and the Johnson noise from the parallel combination of r_d and R_a is represented by an equivalent resistance R_e at some fictitious temperature T_e (noise temperature of amplifier). The signal power from the heterodyne term, equation (5.16), is

given by

$$S = 2\rho_0^2 P_{LO} P_{BS} R_e^2. \tag{5.17}$$

The noise power can be calculated by ignoring the contributions from the background radiation and dark current. The remaining sources of noise are the Johnson noise from the equivalent resistance R_e and the signal shot-noise due to the LO. The noise power becomes[30]

$$N = 2eB\rho_0 P_{LO} R_e^2 + 4BKT_e R_e \tag{5.18}$$

where K is Boltzman's constant, and e is the electronic charge of an electron. The SNR for the ideal case is formed from equations (5.18) and (5.17)

$$SNR_{ideal} = \frac{\rho_0^2 P_{LO} P_{BS}}{B\left[e\rho_0 P_{LO} + \frac{2KT_e}{R_e}\right]}. \tag{5.19}$$

The most optimum operation is to increase the LO power so that the first term in the denominator of equation (5.19) dominates over the Johnson noise. This type of operation is always possible for ideal photodiodes. Under these conditions the heterodyne system is limited only by the shot-noise from the LO, simplifying the SNR to

$$SNR_{ideal} = \frac{\rho_0 P_{BS}}{Be}. \tag{5.20}$$

Unfortunately, the responsivity of actual photodiodes *decreases* as more P_{LO} is

applied. The SNR becomes a function of P_{LO} which has a maximum at a finite P_{LO} level. Therefore, the P_{LO} level that maximizes the SNR curve may not be enough to operate under LO shot-noise limited conditions.

To investigate the photodiode saturation effect, the responsivity for each of the four photodiodes summarized in table 5.2 was measured and plotted in figure 5.15. As more optical power is applied, the photodiode current starts to saturate for each photodiode. Consequently, the responsivity decreases as more power is applied to the photodiode surface. The least squares fit to the diode current vs. the applied power in figure 5.15 is a quadratic of the form

$$i_d(P_{in}) = \rho_0 P_{in} - \rho_0 \alpha P_{in}^2 \tag{5.21}$$

where $P_{in} = P_{LO} + P_{het}$, and ρ_0 and α are found by making a least squares fit to the i_d vs. P_{in} data in figure 5.16. For example, for the least-squares curve of photodiode A in figure 5.16, $\rho_0 = 6.206$ A/W and $\alpha = 198.0$/W.

Table 5.2 Properties of the photodiodes presented in figure 5.15

Photo diode	Manufacturer	Model and Serial #	Material	Dimensions	Photodiode Surface Area
A	Fermionics	FM-PV-K100 1-1238N2-2	Hg-Cd-Te	□ d=0.2mm	A=0.04mm^2
B	NERC	MPV-11-.2-B50 3917-2	Hg-Cd-Te	□ d=0.2mm	A=0.04mm^2
C	Brimrose	MMT 10-3-.45 9204-IR-176	Hg-Mn-Te	◯ r=0.225mm	A=0.16mm^2
D	Fermionics	FM-PV-K100 1-1238N2-1	Hg-Cd-Te	□ d=0.2mm	A=0.04mm^2

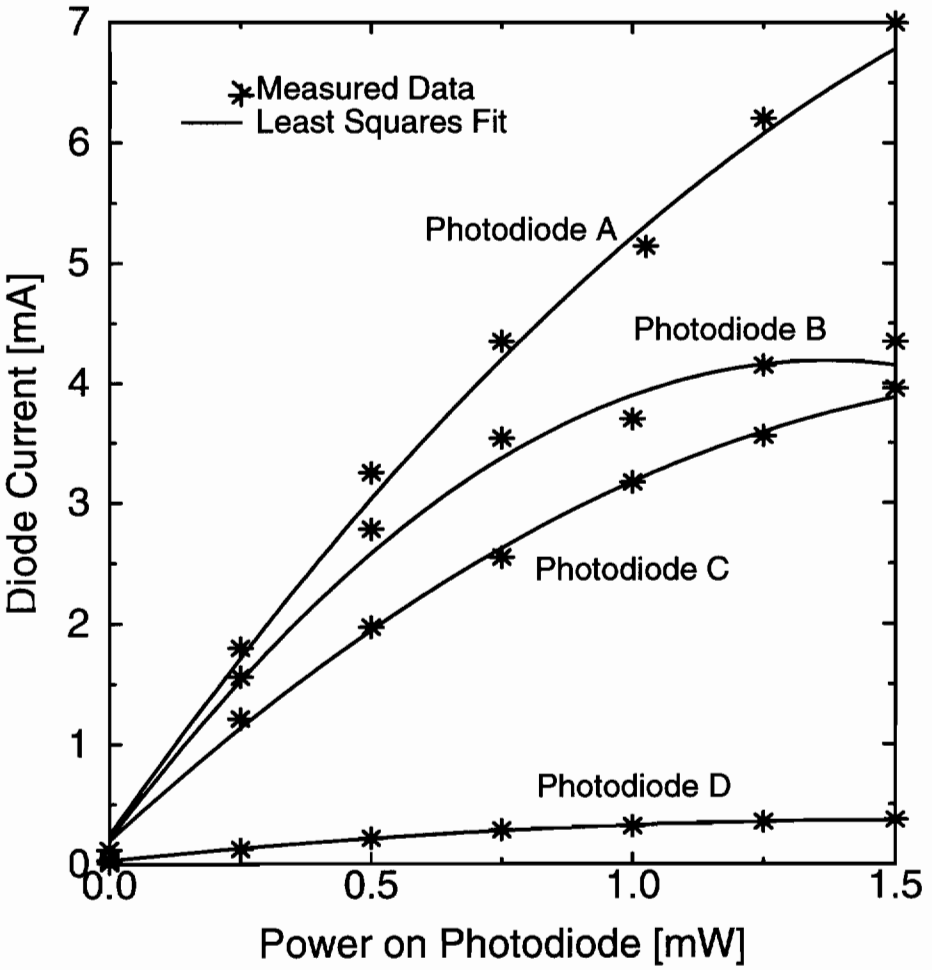


Figure 5.16 Detector current vs. applied optical power, and least squares fit for the four photodiodes characterized in table 5.2.

The effect on the SNR can be evaluated more easily by expanding equation (5.21) in a Taylor series about P_{LO}. This has the effect of linearizing the nonlinear photodiode equation about a power level established by P_{LO}. This approximation is valid because P_{LO} dominates over the noise level and the heterodyne signal power. Expanding equation (5.21) about P_{LO},

$$i_d = i_d(P_{LO}) + \frac{\partial}{\partial P_{LO}}[i_d(P_{LO})]\,P_{het}$$

$$= i_{DC} + i_{BS}\cos[2\pi(f_{het} + f_D)t + \phi_{total}(t)] \tag{5.22}$$

where

$$i_{DC} = \rho_0 P_{LO}(1 - \alpha P_{LO}) \tag{5.23}$$

and

$$i_{BS} = 2\rho_0[1 - 2\alpha P_{LO}]\sqrt{P_{LO}P_{BS}}\,. \tag{5.24}$$

The heterodyne signal power becomes

$$S = 2\rho_0^2[1 - 2\alpha P_{LO}]^2 P_{LO}P_{BS}R_e^2 \tag{5.25}$$

and the noise power converts to

$$N = 2e\rho_0 P_{LO}(1 - \alpha P_{LO})BR_e^2 + 4KT_eBR_e \tag{5.26}$$

transforming the SNR to

$$SNR_{saturation} = \frac{\rho_0 P_{BS}}{Be}\left[\frac{(1-2\alpha P_{LO})^2 P_{LO}}{P_{LO}(1-\alpha P_{LO}) + \frac{2KT_e}{\rho_0 e R_e}}\right] . \qquad (5.27)$$

Figure 5.17 shows the SNR normalized by the constants in front of equation (5.27) for photodiode A. To plot equation (5.27), $T_e = 600^0$F (approximately a 5 dB noise figure) and R_e = 75 ohms. The SNR curve increases linearly for low P_{LO}, but finally reaches a maximum for $P_{LO} \approx 0.44$ mWatts and decreases for P_{LO} beyond that value. Therefore, there is a finite value of P_{LO} that optimizes the SNR due to photodiode saturation.

Another way to view the degradation in photodiode performance due to saturation is to normalize equation (5.27) by equation (5.20) which assumes LO shot-noise limited operation and no diode saturation. The result is a SNR reduction factor of the form

$$RF_{SNR} = \frac{(1-2\alpha P_{LO})^2 (P_{LO} + \frac{2KT_e}{\rho_0 e R_e})}{P_{LO}(1-\alpha P_{LO}) + \frac{2KT_e}{\rho_0 e R_e}} . \qquad (5.28)$$

Figure 5.18 shows the plots of equation (5.28) due to the LO shot-noise limited assumption applied to all four photodiodes for the same values of T_e and R_e used to plot equation (5.27). From figure 5.18, photodiode A performs the best, but is still 3.18 dB short of optimum signal shot-noise limited operation. The worst of the diodes is more than 12 dB below the limit. Based on the results of the SNR reduc-

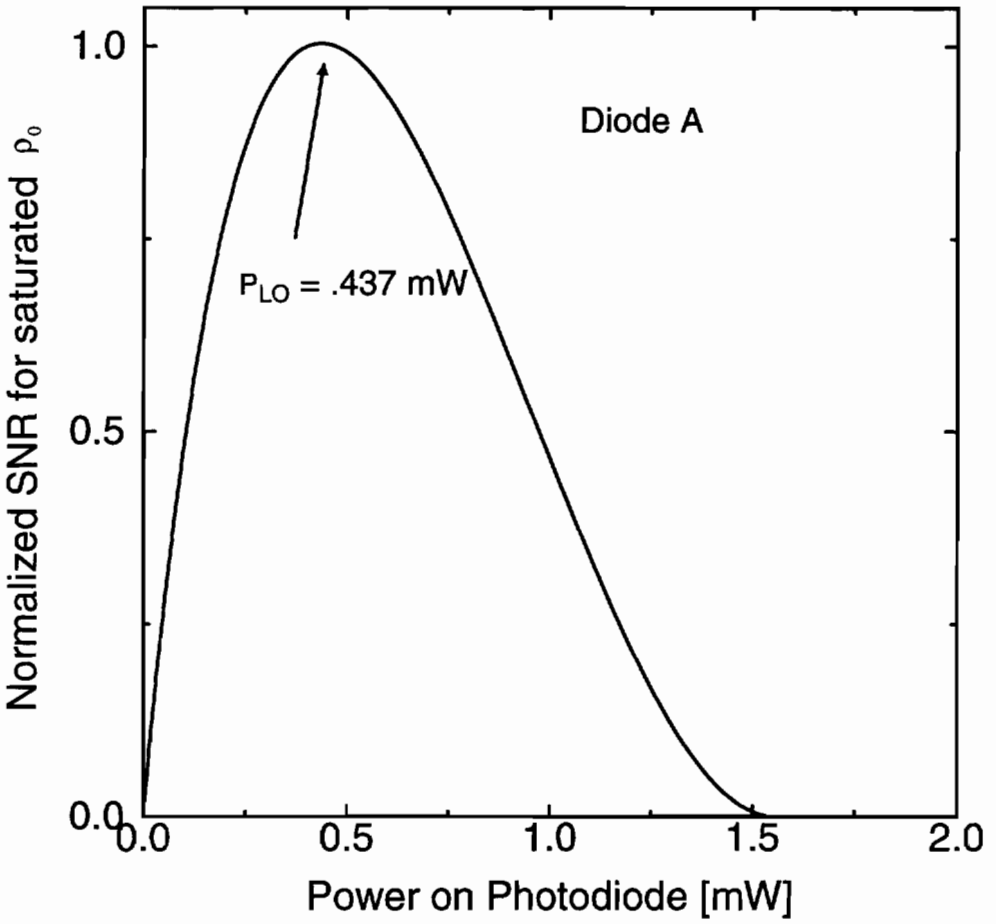


Figure 5.17 Normalized SNR for Photodiode A with saturated responsivity.

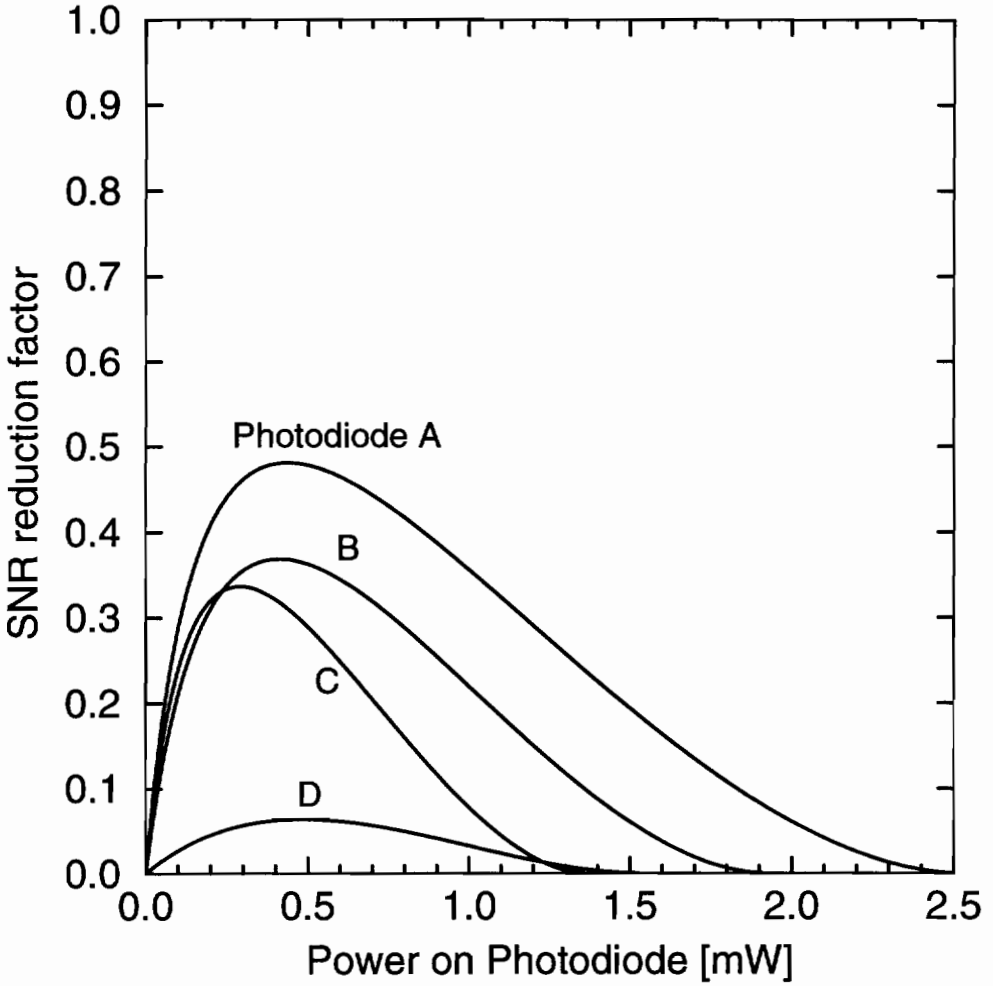


Figure 5.18 SNR reduction factor due to LO shot-noise limited assumption.

tion, photodiodes A and B were installed in the lidar system as these photodiodes had the best overall performance.

Since the intensity and Doppler frequency are taken from the FFT of the digitized backscattered laser signal, it is important to verify that the SNR is above the threshold to estimate the intensity and Doppler frequency from discrete time observations buried in white Gaussian noise. This subject has received much attention.[64,65,66] The best estimate of the Doppler frequency and intensity is obtained from the maximum peak in the discrete Fourier transform of the digitized data. However, the ability to distinguish the peak diminishes as the SNR decreases. Finally, below some threshold of SNR it becomes impossible to identify the peak, and therefore, impossible to determine the Doppler shift.

One immediate solution for enhancing the ability to recognize the Doppler frequency buried in Gaussian white noise is to take a longer data sample length for the FFT. Not only does the frequency resolution improve, but the noise level decreases as well. Since the noise is 0-mean and Gaussian, a longer sample length reduces the noise level. The sample length cannot become too large, however, because of the laser decorrelation time discussed in Chapter 3. The laser used in this project has a decorrelation time of about 50 microseconds. For a sampling rate of 20MHz, the closest sample length with a power of 2 is N = 1024 points (51.2 microseconds). Therefore, the Doppler shift and intensity are obtained from FFT's of length N = 1024 points. Figure 5.19 shows a plot of the relative processed SNR vs. FFT time slice for digitized data of the CO_2 laser backscattered from aerosols. This curve was generated by taking the ratio of the peak in the FFT to the average noise floor level in the FFT spectrum. This ratio is called the processed SNR. This figure should be compared to figure 3.13 in Chapter 3. For FFT time slices greater than the laser coherence time, the processed SNR actually decreases. Figure 5.19 indicates that the FFT length

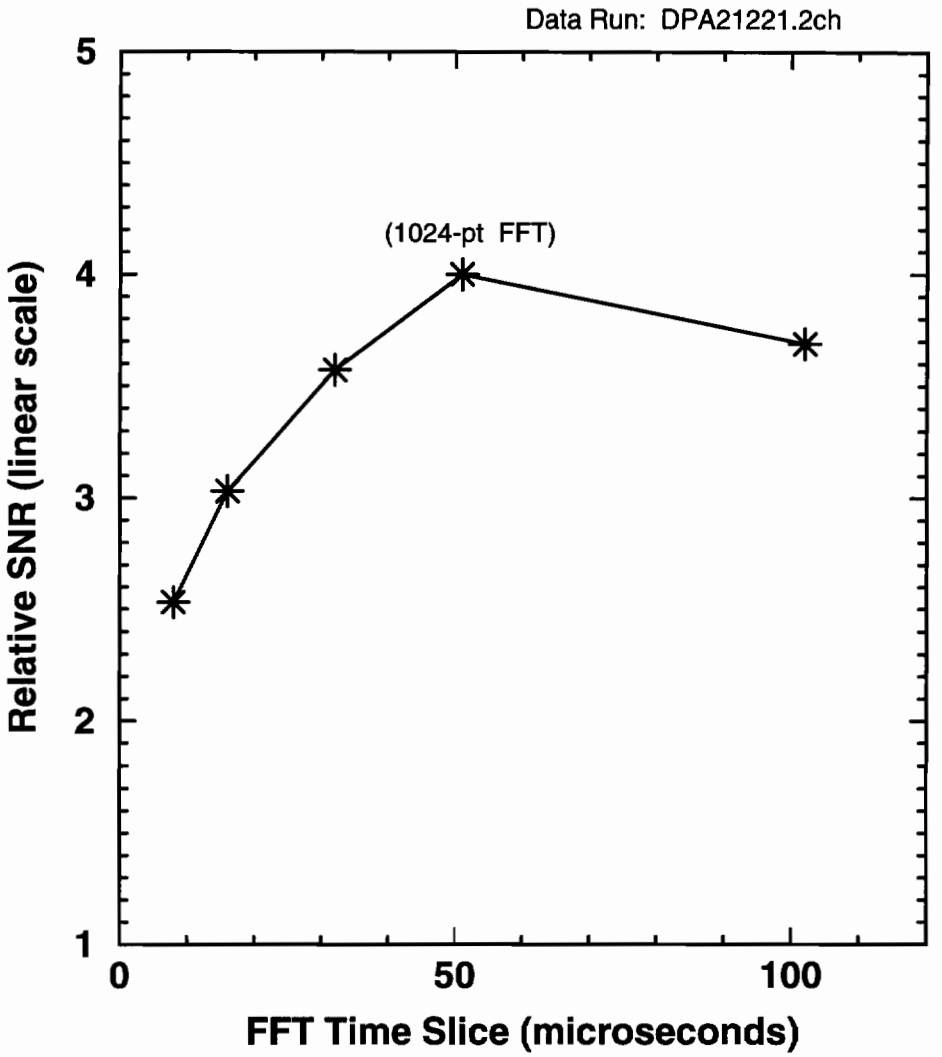


Figure 5.19 Processed relative SNR of signal backscattered from aerosols for different FFT time slices (different sample lengths). For time slices greater than the laser coherence time, the SNR decreases. This figure should be compared to figure 3.13, Chapter 3.

which maximizes the processed SNR is N = 1024 points. As demonstrated in Chapter 3, the only factor which limits the sample length is the laser decorrelation time, not the atmospheric decorrelation or turbulence effects on the phase. Therefore, a laser that has a longer coherence time could greatly improve the processed SNR. It should be noted that all other data sets used to process the SNR vs. time slice resembled figure 5.19, giving the best SNR for a time slice = 51.2 microseconds, providing strong proof that the most significant effect that reduces the processed SNR is the laser coherence time.

The effect of improving the processed SNR by increasing the sample length is studied in detail by Rife and Boorstyn [64]. From their analysis comes figure 5.20, the probability of an outlier or an erroneous choice of the peak frequency vs. SNR for different sample sizes. From the figure, the threshold to choose the correct peak frequency for an N = 1024 point FFT is about SNR = −14dB. For SNR's below -14dB, the peak frequency becomes undetectable. It is therefore imperative that the SNR of the detected signal be greater than -14dB.

Equation (5.27) can be used to estimate the SNR for the actual case of the photodiode under saturation to verify that the SNR is above threshold. First an estimate of P_{BS}, the backscattered laser power must be calculated. A rough estimate of P_{BS} is obtained from the lidar equation[30]

$$P_{BS} \approx \frac{A_{aper} \Delta Z_{rb} U_T^2 \beta}{z_c^2} \tag{5.29}$$

where A_{aper} is the detector aperture area, ΔZ_{rb} is the range bin width, U_T^2 is the transmitted power, z_c is the distance to the center of the range bin, and β is the backscattering coefficient. The aperture area is a half moon with radius 1.05", the transmitted

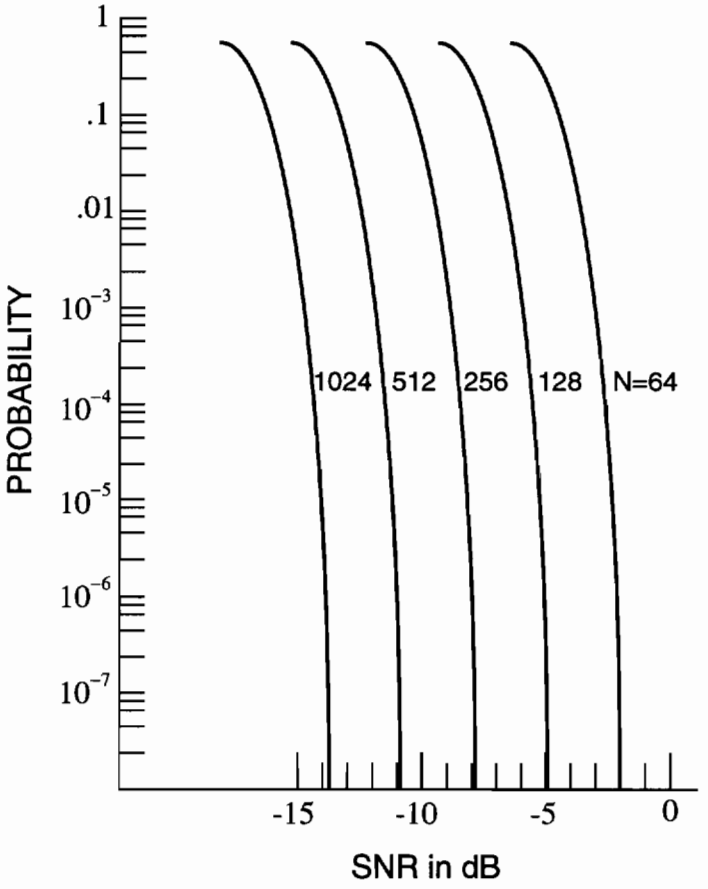


Figure 5.20 Probability of an outlier (erroneous peak frequency) vs. SNR (from Rife and Boorstyn [64]).

power is 3 Watts, and the range bin width is 300 meters, making the received power

$$P_{BS} \approx 0.91 \frac{\beta}{z_c^2}. \tag{5.30}$$

For a backscattering coefficient of $10^{-8} m^{-1}$ (typical for the lower atmosphere) and a path length z_c of 400 meters, a rough estimate of the backscattered power is

$$P_{BS} \approx 5.7 \times 10^{-14} \text{ Watts}$$

Using this in equation (5.27) for photodiode A gives the estimate of the SNR to be

$$SNR \approx 0.124 \text{ or } -9dB.$$

Therefore, the lidar operates above threshold.

5.3 Processing Schemes to Estimate Wind Speed

The expression for the time-filtered time-delayed crosscovariance of intensities between two point detectors is derived in chapter 4 and resulted in equation (4.45). Appendix E shows that equation (4.45) can be simplified further, reducing to

$$C_{N_{lpf}}(\vec{\rho}, \tau) = \exp[4C_\chi(z_c, \vec{\rho}, -\tau)] - 1. \tag{5.31}$$

after normalizing by the average intensities to eliminate the constants in equation (e.5). The subscript N_{lpf} stand for a normalized, low pass filtered result. Since C_χ is small, the exponential can be expanded using $e^x \approx 1 + x$ to simplify equation (5.31) to

$$C_{N_{lpf}}(\vec{\rho}, \tau) \approx 4C_\chi(z_c, \vec{\rho}, -\tau) \tag{5.32}$$

where the log amplitude covariance is defined by equation (4.18), with $\rho = 0$

$$C_\chi(\vec{\rho}, \tau) = 0.132\pi^2 k^2 C_n^2 z_c \int d\omega \int du \; u^{-8/3} \sin^2\left[\frac{\omega(1-\omega)u^2 z_c}{2k}\right]$$

$$\times J_0[u|\omega\vec{\rho} - \langle\vec{v}_\perp\rangle\tau|]. \tag{5.33}$$

The normalized time-delayed crosscovariance of intensities for two point detectors is plotted in figure 5.21 for two different values of the crosswind speed and distance to the center of the range bin. The detector spacing is 1.05 inches, the experimental value. As the crosswind speed gets stronger, the peak of the curve shifts toward the origin. By changing the average crosswind speed, the relationship between the detector spacing, p, the average crosswind speed, $\langle v_\perp \rangle$, and the time delay to the peak, τ_p,

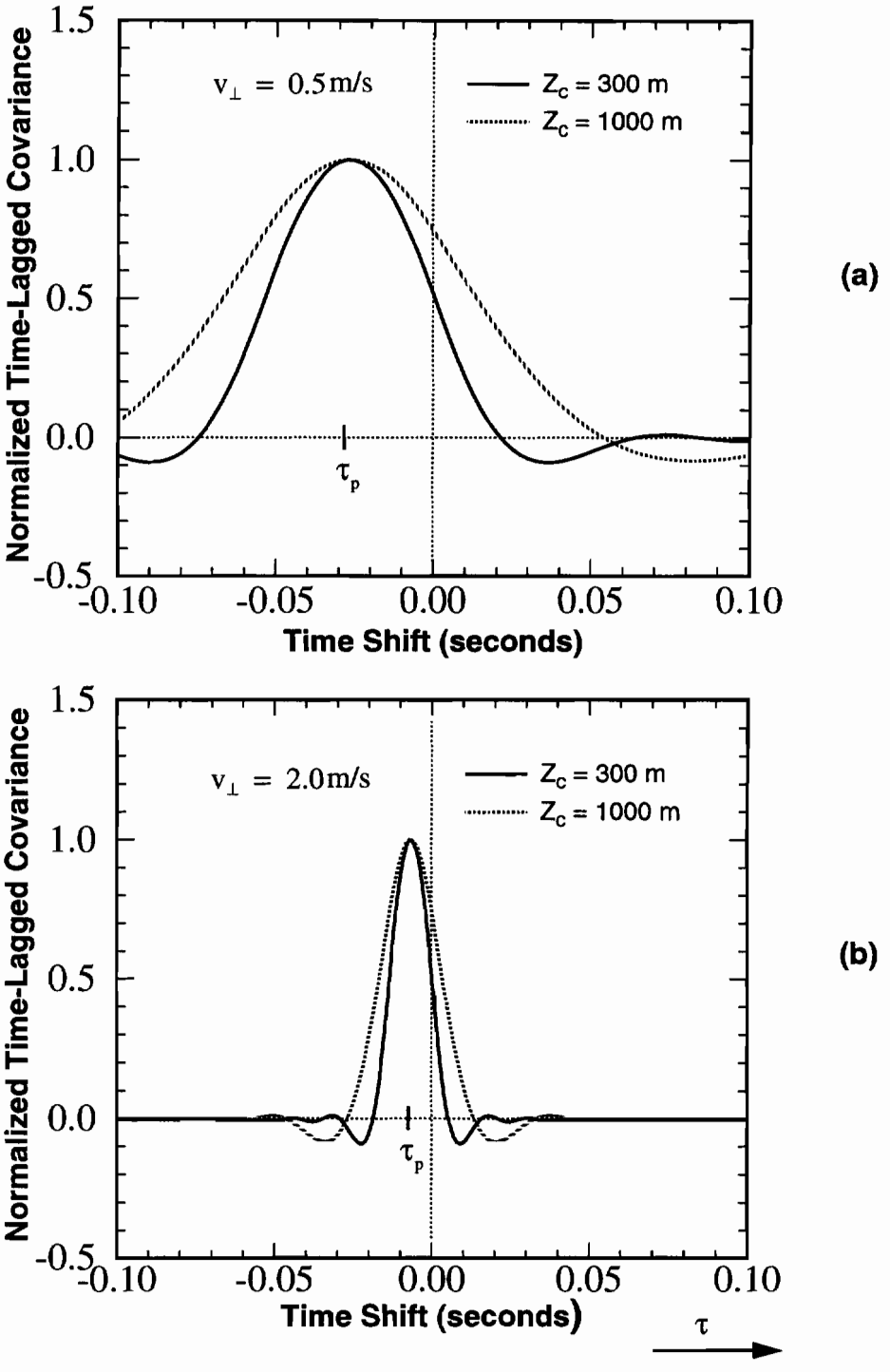


Figure 5.21 Normalized time-delayed crosscovariance of intensities. As the crosswind speed increases, the time delay to the peak shifts toward the origin. (a) crosswind speed = 0.5m/s. (b) crosswind speed = 2.0m/s.

is found to be

$$<v_\perp> = \frac{p}{2\tau_p}. \tag{5.34}$$

One method of estimating the crosswind speed is by finding the time delay to the peak, and then using equation (5.34). The time delay to peak method (TDTP) is attractive because it does not require a knowledge of C_n^2, or any other statistical quantities. The TDTP method also gives the wind direction depending on which side of the origin the peak is shifted.

Another way to estimate the crosswind speed is by computing the slope of the crosscovariance curve at zero time delay (SZTD). The slope at zero time delay is derived by taking the slope of equation (5.32) with respect to τ, then letting τ be zero

$$\left.\frac{\partial C_{N_{ipf}}}{\partial \tau}\right|_{\tau = 0} = \text{SZTD} = (<\vec{v}_\perp> \cdot \vec{p})\, 0.132\pi^2 k^2 C_n^2 z_c \int d\omega \;\; \omega \int du \;\; u^{-8/3} \sin^2\left[\frac{\omega\,(1-\omega)\,u^2 z}{2k}\right]$$

$$\times J_1\left[u|\omega\vec{p}|\right] \tag{5.35}$$

The SZTD can be normalized by the covariance (crosscovariance at zero time delay, $C_{N_{ipf}}(\vec{p}, 0)\,|_{\tau = 0}$ = CZTD) to get rid of C_n^2. Solving for the average crosswind speed gives

$$<v_\perp> = \frac{1}{p}\frac{\text{SZTD}}{\text{CZTD}}\frac{\text{INT1}}{\text{INT2}} \tag{5.36}$$

where

$$\mathrm{INT1} = \int d\omega \int du \; u^{-8/3} \sin^2\left[\frac{\omega(1-\omega)u^2 z_c}{2k}\right] J_0\left[u|\omega\vec{\rho}|\right]$$

and

$$\mathrm{INT2} = \int d\omega \; \omega \int du \; u^{-8/3} \sin^2\left[\frac{\omega(1-\omega)u^2 z}{2k}\right] J_1\left[u|\omega\vec{\rho}|\right].$$

The SZTD method also gives the wind direction, depending on the sign of the slope. Both of these crosswind speed detection techniques, as well as several others, are evaluated by Wang, Ochs, and Lawrence.[67] From computer simulation, they find that both methods give the correct crosswind speed estimate when the crosswind speed and C_n^2 are constant along the whole path. But when the wind fluctuates along the path, the slope method is more accurate. Both methods will be used to estimate the cross-wind speed.

5.3.1 Aperture Averaging

Lidars that use optical heterodyne detection must be designed carefully to avoid aperture averaging. So far, the statistics for a CO_2 laser backscattered from aerosols have been derived for two point detectors. But the physical reality is that actual lidars have some finite receiving area to collect the backscattered radiation. Because heterodyne systems measure the amplitude and phase of the backscattered laser rather than the intensity, a receiving aperture that is too large could cause the heterodyne signal current to decrease because of averaging of different phases of the cosine. Figure 5.22 shows the size of the equivalent receiving aperture of the lidar used in this thesis project. Each detector gets one of the half moons with radius of

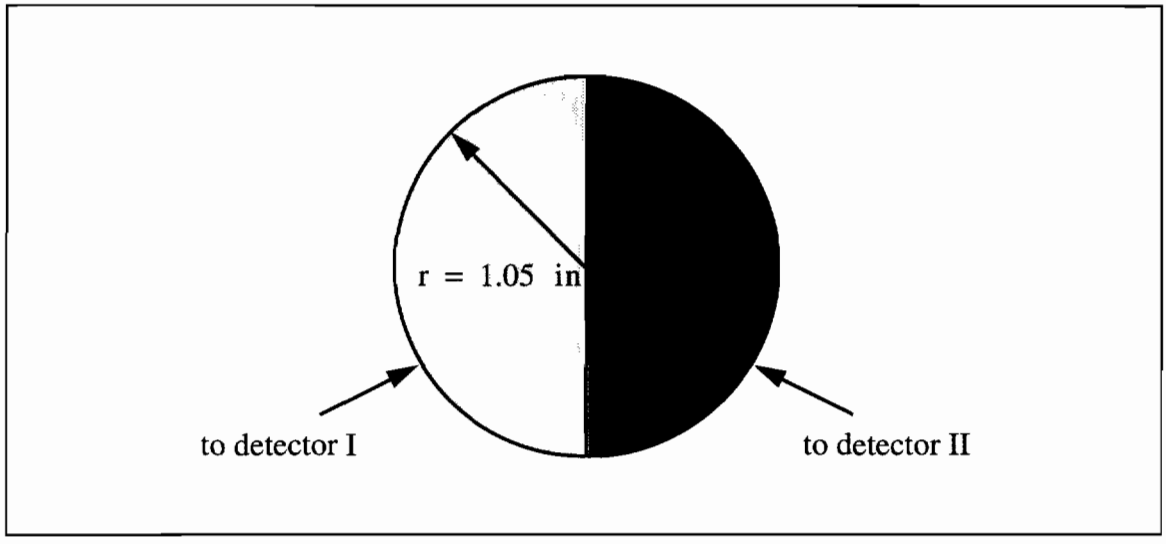


Figure 5.22 The equivalent detector aperture area. Each detector gets one of the half moons with radius 1.05 inches.

1.05 inches. The aperture averaging problem was initially investigated by D. L. Fried for a circular aperture.[68] To perform the analysis correctly for the two half moons, the statistics need to be rederived to include the finite aperture. One way to illustrate the effect of aperture averaging is to examine the effect on the heterodyne signal power defined by

$$\langle i_d^2 \rangle = \iint d\vec{\rho}_1 d\vec{\rho}_2 \Gamma(\vec{\rho}_1, \vec{\rho}_2) U_{LO}(\vec{\rho}_1) U_{LO}^*(\vec{\rho}_2) \tag{5.37}$$

where $U_{LO}(\vec{\rho})$ is the local oscillator electric field and $\Gamma(\vec{\rho}_1, \vec{\rho}_2)$ is the mutual intensity function defined by equation (3.21) with $\tau = 0$. The geometry for the integrations is shown in figure 5.23. The detector variables $\vec{\rho}_1$ and $\vec{\rho}_2$ cover the entire half moon for one of the detectors. Setting $\tau = 0$ and completing the integrations left undone in equation (3.21) gives the mutual intensity function at 0 time delay

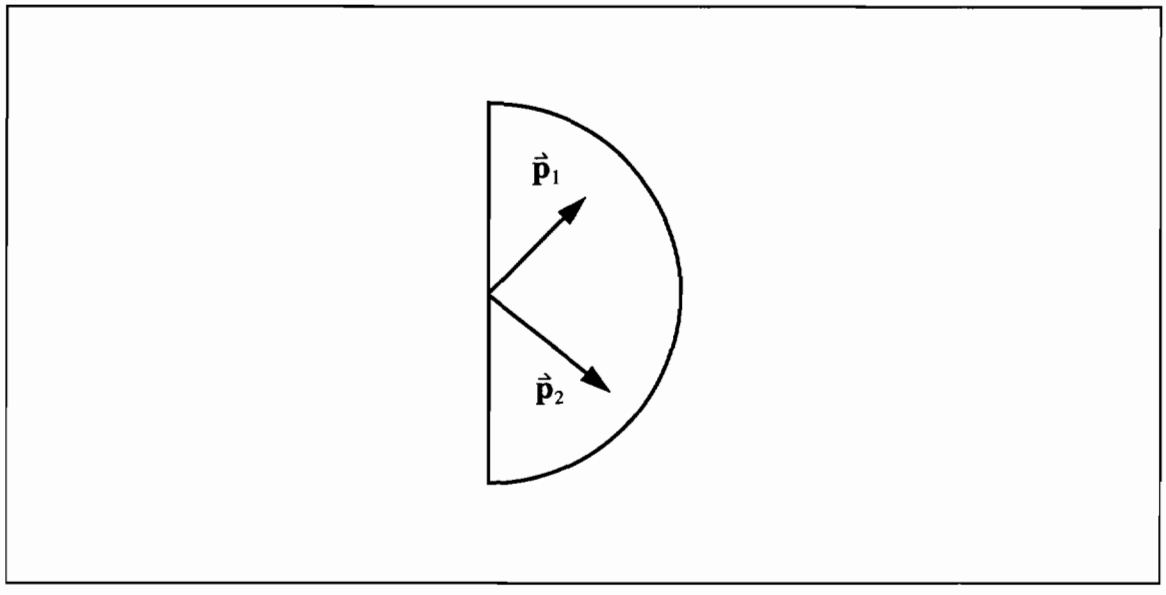


Figure 5.23 Geometry for calculating the average heterodyne signal power to evaluate the effect of aperture averaging.

$$\Gamma(\vec{p}_1, \vec{p}_2) = \frac{kU_T^2\beta}{2\pi}\iint dz d\vec{R}\left(\frac{1}{z}\right)^3 \exp\left[-\frac{R^2}{\alpha_0^2} - \frac{1}{4\alpha_0^2}(p_1^2 - p_2^2)\right]$$

$$\times \exp\left[i\frac{k}{z}\vec{R}\cdot(\vec{p}_1 - \vec{p}_2)\left(1 - \frac{z}{F_T}\right)\right]\exp\left[\frac{ik}{2z}(p_1^2 - p_2^2)\right]$$

$$\times H(\vec{R}, \vec{p}_1, \vec{p}_2) \tag{5.38}$$

where,

$$H(\vec{R}, \vec{p}_1, \vec{p}_2) = \exp\left[-D_\phi(\vec{p}_1 - \vec{p}_2) + D_\phi\left(\vec{R} - \frac{1}{2}(\vec{p}_1 + \vec{p}_2)\right) - D_\phi\left(\vec{R} - \frac{1}{2}\vec{p}_1 - \frac{3}{2}\vec{p}_2\right)\right]$$

$$\times \exp\left[-\frac{1}{2}D_\phi\left(\vec{R} - \frac{3}{2}\vec{p}_1 + \frac{1}{2}\vec{p}_2\right)\right]\exp\left[4C_\chi\left(\vec{R} - \frac{1}{2}(\vec{p}_1 + \vec{p}_2)\right)\right] \tag{5.39}$$

and α_0 and F_T are the transmitted beam radius and focal length, respectively. The local oscillator electric field is described by

$$U_{LO}(\vec{p}) = \exp\left[-\frac{p^2}{2}\left(\frac{1}{\beta_0^2} + i\frac{k}{F_{LO}}\right)\right] \tag{5.40}$$

where β_0 and F_{LO} are the LO beam radius and focal length back propagated through the receiver optics, respectively. Inserting equations (5.40), (5.39) and (5.38) into equation (5.37), and changing the radius of the aperture produces the curves plotted in figure 5.24. The top curve is the signal power if there were no aperture averaging, included as reference for the other heterodyne signal power terms, and is equivalent to the received power for a direct detection system. The detector radius has been normalized by the Fresnel zone scale size, $\sqrt{z_c/k}$, to make the result more general. For small values of the receiver radius, the heterodyne signal power increases at approximately the same rate as the signal power with no aperture averaging. As the radius approaches the Fresnel zone scale size, however, the signal power deviates dramatically from the signal power with no aperture averaging. For detector radii beyond the Fresnel zone scale size, the heterodyne signal power actually decreases. Therefore, increasing the receiving aperture area beyond the Fresnel zone scale size gives no benefit for heterodyne detection systems. For a z_c of 400 meters, the Fresnel zone scale size for λ = 10.6μm is 0.026m or 1.023 inches. The receiving aperture of 1.05 inches for this lidar is ideal to maximize the heterodyne signal power, and therefore, the SNR.

Setting the receiving aperture at 1.05 inches, the path weight function can be computed using equation (5.37) again, varying z_c instead of the detector radius. Figure 5.25 shows the path weight function or the theoretical Intensity vs. Range for a

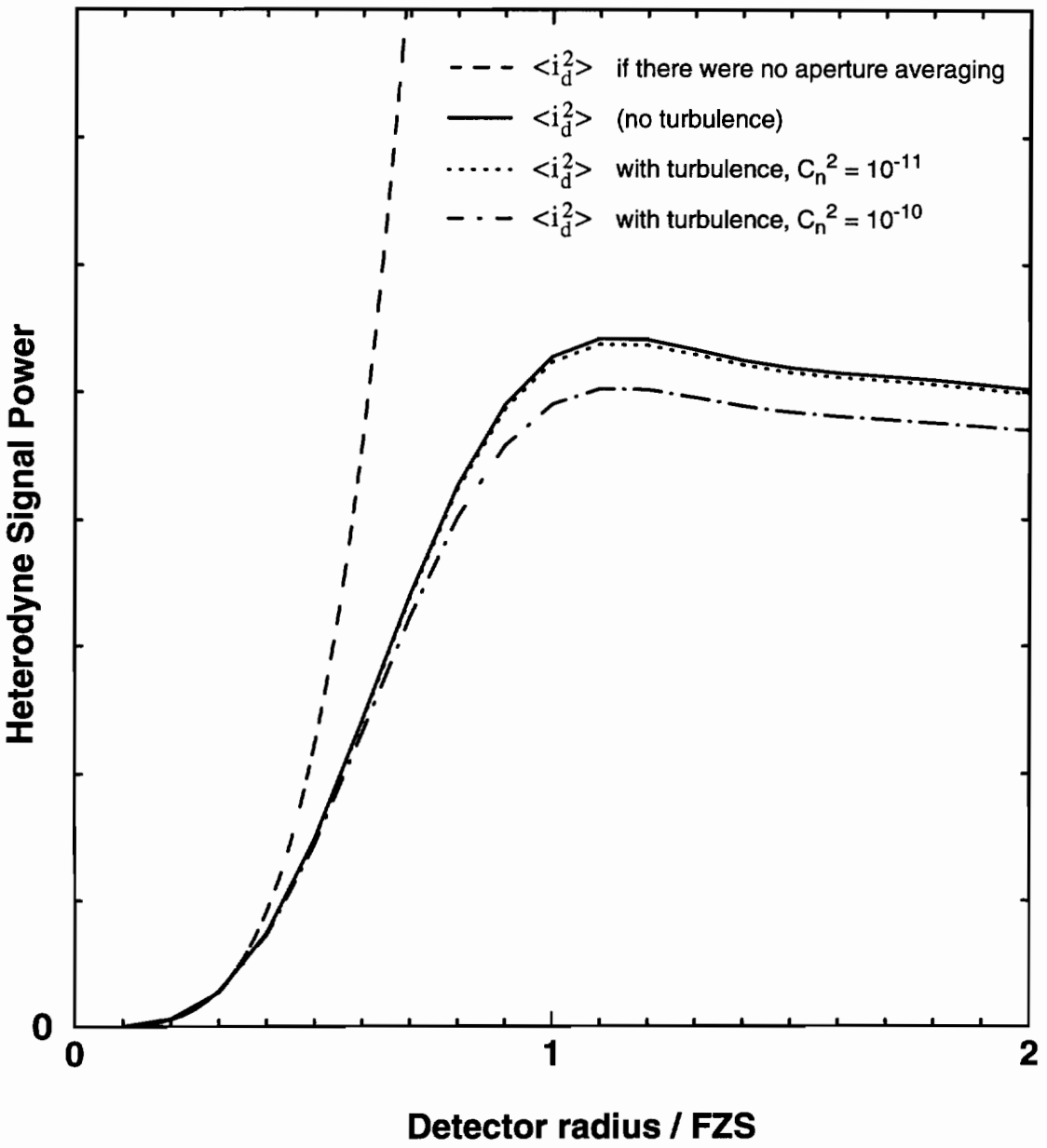


Figure 5.24 Heterodyne signal power plotted as a function of detector radius normalized by the Fresnel zone scale size FZS = $\sqrt{z_c/k}$.

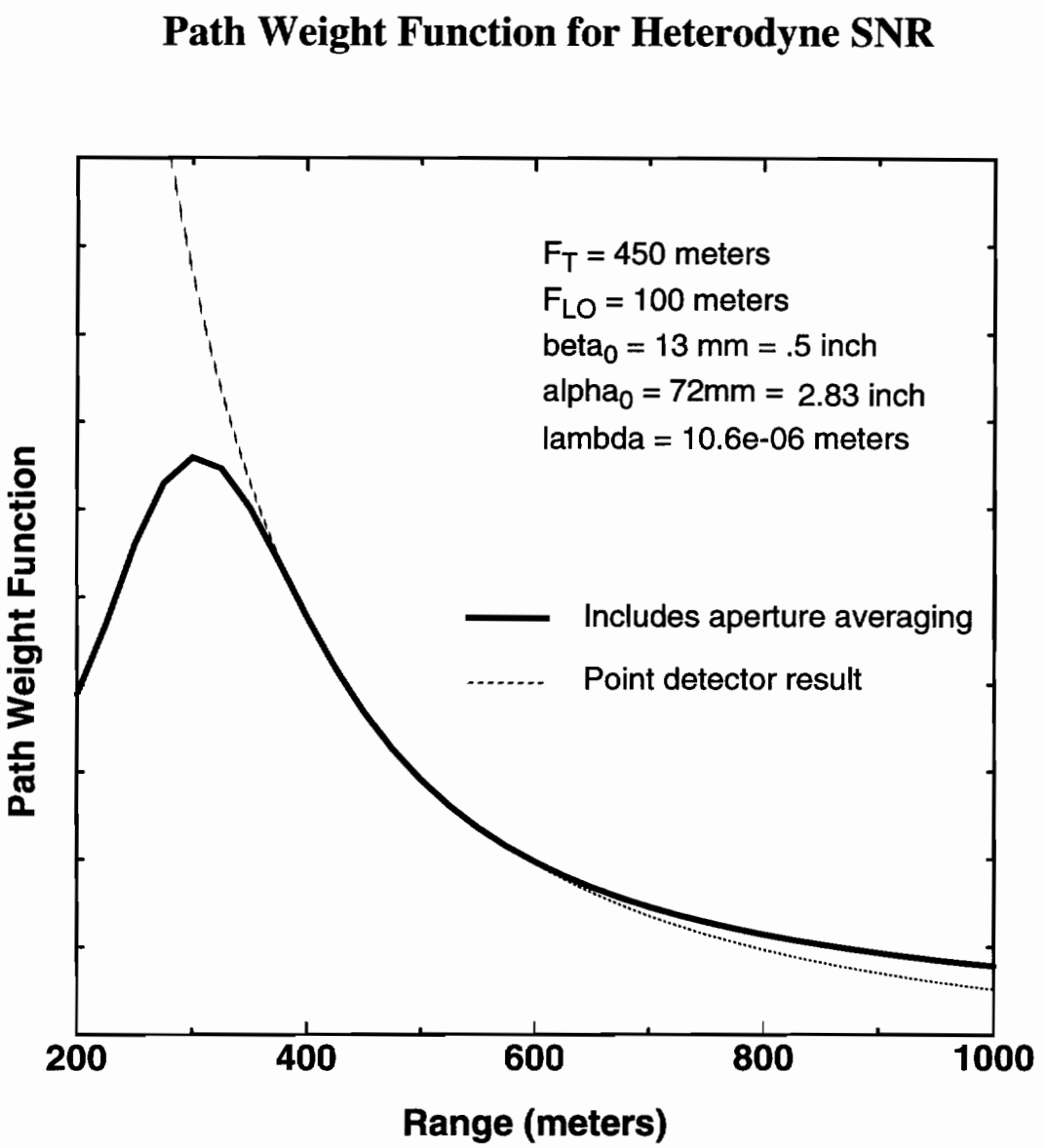


Figure 5.25 Path weight function for a CO_2 heterodyne coaxial lidar backscattered from aerosols. Result considers actual finite receiving aperture. For reference, the point detector path weight function is included.

CO_2 heterodyne coaxial lidar backscattered from aerosols. For reference, the theoretical path weight function for a point detector is included. The difference in the curves is due to the aperture averaging over the detector area, and the fact that the local oscillator and backscattered wavefronts must match to produce a heterodyne signal current for a heterodyne lidar.

The time-delayed crosscovariance of intensities for two point detectors is used as an approximation to develop the crosswind detection schemes. The equivalent detector spacing is the distance between the centers of the two half moons, or $p = 1.05in$. The computer programs used to compute the time-delayed crosscovariance and the aperture averaging effects, as well as the processing programs are listed in Appendix D. The electronics of the PRC generator and of the amplifiers for both channels are sketched in Appendix A.

This chapter introduced a complete description of the experimental lidar used in this thesis project. The laser system is a cw, heterodyne system operating at 10.6 microns. In order to obtain path resolved measurements, the outgoing laser has been pseudo-random code modulated through the use of an acousto-optic modulator to implement the phase changes. The Signal to Noise Ratio (SNR) was examined to include photo diode saturation effects which indicated that heterodyne lidars actually operate below the local oscillator signal shot noise limit, a common assumption when working with heterodyne lidars. Two processing schemes for obtaining crosswind estimates were also developed. Finally, the effects of aperture averaging were examined by including the finite aperture size in the average intensity calculations.

Chapter 6

Experimental Results

The CO_2 lidar described in chapter 5 was designed, constructed, and tested at the Oregon Graduate Institute of Science and Technology over a period of three years, from the summer of 1991 to the fall of 1994. The laser is discharged out of a window at the west end of student trailer #1 and propagates over the terrain presented in the aerial photograph in figure 6.1. The ideal propagation path would be flat and featureless, so that the atmospheric quantities such as C_n^2 and the backscattering coefficient, β, would be constant over the path. As can be seen from figure 6.1, the beam travels over grass for about the first 300 meters, beyond which is a wheat field. The wheat field extends another 500 meters to a grassy pasture outlined by fir trees. The laser is angled up by about 5 degrees to avoid the trees when backscattering from aerosols. There are also three roads encountered along the course. Unfortunately, the terrain is not uniform but it is conveniently located close to the Institute.

Because of the poor SNR at the receiver due to the small fraction of radiation backscattered from aerosols, most days in the Northern part of the Willamette valley of Oregon are not adequate for taking data. The frequent rain washes out the aerosols on most days, and persistent cloudy weather produces a low C_n^2, causing an interaction between the atmosphere and the CO_2 laser beam that is too weak to detect crosswinds. A window of about 2 1/2 months exists from about the middle of July to the end of September with hot, sunny, and breezy weather ideal for aerosol backscattering and crosswind detection. There are some additional days in winter months if there are

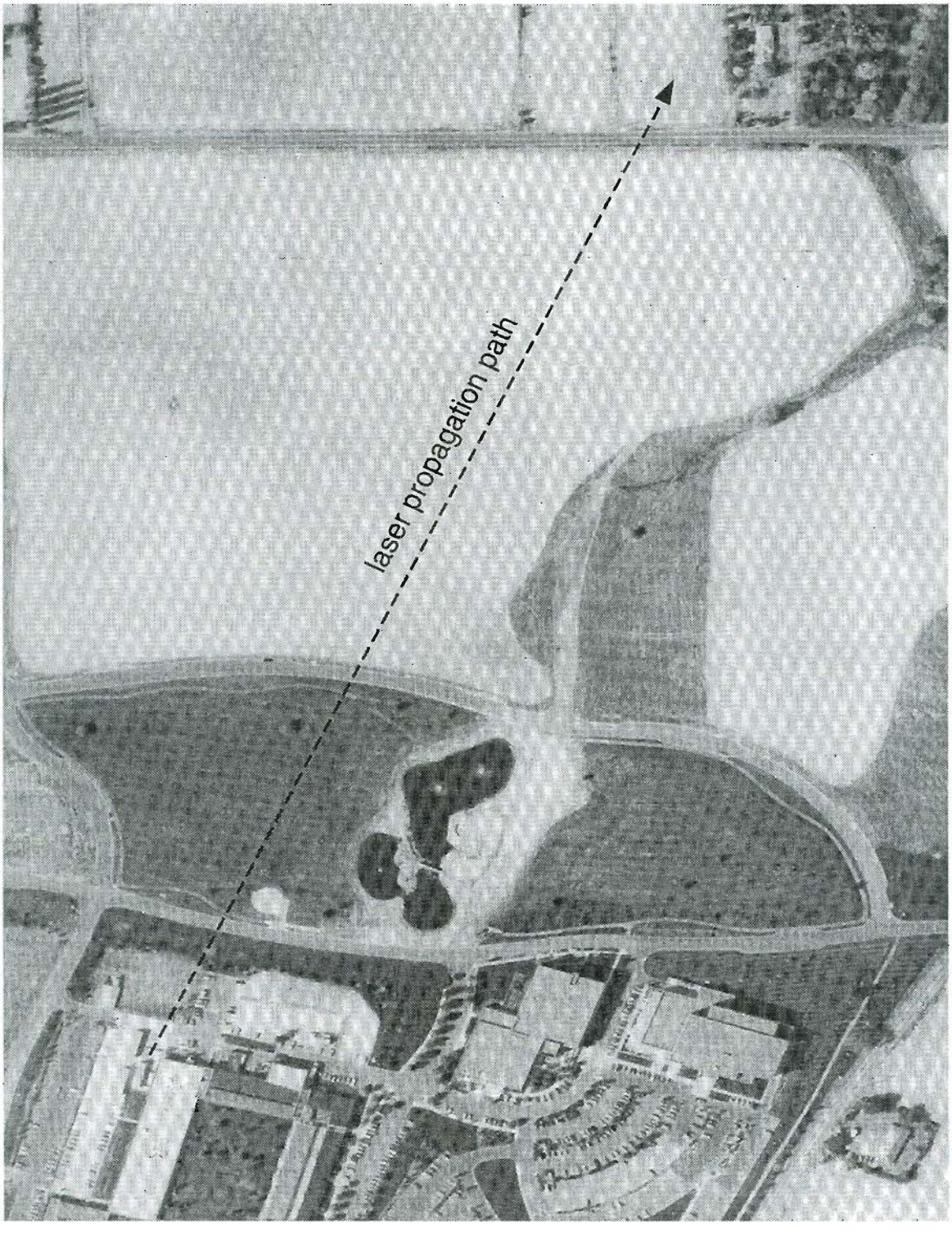


Figure 6.1 Aerial photograph of the Oregon Graduate Institute showing the laser propagation path.

4-6 consecutive days without rain and it is sunny. In general, the best time to take data is during the second half of the summer months. Oregon's clean environment, coupled with the fact that the lidar is operated in a rural location, requires the entire system to be optimized in order to detect enough backscattering and operate above threshold.

The heterodyne signal is proportional to the amplitude of the backscattered radiation, modulated by the PRC, centered at 6 MHz ± the Doppler shift. The PRC has a bandwidth of 4 MHz including the first two zeroes of its spectrum. Depending on the Doppler shift, the 4 MHz wide heterodyne signal moves up and down the frequency spectrum. The amplifiers were designed to detect Doppler shifts up to ±2MHz corresponding to a 10 m/s (22.4 mph) radial wind speed. The required bandpass of the detectors for these specifications is 2 to 10 MHz. The maximum number of data samples that can be recorded by the CS220 digitizer is 4 megabytes in single channel mode, or 2 megabytes for dual channel mode. For a sampling rate of 20 MHz, the sampling time is 100 milliseconds. After the CS220 resident memory is full, the data must be extracted and stored. The amount of time required to transfer both channels of data from CS220 memory to PC hard drive is 10 seconds. Therefore, 2 megabytes of data per channel can be recorded and stored on the PC's hard drive every ten seconds.

The in situ data is taken by a Campbell Scientific CA-9 path averaging cross-wind speed and C_n^2 detector. The Campbell unit is part of a double-ended laser system with a 5 mW HeNe laser at one end as the source and the Campbell unit as receiver at the other. Detecting the radial winds from the Doppler shift is quite reliable and accurate. Therefore, there were no in situ instruments used to measure the radial winds. The HeNe laser and Campbell unit were located about 2 meters above the ground. Because the CO_2 laser needed to be angled up above the tree line, its

path is slightly different than the one used by the Campbell unit.

An example of the heterodyne signal current for the lidar backscattered from aerosols is shown in figure 6.2a. The total time represented in the graph is 40 microseconds, showing several speckle fluctuations centered at about 2, 11, 22, and 31 microseconds. The time scale of the speckle fluctuations is on the order of 3-5 microseconds which coincides with the intensity decorrelation times predicted by the theory of chapter 3. The backscattering range profile for this data set is obtained by computing the processed intensity vs. range, shown in figure 6.2b. Most of the backscattering occurs at about 280 meters. The path weight function is not a perfect triangle, as it would be for a hard target, but most of the return signal comes from a region around 280 meters. The Doppler shift of this data is found by taking a FFT of the digitized data that is multiplied by the PRC delayed to 280 meters. Figure 6.2c presents the FFT spectrum. The FFT length is 1024 points and there are 500 consecutive 1024-point FFT spectra averaged to produce figure 6.2c. Averaging consecutive spectra reduces the noise variance and enhances the peak frequency, making it easier to identify the peak frequency and Doppler shift over the data run. The Doppler shift is 0.2 MHz, indicating a radial wind speed of v_z = 1.06m/s. The humps at ± 1MHz from the peak frequency represent the part of the return signal from different scattering locations other than 280 meters not recovered perfectly by the PRC delayed to 280 meters.

A more interesting range profile is presented in figure 6.3a. The double humped intensity vs. range indicates scattering from two different parts of the propagation path. Figure 6.3b shows the processed Doppler shift vs. range provided as reference to the range profile. The scattering around 100 meters has a Doppler shift of -0.07 MHz, while the scattering from around 320 meters has a Doppler shift of 0.48 MHz. This is interesting because the lidar has detected the z-directed winds blowing

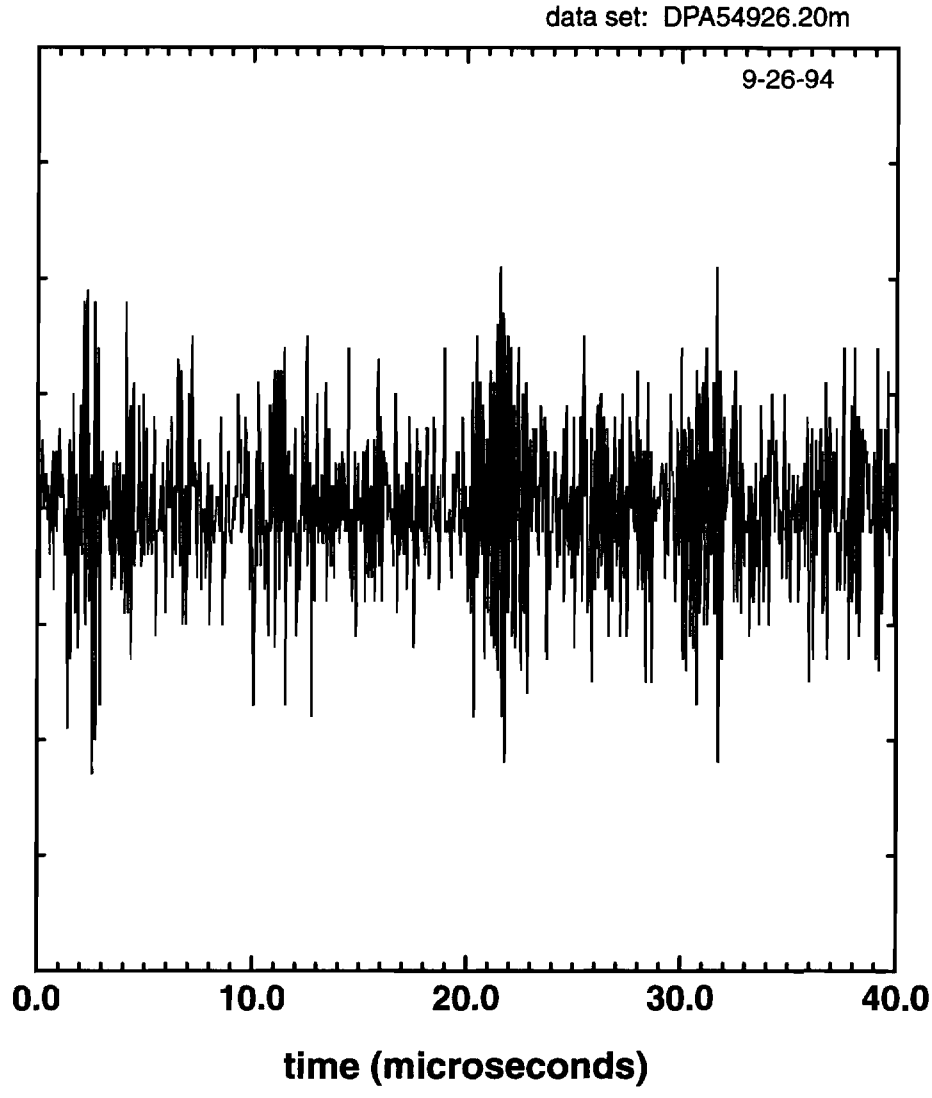


Figure 6.2a Digitized laser signal current backscattered from aerosols. Speckle fluctuations centered at 2, 11, 22, and 31 microseconds indicate the time scale of the speckle fluctuations is on the order of 3-5 microseconds.

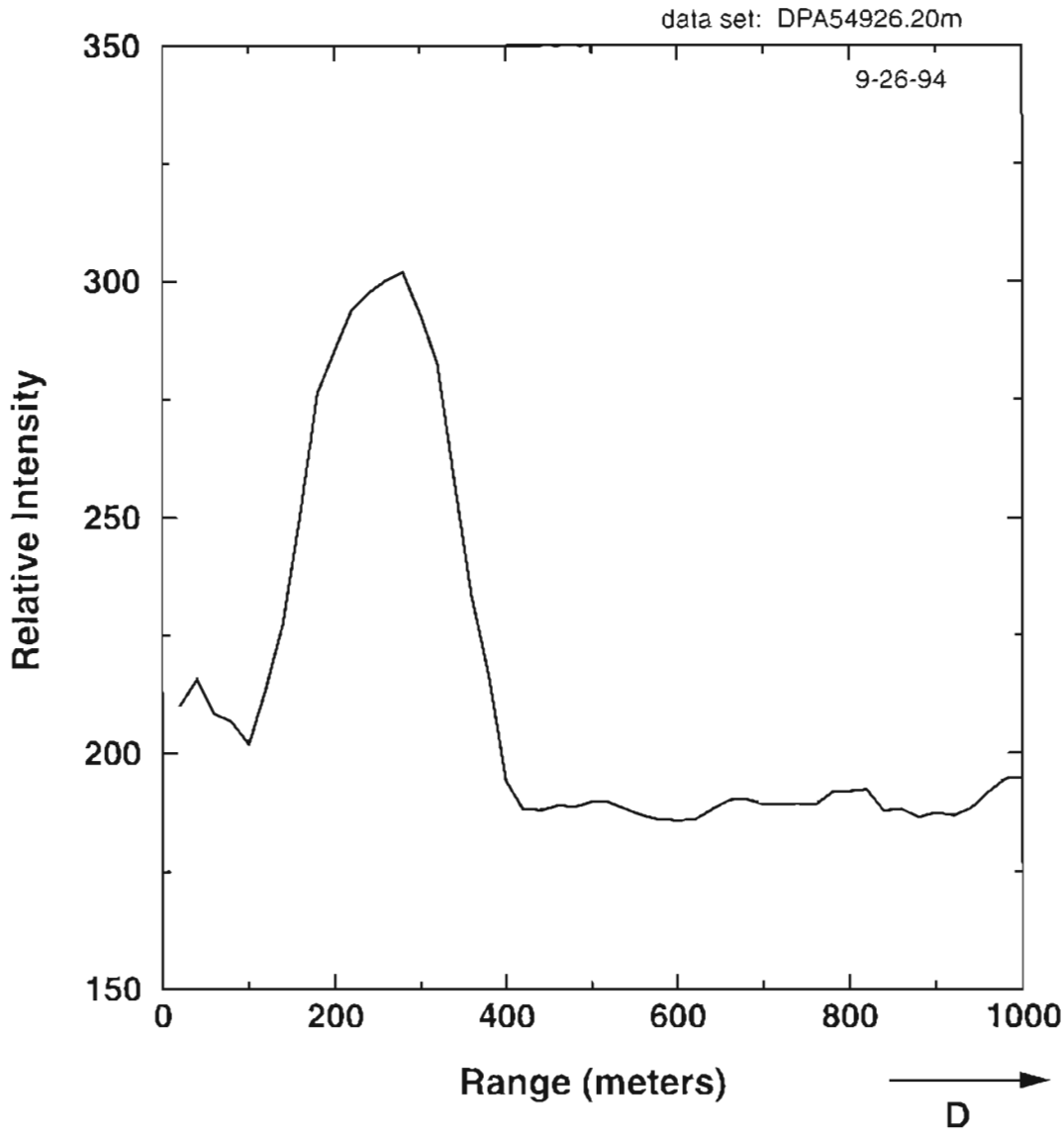


Figure 6.2b Processed Intensity vs. Range indicating that most of the laser radiation backscattered from location z = 280 meters.

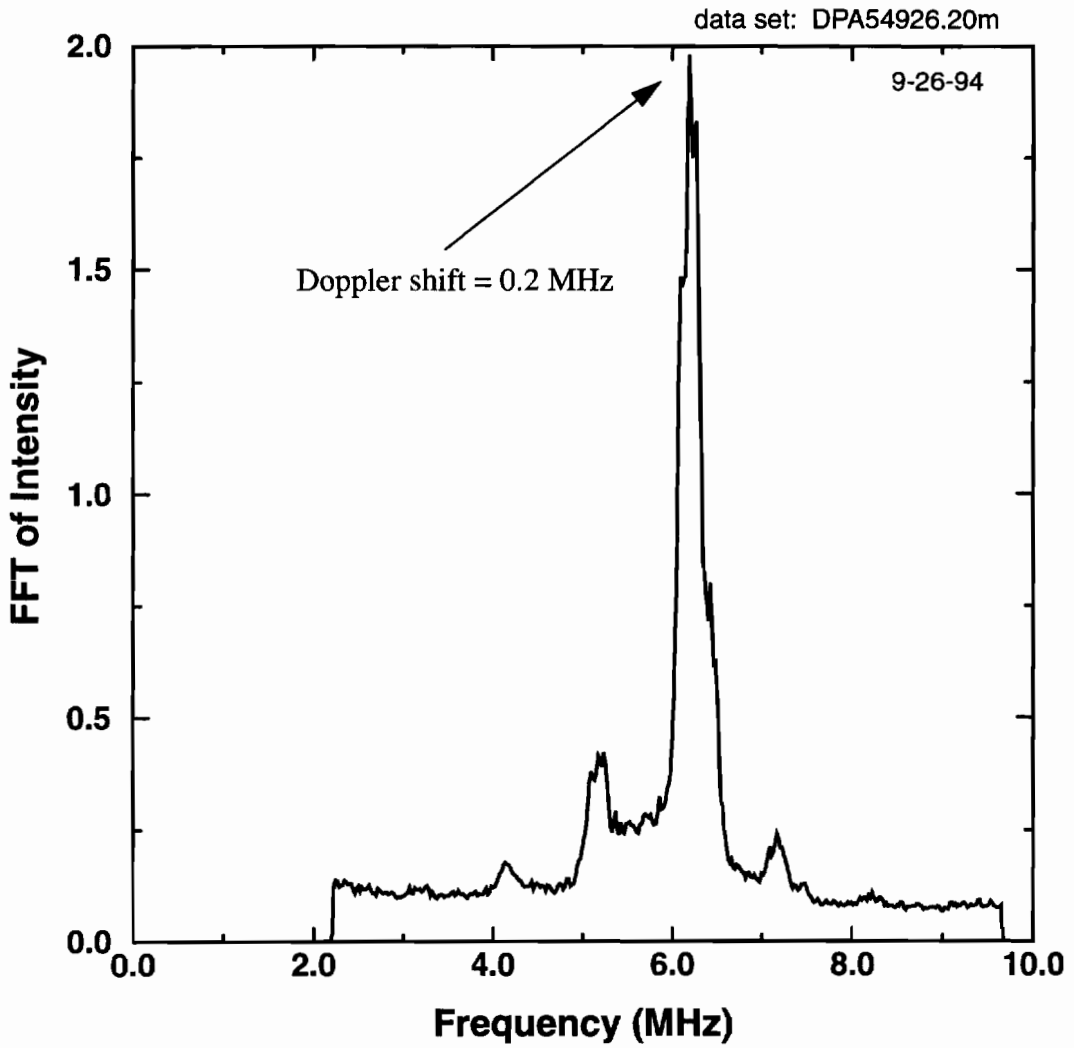


Figure 6.2c Average of 500 contiguous FFT spectra, N = 1024-point FFT's. Data represents an average over 25 milliseconds. Data has been processed with the PRC delayed to 280 meters, as suggested by figure 6.2b.

in opposite directions at different parts of the path. The noisy Doppler shift vs. range beyond 450 meters in figure 6.3b is due to the SNR being below threshold for ranges longer than 450 meters.

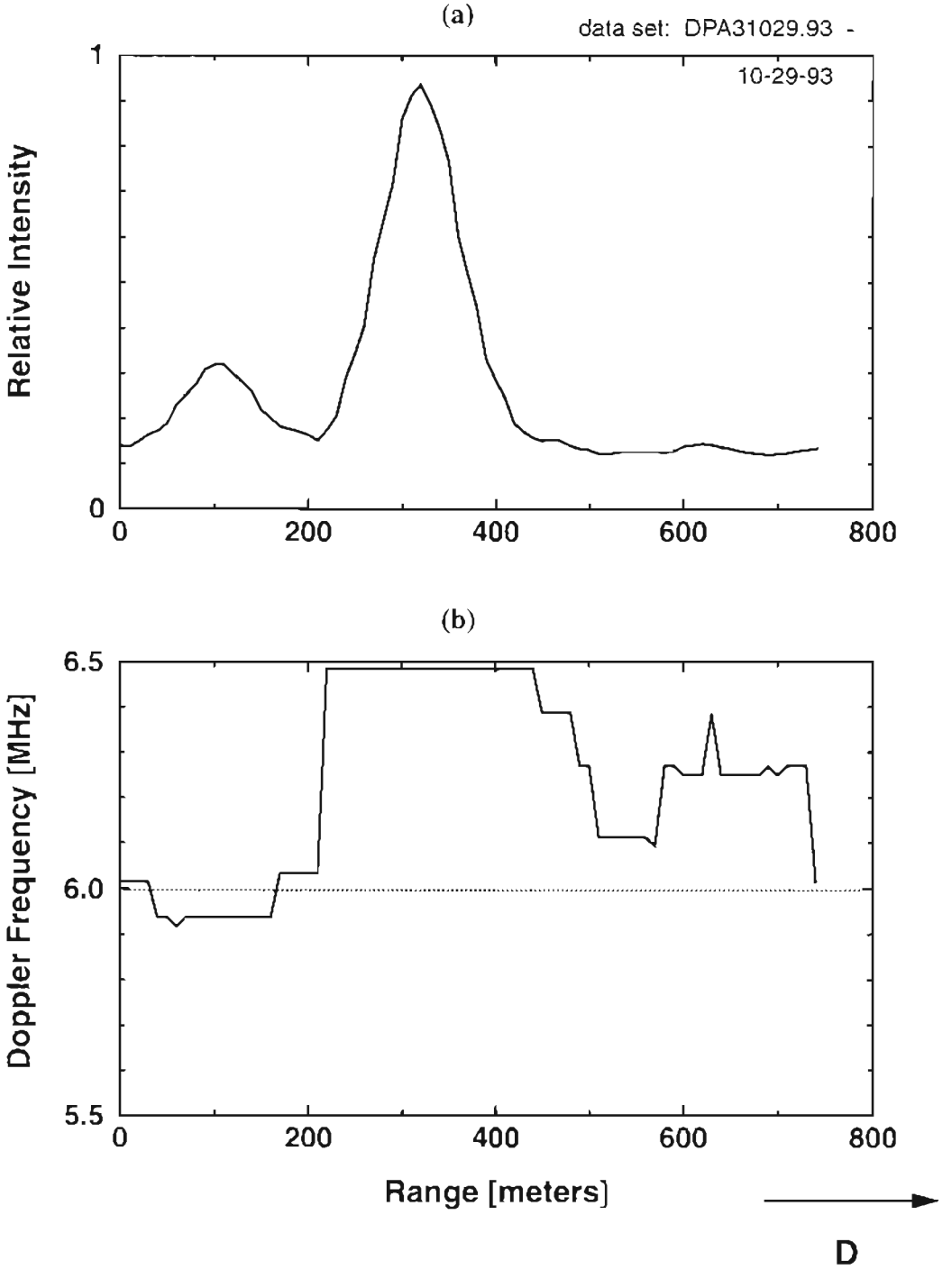


Figure 6.3 Processed intensity vs. range showing backscattering from two different ranges. (a) intensity vs. range (b) Doppler frequency vs. range.

6.1 Lidar Used as Doppler/Range Sensor

If only the Doppler shift and intensity vs. range are desired, then only one detector can be used. Since the backscattered field is kept intact, instead of splitting into two beams, there is twice as much signal directed to a detector than the two receiver system. Less data can be taken than when trying to detect crosswinds because the z-directed wind speed is found from the Doppler shift of the FFT. It was found that running experiments to estimate the intensity vs. range and z-directed wind speed could be performed on most days of the year, including winter days after heavy rains, because the only quantity needed is the Doppler shift from one of the detectors. For these experiments, the split receiver was removed and the entire backscattered field was directed to detector #1. Each data run consisted of 24 sets of 25.6 milliseconds (512k points) of continuous data samples.

An example of a data run to detect the intensity vs. range and Doppler shift is shown in figures 6.4a-g. The intensity vs. range profiles are plotted in figure 6.4a and 6.4b. Each consecutive data set is separated in time by 6 seconds. Sets 1 through 7 had some strong backscattering but the strength dies slightly from sets 8 to 17. The backscattering strength picks up again for sets 18 through 24. The total time represented in figures 6.4a and b is 144 seconds (2.4 minutes). Over this time the center of strongest backscattering meanders between 250 and 420 meters. In sets 7 through 19 there appears to be some backscattered radiation from very short ranges, near 0 to 20 meters. This is not backscattering from aerosols but is actually feedback noise caused by reflections from the transmitter path AOM crystal surface passing through to the local oscillator and on to the detector. Sometimes this feedback noise was so detrimental that experiments needed to be halted until this noise level could be reduced by rotating the AOM slightly off the peak reflection angle. The feedback

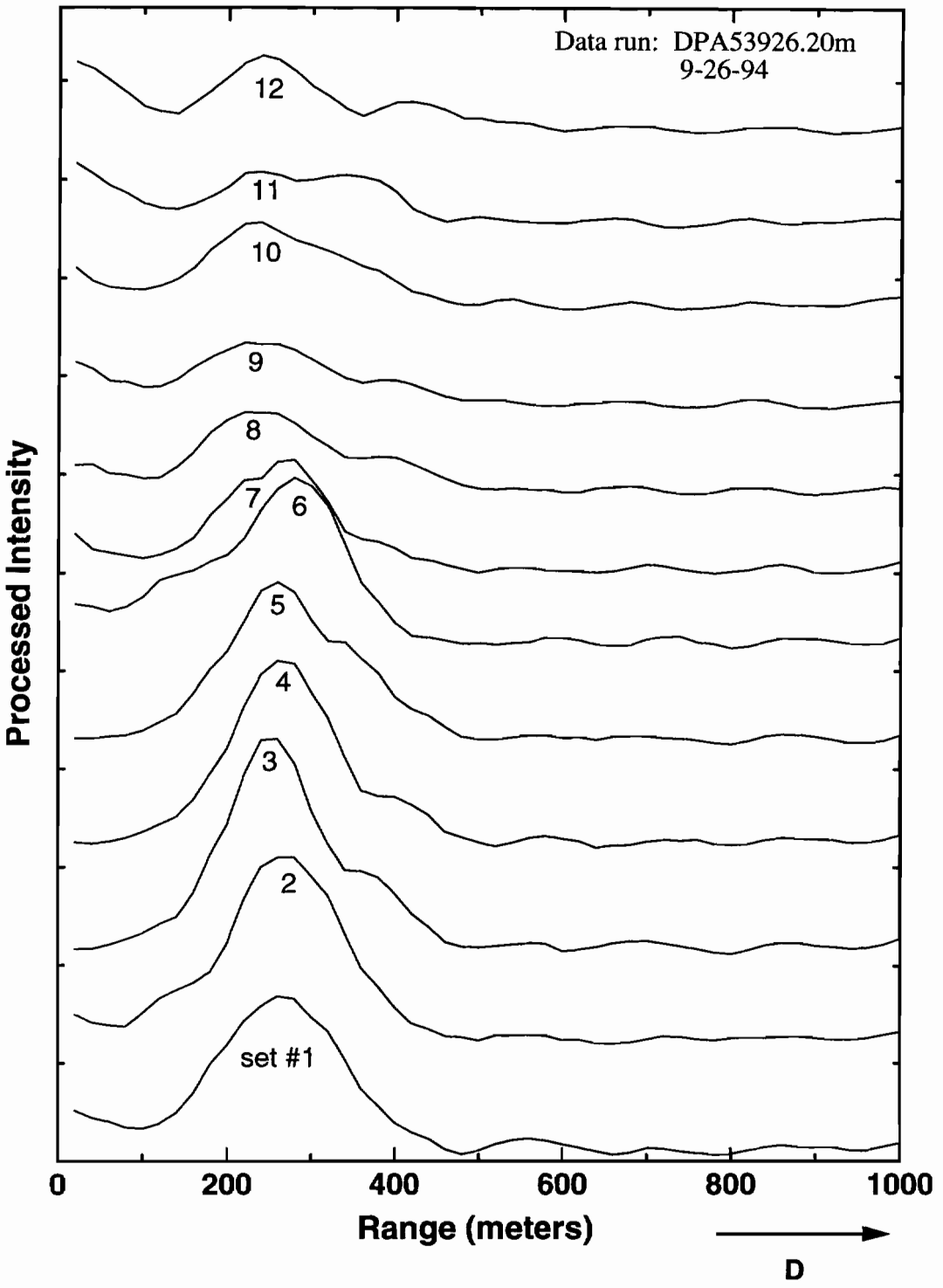


Figure 6.4a Sets 1 through 12 of the intensity vs. range for data run DPA53926.20m. Each set represents an average of 25.6 msec (512K points).

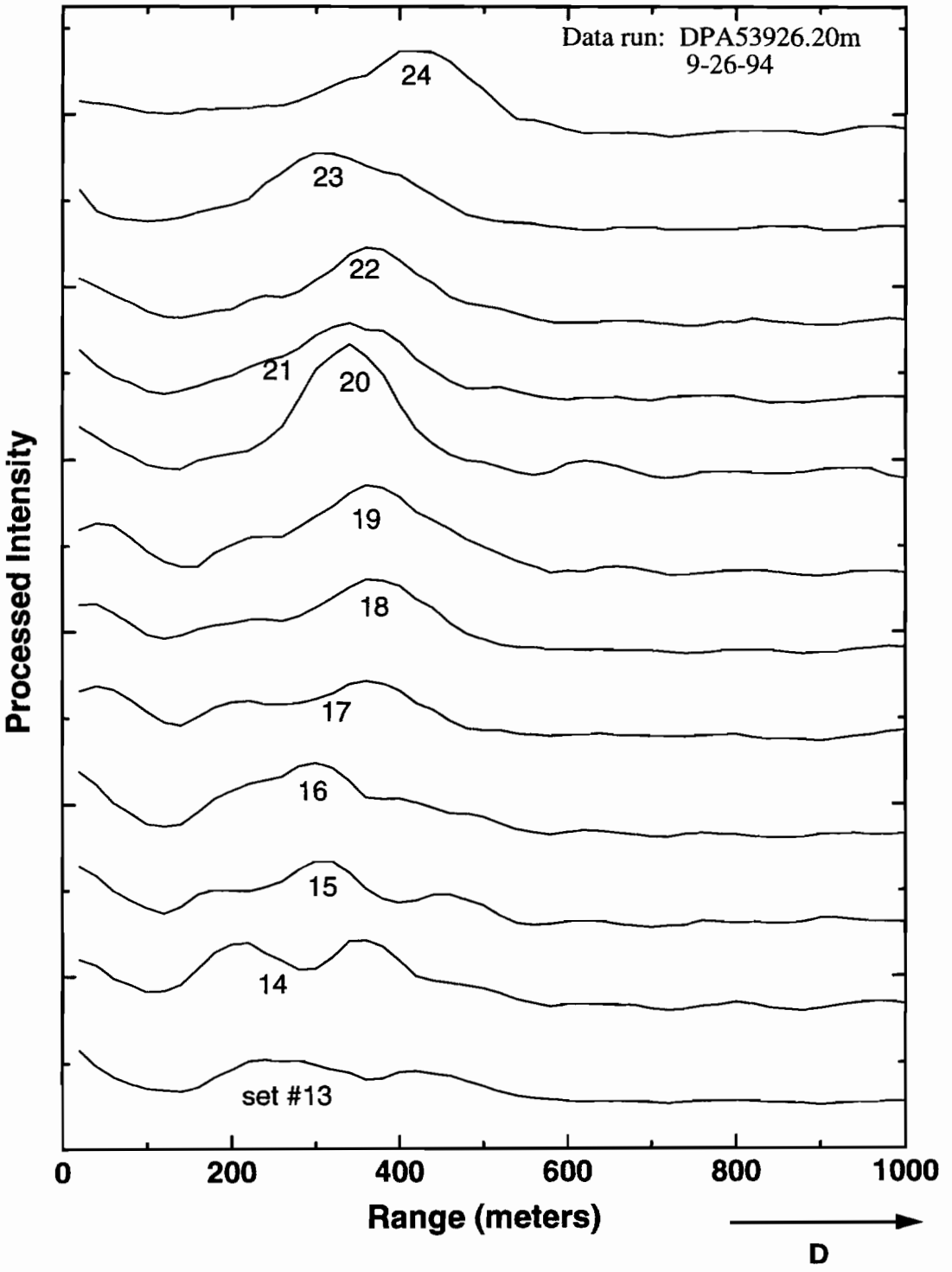


Figure 6.4b Sets 13 through 24 of the same data run as figure 6.4a. Each set represents an average of 25.6 msec (512K points).

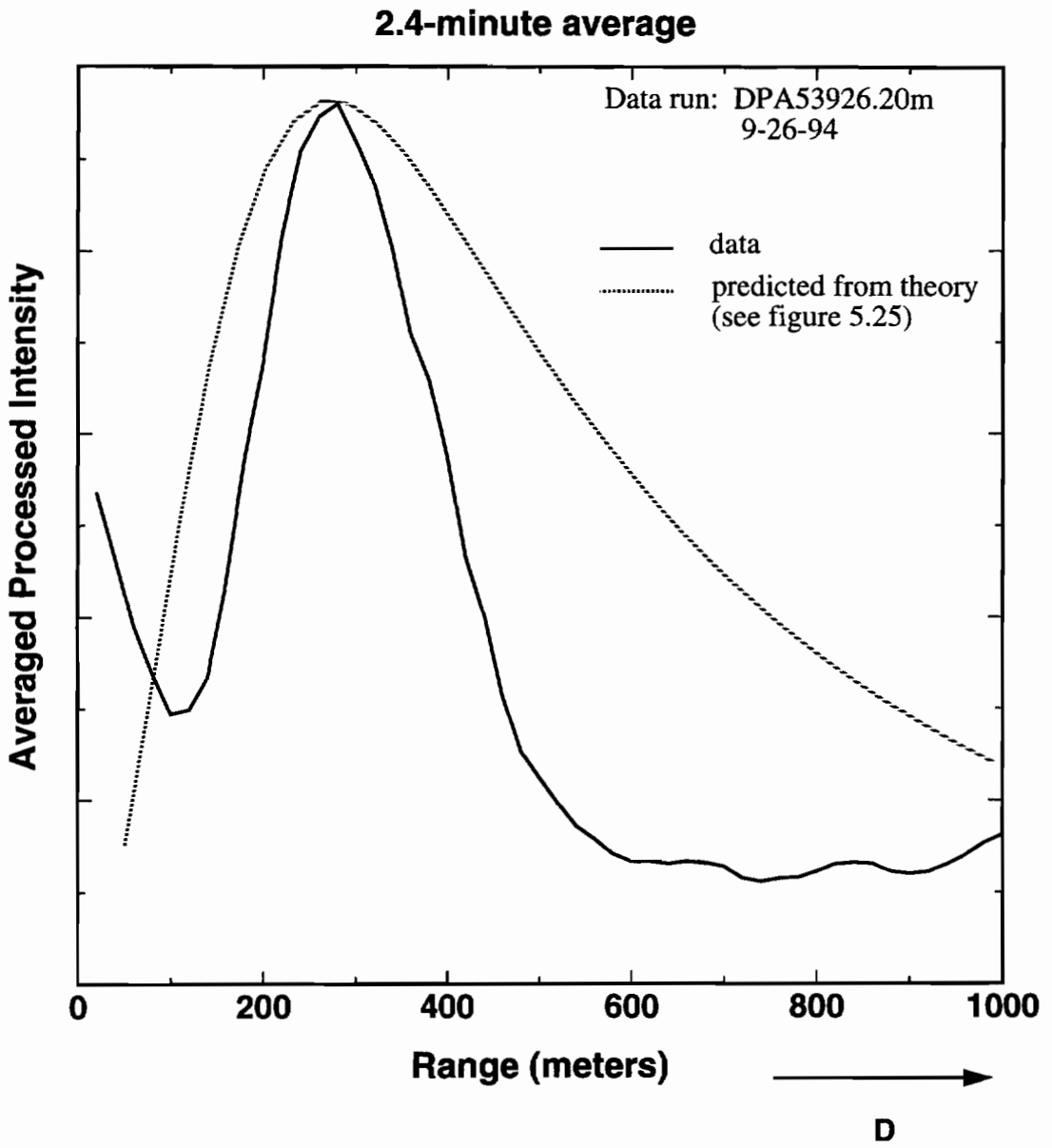


Figure 6.4c The intensity vs. range averaged over 2.4 minutes for data run DPA53926.20m. The dotted line is the path weight function predicted from the theory at the end of Chapter 5.

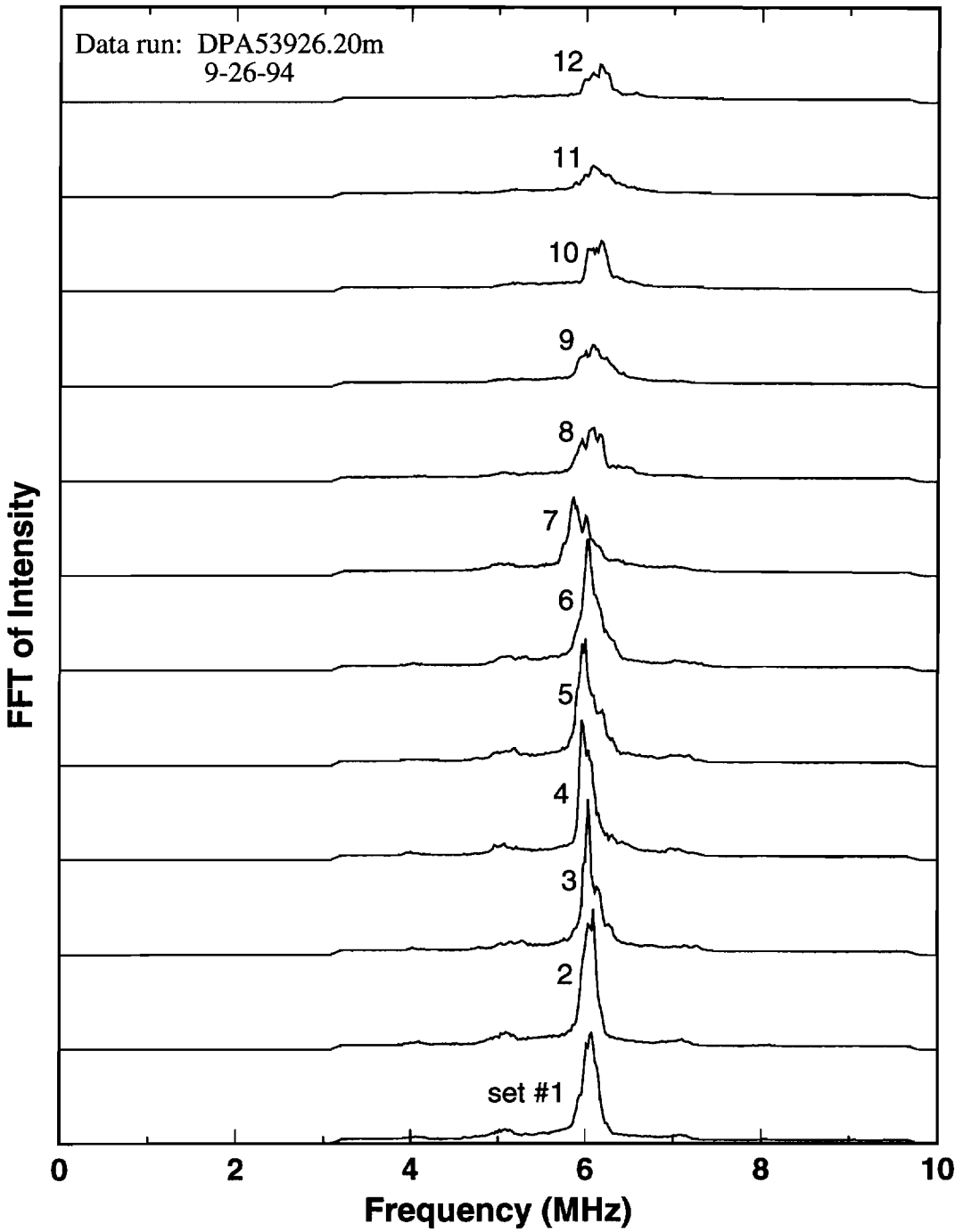


Figure 6.4d FFT spectra of sets 1 through 12 for data run dpa53926.20m. Each set represents an average of 25.6 msec (512K points).

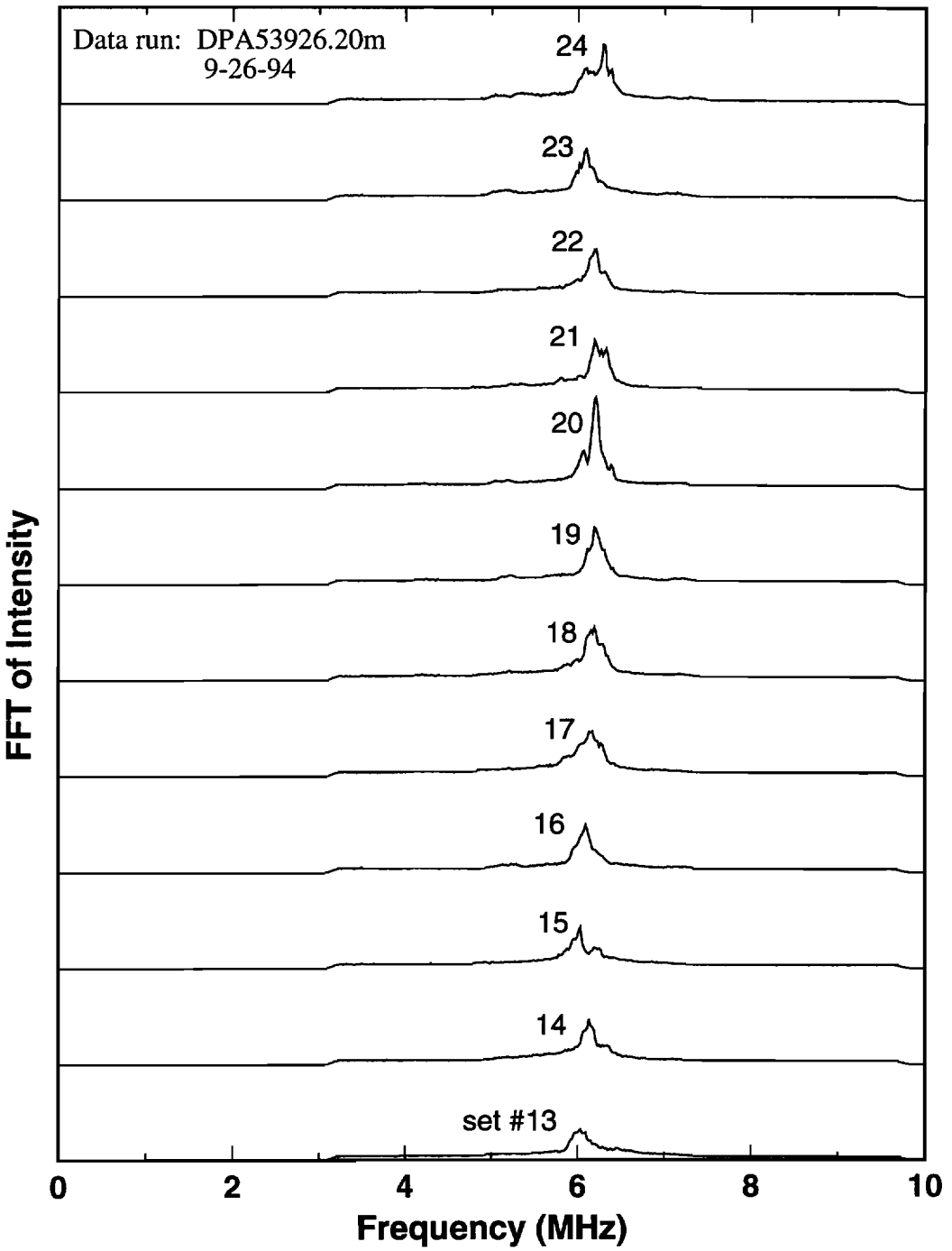


Figure 6.4e FFT spectra of sets 13 through 24 of data run DPA53926.20m. Each set represents an average of 25.6 msec (512K points).

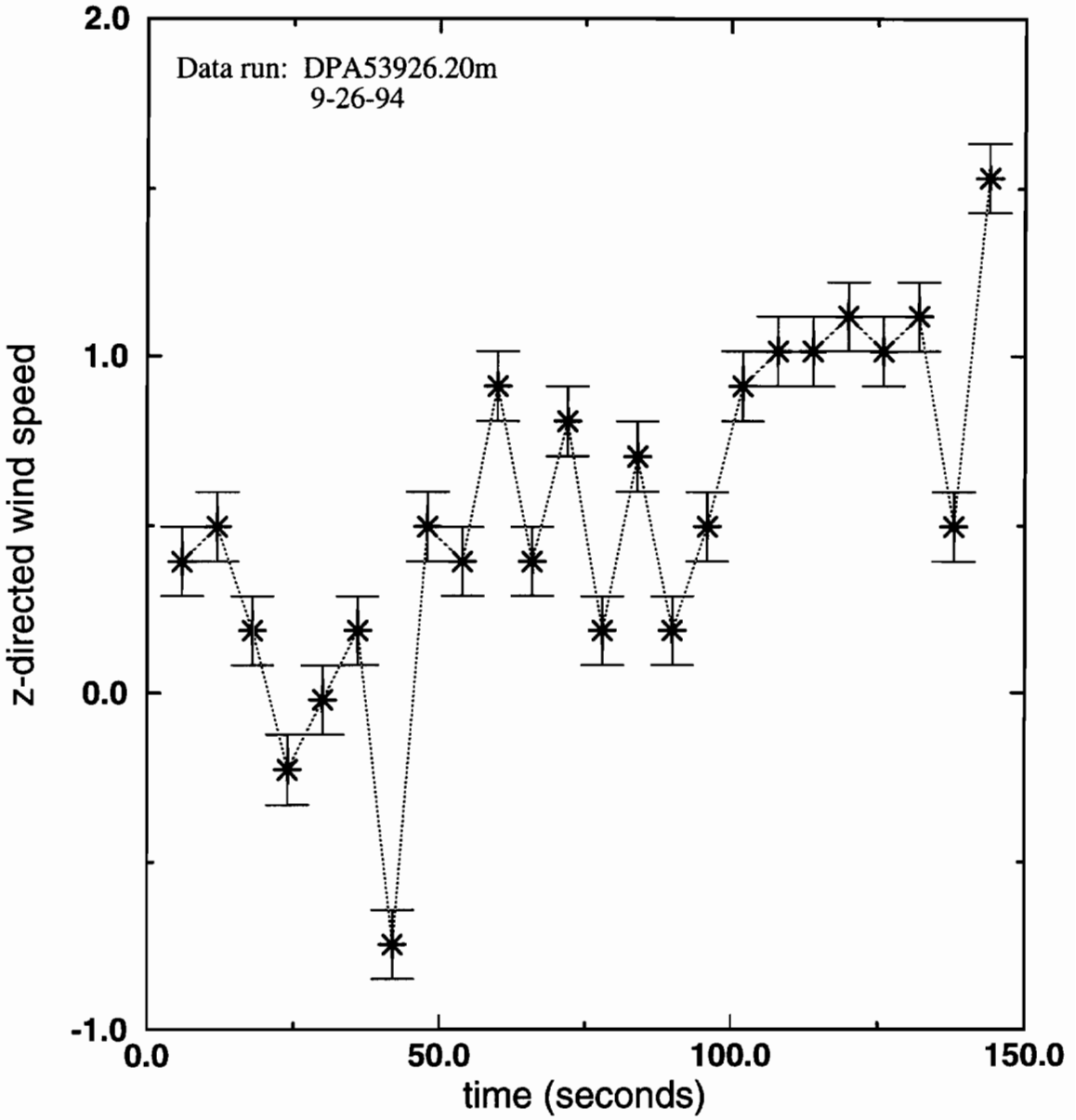


Figure 6.4f The z-directed wind speed vs. time series extracted from the Doppler shifts of each data set from data run DPA53926.20m. Error bars indicate the resolution capability of Doppler processing scheme: wind speed error = 20MHz / 1024pt FFT * 5.3m/s/MHz = 0.103 m/s resulution.

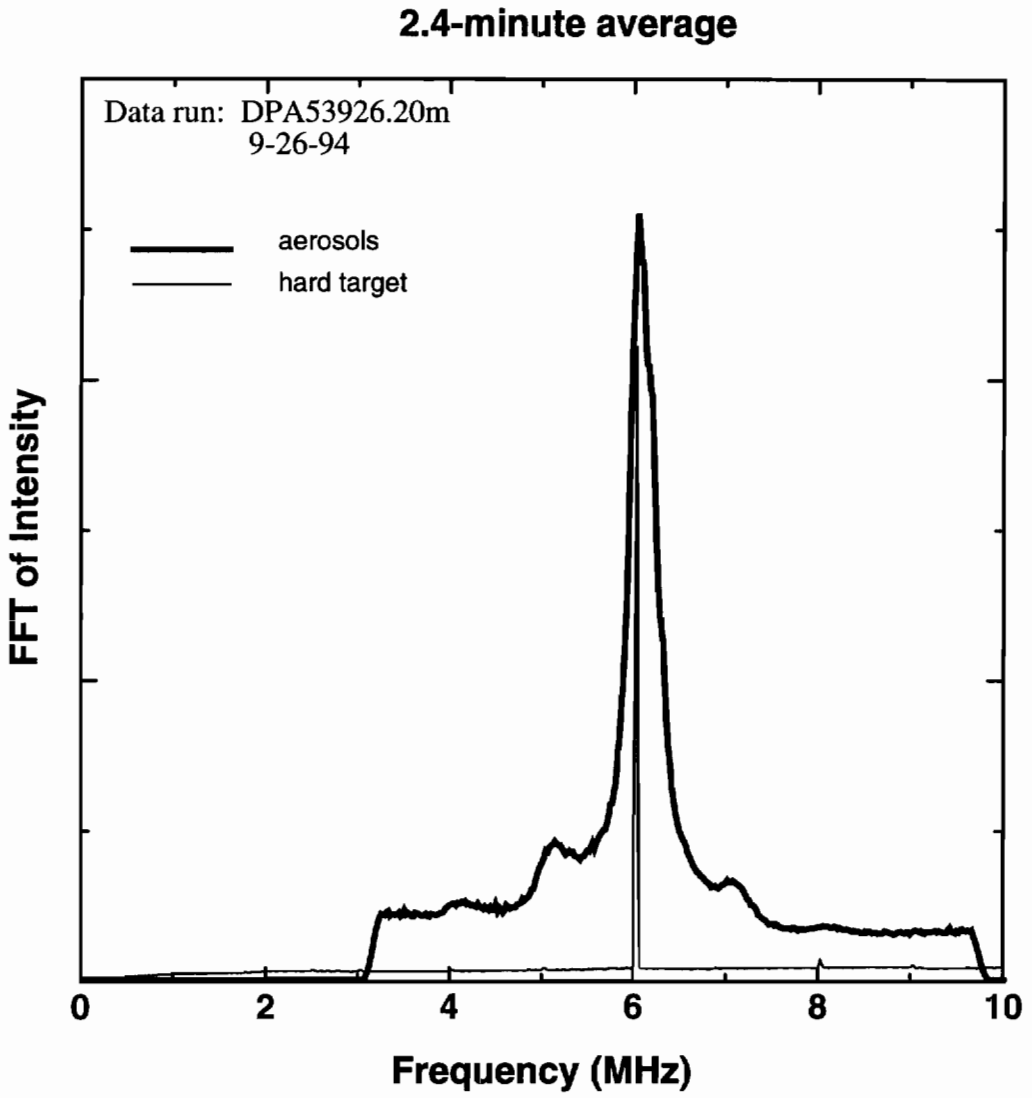


Figure 6.4g 24-set, 2.4 minute average of all the FFT spectra of data run DPA53926.20m. The FFT spectrum from a hard target is plotted for reference against the spectrum from aerosols.

noise level of the data run of figures 6.4a and b is about as low as could be achieved. Figure 6.4c presents the average Intensity vs. Range over the entire data run of 2.4 minutes. For reference, the path weight function derived at the end of chapter 5 is plotted with the center of the range bin at the same location as the data. While the averaged IvsR plot has a width of about 400 meters, it appears more sharply peaked than the theoretical prediction. Several factors may be the cause that the measured IvsR rolls off too quickly. One factor may be that figure 6.4c may not include enough averaging time to get a true description of the actual IvsR. Another reason could be that the backscattering coefficient is not uniform along the entire path. The theoretical curve was derived for uniform backscattering. But the actual propagation path has a terrain that is quite diverse which could contain plumes of aerosols that are more strongly concentrated near ranges 200 to 400 meters. Another reason could be that the theory was derived without the turning mirror blocking the center part of the outgoing laser beam. An additional factor could be that the lidar transmitting and receiving paths may not be perfectly coaxial, causing some off-axis effects. Since the receiving mirror is aligned near the focus, any off-axis effect will cause the system to have a more peaked path weight function.

The processed FFT spectra for the same data run used to produce figures 6.4a-c are plotted in figures 6.4d and e. Sets 1 through 6 are sharply defined corresponding to the IvsR sets with better SNR in figure 6.4a. After set 6, however, the FFT spectra become less sharp, and broaden. This broadening is due to the contributions from ranges other than the one for which the PRC was delayed. Because of this added cross talk, a wider z-directed wind speed variance tends to broaden the peak. Even with the additional Doppler broadening, the central peaks are still definable. The z-directed wind speeds can be extracted from the spectral plots, and plotted vs. time in figure 6.4f. This plot shows an example when the z-directed wind speed

changes direction over the data run. The z-directed wind speed blows away from the laser/receiver when the value drops below zero. This is because the aerosols are moving away from the laser/receiver and create a Doppler shift that lowers the frequency. When the wind speed is above zero, the wind is moving toward the laser/receiver.

A 2.4-minute average of all the spectra is given in figure 6.4g. The average spectrum for the return off a hard target has been included for reference. This plot illustrates the effect of Doppler broadening. The hard target spectrum is narrow because the target does not move with the z-directed wind. Therefore, no Doppler shift results and the peak is narrow and unshifted. However, the spectrum from aerosols shows the shift and broadening due to the variance in the radial winds. The width of the peak has widened to about 0.35 MHz at the full-width half-maximum.

The z-directed wind speed of another data run that has a higher Doppler shift is presented in figures 6.5. Most of the 24 sets have Doppler shifts corresponding to z-directed wind speeds between 2 and 4 meters per second. Toward the end of the data run, however, the winds reached 4.5 m/s from Doppler shifts of over 0.8 MHz.

The total number of data points in each of the two data runs presented in figures 6.4 and 6.5 is 12 megasamples per data run. This number can be reduced significantly, however, depending on the SNR. During the experiments to detect Doppler winds, a high number of samples per subset were used to provide as much averaging as possible. However, the full 25.6 milliseconds (512k points) per subset are really not needed. A good estimate of the Doppler shift can be obtained from 1 FFT, provided the SNR is above threshold. This fact is very important if radial wind detection is needed in real time.

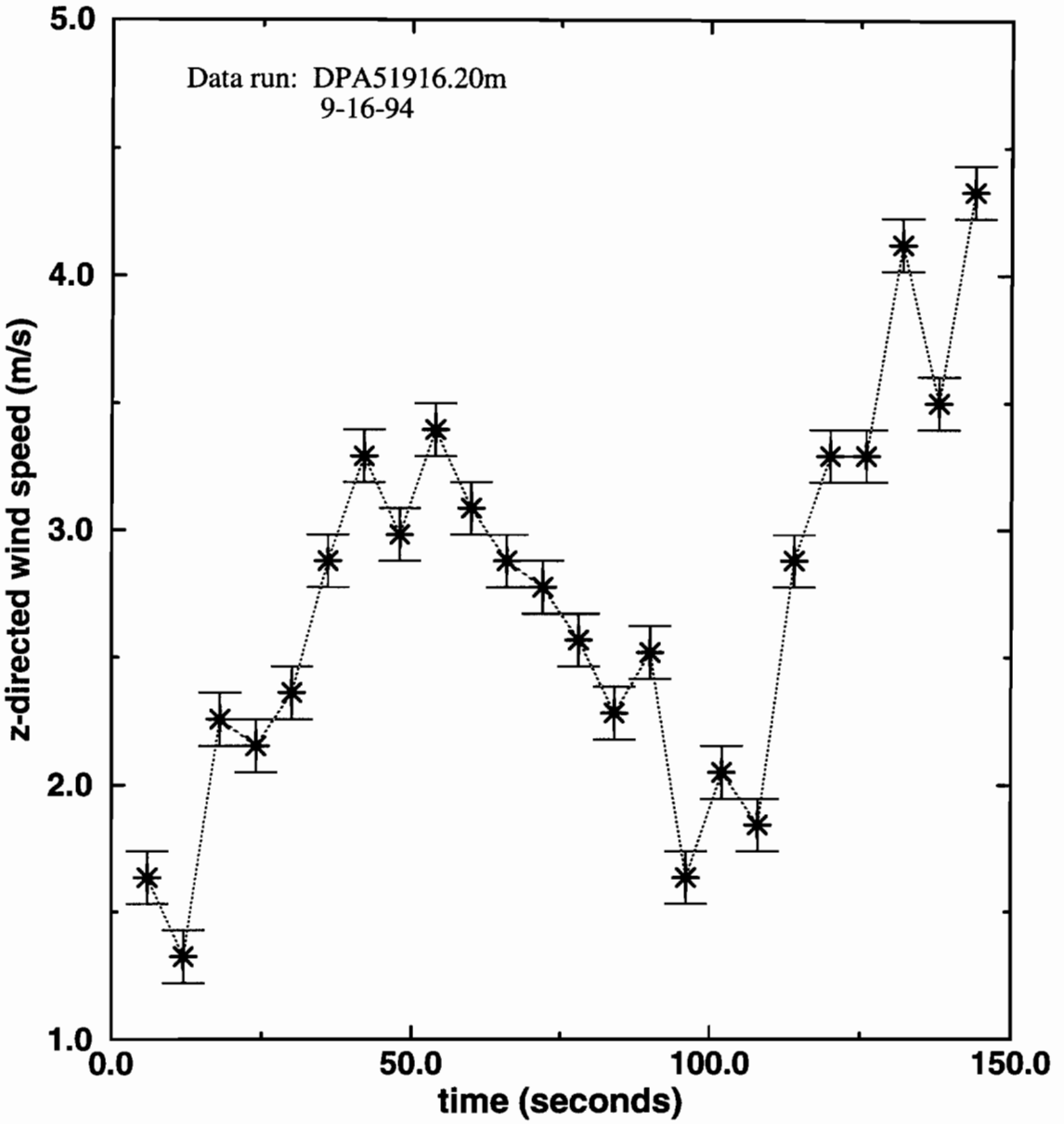


Figure 6.5 The z-directed wind speed vs. time series for data run DPA51916.20m. Error bars indicate resolution in Doppler shift detection. Tolerance = 0.103 m/s (see figure 6.4f).

6.2 Lidar Used to Detect Crosswinds

The crosswinds are much more difficult to detect than the radial winds. Not only are the receiving optics more complicated, there is more signal processing involved than just detecting the Doppler shift of the return signal. To detect the crosswinds, the split mirror is installed to divide the backscattered field into two equal portions. Each portion is directed to its own detector, amplified, and sent to the CS220 digitizer. Two channels of data are recorded and stored. If the frequency of the backscattered signal were constant, as with backscattering from a hard target, then the heterodyne signal could be mixed down to a low frequency and sampled at a much lower rate. Since turbulence causes the intensity to fluctuate at a maximum of 1-2 kHz, the heterodyne signal could be mixed as low as 2 kHz and sampled at 4 KHz. At a sampling rate of 4 kHz, the CS220 could take data for as long as 500 seconds (2 Megasamples per channel) before needing to dump to a hard disk. The use of aerosols as a backscatterer makes the crosswind detection difficult. The unknown Doppler shift prevents any mixing to a lower frequency since the Doppler shift could be as large as 2 MHz on either side of the heterodyne frequency. For a heterodyne frequency of 6 MHz, and a sampling rate of 20MHz, the CS220 fills up in 0.1 seconds. This is a serious limitation for crosswind detection since it is desirable to process data streams on the order of seconds. Even if a digitizer were used with more memory, 1 second of data sampled at 20 MHz is 20 Megasamples *per channel.* Such a huge amount of data makes it difficult for any system to operate in real time for the detection of crosswinds.

In chapter 4, the time-delayed crosscovariance of the intensities was time-filtered to make it easier to develop a crosswind detection scheme. The filter that is used comes from the FFT process. The peak value of the cosine from the FFT is the

average value over the entire FFT length. Therefore, the peak value of the FFT is the magnitude of the cosine averaged over the set length. Extracting the intensities from contiguous FFT's generates a series of intensities vs. time. An intensity estimate is made every 512 microseconds and there are 204 intensity estimates for both channels within 0.1 seconds of total sampling time. The key to estimating the crosswind speed for a particular set is generating the time-delayed crosscovariance of the IvsT of both channels.

Unfortunately, it was never possible to produce a crosswind time series to compare with the in situ data taken by the Campbell unit for any of the data runs with the short sampling time of 0.1 seconds. A percentage of the time, however, a turbulence fluctuation did appear on both channels which allowed a crosswind speed estimate to be made. About 1 in 6 data sets produced an adequate crosscovariance curve from which the crosswind speed could be estimated. An example of the processed intensities vs. time is shown in figure 6.6 with the corresponding time-delayed crosscovariance plotted in figure 6.7. The intensities vs. time series in figure 6.6 show the intensities with their smoothed versions to better visualize time-delayed turbulence fluctuations between the two channels. The intensity time series are noisy because the intensity estimate is taken from the average of only 10, 1024-point FFT's which is only a 512-microsecond average. The crosscovariance in figure 6.7 has been normalized by the average intensities of both channels. This plot was the only one of 6 sets recorded which gave a crosscovariance curve that was adequate enough to extract the crosswind. Because of the low SNR, the crosscovariance needed smoothing to identify the curve and to bring it out of the noise. The peak was shifted correctly to the right of the origin according to the observed crosswind direction. For time shifts beyond the shift to the peak, the curve drops below 0, verifying the behavior of the theoretical curve. The low value of covariance at zero time delay may be due to the

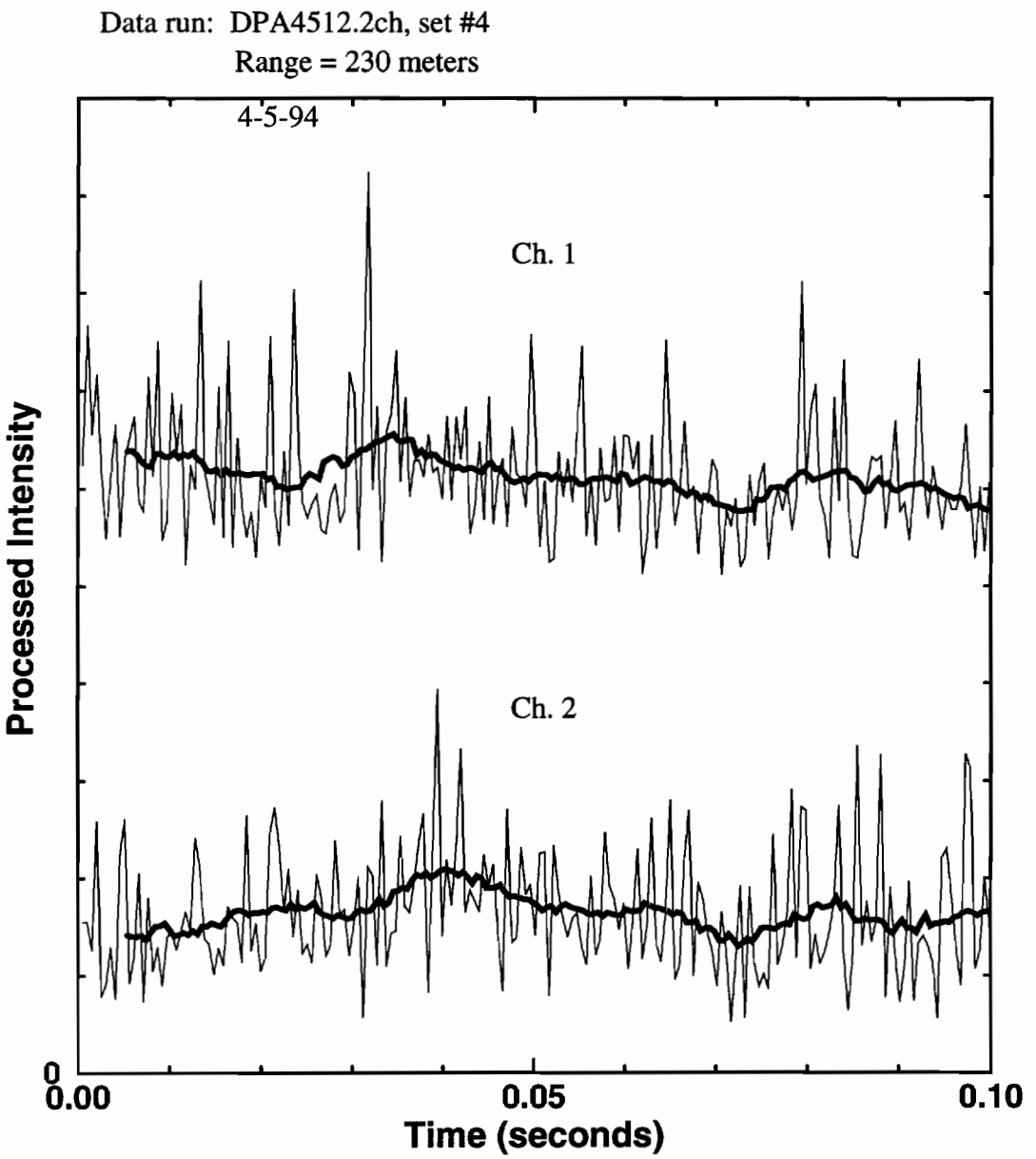


Figure 6.6 Processed intensities vs. time for data run DPA4512.2ch used to compute time-delayed crosscovariance in figure 6.7. Intensities have been smoothed to better visualize time-delayed turbulence fluctuations between the two channels.

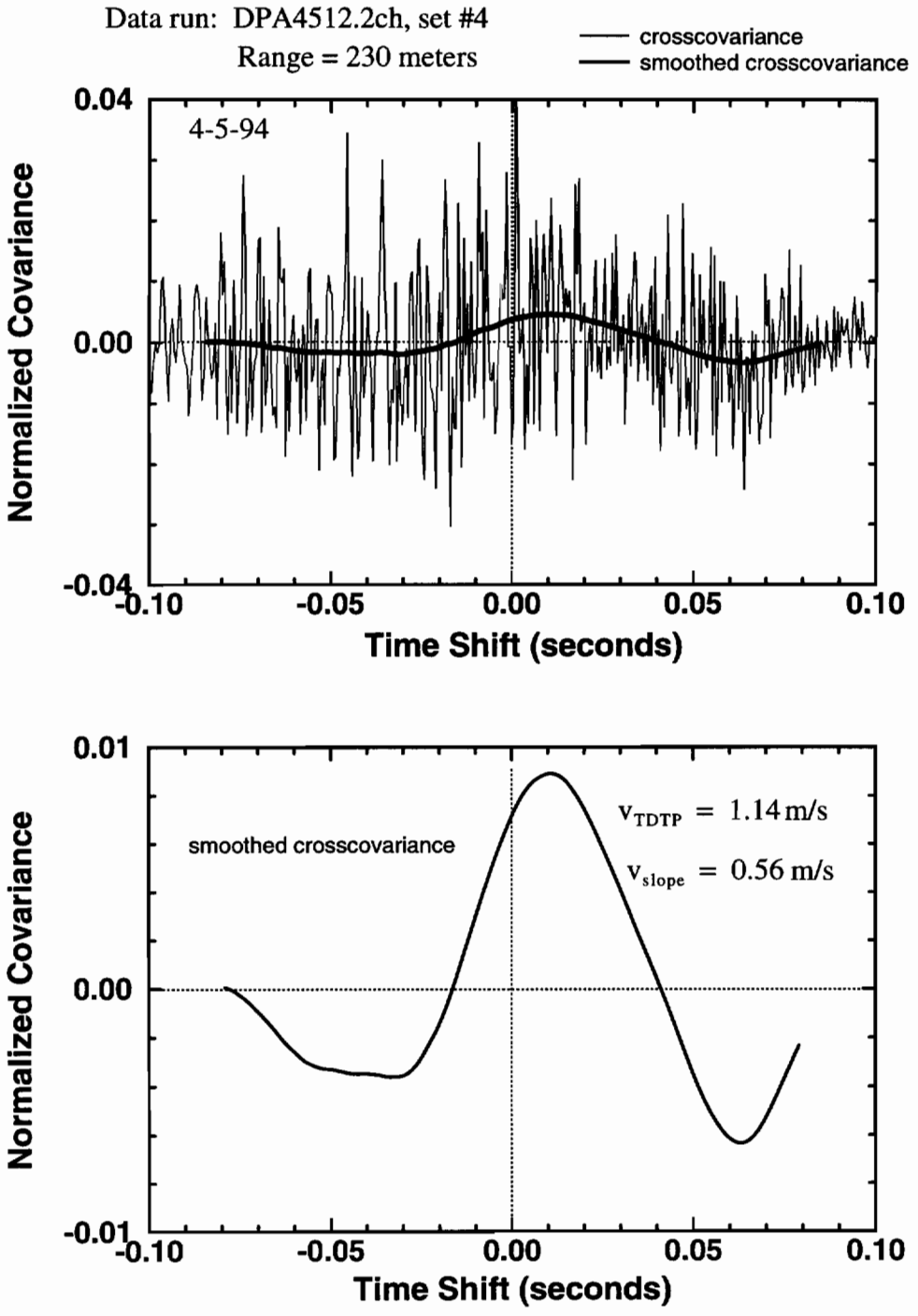


Figure 6.7 Processed time-delayed crosscovariance of intensities normalized by the average intensities from both channels for data run DPA4512.2ch.

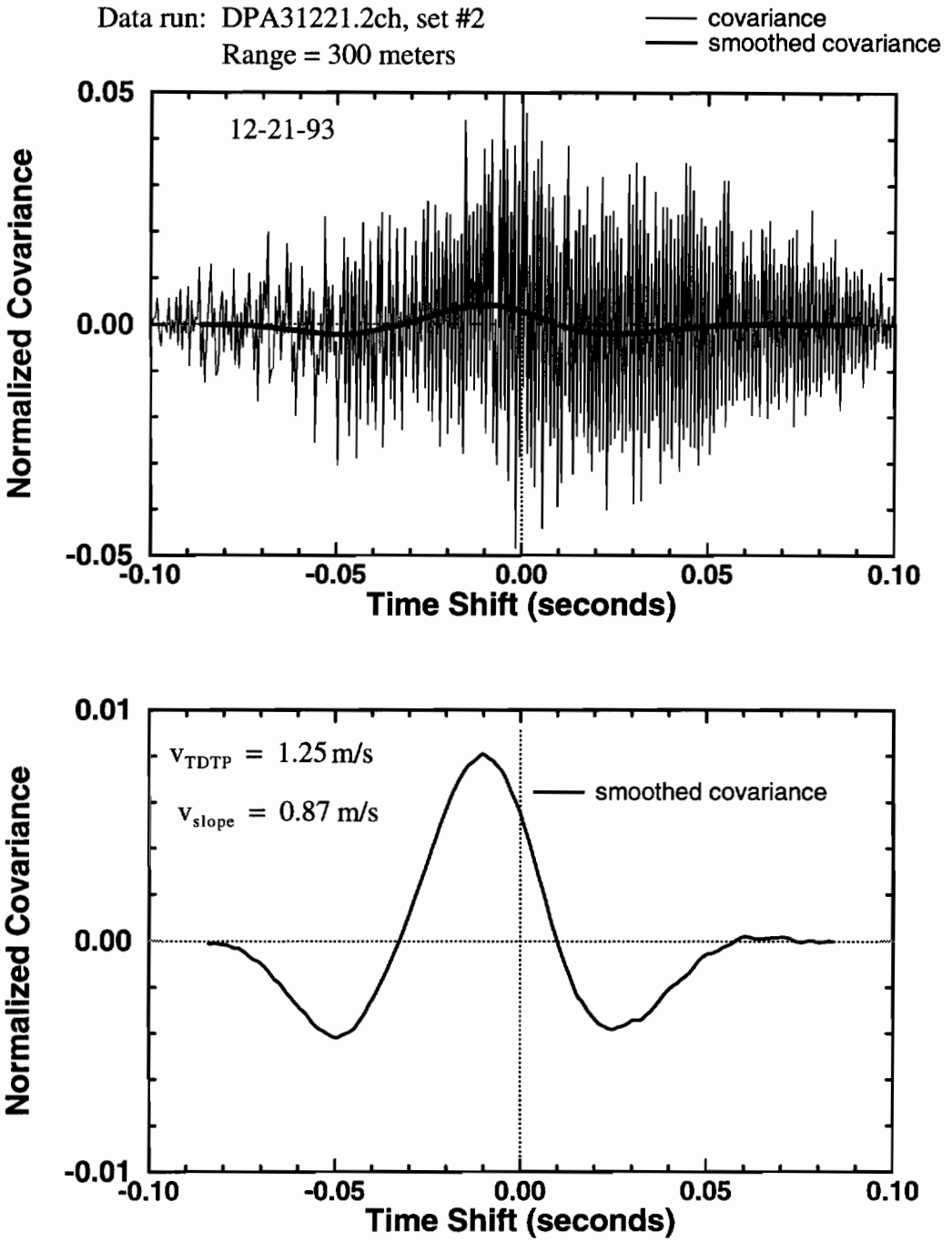


Figure 6.8 Processed time-delayed crosscovariance of intensities normalized by the average intensities from both channels for data run DPA31221.2ch.

small interaction between the turbulence and the CO_2 laser beam. Aperture averaging also tends to reduce the covariance because the intensity fluctuations are averaged. The time delay to peak method gives a crosswind speed of 1.18m/s while the slope method estimates a crosswind speed of 0.54 m/s. The smoothed covariance curves were used for both crosswind detection schemes.

An example of another data run is shown in figure 6.8. Again, smoothing is necessary to outline the crosscovariance curve and bring it out of the noise. This curve also follows the shape predicted by the theory, dropping below 0 for time shifts beyond the peak value. The time delay to peak method indicates a crosswind speed of 1.25 m/s while the slope method gives a crosswind speed of 0.87 m/s. No in situ data was taken to compare with the data of figures 6.7 and 6.8.

The crosscovariance curves of figures 6.9 and 6.10 were taken with the Campbell unit to provide in situ data. Both figures offer examples of data sets that contain turbulence fluctuations on both channels within the 0.1 seconds of sampling time. When a turbulence fluctuation appears on both channels, each crosscovariance curve gives a fairly accurate estimate of the crosswind compared to the in situ measurement. For example, the crosswind speed estimate of the data set presented in figure 6.9 is 1.49 m/s using the time delay to peak method, and 1.31 m/s using the slope method. These values compare well with the in situ estimate of 1.25 m/s. The crosswind speed estimate of the data presented in figure 6.10 is 2.42 m/s using the time delay to peak method and 2.11 m/s using the slope method. The in situ value is 2.33 m/s which compares well with the experimental values.

A better percentage of adequate time-delayed crosscovariance curves per data run are needed to give more validity to the experimental measurements. However, the fact that crosscovariance curves are computable with only 0.1 seconds of sampling time shows promise that a better percentage is possible if the sampling time could be

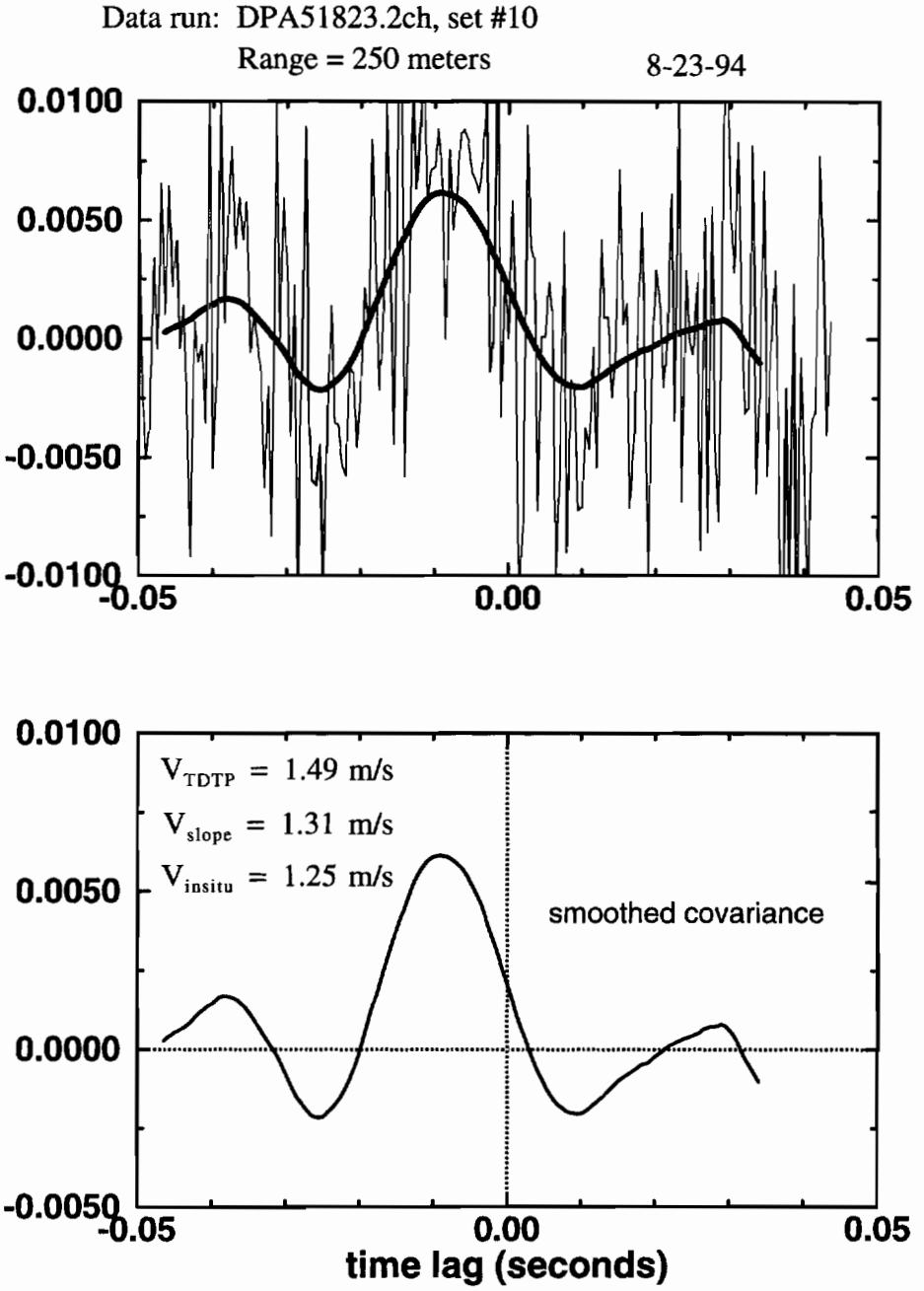


Figure 6.9 Time-filtered, time-delayed crosscovariance of intensities for data run DPA51823.2ch.

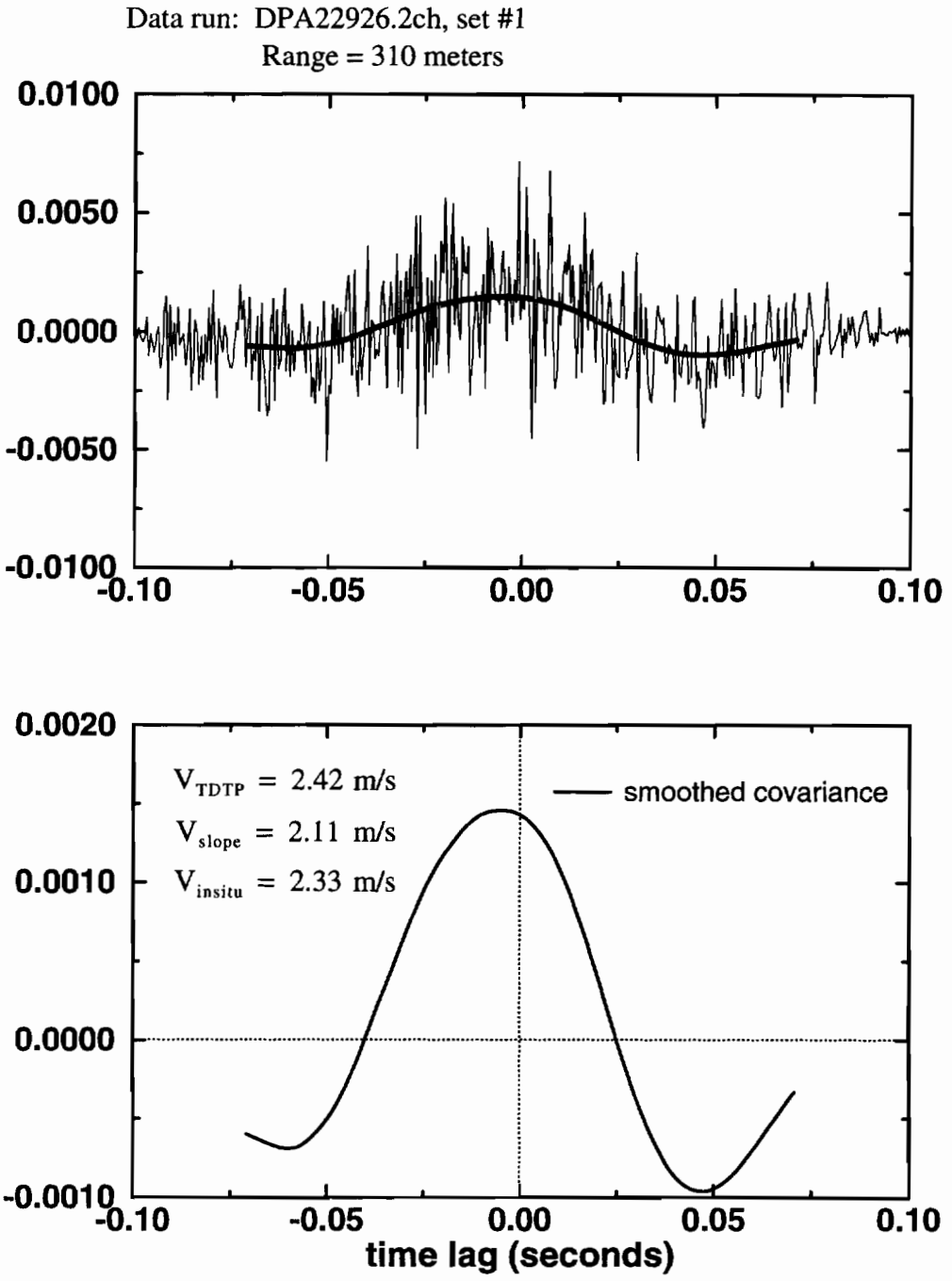


Figure 6.10 Time-filtered, time-delayed crosscovariance of intensities for data run DPA22926.2ch.

increased to the order of seconds per data run. Improving the SNR can only help as well since the crosscovariance curves need smoothing to outline their shape.

6.3 Modified Data Acquisition Techniques

After a large portion of the data had been processed and analyzed, it became clear that the data could be sampled differently to give better representations of the wind speed. Two new sampling routines resulted. The first was applied to the sensing of radial winds. Since the only desired quantity was the Doppler shift, less data could be taken per subset while extending the number of subsets to include more total sampling time. The Doppler wind sensing routine was changed to taking data samples every 2 seconds, taking 128K points per subset. The number of subsets per data run was extended to 600, covering 20 minutes of sampling time.

An example of a data run which was taken using this new Doppler wind sensing data acquisition routine is presented in figures 6.11 a-g. Figures 6.11a and b show the processed IvsR averaged over 1 minute. Each curve is an average of 30 IvsR subsets consisting of 128K points. Over the 20-minute span, the center of strongest backscattering meanders between about 180 and 280 meters. The 20-minute average of all the IvsR curves is presented in figure 6.11c, indicating that the center of strongest backscattering occurs near 200 meters. The IvsR predicted from the theory at the end of Chapter 5 is sketched with the measured IvsR for reference. Even with the longer averaging time of 20 minutes, the measured IvsR curve is too sharply peaked, and rolls off too quickly. The cause is probably not a lack of time in the average, since the 20-minute IvsR average in figure 6.11c is not much different than the 2.4-minute IvsR average in figure 6.4c. The most likely factor is that the lidar is probably not perfectly coaxial.

The processed FFT spectra for the same data run used to produce figures 6.11a-c are plotted in figures 6.11d and e. Most of the spectra indicate a significant Doppler shift on the order of 1 MHz. This is one reason why the peaks are much

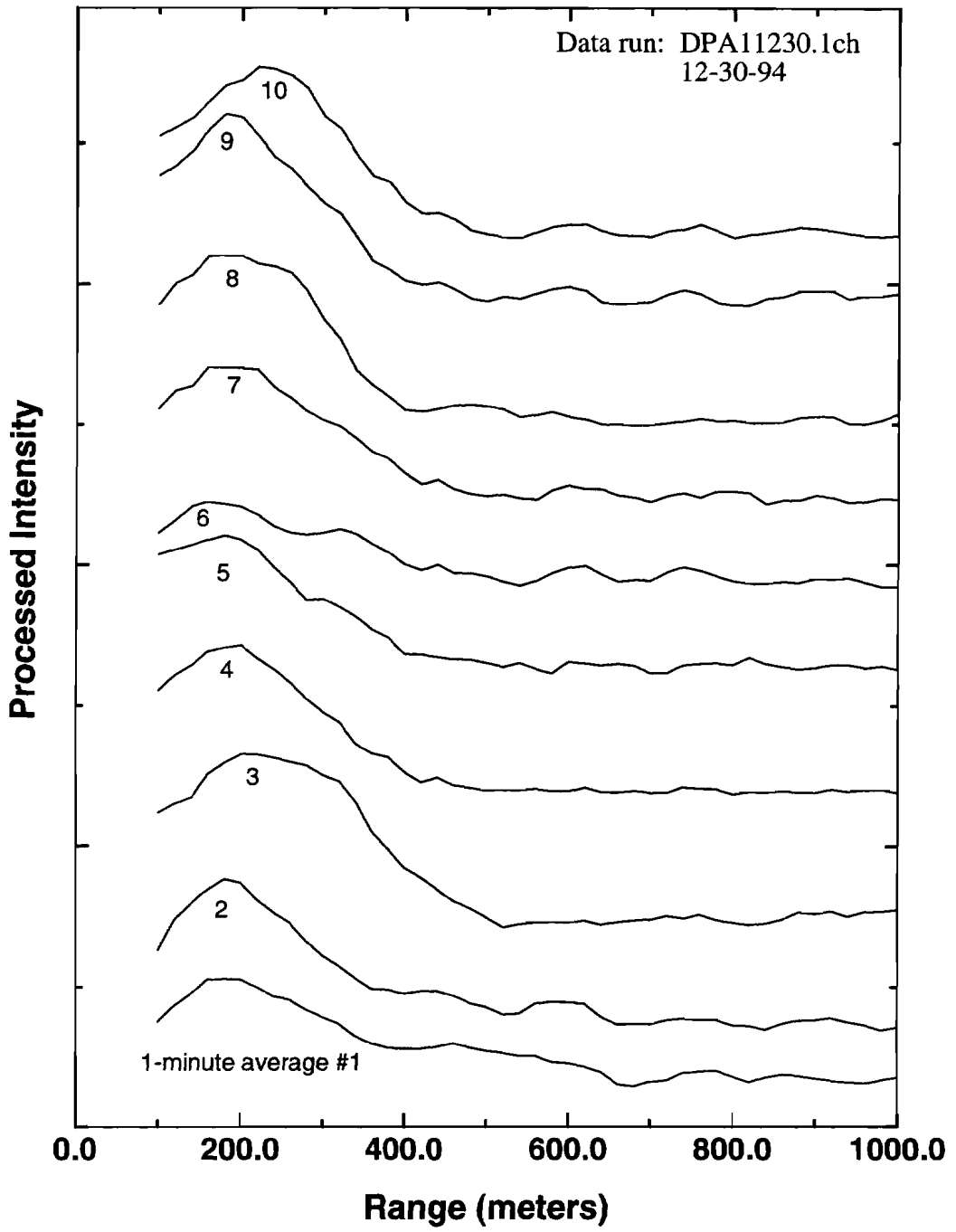


Figure 6.11a Sets 1 through 10 of 1-minute averages of the processed intensity vs. range for data run DPA11230.1ch. Each curve is an average of 30 subsets of 128K points.

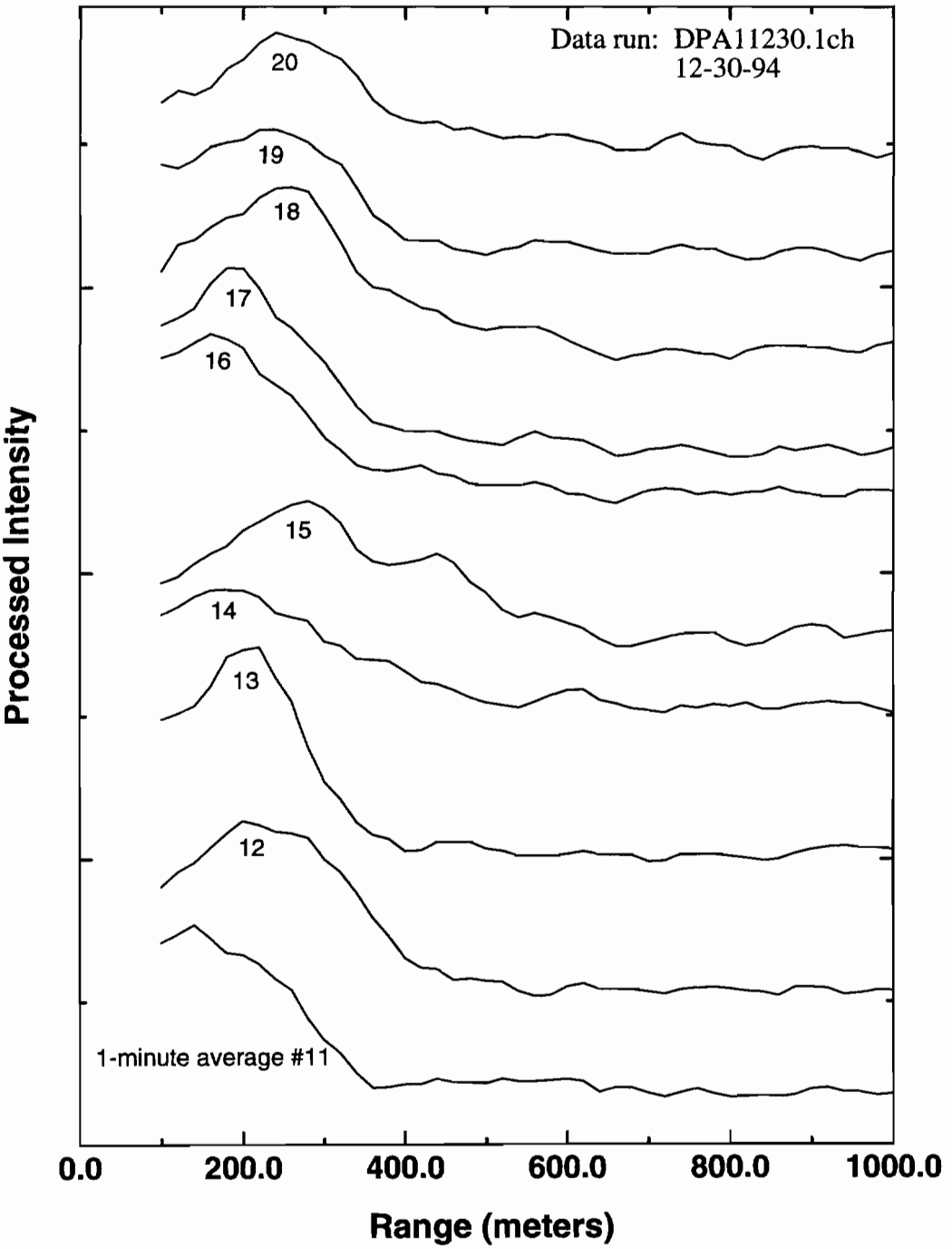


Figure 6.11b Sets 11 through 20 of 1-minute averages of the processed intensity vs. range for data run DPA11230.1ch. Each curve is an average of 30 subsets of 128K points.

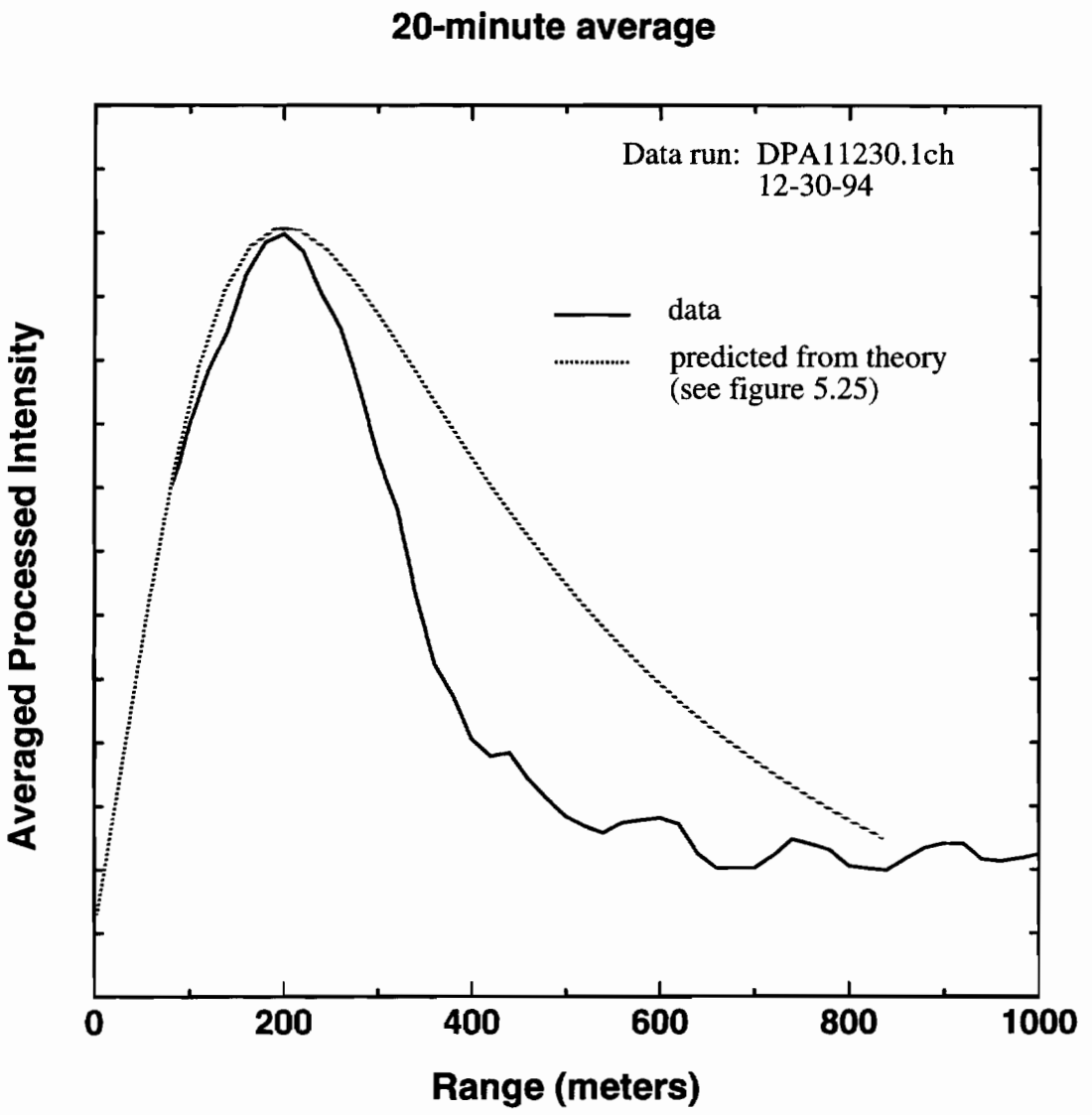


Figure 6.11c The intensity vs. range averaged over 20 minutes for data run DPA11230.1ch. The dotted line is the path weight function predicted from the theory at the end of Chapter 5.

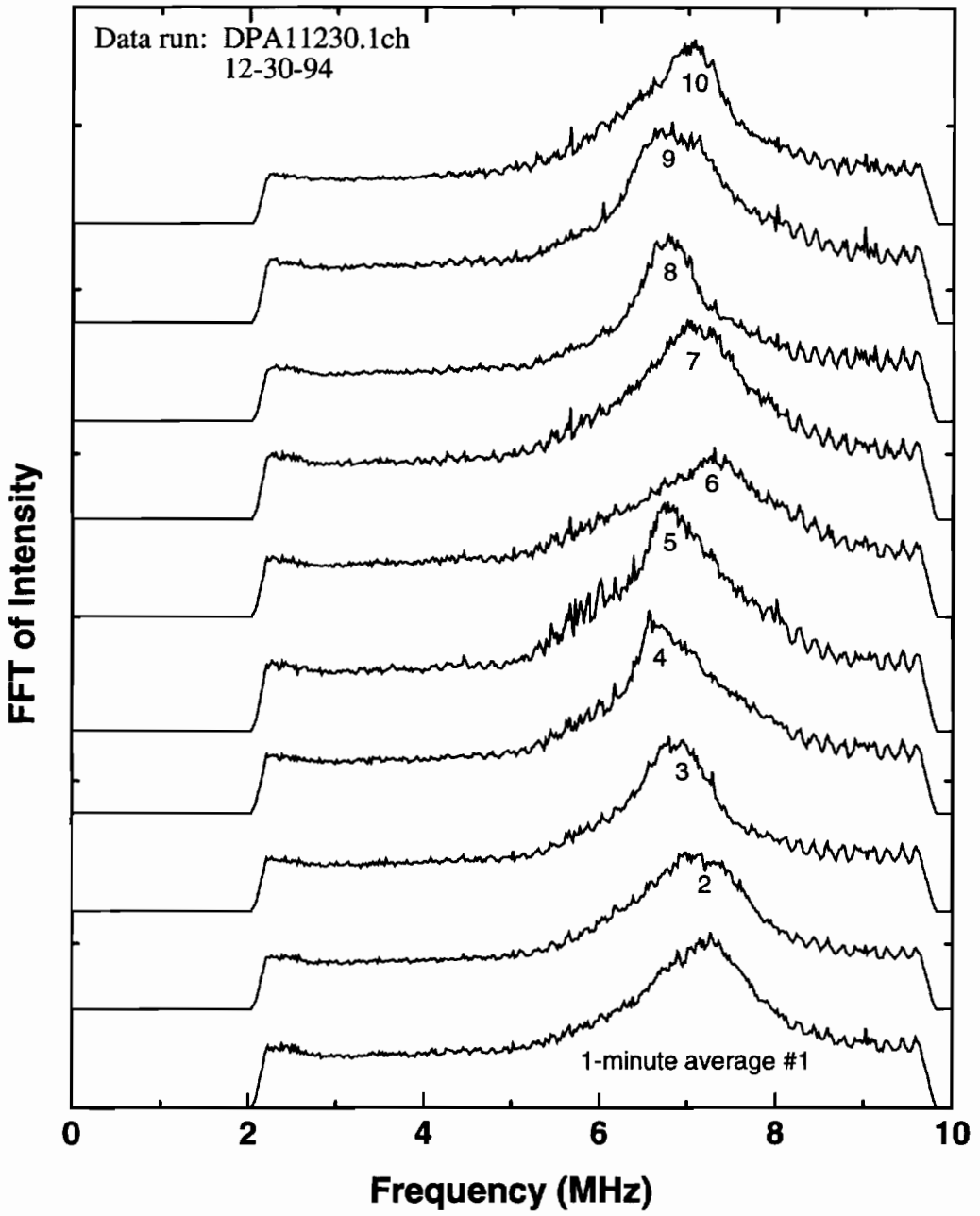


Figure 6.11d Sets 1 through 10 of 1-minute averages of the FFT spectra for data run DPA11230.1ch. Each curve is an average of 30 subsets of 128K points.

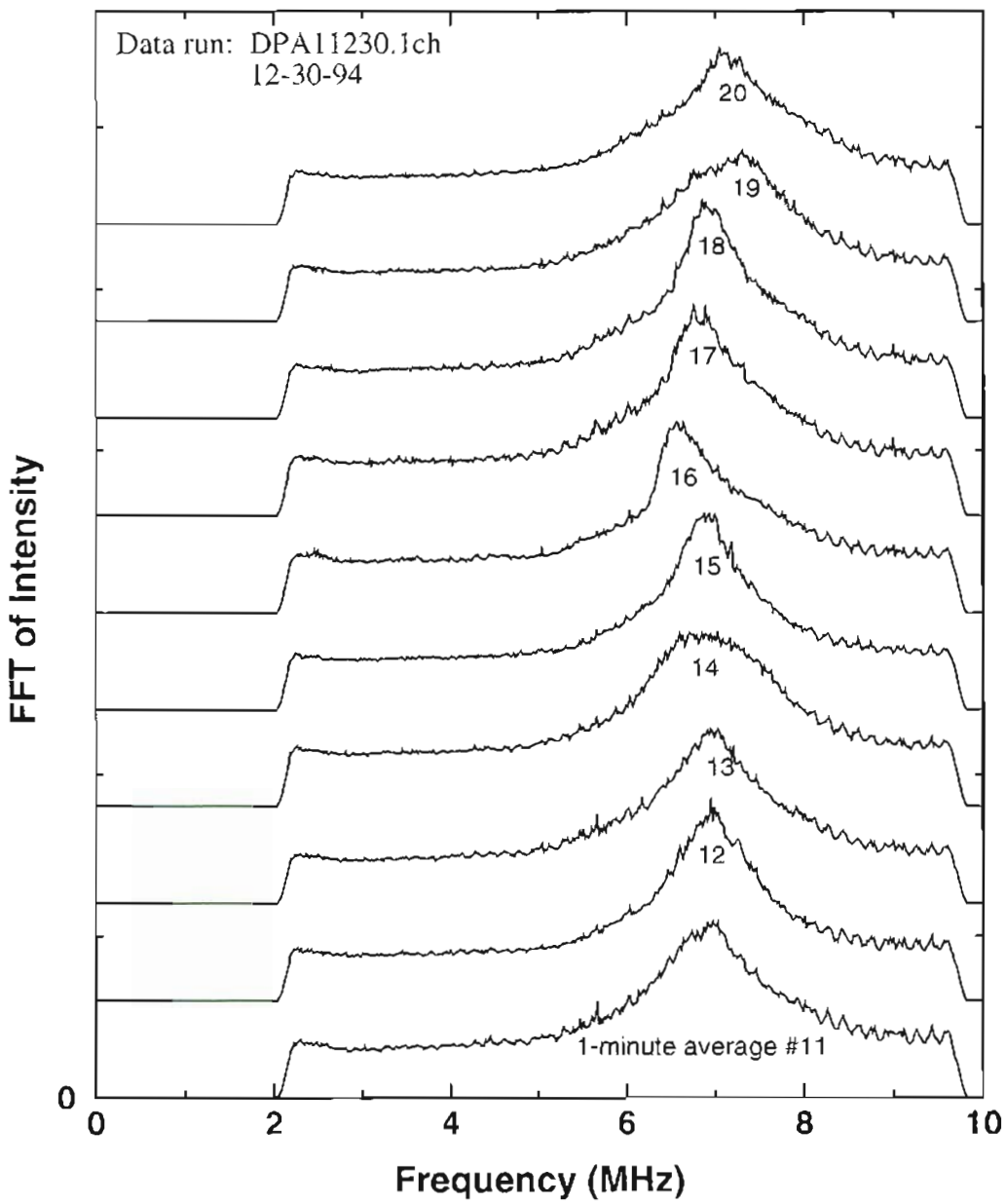


Figure 6.11e Sets 11 through 20 of 1-minute averages of the FFT spectra for data run DPA11230.1ch. Each curve is an average of 30 subsets of 128K points.

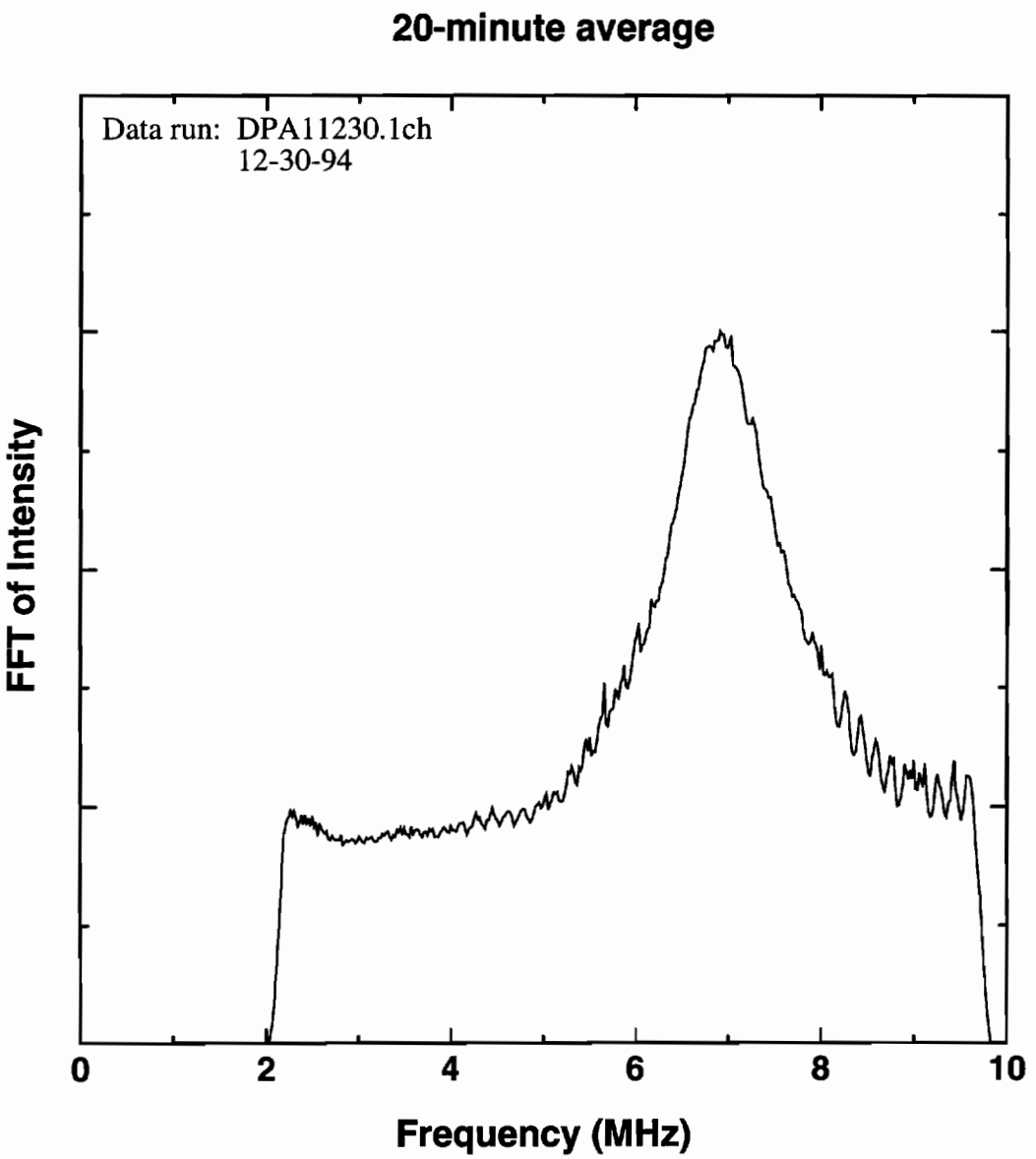


Figure 6.11f 20-minute average of the FFT spectra of data run DPA12230.1ch.

broader than the peaks of the data presented in figures 6.4d and e. Since the variance of the wind speed is proportional to the average wind speed, the peaks are broadened due to the wide distribution of z-directed winds. Each curve in figures 6.11d and e also represent a full 1 minute of averaging time which also contributes to the broadening of the peaks. Some of the peaks are over 1 MHz wide at the full-width half-maximum, indicating a large distribution of radial wind speeds. The 20-minute average of all the spectra for this data run is presented in figure 6.11f. The average Doppler shift is about 1 MHz and the peak is broadened to about 1 MHz at the full-width half-maximum.

The radial wind speed vs. time series is plotted in figure 6.11g. Each radial wind speed estimate is separated in time by 2 seconds and covers an average of 128K points. The smoothed wind speed time series is plotted to outline the general trends of the z-directed wind speed throughout the data run. A 1-minute averaging time is used for the smoothing. Because the data was taken in the winter, there was a much lower concentration of aerosols, leading to a much lower SNR compared to the data run of figure 6.4. The low SNR also contributes to the noisy radial wind speed estimates. Smoothing helps to eliminate the noisy fluctuations in the radial winds due to the low SNR. This data run was taken during one of the windiest periods observed for the past 3 years. The strong pressure gradient in the weather pattern persisted for days during the latter part of December, 1994, creating some strong east winds. The radial wind speed time series contains winds approaching 10 m/s (over 22 mph), with a mean wind speed of about 5.3 m/s (11.8 mph).

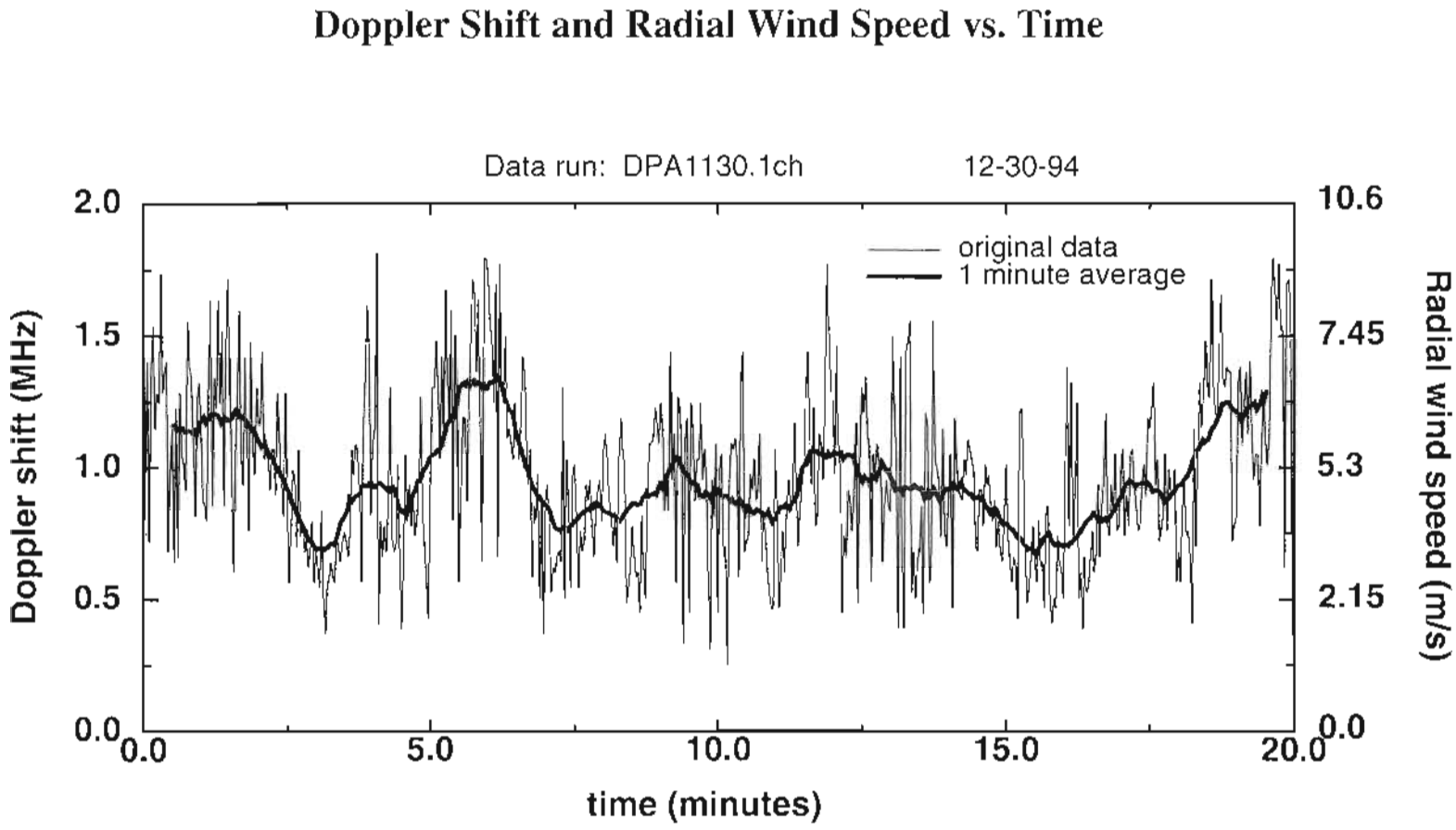


Figure 6.11g The radial wind speed time series for data run DPA12230.1ch. Each wind speed estimate consists of 128K points and occurs every 2 seconds, covering 20 minutes. The smoothed time series outlines the general trends of the radial winds over the entire data run.

Chapter 7

Conclusions

The goal of this research project was to design, construct, and operate a cw, CO_2, optical heterodyne lidar to measure the 3-D atmospheric wind speed using back-scattering from aerosols. The use of aerosols as a distributed, random, moving target also required new theoretical analysis of the time-delayed statistics of the received intensity. Since it was desired to operate the laser cw, instead of pulsed, some kind of pseudo-random modulation of the laser source was needed in order to confine the wind speed estimates to desired range bins while suppressing contributions from other ranges. The lidar constructed over the past three years from 1991 to 1994 demonstrated the first successful operation of a pseudo-random code, cw, optical heterodyne lidar at 10.6 microns. During this time, several major advancements were made which can be used in future systems.

Using an acoustooptic modulator to implement the pseudo-random code onto the outgoing laser beam was the first achievement. In order to produce path resolved measurements with a cw system, some kind of pseudo-random modulation of the laser source was necessary. Instead of just turning the laser source on and off with the PRC which would have meant a 3dB reduction in signal-to-noise ratio, the outgoing laser beam was successfully diphase modulated using an AOM for the first time. It was shown that sending a diphase modulated electrical drive signal to the AOM was converted to a diphase acoustic wave by the transducer which correctly diphase modulated the CO_2 laser beam. Although there was some finite transition time during the

phase changes where the phase of the beam was not defined, there was only a 9 percent drop in received power during diphase operation as compared to the operation without diphase modulation. The measured transition time corresponded with the rise time of the AOM. The 300 ns transition time was adequately small enough for this lidar with a range resolution of 300 meters. The range resolution is proportional to the time of the PRC minimum gate width which was 1μs in this case. Consequently, an AOM with a faster rise time may be necessary if smaller range bins are desired. For example, for a 30-meter range bin, the minimum PRC gate width needs to be 100 ns. Therefore, an AOM with a faster rise time would probably be necessary to facilitate a range bin of 30 meters.

Another advancement was the realization of an optimum local oscillator power level which maximized the SNR using photodiodes operated in the reverse biased mode as detectors for heterodyne systems. The diode responsivity was found to saturate and become nonlinear as more LO power was applied to the photodiode surface. Instead of treating the diode responsivity as a constant, the diode responsivity was modeled as a nonlinear function of the LO power which saturated as more LO power was applied. Four separate photodiodes were examined and each exhibited the saturation effects as more LO power was applied. A more correct SNR equation was derived which included the saturation effects of real photodiodes and showed that the SNR peaked at a finite LO power level. By comparing the more correct SNR to the SNR derived using LO shot noise limited operation (no saturation), an SNR reduction factor resulted for each photodiode showing how much under the signal shot noise limit the lidar actually operated.

Deriving the time-delayed statistics of the backscattered laser beam intensity off of aerosols also provided some new insight into laser beam propagation. The received mutual intensity function, as well as the time-delayed crosscovariance of the

intensities backscattered from aerosols became a function of speckle and turbulence with drastically different decorrelation times. Since aerosols move in a random manner relative to one another, the decorrelation time due to speckle was much more rapid than the decorrelation due to turbulence. This result was not new. However, the application of the proper time filter successfully removed the high frequency speckle from the analysis, leaving only the turbulence perturbation on the time delayed statistics. Under these circumstances, the coherent heterodyne lidar acted as if it were incoherent because the expression for the time-delayed crosscovariance resembled the one for an incoherent system. This result made it easier to develop a crosswind detection scheme, and simplified the expressions of the time delayed statistics considerably. Since the laser beam intensity is taken from the peak value of a FFT, the intensity was filtered by the FFT process because the peak value was the average over the entire FFT length. The total time in the FFT determined the cutoff frequency to be about 20 kHz (1024-pt FFTs at 20 MHz sampling rate). This cutoff frequency was nearly ideal to filter high frequency speckle fluctuations on the order of megahertz but leave the lower turbulence fluctuations on the order of several hundred hertz intact.

Perhaps the most significant discovery was realized with respect to the decorrelation time of the *phase* of the backscattered beam. The decorrelation time of the phase determined the maximum FFT window from which to compute estimates of the backscattered intensity and Doppler shift. Contrary to popular opinion, the only factor which determined the coherence time was that due to the laser. The rapid speckle decorrelation time only applied to the intensity, but had no effect on the phase. This result was both predicted from theory, as well as experimentally verified. By computing the SNR vs. different FFT time slices (different FFT lengths) a maximum SNR was found for every processed data set at 50 microseconds which corresponded with the 50-microsecond coherence time of the laser. Previous theories had put the coher-

ence time due to speckle as low as 0.5 microseconds. This advancement should have a significant impact on future design of coherent heterodyne lidar systems.

The lidar was used to produce the first path resolved measurements of radial wind velocities by detecting the Doppler shift of the moving aerosol particles using a cw, pseudo-random modulated lidar. It was found that operating the system to detect radial winds could occur on most days of the year, even in winter months, because all of the backscattered radiation was directed to 1 detector, and the only desired quantity was the Doppler shift. A 20-minute profile of the radial winds was taken during an exceptionally blustery weather pattern which showed that the Doppler spread was much more significant for the higher wind speeds as compared to spectra of data taken in less windy weather. This observation corresponded with theory, which predicted that the standard deviation of the winds is proportional to the average wind speed. Therefore, there was more Doppler spread as the radial winds became stronger. The lidar was shown to operate out to a maximum range of about 600 meters. Most of the backscattered radiation occurred from ranges between 200 and 400 meters. The lidar could operate to a longer range by redesigning the transmitting optics to push the focal point out to a further range.

In addition to the Doppler winds, the lidar was used to take the first path resolved measurements of crosswinds using speckle-turbulence interaction and a cw, pseudo-random modulated lidar. The key to obtaining crosswind estimates was to produce the time-delayed crosscovariance of the intensities. Unfortunately, a crosswind time series like those for the Doppler winds was never achieved. However, several important advancements were made which showed that crosswind detection using this kind of a lidar was headed in the right direction. When the time-delayed crosscovariance was produced, the shape of the curve matched that from theory, having a maximum value that was time shifted from the origin depending on the crosswind speed,

and decorrelated for time shifts beyond the shift to the peak. The curves also dropped below 0 with increasing time shift which matched the theoretical curves. Only about 1 in 5 or 6 data sets processed gave a time-delayed crosscovariance curve adequate to estimate the crosswind. Part of the lack of success was due to the short sampling time of 0.1 seconds used to produce the intensity vs. time series and the time-delayed crosscovariance. The other factors were due to the low SNR, and weak turbulence interaction at 10.6 microns.

New cw, pseudo-random lidars could be designed to provide better representations of the atmospheric winds. Using a wavelength around 1.06 microns could push the focus distance out a factor of 10 over the 10.6 micron system. In addition, the interaction with turbulence is much stronger at 1.06 microns which would benefit the crosswind detection. Using an array of detectors would also increase the crosscovariance between two different backscattered laser signals because aperture averaging would be less of a problem. Vertical winds would also be detectable if the array contained detectors arranged in a rectangular grid.

References

[1] "New Space Tool to Study Earth Warming," Article from The Oregon Scientist, Vol. VII, Winter issue, 1994-1995, pg. 1.

[2] M. Born and E. Wolf, Principles of Optics, Pergamon Press Inc., 1959, Ch. 8, pgs. 370-458, Ch. 13, pgs. 635-658.

[3] A. N. Kolmogorov, "Local Structure of Turbulence in an Incompressible Liquid at High Reynolds Number," Dokl. Akad. Nauk SSSR, Vol. 30, 1941, pg. 299.

[4] V. I. Tatarskii, Wave Propagation in a Turbulent Medium, Translated by R. A. Silverman, McGraw-Hill, New York, 1961, Ch. 3, pgs. 40-58.

[5] R. F. Lutomirski and H. T. Yura, "Propagation of a Finite Optical Beam in an Inhomogeneous Medium," Applied Optics, July, 1971, pgs. 1652-1658.

[6] Z. I. Feizulin and Y. A. Kravtsov, "Expansion of a Laser Beam in a Turbulent Medium," Izv. Vyssh. Uchebn. Zaved. Radiofiz., Vol. 10, 1967, pgs. 68-73.

[7] Z. I. Feizulin and Y. A. Kravtsov, "Some Consequences of the Kirchoff-Huygens Principle in a Smoothly Inhomogeneous Medium," Izv. Vyssh. Uchebn. Zaved. Radiofiz., Vol. 12, 1969, pg. 886.

[8] A. S. Turvich, A. I. Kon, V. L. Mironov and S. S. Khmelevtsov, Laser Radiation in a Turbulent Atmosphere, Nauka, Moscow, 1976.

[9] J. L. Codona, D. B. Creamer, S. M. Flatte, R. G. Frehlich, and F. X. Henyey, "Moment-Equations and Path-Integral Techniques for Wave Propagation in Random Media," J. Math. Phys., Vol. 27, 1986, pgs. 171-177.

[10] R. Dashen, "Path Integrals for Waves in Random Media," J. Math. Phys., Vol 20, 1979, pgs. 894-920.

[11] Ting-i Wang, S. F. Clifford, and G. R. Ochs, "Wind and Refractive-Turbulence Sensing Using Crossed Laser Beams," Applied Optics, Vol. 13, November 1974, pgs. 2602-2608.

[12] R. S. Lawrence, G. R. Ochs, and S. F. Clifford, "Use of Scintillations to Measure Average Wind Across a Light Beam," Applied Optics, Vol. 11, February 1972, pgs. 239-243.

[13] J. Fred Holmes, J. Richard Kerr, Richard A. Elliot, Myung H. Lee, Philip A. Pincus, and Michael E. Fossey, "Experimental Pulsed Laser Remote Crosswind Measurement System-Feasibility and Design (Part V)," U.S. Army Armament Research and Development Command Report ARSCD-CR-79-007, Dover, New Jersey, 07801, September, 1978.

[14] Myung Hun Lee, J. Fred Holmes, and J. Richard Kerr, "Statistics of Speckle Propagation Through the Turbulent Atmosphere," J. Opt. Soc. Am., Vol. 66, November 1976, pgs. 1164-1172.

[15] Philip A. Pincus, Michael E. Fossey, J. F. Holmes, and J. R. Kerr, "Speckle Propagation Through Turbulence: Experimental," J. Opt. Soc. Am., Vol. 66, June 1978, pgs. 760-762.

[16] J. Fred Holmes, Myung Hun Lee, and J. Richard Kerr, "Effect of the Log-Amplitude Covariance Function on the Statistics of Speckle Propagation Through the Turbulent Atmosphere," J. Opt. Soc. Am., Vol. 70, April 1980, pgs. 355-361.

[17] J. Fred Holmes, "Applications of Speckle Phenomena," Reprint from the Proceedings of the Society of Photo-Optical Instrumentation Engineers, Vol. 243, July 1980, pgs. 20-27.

[18] J. Fred Holmes, Farzin Amzajerdian, Rao V. S. Gudimetla, and John M. Hunt, "Remote Sensing of Atmospheric Winds Using Speckle-Turbulence Interaction, a $C0_2$ Laser, and Optical Heterodyne Detection," Applied Optics, Vol. 27, June 1988, pgs. 2532-2538.

[19] V. S. Rao Gudimetla, J. Fred Holmes, and Richard A. Elliott, "Two-Point Joint-Density Function of the Intensity for a Laser-Generated Speckle Field After Propagation Through the Turbulent Atmosphere," J. Opt. Soc. Am., Vol. 7, June 1990, 1008-1014.

[20] V. S. Rao Gudimetla, J. Fred Holmes, P. A. Pincus, and M. E. Fossey, "Time Delayed Covariance of the Received Intensity of a Monochromatic Speckle Pattern in the Turbulent Atmosphere," Applied Optics, Vol. 29, July 1990, pgs. 2872-2875.

[21] S. F. Clifford, *The Classical Theory of Wave Propagation in a Turbulent Medium*, Ch. 2 of Laser Beam Propagatin in the Atmosphere, Springer-Verlag, J. W. Strohben, ed., New York, 1978, pgs. 9-41.

[22] V. A. Banakh and V. L. Mironov, Lidar in a Turbulent Atmosphere, Transleted by S. Chomet, Artech House, Boston MA, 1987, pg. 12.

[23] Peter W. Milonni and Joseph H. Eberly, Lasers, John Wiley & Sons, New York, 1988, Ch. 14, pgs. 480-490.

[24] M. H. Lee, J. F. Holmes, and J. R. Kerr, "Generalized Spherical Wave Mutual Coherence Function," J. Opt. Soc. Am., Vol. 67, September 1977, pgs. 1279-1281.

[25] J. W. Goodman, "Some Fundamental Properties of Speckle," J. Opt. Soc. Am., Vol. 66, November 1976, pgs. 1145-1149.

[26] J. W. Goodman, *Statistical Properties of Laser Speckle Patterns*, Ch. 2 of Laser Speckle and Related Phenomena, Springer-Verlag, J. C. Dainty, ed., New York, 1975, pgs. 9-75.

[27] Douglas C. Draper, J. Fred Holmes, and John Peacock, "Unwrapped-Phase Distribution Model for Speckle and Turbulence," Applied Optics, Vol. 31, June 1992, pgs. 3481-3487.

[28] Christian E. Junge, Air Chemistry and Radioactivity, Academic Press, New York, 1963, pg. 130.

[29] Geoge M. Hidy, Aerosols, an Industrial and Environmental Science, Academic Press, Inc., New York, 1984, pgs. 42-44.

[30] Raymond M. Measures, Laser Remote Sensing, John Wiley & Sons, New York, 1984, pg. 48.

[31] S. Twomey, Atmospheric Aerosols, Elsevier Scientific Publishich Company, New York, 1977, pg. 15.

[32] William C. Hinds, Aerosol Technology, Properties, Behavior, and Measurement of Airborne Particles, John Wiley & Sons, New York, 1982, pg. 5.

[33] H. C. Van De Hulst, Light Scattering by Small Particles, John Wiley & Sons, New York, 1957, pgs. 114-130.

[34] James M. Rosen and David J. Hofmann, "Optical Modeling of Stratopheric Aerosols: Present Status," Applied Optics, Vol. 25, 1 February 1986, pgs. 410-419.

[35] M. L. Wright, E. K. Proctor, L. S. Gasiored, and E. M. Liston, "A Preliminary Study of Air-Pollution Measurement by Active Remote Sensing Techniques," NASA CR-132724, 1975.

[36] David M. Tratt and Robert T. Menzies, "Recent Climatological Trends in Atmospheric Aerosols Backscatter Derived from the Jet Propulsion Laboratory Multiyear Backscatter Profile Database," Applied Optics, Vol. 33, 20 January 1994, pgs. 424-430.

[37] M. J. Post, F. F. Hall, R. A. Richter, and T. R. Lawrence, "Aerosols Backscattering Profiles at λ = 10.6μm," Applied Optics, Vol. 21, 1 July 1982, pgs. 2442-2446.

[38] J. H. Churnside and H. T. Yura, "Speckle Statistics of Atmospherically Backscattered Laser Light," Applied Optics, Vol. 22, September, 1983, pgs. 2559-2565.

[39] S. R. Murty, "Aerosol Speckle Effects on Atmospheric Pulsed Lidar Backscattered Signals," Applied Optics, Vol. 28, March, 1989, pgs. 875-878.

[40] John A. Dutton and Jorgen Hojstrup, "A Model for the Probability Structure of Atmospheric Turbulence," in Preceedings, Conference and Workshop on Wind Energy Characteristics and Wind Energy Siting, Portland, OR, 19-21 June 1979, pgs. 59-67.

[41] Clare D. McGillem and George R. Cooper, Continuous and Discrete Signal and System Analysis, Holt, Rinehart and Winston, 1984, pg. 136.

[42] R. M. Hardesty, R. J. Keeler, M. J. Post, and R. A. Richter, "Characteristics of Coherent Lidar Returns from Calibration Targets and Aerosols," Applied Optics, Vol. 20, November, 1981, pgs. 3763-3769.

[43] J. M. Vaugn, D. Brown, R. D. Callan, P. H. Davies, R. Foord, and W. R. M. Pomeroy, "Correlation Analysis of Fluctuations in Coherent Lidar Signals at 10.6μm," Journal of Modern Optics, Vol. 38, 1991, pgs. 623-648.

[44] R. M. Schotland, P. V. Cvijin, Y. Zhao, and A. M. Hudor, "Conceptual Test of Frequency Diversity Lidar," Applied Optics, Vol. 27, 1988, pgs. 441-443.

[45] G. M. Ancellet and R. T. Menzies, "Atmospheric Correlation-Time Measurements and Effects on Coherent Doppler Lidar," J. Opt. Soc. Am., Vol. 4, 1987, pgs. 367-373.

[46] Peter W. Milonni and Joseph H. Eberly, Lasers, John Wiley & Sons, New York, 1988, Ch. 1, pgs. 5-6.

[47] J. Uozumi and T. Asakura, "First-Order Probability Density Function of the Laser Speckle Phase," Optical and Quantum Electronics, Vol. 12, 1980.

[48] D. C. Draper, J. F. Holmes, and J. Peacock, "Unwrapped-Phase Distribution Model for Speckle and Turbulence," Applied Optics, Vol. 31, 20 June 1992, pgs. 4777-494.

[49] J. Fred Holmes, John J. Peacock, and Douglas C. Draper, "Optical Remote Sensing of Surface Roughness Through the Turbulent Atmosphere," Applied Optics, Vol. 11, 20 December 1994, pgs. 7770-7776.

[50] Robert S. Lawrence and John W. Strohbehn, "A Survey of Clear-Air Propagation Effects Relevant to Optical Communications," Preceedings of the IEEE, Vol. 58, No. 10, October 1970, pgs. 1523-1545.

[51] J. Fred Holmes, "Applications of Speckle Phenomena," Proceedings of the Society of Photo-Optical Instrumentation Engineers, Vol. 243, July 30, 1980, pgs. 20-27.

[52] J. D. Jackson, Classical Electrodynamics, John Wiley & Sons, New York, 1975, pgs. 29-30.

[53] I. S. Gradshteyn and I. M. Ryzhik, Table of Integrals, Series, and Products, Academic Press, New York, 1980, formula 3.322, pg. 307.

[54] Athanasios Papoulis, Probability, Random Variables, and Stochastic Processes, McGraw-Hill, New York, 1984, pg. 272.

[55] V.S. Rao Gudimetla, "Statistics of Polychromatic Speckle Propagation Through the Turbulent Atmosphere," Ph.D. dissertation, Oregon Graduate Institute of Science & Technology, 1982, p. 56.

[56] V. S. Rao Gudimetla, J. Fred Holmes, M. E. Fossey, and P. A. Pincus, "Covariance of the Received Intensity of a Partially Coherent Laser Speckle Pattern in the Turbulent Atmosphere," Applied Optics, Vol. 30, 20 March 1992, pgs. 1286-1293.

[57] J. Fred Holmes and V. S. Rao Gudimetla, "Variance of Intensity for a Discrete-Spectrum, Polychromatic Speckle Field After Propagation Through the Turbulent Atmosphere," J. Opt. Soc. Am., Vol. 71, No. 10, October 1981, pgs. 1176-1180.

[58] David A. Bowdle, Jeffry Rothermel, J. Michael Vaughan, Derek W. Brown, and Madison J. Post, "Aerosol Backscatter Measurements at 10.6 Micrometers with Airborne and Ground-Based CO_2 Doppler Lidars over the Colorado High Plains 1. Lidar Intercomparison," Journal of Geophysical Research, Vol. 96, March 20, 1991, pgs. 5327-5335.

[59] J. F. Holmes, F. Amzajerdian, and J. M. Hunt, "Improved Optical Local Oscillator Isolation Using Multiple Acousto-Optic Modulators and Frequency Diversity," Optics Letters, Vol. 12, August 1987, pg. 637-639.

[60] F. Jessie MacWilliams and Neil J. A. Sloane, "Pseudo-Random Sequences and Arrays," Proc. of the IEEE, Vol. 64, December 1976,pgs. 1715-1729.

[61] N. Takeuchi, N. Sugimoto, H. Baba, and K. Sakurai, "Random Modulation CW Lidar," Applied Optics, Vol. 22, 1 May 1983, pgs. 1382-1386.

[62] Nobuo Takeuchi, Hiroshi Baba, Katsumi Sakurai, and Toshiyuki Ueno, "Diode-Laser Random-Modulation CW Lidar," Applied Optics, Vol. 25, 1 January 1986, pgs. 63-67.

[63] Chikao Nagasawa, Makoto Abo, Hideki Yamamoto, and Osamu Uchino, "Random Modulation CW Lidar Using New Random Sequence," Applied Optics, Vol. 29, 1 April 1990, pgs. 1466-1470.

[64] David C. Rife and Robert R. Boorstyn, "Single-Tone Parameter Estimation from Discrete-Time Observations," IEEE Transactions of Information Theory, Vo. IT-20, September 1974, pgs. 591-598.

[65] D. Slepian, "Estimation of Signal Parameters in the Presence of Noise," IRE Transcations of Information Theory, PGIT-3, March 1957, pgs. 68-89.

[66] David C. Rife and Robert R. Boorstyn, "Multiple Tone Parameter Estimation from Discrete-Time Observations," Bell System Technical Journal, Vol. 55, November 1976, pgs. 1389-1409.

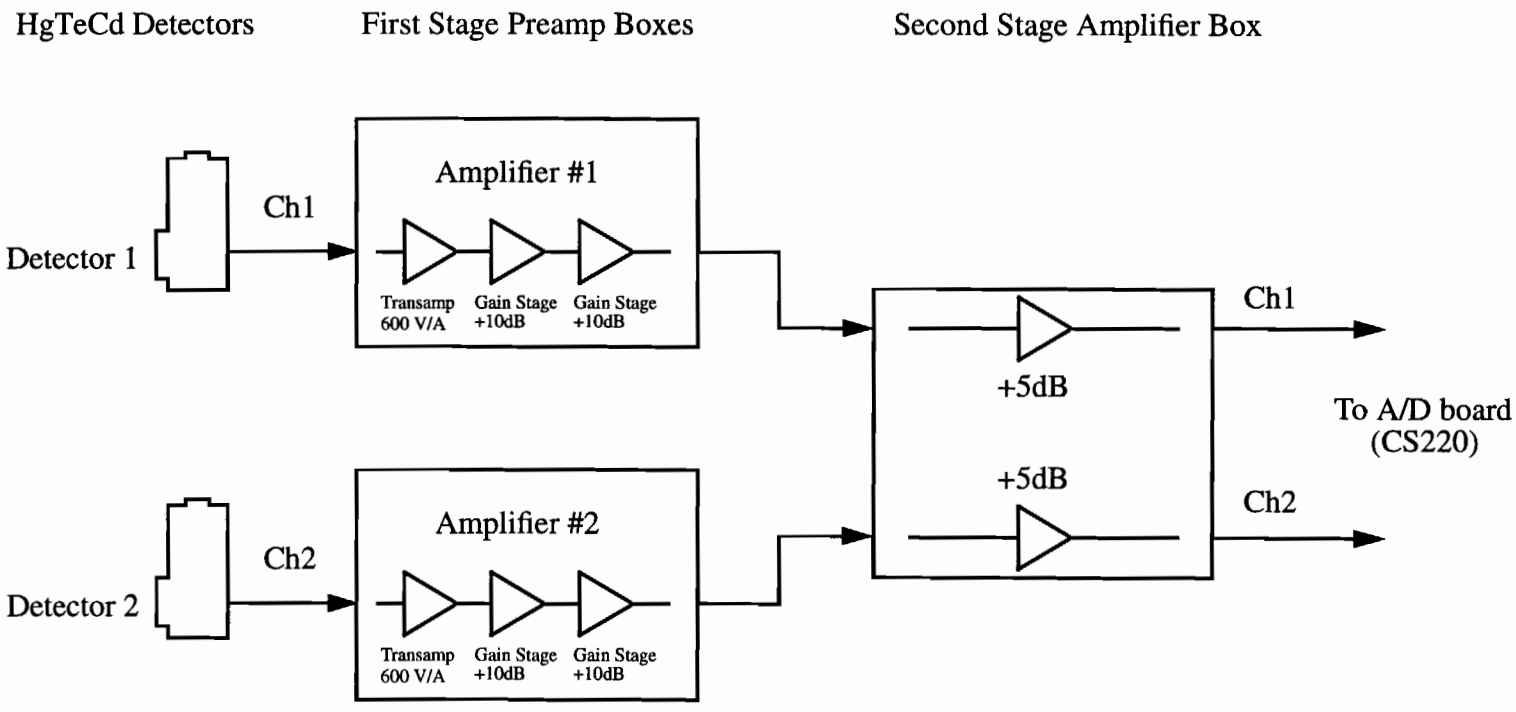
[67] Ting-i Wang, G. R. Ochs, and R. S. Lawrence, "Wind Measurements by the Temporal Cross-correlation of the Optical Scintillations," Applied Optics, Vol. 20, 1 December 1981, pgs. 4073-4081.

[68] D. L. Fried, "Aperture Averaging of Scintillation," J. Opt. Soc. Am., Vol. 57, February 1967, pgs. 169-175.

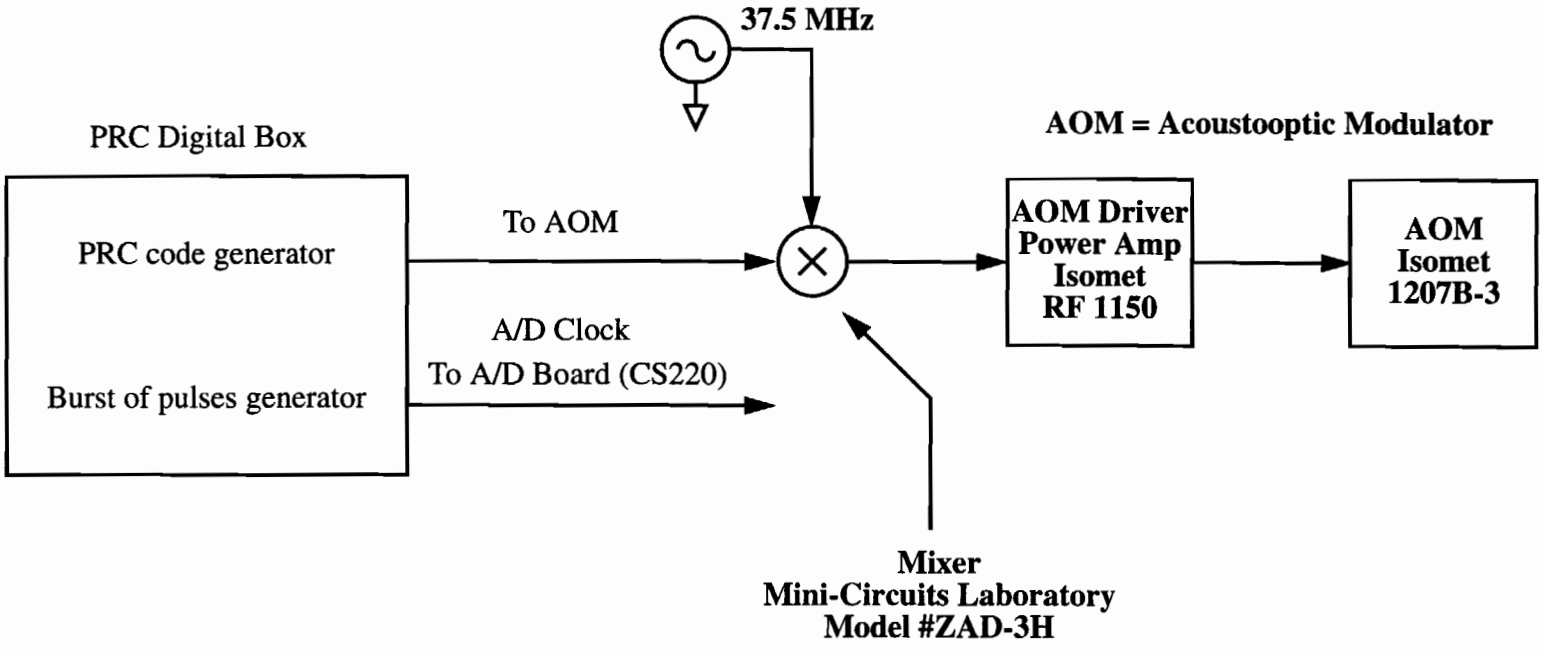
Appendix A

Appendix A describes the electronics of the PRC generator and of the amplifiers for both channels. The clock of the PRC generator comes from the burst of pulses generator. The burst of pulses generator is part of the new data acquisition schemes for the crosswinds in order to spread out the intensity estimates in time. The PRC generator needs a 40 MHz clock which is provided by the burst generator.

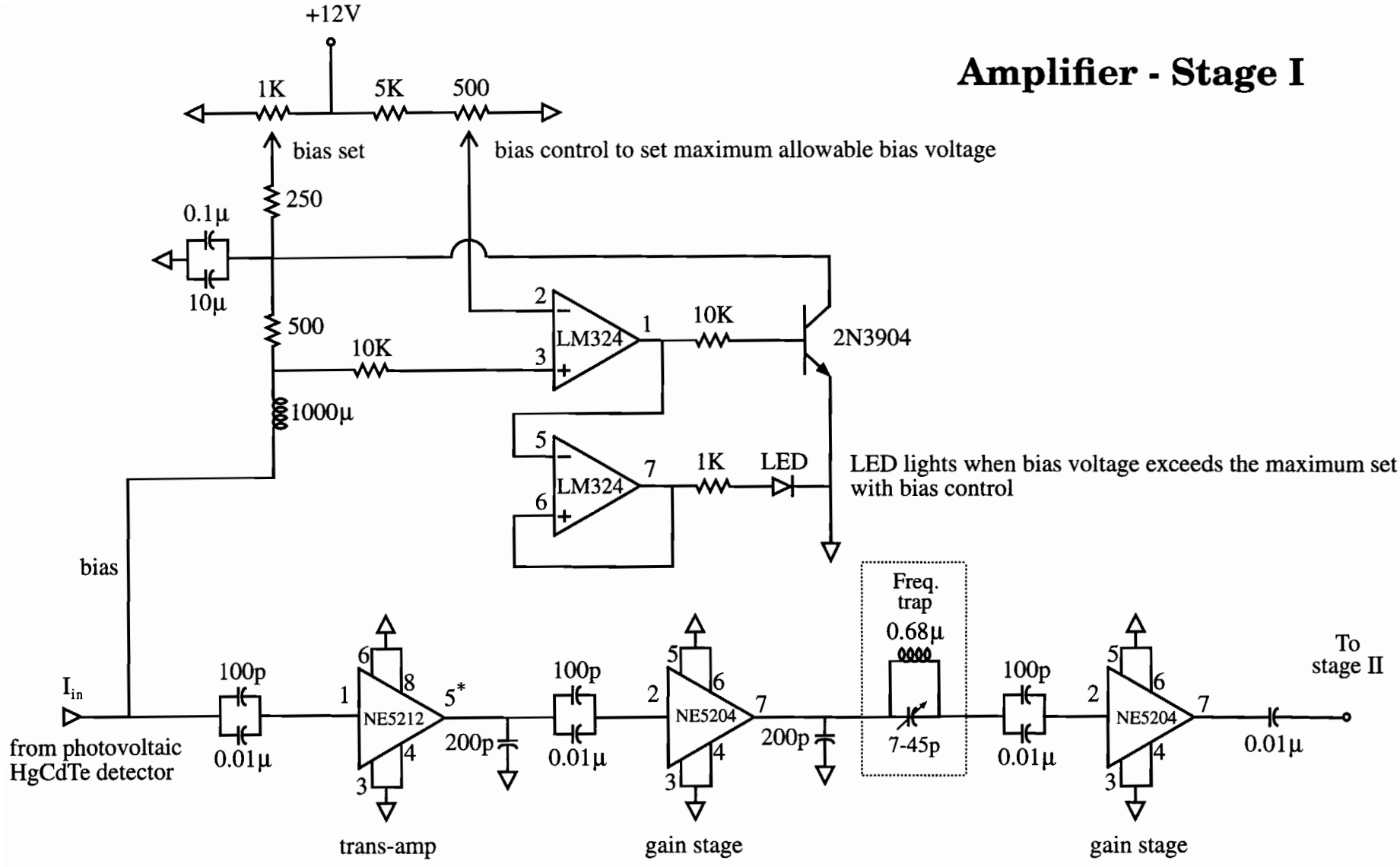
Detector and Amplifier Block Diagram



PRC Code Generator and AOM Block Diagram



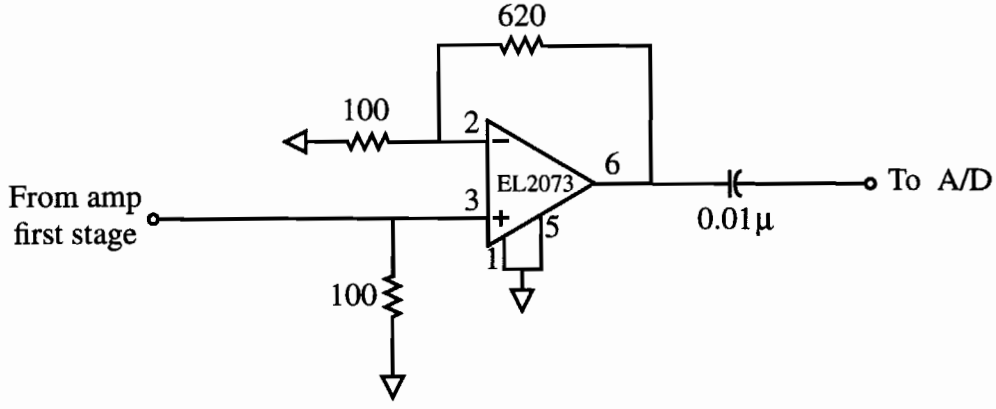
Amplifier - Stage I



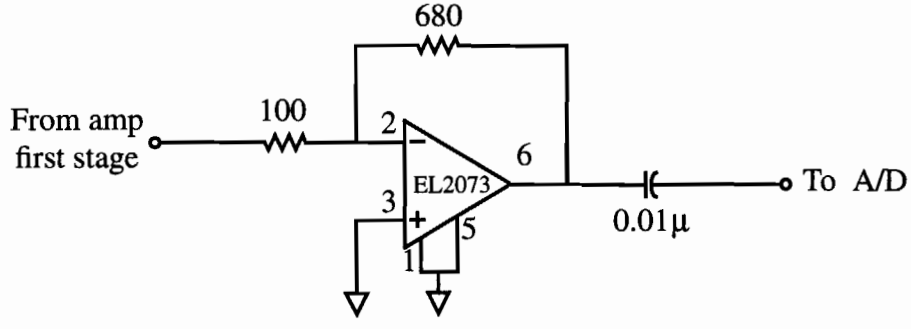
* Ch.2 uses pin 7, inverting output

Amplifier - Stage II

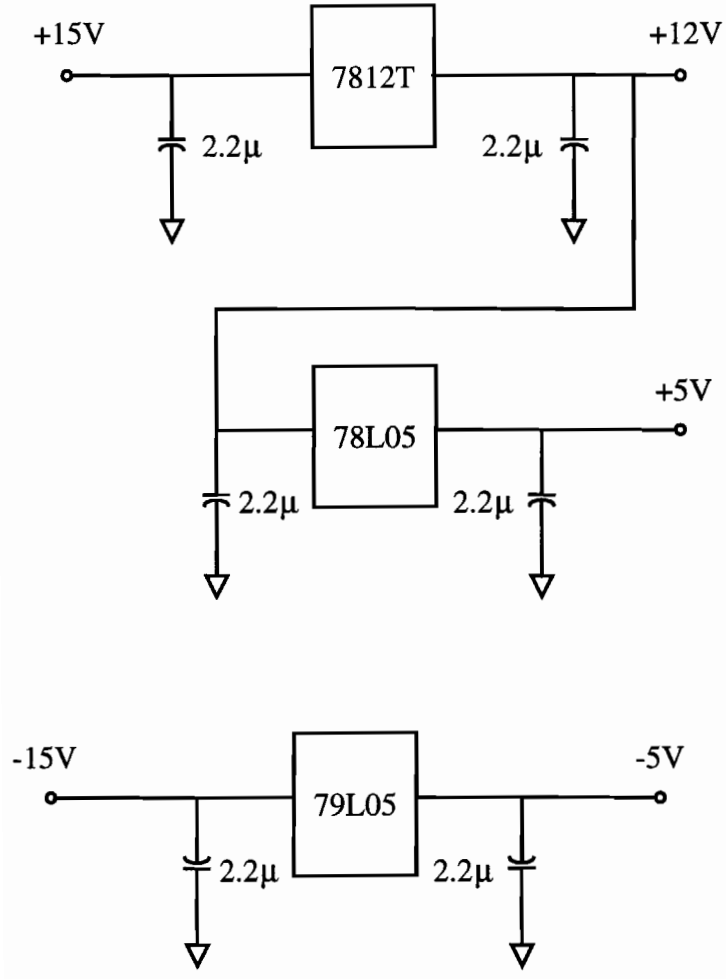
Ch. 1

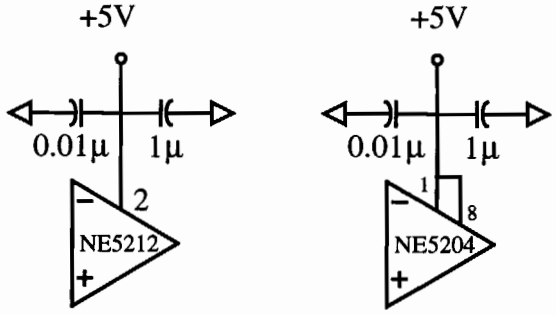


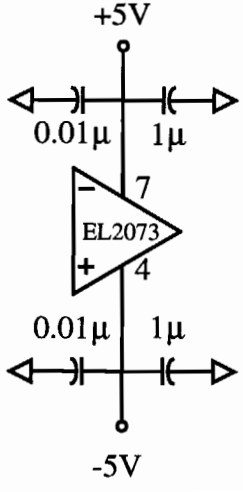
Ch. 2



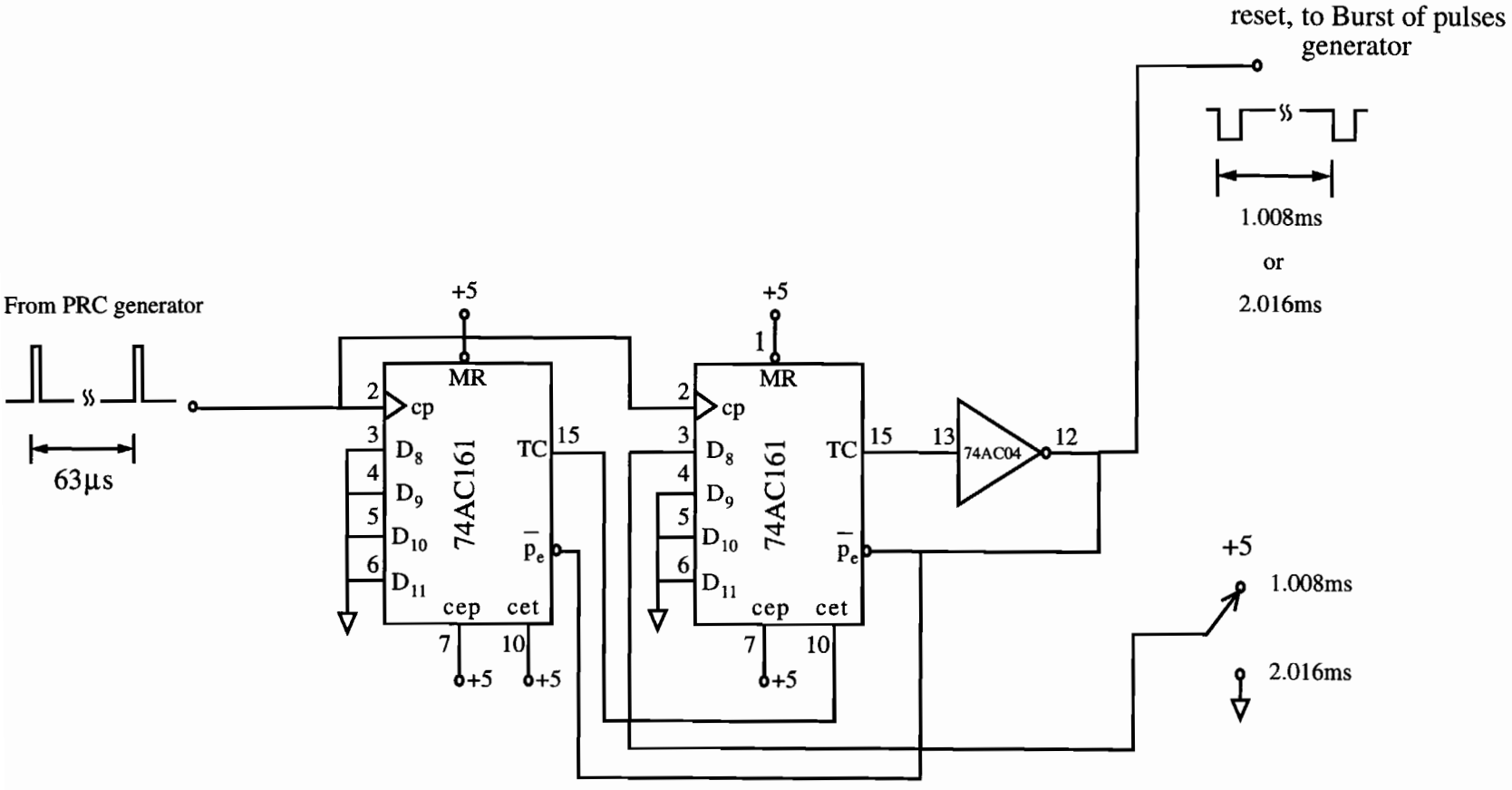
POWER FOR AMPLIFIERS



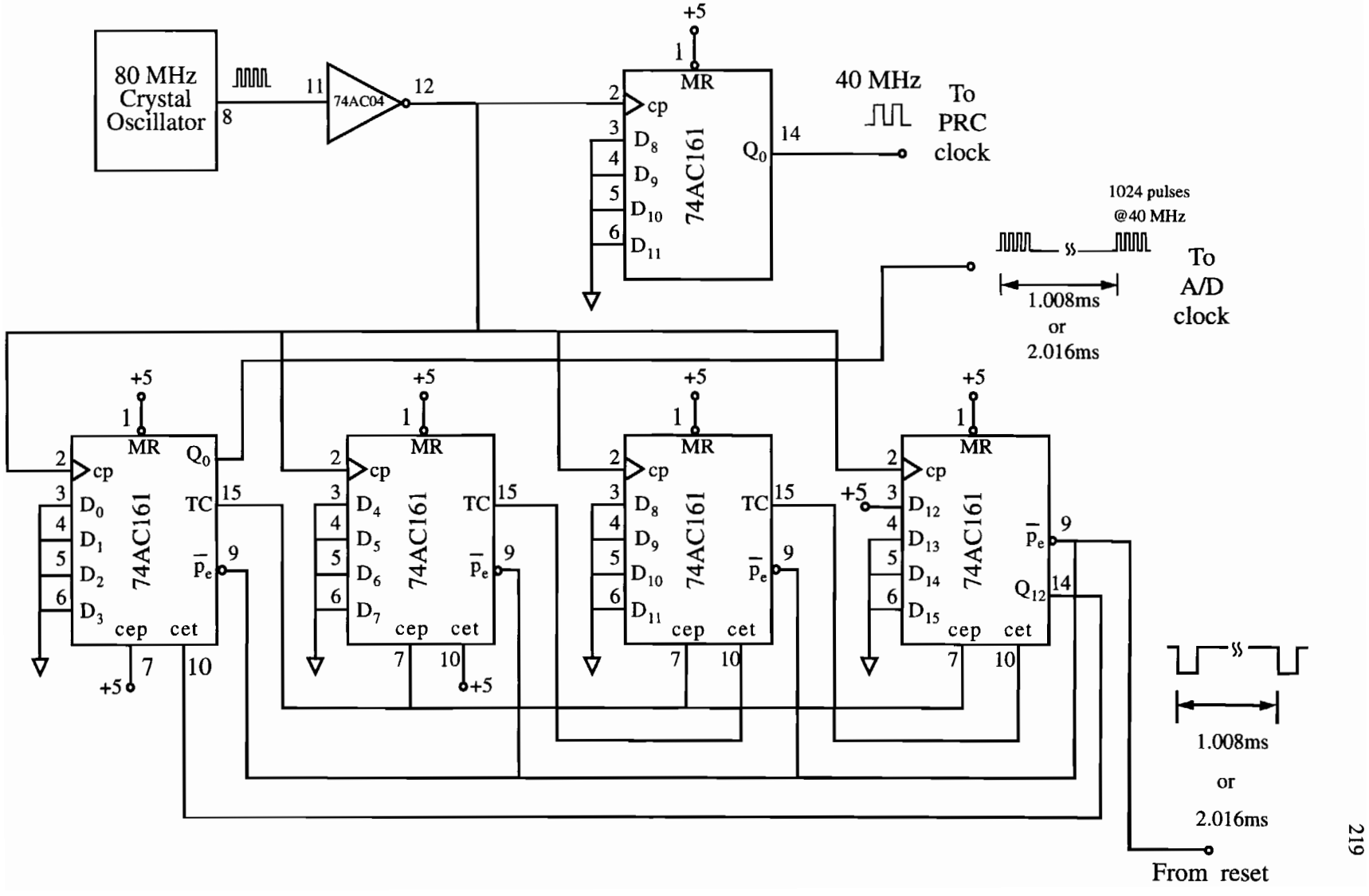




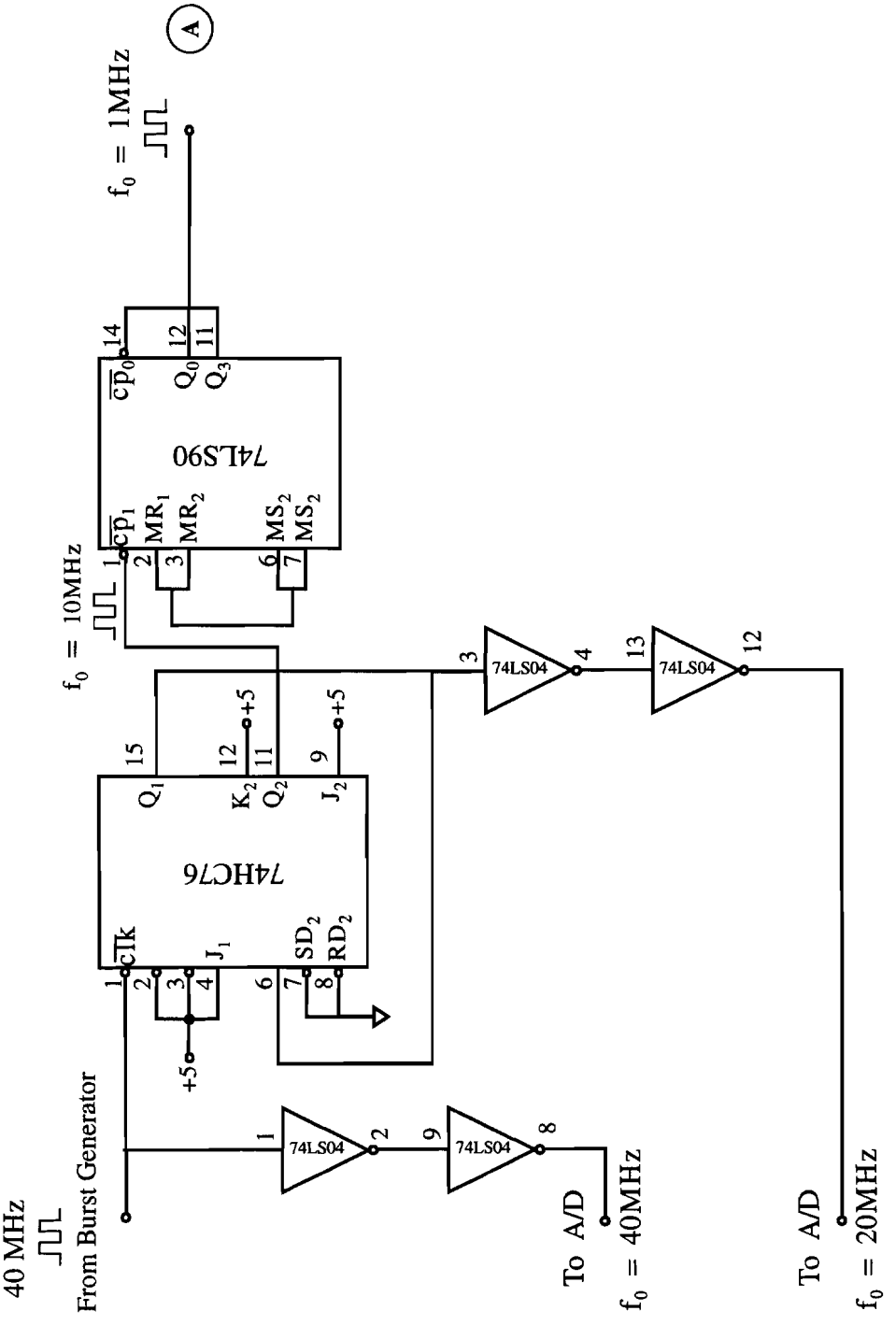
Reset Network, Used by Burst of Pulses Generator



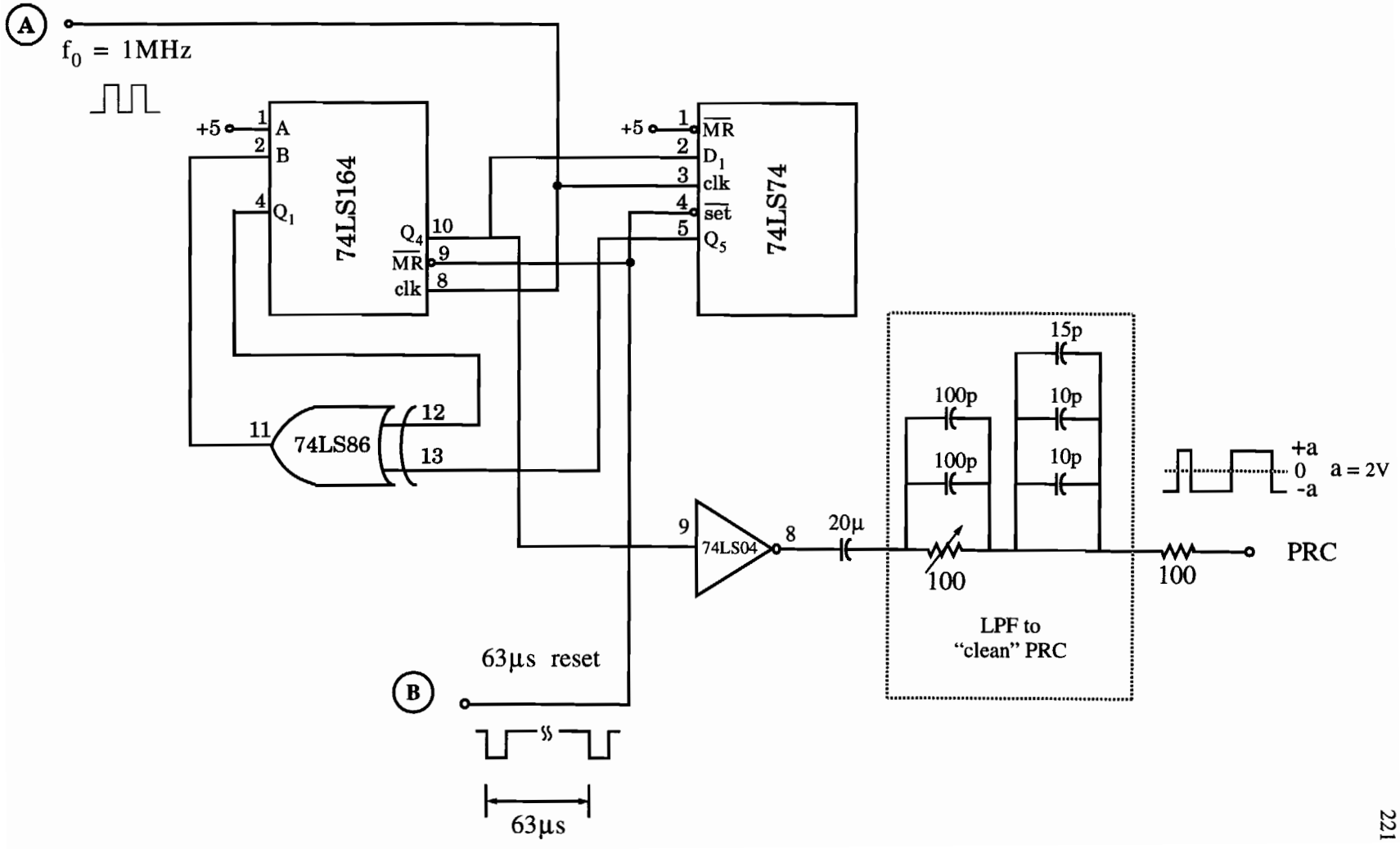
Burst of Pulses A/D Clock, PRC Clock, Generator



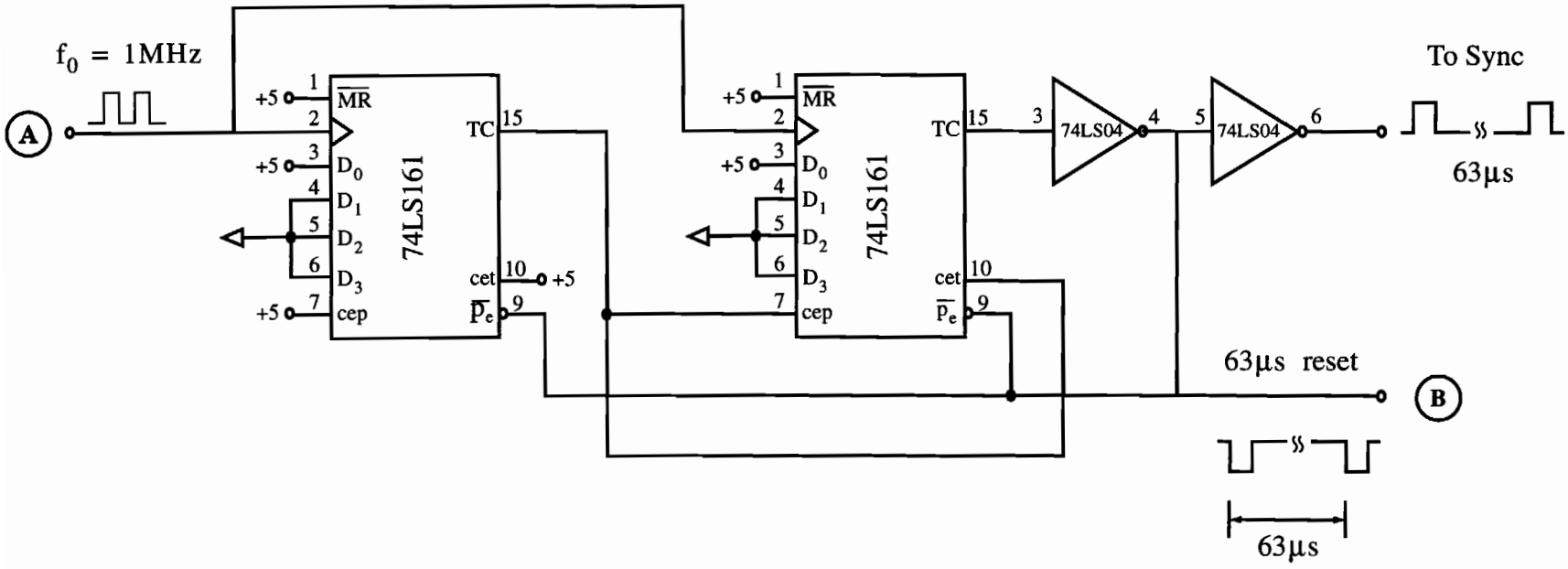
Frequency Divide by 40 Network Used by PRC Generator



63-State PRC Code Generator



63µs Count Reset



POWER AND BYPASS FOR IC'S

Chip	Pin		Bypass
	+5	GND	
74AC161	16	8	1μF || 0.01μF on Vcc
74AC04	14	7	1μF || 0.01μF on Vcc
74HC76	5	13	0.1μF on Vcc
74LS90	5	10	0.1μF on Vcc
74LS164	14	7	0.1μF on Vcc
74LS161	16	8	0.1μF on Vcc
74LS04	14	7	0.1μF on Vcc
74LS74	14	7	0.1μF on Vcc
74LS86	14	7	0.1μF on Vcc
80 MHz Crystal	14	7	0.1μF on Vcc

Appendix B

Appendix B contains a sampling of the data acquisition and storage programs. The four main data acquisition and storage programs are: "data2hd.c," "datadopp.c," "hdto-tape.c," and "tapetohd.c." Program "data2hd.c" is the program to control the CS220 to take 2-channel data used to estimate the crosswinds and to control the CI08 board to take the in situ data. Program "datadopp.c" is the program used to take streams of Doppler data. Program "hdtotape.c" is the program to write data from the PC hard drive to the Exabyte 8500 tape drive. Program "tapetohd.c" retrieves the data from the Exabyte 8500 to the hard drive. Included in this appendix are programs "data2hd.c" and "hdtotape.c" to give examples of the data acquisition and storage process. The remaining programs can be found on a floppy disk labeled "DATAQ/STORE PROGRAMS" and are also listed as part of a hard copy of all of the programs used throughout this thesis project titled "CO2 LIDAR PROGRAMS."

```
/*****************************************************************************\
*                Program :  data2hd.c        version 1                        *
*                                                                             *
*                LANGUAGE: C                                                  *
*                                                                             *
*                written by                                                   *
*                Badih John Rask                                              *
*                8-31-93                                                      *
*                                                                             *
* --- revised 9-30-93 to add CI08 insitu A/D board commands --                 *
*                                                                             *
*     This program controls the cs220 compuscope to record the data           *
*     from a hardtarget or aerosols. Program has been written in the          *
*     C language; therefore, as many comment lines as possible have           *
*     been included.  datatohd.c, "data to hard drive," transfers data to     *
*     c: hard drive.  Each channel records two MEG points.  The 2 MEG         *
*     points per channel are transfered as follows: First, 34 sets            *
*     of 60K points are transfered.  The data is transfered from PC's RAM to  *
*     binary files on the c: hard drive in 60K chunks.  Program also          *
*     stores data from the Campbell Scientific unit on Channel B.  The        *
*     data from the Campbell unit is the wind speed.                          *
*                                                                             *
*     The PRC code generator supplies the sync and clock signals used         *
*     by the CS220.                                                           *
\*****************************************************************************/
#include      <malloc.h>
#include      <conio.h>
#include      <string.h>
#include      <stdlib.h>
#include      <process.h>
#include      <stdio.h>
#include      <graph.h>
#include      <dos.h>
#include      <memory.h>
#include      "cs220drv.h"   /* This library has the cs220 drive commands      */

float sumChA, sumChB;        /* from CI08 board, averaged data                 */
float Vpave, Cn2;            /* data from CI08 board, in volts                 */
unsigned char huge *data;    /* array that contains the data read from CS220   */
int i;                       /* counter                                        */
int numchan;                 /* number of channels user wants to record        */
int numsets;                 /* number of sets to record                       */
char *p;                     /* pointer used by strcat and strcpy              */
char samprate[3];            /* sampling rate as a character string            */
char RATE[7];                /* used by calling routine set_capture_mode to
                                set sampling rate                              */
char line[100];              /* input character strings                        */
int sr;                      /* RATE_XX as integer                             */
int nblcks;                  /* the number of blocks of blsize bytes           */
int ierr;                    /* error status used by tape drive commands       */
unsigned int npoints;        /* number of points per data set                  */
int numsets;                 /* number of sets to record                       */
FILE *fp, *fi, *fopen ();    /* binary file on hard drive that holds data
                                and ASCII file that holds insitu data          */
char fileout[14];            /* name of output binary file                     */
char fileinsitu[14];         /* name of insitu data filename                   */

unsigned int j,k;            /* counter                                        */
int npoints_offset;          /* number of remaining points (see comments below)*/
unsigned char npoints4K;     /* number of 4K-byte blocks to
                                move from CS220 to array "data"                */
long count0;                 /* count is starting address of
                                data where trigger occurred.                   */
long count;                  /* there will be 34 sets of 60K
                                points moved. count is the
                                starting place for each 60K chunk.             */
long total_points_moved;     /* no explanation needed                          */
long npoints_moved;          /* number of points moved within the 60K chunk    */
int npoints_to_move;         /* number of bytes to move used by
                                memmove. 1 < npoints_to_move < 4K              */
int count4K;                 /* number of multiples of 4K of
                                count. example:
                                count = 5000
                                count4K = 5000/4K = 1                          */
int count_offset;            /* number of remaining points (see
                                comments below)                                */
int nblcks;                  /* used by XWRDMA                                 */
word blocknumber;            /* used by set_block_number                       */
word block_offset;           /* If chan = CHAN_A, block_offset=0
                                If chan = CHAN_B, block_offset=1024
                                used to access correct channel                 */
char *data_address;          /* address of data array                          */
char *src_address;           /* address of source to transfer
                                to data array, used by memmove                 */
unsigned int byteswritten;   /* number of bytes written to file, return
                                value from write command                       */
float Vpave;                 /* the crosswind speed from the Campbell
                                unit. The insitu data.                         */
float scale;                 /* specifies digitization range                   */
/*-----------------------------------------------------------------------------*/

main()

{

      data = (unsigned char huge *)halloc(61440, sizeof(unsigned char));
       if (data == NULL)
              {
                printf("halloc failed\n");
                exit (1);
              }
```

```
    _clearscreen( _GCLEARSCREEN ); /* clears the screen */

    printf("\n");
    printf("\n");
    printf("*********************************************************\n");
    printf("* Welcome to the CS220 Compuscope Controlling Program *\n");
    printf("*********************************************************\n\n");
    printf("-----------------------------------------------------\n");
    printf("\ program: data2hd.c / \n");
    printf(" ----------------------------------------------------- \n");
    printf("\n\n");
    printf("****This program records 2-channel data to EXABYTE tape drive****\n");
    printf("\nmultiple 2 Meg sets per channel is sent to hard drive \n");
    printf("in one binary file. The program also records 2-channel insitu\n");
    printf("with the CI08 board and stores insitu data in 1 insitu file\n");


/***  start input section ***/


    printf("\n");
    printf("\n");
    printf("Program INPUT section:\n\n\n\n");

    printf("Input the number of channels to record\n");
    scanf("%d", &numchan);

    printf("INPUT the number of sets of 2 Meg points/channel to record\n");
    scanf("%d", &numsets);

    printf("\nINPUT the binary output filename for storage on hard drive\n");
    scanf("%s", fileout);

    printf("\nINPUT the output filename for the insitu wind speed\n");
    printf("and Cn2\n");
    scanf("%s", fileinsitu);

    printf("\nINPUT the range over which to digitize\n");
    printf(" INPUT FOR\n");
    printf("==================\n");
    printf(" .1 +-10 Volts\n");
    printf(" .2 +-5 Volts\n");
    printf(" .5 +-2 Volts\n");
    printf(" 1 +-1 Volts\n");
    printf(" 2 +-500 mVolts\n");
    printf(" 5 +-200 mVolts\n");
    printf(" 10 +-100 mVolts\n");
    fgets(line, sizeof(line), stdin);
    sscanf(line, "%f", &scale);

    fp = fopen ( fileout, "w+b");          /* w is write mode for outputfiles.
                                              b means binary.                  */
    fi = fopen ( fileinsitu, "w");

/***          Done with input section                    ***/
/***          Initialize the CS220 Data Acquisition Board ***/
/***          This is for one or two channel data capture ***/

    initialize_cs220 ();                   /* This initializes cs220 hardware
                                              to default settings              */
    start_system_clock ( TRUE );           /* Starts system clock              */
    set_capture_mode (DUAL_CHAN, RATE_20, MHZ);
    needram(TRUE);                         /* Connect CS220 to PC bus          */
    setcoupling(CHAN_B, AC);               /* Set coupling of Ch.B             */
    setcoupling(CHAN_A, AC);               /* Set coupling of Ch.A             */
    select_trigger(EXT);                   /* The trigger is supplied from
                                              PRC sync pulse                   */
    settriggerlevel ( 0xc0 );              /* corresponds to 2.5 volts         */
    clkregs(0,0,0);                        /* select external clock to be
                                              supplied by PRC code generator   */
    setgain(CHAN_A, scale);                /* specify digitization range       */
    setgain(CHAN_B, scale);                /* specify digitization range       */

    if ( (setsegment(0xD000) != 0 ) )
          printf("setsegment did not set properly\n");

/***          Done with Initialize CS220                 ***/

/*-------------------------------------------------------------------------------*/
/* set-up the CI08 board (takes insitu data) */

          initialize_ci08();  /* set-up the CI08 insitu board */
/*-------------------------------------------------------------------------------*/

    /* start two channels recorded, A and B */

    _clearscreen( _GCLEARSCREEN );
    printf("\n\n");
    printf("Data Acquisition mode is:\n\n");
    printf(" Data recorded to binary file: %s \n", fileout);
    printf(" Dual Channels recorded, Trigger source Ext. Channel\n");
    printf(" Concurrent sampling\n");
    printf(" Sampling rate = %s MHz\n", samprate);
    printf(" Number of Points/Set = 2 Meg\n");
    printf(" Number of Data Sets per channel = %d\n", numsets);

    printf("\n\n");
    printf("CHECK that signals are connected to Channels A & B\n");
    printf("on the CS220\n");
    printf("CHECK that trigger source is connected to External\n");
    printf("on the CS220\n");
```

```
printf("CHECK that Vcross_ave is connected to Ch0 and Cn2 is\n");
printf("connected to Ch1\n");
printf("\n\n");
printf("Press <return> to activate DATA RECORD procedure\n");
getch();                    /* wait for a keypress                                  */

printf("\n\n\ntransfering 2 Meg set #:\n");

/***     start data capture for two channels                          ***/

for (i = 0 ; i < numsets ; i++ )
        {
        /*** first the CS220 A/D ***/

        needram ( FALSE ); /* Disconnects CS220 RAM from PC */

        /*-------------------------------------------------------------------------*/
        getsample();    /* Captures data to A and B registors
                           at specified sampling rate          */

        do
                {
          /* printf("\r Wait for trigger event");             */
                } while ( !check_if_trigger () );
        do
                {
                   /* printf("\r Check if done recording data"); */
                } while ( !check_if_notbusy () );

        /*-------------------------------------------------------------------------*/

        needram (TRUE); /* reconnects cs220 to PC bus        */

        /*** now for the CI08 A/D board ***/
        /*    the function "take_data" is the CI08 routine that takes the in situ data */

        sumChA = sumChB = 0.0;

        for (j=0; j<10; j++)
                {
                   Vpave = 1.25*(take_data(1)-2048.)/2048.0;
                   sumChA = sumChA + Vpave;
                   Cn2 = 1.25*(take_data(2)-2048.)/2048.;
                   sumChB = sumChB + Cn2;
                }
         Vpave = sumChA/10.;
         Cn2 = sumChB/10.;

        /*** store the system data ***/

         printf("\r%d, Ch. 1", i+1);
                store_data_hd ( CHAN_A,data);  /* subroutine to store data tape      */
                printf("\r%d, Ch. 2", i+1);

                store_data_hd ( CHAN_B,data); /* subroutine to store data to tape    */

                /*** now write the insitu data ***/

                fprintf(fi, "%d, %f, %f\n", i, Vpave, Cn2);

        /***     End of data capture for two channels ***/
        }       /* end of i loop */
                printf("\n");


printf("\n\n\n\n\t---End of program 'data2hd.c'---\n");

fclose(fi);
fclose (fp);

} /* end of main of datatohd.c */

/*************************************************************************/
        store_data_hd ( int channel, unsigned char huge *data)
/*************************************************************************/
{
/*
        Subroutine that stores one channel's data to the PC hard drive.
        The CS220 stores data in its own RAM, however, it is memory mapped,
        and can be reassigned to a section of the PC's RAM. The
        PC can only access a 4K-byte window; to access another 4K bytes the
        window must be incremented."setsegment(address)" is used to specify the
        address of the first byte of the PC's RAM to which the CS220's RAM
        is reassigned. "setblocknumber" is used to move the CS220's 4K
        RAM window so that the PC can access a new 4K page of CS220 memory.
        With the Microsoft C command, "memmove," the 4K bytes are moved
        from the address specified by setsegment to the starting address of
        a character array.Finally, the character array is stored onto the
        PC hard drive with the "fwrite" command. This subroutine
        records 34 chunks of 60K points (which is the largest integer of 60K
        points which does not exceed 2 Meg points) for each channel.
*/

/*-------------------------------------------------------------------------------*/

        if (channel == CHAN_A)
                block_offset = 0;
        else
                block_offset = BANK_OFFSET_VALUE;

        find_trigger_address( &count0 );
```

```
    npoints = 61440;

    count4K = count0/4096;              /* if count0 = 5000,
                                           count4K = 1. The
                                           number of 4K chunks
                                           in count0.                    */
    count_offset = count0 - count4K*4096;  /* This is the number
                                           between 4K boundaries
                                           of count0. If count =
                                           5000, ct_off = 904.           */
    blocknumber = count4K;              /* data for fast transfer
                                           is moved in 4K blocks.
                                           blocknumber is the
                                           block that contains
                                           count0.  If count0 =
                                           5000, blnum = 1 (this
                                           is the second block)          */
    npoints_moved = 0;                  /* this variable is the
                                           number of points
                                           moved and the
                                           array element address
                                           of the destination.           */
    npoints_to_move = (count4K+1)*4096 - count0; /*This is the number of
                                                   points to move for
                                                   the first block move.
                                                   If count = 5000,
                                                   nps_to_mv = 8192 -
                                                   5000                  */
    total_points_moved = 0;

for (j = 0; j < 34; j++)

    {       /* transfer 34 chunks of 60K points to array                */

            count = count0 + total_points_moved;   /* count0 has the trigger
                                                      address. count is the
                                                      starting address of
                                                      each 60K chunk       */
            count4K = count/4096;                  /* if count = 5000,
                                                      count4K = 1. The
                                                      number of 4K chunks
                                                      in count.            */
             count_offset = count - count4K*4096;  /* This is the number
                                                      between 4K boundaries
                                                      of count. If count =
                                                      5000, ct_off = 904.  */
            blocknumber = count4K + block_offset;  /* data for fast transfer
                                                      is moved in 4K blocks.
                                                      blocknumber is the
                                                      block that contains
                                                      count.  If count =
                                                      5000, blnum = 1 (this
                                                      is the second block).
                                                      block_offset gets
                                                      channel A or B       */
            npoints_moved = 0;                     /* this variable is the
                                                      number of points
                                                      moved within the
                                                      60K chunk and the
                                                      array element address
                                                      of the destination.  */

            src_address = (char *)( (unsigned long)(0xD000)*65536
                        + count_offset);           /* The source address
                                                      is the segment +
                                                      count_offset only for
                                                      the first chunk of
                                                      data moved.          */
            npoints_to_move = (count4K+1)*4096 - count;  /* This is the number of
                                                      points to move for
                                                      the first block move.
                                                      If count = 5000,
                                                      nps_to_mv = 8192 -
                                                      5000                 */

/*-------------------------------------------------------------------------------------------*/
/*---- move data from cs220 to array, data in 60K blocks ----*/

        do
                {
                 if (blocknumber == 1024 && channel == CHAN_A)
                {
                 blocknumber = 0;              /* there are 500 blocks 4096
                                                  bytes in 2 Meg points. When
                                                  the blocknumber gets to 500,
                                                  must wrap around to the 0th
                                                  block.                   */
                }
        if (blocknumber == 2048 && channel == CHAN_B)
                {
                 blocknumber = 1024;           /* same reasons as above    */
                }

        set_block_number(blocknumber);         /* get this block of 4096
                                                  bytes from CS220's RAM.
                                                  If count = 5000, want 1st block
                                                  If count = 24, want 0th, etc. */

        data_address = &data[npoints_moved];   /* destination              */

        memmove(data_address, src_address,
```

```
            npoints_to_move);           /* moves the data                          */

    npoints_moved = npoints_moved +
            npoints_to_move;            /* keeps track of the # of
                                           points moved.                           */
    blocknumber = blocknumber + 1;
    src_address = (char *)( (unsigned long)(0xD000)*65536);
                                        /* only the first
                                           chunk of data
                                            needs to
                                            consider
                                            count_offset                           */

    npoints_to_move = 4096;             /* after the first chunk
                                           of data, move 4096 points
                                           until "while" condition.
                                           Example:
                                           if  npoints = 61440
                                           count = 5000 :
                                           then data will be moved
                                           in 1) 8192-5000 pts.
                                              2) 4096 pts. for 14 times
                                           3) 61440- 14*4096 - (8192-5000) = 904 pts.
                                           1) and 2) are in while loop but
                                           3) is outside while
                                              loop below                           */

} while (npoints - npoints_moved >= 4096);


    if (blocknumber == 1024 && channel == CHAN_A)
            {
              blocknumber = 0;          /* When the blocknumber gets to 512,
                                           must wrap around to the 0th block.       */
            }

    if (blocknumber == 2048 && channel == CHAN_B)
            {
              blocknumber = 1024;       /* same reasons as above                    */
    }

    set_block_number(blocknumber);

    npoints_to_move = npoints -npoints_moved;          /* If npoints = 61440, and
                                                          count = 5000, then
                                                          nps_to_mv = 61440-
                                                          14*40 - (8192-5000)
                                                          = 904 pts. */

    data_address = &data[npoints_moved];
    src_address= (char *)( (unsigned long)(0xD000)*65536);

    memmove(data_address, src_address,
            npoints_to_move);           /* gets the rest ( from
                                           3) above )                              */

    /*=================================================*/
    if ((byteswritten = fwrite(data, sizeof(unsigned char), npoints, fp)) == -1)
            printf("write failed\n");
    /*=================================================*/

    total_points_moved = total_points_moved + npoints;

  }         /* loop j, end of 34*60K points to file                               */

}                       /* end of store_data_hd subroutine                        */
```

```
/*****************************************************************************\
*                          Program          hdtotape.c        version 1       *
*                                                                             *
*                          written by                                         *
*                          Badih John Rask                                    *
*                           12-19-92                                          *
*                                                                             *
* This program transferes data from the hard drive to the tape                *
* drive (Exabyte 8500).  Data has been stored on hard drive in                *
* one binary data file.                                                       *
\*****************************************************************************/
#include          <malloc.h>
#include          <conio.h>
#include          <string.h>
#include          <stdlib.h>
#include          <process.h>
#include          <stdio.h>
#include          <graph.h>
#include          <dos.h>
#include          <memory.h>

FILE *fp, *fopen();
unsigned char  *data;                  /* array that contains the data read from HD      */
unsigned int bytesread;                /* number of bytes to read from binary file       */
int npts10K;                           /* number of 10K chunks in npoints                */
unsigned int tenK;                     /* 10240 as unsigned integer                      */
int i;                                 /* counter                                        */
int nblcks;                            /* # of 1K blocks to transfer to tape             */
int nsets;                             /* # of sets of 100*20K = 1024000 points          */
int numsets;                           /* number of sets to record                       */
int ierr;                              /* error status used by tape drive commands       */
char *filename;                        /* binary filename                                */
char answer[12];                       /* answer to questions                            */
int  compresult;                       /* compare two character strings                  */
long start_block_number;               /* starting 1024-byte block number, used as a
                                          counter for the data set number              */
int set_number;                        /* the particular 2 MEG set number                */
/*-----------------------------------------------------------------------------*/

main()

{

  data = (unsigned char *)calloc(10240, sizeof(unsigned char));
      if (data == NULL)
              {
                printf("halloc failed\n");
                exit (1);
              }

      _clearscreen( _GCLEARSCREEN );     /*  clears the screen   */

      printf("\n");
      printf("\n");
      printf("*************************************************************\n");
      printf("* Welcome to the data-to-EXABYTE-tape-drive transfer program *\n");
      printf("*************************************************************\n\n");
      printf("-----------------------------------------------------\n");
      printf("\          program:  hdtotape.c          / \n");
      printf(" ----------------------------------------------------- \n");
      printf("\n\n");
      printf("****This program transfers data to the EXABYTE tape drive****\n");
      printf("\n");


/***           start input section                    ***/

      start_block_number = 1;
      set_number = 1;

      printf("\n");
      printf("\n");
       printf("Program INPUT section:\n\n\n\n");
      printf("\n\nWhat is input binary filename? ");
      scanf("%s", filename);

      printf("\n\nHow many 2-Meg sets to transfer from HD to tape? ");
      scanf("%d", &nsets);
      npts10K = nsets*204;
      nblcks = 10;
      tenK = 10240;

      fp = fopen(filename, "r+b");

      initialize_tape();                                   /*  subroutine that initializes the
                                                               tape drive to take data          */

/***           Done with input section    ***/


      printf("\n\nPress <return> to transfer data to tape drive\n");
      getch();                    /*  wait for a keypress                 */

/***           start data transfer                    ***/

      for (i = 0 ; i < npts10K ; i++ )
         {
              /*=======================================================*/
               if ((bytesread = fread(data, sizeof(unsigned char), tenK, fp)) == -1)
                              printf("write failed\n");
              /*=======================================================*/
```

```
        /*================================*/
         XWRDMA(data, &nblcks, &ierr);              /* write data to tape          */
        /*================================*/

        printf("percent done = %f \r", (i+1)*100.0/npts10K);

        }

        printf("\n\n\n\n\t---End of program 'hdtotape.c'---\n");

} /* end of main of hdtotape*/


/*************************************************************************/
                        initialize_tape()
/*************************************************************************/
{
/*  Subroutine that initializes the tape drive to accept new data.  */

    int k, l;                           /* counters                          */
    int id = 0;                         /* id sets the destination for the data.  */
    int itype=0;                        /* returned by SETTGT. itype = 0
                                           means the EXABYTE 8500            */
    int compresult;                     /* compare two strings               */
    int sense[18];                      /* array that contains the mode
                                           parameters. See APTEK Drivers
                                           description for XMSENS to learn
                                           the meaning of each element
                                           of the array.                     */

    /* Set parameters used by xsmode: */

      int ibsize = 1024;                /* # of bytes in a block             */
      int ibmode = 1,                   /* buffered mode on                  */
      ipe   = 0,                        /* parity enable off                 */
      inal  = 0,                        /* auto-load enable on               */
      iebd  = 0,                        /* even byte disconnect off          */
      inbd  = 0,                        /* report busy status on             */
      ieur  = 0,                        /* use P6 cartridge                  */
      ibend = 128,                      /* motion threshold (see Aptek util. lib. p. 9)  */
      ibbeg = 128,                      /* reconnect threshold (see Aptek " " ")  */
      ibgap = 7;                        /* gap threshold, # of blocks padded after a data set  */
      int numsets;                      /* number of sets to position tape   */
      int numblocks;                    /* number of blocks to position tape */
/*-------------------------------------------------------------------------*/
            SETTGT( &id , &itype );             /* sets the id for all future
                                                   calls                     */

            XSMODE(&ibmode,&ibsize,&ipe,&inal,
                   &iebd,&inbd,&ieur,&ibend,
                   &ibbeg,&ibgap,&ierr);        /* sets the drive mode, which
                                                   controls the read/write commands*/

if (ierr > 0 && ierr != 0x40 && ierr != 0x46)
  {
    printf("error with setting the mode, ierr = %x hex", ierr);
    exit (1);
  }

XMSENS( sense, &ierr );                 /* checks to see if XSMODE set
                                           parameters correctly              */

menu:

_clearscreen ( _GCLEARSCREEN );
printf("Block size        = %d bytes\n",256*256*sense[10] +
                256*sense[11] + sense[12]);
printf("Motion threshold   = %d Kbytes\n", sense[15]);
printf("Reconnect threshold = %d Kbytes\n", sense[16]);
printf("Gap threshold  = %d blocks\n\n\n", sense[17]);
printf("starting block number = %ld\n", start_block_number);
printf("\n2-Meg set number = %d\n", set_number);

if (sense[10] && sense[11] && sense[12] && sense[15]
                && sense[16] && sense[17]  == 0)
                {
                  printf("\n\n\nError with tape drive!\n");
                  printf("Reset by turning tape drive power off then on\n");
                  abort();
                }

printf("\n\nENTER option desired:\n\n\n\n");
printf("\t reset tape..... rs \n");
printf("\t rewind tape..... rw \n");
printf("\t move to end of last data..... mvld \n");
printf("\t position tape to desired location..... pt \n");
printf("\t write data to tape..... wt \n");
printf("\n\n");
scanf("%s", answer);

if ( (compresult = strcmp(answer,"rs")) == 0)       /* is answer = rs?       */
                goto reset;
 if ( (compresult = strcmp(answer,"rw")) == 0)      /* is answer = rw?       */
                goto rewind;
if ( (compresult = strcmp(answer,"mvld")) == 0)     /* is answer = mvld?     */
                goto move_to_last_data;
if ( (compresult = strcmp(answer,"pt")) == 0)       /* is answer = pt?       */
                goto position_tape;
if ( (compresult = strcmp(answer,"wt")) == 0)       /* is answer = wt?       */
                goto initialize_tape_end;

printf("\n\n\nYour input matches no choices, try input again\n");
for (k = 0; k<11000; k++)
```

```
                {
                  for (l = 0; l<1500; l++)  /*  This is a delay routine   */
                      {         ;}
                }
              goto menu;

reset:
      {
              printf("\n\nResetting...\n");

              XRESET(&ierr);

              start_block_number = 1;
              set_number = 1;

              if (ierr > 0 && ierr != 0x40)
                        printf("error in resetting, ierr = %d\n", ierr);
              goto menu;
      }

rewind:
      {
              printf("\n\nRewinding...\n");

               XREWND(&ierr);  /*  Rewind the Tape                 */

              start_block_number = 1;
              set_number = 1;

               if (ierr > 0 && ierr != 0x40)
                        printf("\n\nerror with rewind, ierr = %d\n\n", ierr);
              goto menu;
      }

move_to_last_data:
      {
              printf("\n\nMoving to end of last data recorded on tape...\n");

              XEOD(&ierr);   /*  Move to end of last data on tape     */

              if (ierr > 0)
                        printf("\n\nerror with move to last data, ierr = %d", ierr);
              goto menu;
      }

position_tape:
      {
              _clearscreen( _GCLEARSCREEN );
              printf("\n\n INPUT:\n ");
              printf("\t # of blocks  > 0  drive moves forward  # of blocks\n");
              printf("\t            < 0  drive moves backward # of blocks\n");
              printf("\t            = 0  no movement                \n\n\n");
              printf("----Block size is 1024 bytes-----\n\n");
              printf("INPUT # of 2_Meg sets to move:\n");
              printf("[must not exceed 16, or be less then -16]\n\n");
              scanf("%d", &numsets);
              numblocks = numsets * 2040;
              start_block_number = start_block_number + numsets*2040;
              set_number = set_number + numsets;

              printf("\n\nSpacing the desired # of blocks...\n");

              XSPACE(&numblocks, &ierr);   /* moves position on tape nblks blocks */

              if (ierr > 0)
                        printf("error with space blocks, ierr = %d", ierr);
              goto menu;
      }

initialize_tape_end:
      {
              start_block_number = start_block_number + nsets*2040;
              set_number = set_number + nsets;
      }
}     /*      end of initialize_tape subroutine  */
```

Appendix C

Appendix C contains a sampling of the data processing and analysis programs. The four main data processing programs are: "doppler.f," "cross.f," "cov.f" and "Vcwslope.f." Program "doppler.f" is written for use on the San Diego Supercomputer facilities and processes data taken to estimate the Doppler winds. Program "cross.f" is also used on the SDSC and processes data taken to estimate crosswinds. Program "cov.f" computes the time-delayed covariance from the intensity vs. time series for two channels provided by "cross.f." Program "Vcwslope.f" computes an estimate of the crosswind speed from the slope of the crosscovariance curve at 0 time delay. Included in this appendix are programs "doppler.f" and "Vcwslope.f" to give examples of the data processing programs. The remaining programs can be found on a floppy disk labeled "PROCESSING PROGRAMS" and are also listed as part of a hard copy of all of the programs used throughout this thesis project titled "CO2 LIDAR PROGRAMS."

```
ic*******************************************************************************
c                       Program         doppler.f                                *
c                       BJ Rask  9-17-94                                         *
c                       *** revised for SDSC 10-4-94 ***                         *
c                                                                                *
c                       LANGUAGE: Fortran 77                                     *
c*******************************************************************************
c
c     *** Program to be used on San Diego Supercomputer ***
c
c  Program that computes:
c    1) the Intensity vs. range for 1 channel for all subsets.
c    2) the Doppler shift at best range for all subsets.

      integer N
      parameter (N=1024)
      real option
      real fftave(0:511)
      integer PRC(0:62)
      real fftwt(512)                          !fft of filter weights to filter averaged spectra
      real fta(0:511)
      real work(2*(N-1)*128+4*128)             !used as workspace by c90 FFT routine
      real table(0:100+2*(N-1))                !holds the trig factors for FFT
      real data1(0:1023,0:128)                 !huge array that contains the data for Ch1
      real Dkeep(131072)
      Complex Xkdata(0:512,0:127)              !transform of data1
      character*1 diphase(0:1023)              !local array that holds temp data for ch1
      character*14 binfile                     !binary file that contains the diphase data
      integer fnum                             !# of frequencies on each side of Fmid
      integer lot                              !# of simultaneous transforms
      integer isign                            !+1 for inverse trans, -1 for trans
      integer isys(0:0)                        !needed by FFT routine
      integer index                            !used as index for huge array, Ndata
      real SPN(1210)                           !array containing max signal + noise for
                                               !a particular set
      real SPNave(1210)                        !average of SPN sets
c-------------------------------------------------------------------------------

      COMMON /arrays/ PRC
      Common /params/ nstart,nend,nstep

c-------------------------------------------------------------------------------
c---  start program input section
c
c
      lot = 128                                !number of simultaneous transforms
      scale = 1.0/float(N)                     !scale factor used by FFT routine
      isys(0) = 0
      ldx = N
      ldy = N/2+1
      n1k = 128                                !number of 1k chunks of data in 128K set

      print*,'*****************************************************'
      print*,'*  Program       dop128k.f                *'
      print*,'*****************************************************'
      print*
      print*
      print*,'Program INPUT section:'
      print*
      print*,'INPUT the binary FILENAME of diphase data'
      read(5, '(a14)' ) binfile
      print*

      print*
      print*,'INPUT the number of subsets in data file'
      print*,'Each subset = continuous run of ch1 and ch2 data'
      read*, nsubsets

      print*
      print*,'On which subset to start?'
      read*,nsubstart

c     print*
c     print*,'INPUT the starting range, ending range, and'
c     print*,'range step size'
c     read*,nstart,nend,nstep
      nstart = 100
      nend = 1000
      nstep = 20

      OPEN(12, file = 'code63.dat', status = 'old')
      OPEN(10, file = binfile, access = 'direct', recl = 1024,
     >     form = 'unformatted')
      OPEN(14, file = 'fftwt.dat', status = 'unknown')
      OPEN(15, file = 'fDopp.dat', status = 'unknown')

c---  Read in the PRC code for 63-state PRC, and filter weights file

       read(12,*) (PRC(i), i = 0, 62)

      do i = 1, 512
            read(14,*) x, fftwt(i)
      end do
c-------------------------------------------------------------------------------
c--- start to read in the data from binary data file, binfile
c--- and PRC file.  Put specified # of 10K sets in Ndata (extended mem).
c--- Data is read from a binary file of 8-bit (1-byte) data.  The
c--- reading routine below reads 8-bit characters from binfile,
c--- 10240 points at a time.  Ndata stores all the data in char format.
c
c
      isign = 0                                !first generate the trig factors needed by FFT
```

```
      call SCFFTM(isign,N,lot,scale,data1,ldx,Xkdata,
     >                                        ldy,table,work,isys)

      do i = 0,511
            fta(i) = 0.0
      end do

      do i = nstart, nend, nstep
            SPNave(i) = 0.0
      end do

      do 10 iii = nsubstart, nsubsets+nsubstart-1

            print*
            print*,'Processing subset # ',iii
c
c--- Read in the data from binary data file:
c
            print*,'  Reading in Ch1 data...'
            do i = 1, n1k
                  indexch = i+(iii-1)*n1k
                  read(10, rec=indexch ) diphase
                  do j = 0,1023
                     index = (i-1)*1024 + j + 1
                     Dkeep(index) = ichar(diphase(j)) -128
                  end do              !end of loop j
            end do   !end of loop i

c-------------------------------------------------------------------
c--- Compute an estimate of the intensity vs. range.
c
            do 30 ii = nstart, nend, nstep

                  do i = 0,n1k-1
                     do j = 0,1023
                        index = i*1024+j + 1
                        data1(j,i) = Dkeep(index)
                     end do
                  end do

                  range = ii
                  deltab = 2.4                  !time delay of the table in microseconds
                  tdelay = deltab + (2.0*range/300.)
                  delay = tdelay * 20 !convert flight plus table delay
                  ndelay = delay

                  !=============================================
                   call rangebin(ndelay, data1, n1k)
                  !---------------------------------------
                  !--- multiplies by the delayed PRC on ch1
                  !=============================================
                  isign = -1
                  !=============================================
                   call SCFFTM(isign,N,lot,scale,data1,ldx,Xkdata,ldy,
     >                              table,work,isys)
                  !-----------------------------------------------------
                  !supplied by SDSC fft libraries.  Computes FFT on n1k
                  !dimensions.  Results are stored in Xkdata1.
                  !=============================================

                  do i = 0, 511
                    fftave(i) = 0.0       !initialization
                  end do

                  do i = 0, 127
                     do  j = 0, 511
                        fftave(j) = fftave(j) + abs(Xkdata(j,i))
                     end do
                  end do

                  !=============================================
                   call find_SPNmax(ii, fftave, SPN)
                  !=============================================
                  !--- computes max signal + noise for
                  !--- selected averaged set for Ch1
                  !=============================================

30    continue    !end of loop ii, the iteration of the range

            call find_maxrange(SPN, maxrange)
            print*,'  Maxrange = ',maxrange

            do i = nstart, nend, nstep
                  SPNave(i) = SPNave(i) + SPN(i)
            end do

            if ((iii/10.) .lt. (iii/10)) then
                  print*,'storing ivsr'
                  call store_ivsr(SPNave, iii)
                  do i = nstart, nend, nstep
                     SPNave(i) = 0.0
                  end do
            endif

c-------------------------------------------------------------------
c--- Now want to go back to original diphase data, multiply by delayed PRC
c--- to center of range bin.
c
            tdelay = deltab + (2.0*maxrange/300.)
            delay = tdelay * 20                   !convert flight plus table delay
            ndelay = delay
```

```
      do i = 0,n1k-1
                  do j = 0,1023
                     index = i*1024+j + 1
                     data1(j,i) = Dkeep(index)
                  end do
      end do

      !=============================================
      call rangebin(ndelay, data1, n1k)
      !-------------------------------------------------------------
      !--- multiplies by the delayed PRC on ch1
      !=============================================

c--- Compute the averaged spectrum at maxrange

      !=============================================================
      call SCFFTM(isign,N,lot,scale,data1,ldx,Xkdata,ldy,
     >                  table,work,isys)
      !-------------------------------------------------------------------------------
      !supplied by SDSC fft libraries. Computes FFT on n1k
      !dimensions. Results are stored in Xkdata.
      !=============================================================

      do i = 0, 511
                  fftave(i) = 0.0                  !initialization
      end do

      do i = 0, n1k-1
                  do  j = 0, 511
                     fftave(j) = fftave(j) + abs(Xkdata(j,i))
                  end do
      end do

      do i = 0, 511
                  fta(i) = fta(i) + fftave(i)
      end do

      call find_Dopp(fftave,fDopp)
      print*,'  fDopp=',fDopp
      write(15,*) iii, fDopp

      if ((iii/10.) .lt. (iii/10)) then
                  print*,'storing spec'
                  call store_spec(fta,fftwt,iii)
                  do i = 0, 511
                     fta(i) = 0.0
                  end do
      endif
      print*

10    continue  !end of loop iii, the subsets

      close(14)
      stop
      end


*********************************************************************
      subroutine          rangebin(ndelay, data1, n1k)
*********************************************************************
c
c--- Subroutine that multiplies the diphase data with the delayed PRC.
c
c--- ndelay is delay in samples to center of desired range bin

      real data1(0:1023,0:128)

      COMMON /arrays/ iPRC(0:62)

c
c--- The actual delay in samples is ndelay.  The PRC code will be
c--- circularly shifted by ndelay, then multiply diphase by
c--- circularly shifted PRC.
c

      do j = 0, 1023
            do i = 0, n1k-1
               index = i*1024 + j
               narg = MOD( (63*20-ndelay+index)/20, 63 )
               if ( iPRC( narg ) .eq.-1) then
                        data1(j,i) = -data1(j,i)
               endif
            end do
      end do

      return
      end

*********************************************************************
      subroutine          find_SPNmax(ii, fftave, SPNmax)
*********************************************************************
c
c--- Subroutine that finds the maximum Signal+noise for the given range
c--- of interest.  The User has input the Doppler Frequency
c--- stored in Fmid.  This subroutine searches 5 points below Fmid and
c--- 5 points above Fmid to find the true maximum intensity of the
c--- smoothed spectrum.
c
c
      real fftave(0:511)                  !contains the spectrum of the signal + noise
      real SPNmax(1210)                   !maximum peak at particular range.  Because
```

```
c                                       of the processing technique, the max peak
c                                       will have signal + DC offset due to the
c                                       noise which must be subtracted.
      real Fmid                         !frequency of max Signal+noise at particular
c                                        range bin
      integer fnum                      !number of frequencies to search on each
c                                        side of Fmid

      fnum = 100
      Fmid = 6.0
      imid = Fmid * 1024.0/20.0
      istart = imid - fnum
      iend = imid + fnum


      SPNmax(ii) = -1

      do 10 i = istart, iend
            if (SPNmax(ii).lt.fftave(i)) then
                  SPNmax(ii) = fftave(i)
            endif
10    continue

c
c--- SPNmax(ii) will be returned to the main program which is the maximum
c--- value of the signal + noise for a particular range.
c

      return
      end

*********************************************************************
      subroutine          find_maxrange(SPN, maxrange)
*********************************************************************
c
c--- Subroutine that finds the range at which the maximum signal +
c--- noise occurs.
c
      real SPN(1210)
      Common /params/ nstart,nend,nstep

c--- Now find the range of maximum intensity

      SPNmax = -1
      do ii = 140,nend,nstep
            if (SPNmax .lt. SPN(ii)) then
                  SPNmax = SPN(ii)
                  maxrange = ii
            endif
      end do

      print*,'  max Intensity for Ch1 found at ',maxrange,'m'

      return
      end

*********************************************************************
      subroutine          store_ivsr(SPNave, iii)
*********************************************************************
c
c--- Subroutine that writes the intensity vs. range for particular subset
c--- to output files.
c
      character*12 file
      character*3 siii                        !variable iii as a character string
      real SPNave(1210)
      integer iii
      Common /params/ nstart,nend,nstep

c--- want to convert integer iii to a string

      write(siii, '(i3.3)' ) iii+300

c--- siii is now a string of numbers: 001, 002, 003, ..., 100, 101...
c--- Want to remove the zeroes in front of the integers.

      m = 0
3     m = m + 1
      if (siii(m:m) .eq. '0') goto 3
      nnum = m

c--- nnum is the starting place of the numbered string, siii

      file = 'ir'//siii(nnum:3)//'.dat'

      OPEN(20, file = file, status = 'unknown')

c--- Store the IvsR to output files
      print*,'  writing IvsR to file ',file

      do ii = nstart,nend,nstep
            write(20,*) ii, SPNave(ii)
      end do
      close(20)

      return
      end

c*********************************************************************
      subroutine          find_Dopp(fftave, fDopp)
c*********************************************************************
c
```

```
c--- Subroutine that finds the Doppler shift for each subset.
c
      real fftave(0:511), Imax, fDopp

      Imax = -1.
      do i = 320,400
            if (Imax .lt. fftave(i-1)**2) then
                  fDopp = i*20./1024.0
                  Imax = fftave(i-1)**2
            endif
      end do

      return
      end

c**********************************************************************
      subroutine          store_spec(fta,fftwt,iii)
c**********************************************************************
c
c--- Subroutine that stores the averaged spectra to output files.
c
      real fta(0:511), fftwt(512)
      real fDopp, Imax
      character*14 file
      character*3 siii

      write(siii, '(i3.3)' ) iii+300

c--- siii is now a string of numbers: 001, 002, 003, ..., 100, 101...
c--- Want to remove the zeroes in front of the integers.

      m = 0
3     m = m + 1
      if (siii(m:m) .eq. '0') goto 3
      nnum = m

c--- nnum is the starting place of the numbered string, siii

      file = 's'//siii(nnum:3)//'.dat'

      OPEN(20, file = file, status = 'unknown')
      do i = 1, 512
            write(20,*) i*20./1024.0 , fftwt(i)*fta(i-1)**2
      end do
      close(20)

      return
      end
```

```
c**********************************************************************************
c                        Program        Vcwslope.f                                 *
c                        12-5-93        BJ Rask                                    *
c                                                                                  *
c Program that calculates the average crosswind speed for the return               *
c Intensity backscattered from aerosols. The solution was derived                  *
c without using the Joint Gaussian assumption for the electric fields              *
c at the receiver. The crosswind detection scheme was derived using                *
c the time-averaged form of the CrossCovariance of the Intensities.                *
c The slope of the CrossCovariance is proportional to the average                  *
c wind speed. The slope of the CC is also dependent upon the                       *
c the turbulence, Cn2. However, since the normalized Covariance is                 *
c also proportional to Cn2, we can divide the slope of the CC by the               *
c normalized Cov. to yield a result only dependent upon the average                *
c cross wind speed, and the ratio of two integrals. The integrands of              *
c these integrals only depend upon system parameters and the propagation           *
c distance, so there are no unknowns. The average Crosswind speed is               *
c obtained by dividing the ratio of the slope at 0 time delay by the               *
c ratio of the two integrals.                                                      *
c **********************************************************************************
c Definition of variables:
c =========================
c
c           Cn2 = stringth of turbulence
c           Zc = distance to center of range bin (meters)
c           P = detector spacing
c           k = wave number = 2 * pi / lambda
c           Ich1 = intensities vs. time for Channel 1
c           Ich2 = intensities vs. time for Channel 2
c           Ncov = normalized covariance of intensities
c           SNcov = slope of norm. cov. at 0 time delay

c
c Program inputs:
c ===============
c
c Programmer inputs the processed time delayed crosscovariance curve.
c
c Program outputs:
c ================
c
c Program outputs 1 crosswind speed estimate per crosscovariance curve.
c

      real k, Zc, P, SNcov, Ncov
      real INT1, INT2
      real Cov(2100)
      real lambda
      real pi
      Double Precision XX          !Zc/(k*P*P)
      character*14 file1, file2
```

```
      parameter (lambda = 10.6e-06)

      Common /parameters/ k, P, Zc, pi, XX

      pi = 3.1415926
      k = 2 * pi / lambda
      P = 1.05*2.54/100.      !detector spacing in meters

c-------------------------------------------------------------------------
c  start program input section
c-------------------------------------------------------------------------
c
      print*,'************************************************'
      print*,'*          Program  Vcwslope.f             *'
      print*,'************************************************'
      print*,'*  program that computes the cross-wind speed *'
      print*,'************************************************'
      print*
      print*
      print*,'---------- Program input section ------------'
      print*
      print*,'INPUT the distance to center of range bin (meters)'
      read*, Zc
      print*
      XX = Zc/(k*P*P)         !needed for ratio of two integrals


      print*,'INPUT the Covariance curve'
      read*, file1

      OPEN(10, file = file1, status = 'unknown')
c
c *** read in the covariance plot
c
      icount = 0
5     READ(10,*,end = 99) x, Cov(icount+1)
       if (x .eq. 0.0) then
         Ncov = Cov(icount+1)
         nkeep = icount+1
       endif
       icount = icount + 1
       goto 5
99     continue
      ntotal = icount
      Print*,'Ncov=',Ncov
c-------------------------------------------------------------------------
c  end of program input section.  Start to compute the covariance, the
c  slope at 0 time lag, and the Crosswind speed.
c-------------------------------------------------------------------------
c
      print*,'Computing the Crosswind speed...'
      print*

c
c *** first compute the parameters and constants used by INT1
c *** and INT2.
c
      call initialize
c
c *** compute the slope at 0 time lag
c
      call Slope(Cov, nkeep, SNcov, ntotal)
c
c *** compute the integrals
c
      call Integrals(INT1, INT2)
c
c *** Back from all subroutines.  Now calculate the wind speed.
c *** Wind speed will be calculated by dividing the Slope at
c *** 0 time delay by the Normalized Covariance and then multiplying
c *** by the ratio of INT2 to INT1.
c
      print*,'INT1,INT2,I1/I2,=',INT1,INT2,INT1/INT2
      print*,'SNcov/Ncov=',SNcov/Ncov
      print*,'P=',P
      Vpave = P * SNcov / Ncov * INT2 / INT1

      print*,'Vpave = ',Vpave

      stop
      end

c*************************************************************************
      subroutine Slope(Cov, nkeep, SNcov, ntotal)
c*************************************************************************
c
c  subroutine that calculates the slope of the normalized covariance
c  of intensities at 0 time delay.  The slope is calculated using the
c  NAG routine E02ACF.
c
      Double Precision x(10), y(10), A(6), REF
      real Ncov, Cov(2100)
      real SNcov

c  *** Use the Covariance at 0 time delay to compute the slope
c  *** 0 time lag to two lags beyond 0 time lag.  Each lag is
c  *** separated by 512 microseconds.
c
c  *** Define the x and y values for the slope, and calculate
c  *** the slope using NAG routine E02ACF (polynomial fit).
c
      y(1) = Cov(nkeep-1)
```

```
      y(2) = Cov(nkeep)
      y(3) = Cov(nkeep + 1)
      x(1) = 0 - 1 * 512e-06
      x(2) = 0
      x(3) = 0 + 1 * 512e-06

      OPEN(11, file = 'points.dat', status = 'unknown')
      do i = 1, 3
        write(11,*) x(i), y(i)
      end do
      close(11)

      N = 3         !three points used to fit line
      norder = 1    !fit a linear curve
      M1 = norder + 1

      call E02ACF(x, y, N, A, M1, REF)    !slope is A(2)

      SNcov = A(2)

      print*,'SNcov=',SNcov

      OPEN(11,file = 'lin.dat', status = 'unknown')
      do i = 1, 3
        write(11,*) x(i), x(i)*A(2)+A(1)
      end do
      close(11)


      return
      end

c*********************************************************************
      subroutine Integrals(INT1, INT2)
c*********************************************************************
c
c  subroutine that calculates the integrals in the time-averaged
c  time-lagged normalized covariance of Intensities.  The Integrands
c  are a function of Confluent hyper-geometric functions.
c  The integration variable is w, the normalized path length from
c  laser/receiver to scattering location.  The integrals are
c  from w = 0 to w = 1.  (See theory development for more details
c  and exact form of the integrals).  The algorithm used to calculate
c  the integrals is found in the appendix of "Two-point joint-density
c  function for a laser-generated speckle field after propagation
c  through the turbulent atmosphere,"  V.S. Rao Gudimetla, J. F. Homles,
c  and R. A. Elliot.  The NAG routine D01BDF is used to numerically
c  approximate the integrals.
c
      real INT1, INT2
      Double Precision A, B, epsabs, epsrel, abserr
      Double Precision Result1, Result2
      External Integral1, Integral2


      A = 0.0d0         !lower limit of integration
      B = 1.0d0         !upper limit of integration
      epsabs = 0.0d0    !used by D01BDF
      epsrel = 1.0e-05  !used by D01BDF

      call D01BDF(Integral1, A, B, epsabs, epsrel, Result1, abserr)
      call D01BDF(Integral2, A, B, epsabs, epsrel, Result2, abserr)

      INT1 = Result1
      INT2 = Result2

      return
      end

c*********************************************************************
      Double Precision function Integral1(w)
c*********************************************************************
c
c  Subfunction that calculates INT1, and puts the result in Result1.
c  Used by NAG routine D01BDF.
c
c                XX = Zc/(k*P*)
c
      real k, P, Zc, pi
      Double Precision X, XX
      Double Precision Cfreg1(0:50),Cfas11(0:50),Cfas21(0:50)
      Double Precision gam1_3, gam1_6, gam11_6
      Double Precision gama1, gamb1, gambma1
      Double Precision a, b, SUM, w
      Double Precision fact0, fact1, fact2, fact3, fact4, fact5, fact6

      Common /coefficients1/ Cfreg1, Cfas11, Cfas21
      Common /parameters/ k, P, Zc, pi, XX
      Common /gamscommon/ gam1_3, gam1_6, gam11_6
      Common /constants1/ gama1, gamb1, gambma1
c---------------------------------------------------------------------

      a = sqrt(w*(1-w)*XX/2.)
      b = w
      X = b*b/(8.*a*a)

      fact0 = w
      fact1 = b**(-1./3.)/(4.*pi**(1./2.))*gam1_3**2*cos(pi/6)/gam11_6
      fact2 = -gam1_6/(2**(1./6.)*a**(1./3.)*16.)

c
c *** If X > 4pi, must use the asymptotic approx. for the summations
```

```
c
      IF (X .gt. (4.*pi) ) then

c
c *** IF HERE, THEN COMPUTING THE ASYMPTOTIC SUMMATION
c *** ================================================
c *** There are 4 summations to compute (two odd, and two even)
c
c *** compute the first odd summation (asymptotic):
c
      ioe = 1       !odd summation
      ia = 1        !asymptotic summation

      call summation(Cfas11, ioe, ia, X, SUM)

      alpha = (pi/12.)*(1.-6*1./6.)
      fact3 = 2.*gamb1/gambma1*X**(-1./6.)*sin(alpha)*SUM
c
c *** compute the second odd summation (asymptotic):
c
      ioe = 1       !odd summation
      ia = 1        !asymptotic summation

      call summation(Cfas21, ioe, ia, X, SUM)

      beta = -X + pi/12.*(1.-6*(1./6.-2.))
      fact4 = 2.*gamb1/gama1*X**(1./6.-2.)*sin(beta)*SUM
c
c *** compute the first even term in the summation (asymptotic):
c
      ioe = 0       !even summation
      ia = 1        !asymptotic summation

      call summation(Cfas11, ioe, ia, X, SUM)

      fact5 = 2.*gamb1/gambma1*X**(-1./6.)*cos(alpha)*SUM
c
c *** compute the second even term in the summation (asymptotic):
c
      ioe = 0       !even summation
      ia = 1        !asymptotic summation

      call summation(Cfas21, ioe, ia, X, SUM)

      fact6 = 2.*gamb1/gama1*X**(1./6.-2.)*cos(alpha)*SUM
c
c *** compute the INT1 (asymptotic):
c
      Integral1 = fact0*(fact1 + fact2*(fact3+fact4+fact5+fact6))

      ELSE

c
c *** IF HERE, THEN COMPUTING THE REGULAR SUMMATION (NOT ASYMPTOTIC FORMS)
c *** ====================================================================
c *** There are 2 summations to compute, 1 odd and 1 even
c
c *** compute the odd term in the summation.
c
      ioe = 1       !odd summation
      ia = 0        !regular summation

      call summation(Cfreg1, ioe, ia, X, SUM)

      fact3 = 2.*sin(pi/12.)*SUM
c
c *** compute the even term in the summation.
c
      ioe = 0       !even summation
      ia = 0        !regular summation

      call summation(Cfreg1, ioe, ia, X, SUM)

      fact4 = 2.*cos(pi/12.)*SUM
c
c *** compute INT1 (regular):
c
      Integral1 = fact0*(fact1 + fact2*(fact3 + fact4))


      ENDIF

      return
      end

c*************************************************************************
      Double Precision function Integral2(w)
c*************************************************************************
c
c  Subfunction that calculates INT2, and puts the result in Result2.
c  Used by NAG routine D01BDF.
c
      real k, P, Zc, pi
      Double Precision X, XX
      Double Precision Cfreg2(0:50),Cfas12(0:50),Cfas22(0:50)
      Double Precision gam1_3, gam1_6, gam11_6
      Double Precision gama2, gamb2, gambma2
      Double Precision a, b, SUM, w
      Double Precision fact1, fact2, fact3, fact4, fact5, fact6
```

```
      Common /coefficients2/ Cfreg2, Cfas12, Cfas22

      Common /parameters/ k, P, Zc, pi, XX
      Common /gamscommon/ gam1_3, gam1_6, gam11_6
      Common /constants2/ gama2, gamb2, gambma2
c-------------------------------------------------------------------------

      a = sqrt(w*(1.-w)*XX/2.)
      b = w
      X = b*b/(8.*a*a)

      fact1 = -.15*b**(5./3.)/pi**(1./2.) * gam1_3**2*cos(pi/6.)/
     >        gam11_6
      fact2 = .3*gam1_6*a**(5./3.)/2.**(1./6.)
c
c *** If X > 4pi, must use the asymptotic approx. for the summations
c


      IF (X .gt. (4.*pi) ) then

c
c *** IF HERE, THEN COMPUTING THE ASYMPTOTIC SUMMATION
c *** ================================================
c *** There are 4 summations to compute (two odd, and two even)
c
c *** compute the first odd summation (asymptotic):
c
      ioe = 1        !odd summation
      ia = 1         !asymptotic summation

      call summation(Cfas12, ioe, ia, X, SUM)

      alpha = (pi/12.)*(5+6*(-5./6.))
      fact3 = 2.*gamb2/gambma2*X**(5./6.)*sin(alpha)*SUM
c
c *** compute the second odd summation (asymptotic):
c
      ioe = 1        !odd summation
      ia = 1         !asymptotic summation

      call summation(Cfas22, ioe, ia, X, SUM)

      beta = X + (pi/12.)*(5+6.*(-5./6.-1))
      fact4 = 2.*gamb2/gama2*X**(-5./6.-1.)*sin(beta)*SUM
c
c *** compute the first even term in the summation (asymptotic):
c
      ioe = 0        !even summation
      ia = 1         !asymptotic summation

      call summation(Cfas12, ioe, ia, X, SUM)

      fact5 = 2.*gamb2/gambma2*X**(5./6.)*cos(alpha)*SUM
c
c *** compute the second even term in the summation (asymptotic):
c
      ioe = 0        !even summation
      ia = 1         !asymptotic summation

      call summation(Cfas22, ioe, ia, X, SUM)

      fact6 = 2.*gamb2/gama2*X**(-5./6.-1.)*cos(alpha)*SUM

c
c *** compute INT2 (asymptotic):
c
       Integral2= fact1 + fact2*(fact3+fact4+fact5+fact6)


      ELSE


c
c *** IF HERE, THEN COMPUTING THE REGULAR SUMMATION (NOT ASYMPTOTIC FORMS)
c *** ====================================================================
c *** There are 2 summations to compute, 1 odd and 1 even
c
c *** compute the odd term in the summation.
c
      ioe = 1        !odd summation
      ia = 0         !regular summation

      call summation(Cfreg2, ioe, ia, X, SUM)

      fact3 = 2.*cos(pi/12.)*SUM
c
c *** compute the even term in the summation.
c
      ioe = 0        !even summation
      ia = 0         !regular summation

      call summation(Cfreg2, ioe, ia, X, SUM)

      fact4 = 2.*sin(pi/12.)*SUM
c
c *** compute INT2 (asymptotic):
c
      Integral2 = fact1 + fact2*(fact3 + fact4)
```

```
      ENDIF

      return
      end

c**********************************************************************
      subroutine summation(Coeff, ioe, ia, X, SUM)
c**********************************************************************
c
c  subroutine that calculates the summations.
c
      Double Precision Coeff(0:50)
      Double Precision sumeven, sumodd, SUM
      Double Precision X
      Integer ioe          !odd or even summation
      Integer ia           !asymptotic or regular summation
c----------------------------------------------------------------------


      IF (ioe .eq. 0) then
c
c *** even summation
c
         sumeven = 1.0

         if (ia .eq. 0) then
          !use regular summation
          do i = 2, 24, 2
            sumeven = sumeven + Coeff(i)*X**(i)
          end do
         else
          !use asymptotic summation
          do i = 2, 24, 2
            sumeven = sumeven + Coeff(i)*X**(-i)
          end do
         endif

       SUM = sumeven

      ELSE

c
c *** odd summation
c

       if (ia .eq. 0) then
         !use regular summation
         sumodd = Coeff(1)*X**(1)
       else
         !use asymptotic summation
         sumodd = Coeff(1)*X**(-1)
       endif

         if (ia .eq. 0) then
          !use regular summation
          do i = 3, 25, 2
            sumodd = sumodd + Coeff(i)*X**(i)
          end do
         else
          !use asymptotic summation
          do i = 3, 25, 2
            sumodd = sumodd + Coeff(i)*X**(-i)
          end do
         endif

         SUM = sumodd

      ENDIF    !odd summation or even summation IF

      return
      end

c**********************************************************************
      subroutine Nfactorial(Nfact)
c**********************************************************************
c
c  Subroutine that computes the coefficients of the factorial function.
c
      Double Precision Nfact(0:50)

      N = 50
      Nfact(0) = 1

      do 10 i = 1, N
        Nfact(i) = Nfact(i-1)*i
10    continue

      return
      end

c**********************************************************************
      subroutine ABcoeff(atemp, btemp, Afact, Bfact)
c**********************************************************************
c
c  subroutine that computes the Pochhammer constants of the confluent
c  hypergeometric series summation.
c
      Double Precision atemp, btemp
      Double Precision Afact(0:50), Bfact(0:50)

      N = 50
      Afact(0) = 1.0
```

```
      Bfact(0) = 1.0
      Afact(1) = atemp
      Bfact(1) = btemp

      do 20 i = 2, N
        Afact(i) = Afact(i-1)*(atemp + i-1)
        Bfact(i) = Bfact(i-1)*(btemp + i-1)
20    continue

      return
      end

c*************************************************************************
      subroutine initialize
c*************************************************************************
c
c  subroutine that calculates the constants needed for INT1, INT2.
c
c
      Double Precision atemp, btemp
      Double Precision Afact(0:50), Bfact(0:50)
      Double Precision Cfreg1(0:50),Cfas11(0:50),Cfas21(0:50)
      Double Precision Cfreg2(0:50),Cfas12(0:50),Cfas22(0:50)
      Double Precision Nfact(0:50)
      Double Precision arg1_3, arg1_6, arg11_6
      Double Precision gam1_3, gam1_6, gam11_6
      Double Precision arga, argb, argbma
      Double Precision gama1, gamb1, gambma1
      Double Precision gama2, gamb2, gambma2
      Double Precision S14AAF
      Integer ifail
      External S14AAF          !NAG routine to get the gamma fcn.

      Common /coefficients1/ Cfreg1, Cfas11, Cfas21
      Common /coefficients2/ Cfreg2, Cfas12, Cfas22
      Common /gamscommon/ gam1_3, gam1_6, gam11_6
      Common /constants1/ gama1, gamb1, gambma1
      Common /constants2/ gama2, gamb2, gambma2

c
c *** define the arguments of the gamma functions, and compute them
c
        arg1_3 = 1./3.
        arg11_6 = 11./6.
        arg1_6 = 1./6.
        gam1_3 = S14AAF(arg1_3, ifail)
        gam11_6 = S14AAF(arg11_6, ifail)
        gam1_6 = S14AAF(arg1_6, ifail)
c
c *** define and compute the gamma functions used by each INT1 and INT2
c
c  INT2:
        arga = -5./6.
        argb = 1.0
        argbma = argb - arga
        gama2 = S14AAF(arga, ifail)
        gamb2 = S14AAF(argb, ifail)
        gambma2 = S14AAF(argbma, ifail)
c
c  INT1:
        arga = 1./6.
        argb = 2.0
        argbma = argb - arga
        gama1 = S14AAF(arga, ifail)
        gamb1 = S14AAF(argb, ifail)
        gambma1 = S14AAF(argbma, ifail)

c
c *** compute the coefficients of the factorial function
c
        call Nfactorial(Nfact)
c
c *** Compute the coefficients of the summations which contained
c *** the hypergeometric function.  These summations are real,
c *** and oscillate in sign, but converge quickly.  First compute
c *** the coeeficients for the regular approximation to the summations,
c *** then compute the coeffients for the asymptotic approx. to the
c *** summations.  We need to calculate the Coefficients for both
c *** INT1 and INT2.

c
c *** the regular approximation for the Coefficients:
c *** INT1:
c
        a1 = 1./6.
        b1 = 2.0
        atemp = a1
        btemp = b1
        call ABcoeff(atemp, btemp, Afact, Bfact)

         Cfreg1(0) = 1.0
        do i = 1, 29, 2
         Cfreg1(i) = (-1)**(i/2)*Afact(i)/(Nfact(i)*Bfact(i))
        end do

        do i = 2, 30, 2
         Cfreg1(i) = (-1)**(i/2)*Afact(i)/(Nfact(i)*Bfact(i))
        end do

c
c *** the regular approximation for the Coefficients:
c *** INT2:
```

```
c
      a1 = -5./6.
      b1 = 1.0
      atemp = a1
      btemp = b1
      call ABcoeff(atemp, btemp, Afact, Bfact)

       Cfreg2(0) = 1.0
      do i = 1, 29, 2
       Cfreg2(i) = (-1)**((i+1)/2)*Afact(i)/Nfact(i)**2
      end do

      do i = 2, 30, 2
       Cfreg2(i) = (-1)**(i/2)*Afact(i)/Nfact(i)**2
      end do

c
c *** the asymptotic approximation for the Coefficients:
c *** asymptotic set #1, INT1
c
      a1 = 1./6.
      b1 = 2.
      atemp = a1
      btemp = 1.0+a1-b1
      call ABcoeff(atemp, btemp, Afact, Bfact)

       Cfas11(0) = 1.0
      do i = 2, 30, 2
       Cfas11(i) = (-1)**(i/2)*Afact(i)*Bfact(i)/Nfact(i)
      end do

      do i = 1, 29, 2
       Cfas11(i) = (-1)**(i/2)*Afact(i)*Bfact(i)/Nfact(i)
      end do

c
c *** the asymptotic approximation for the Coefficients:
c *** asymptotic set #1, INT2
c
      a1 = -5./6.
      b1 = 1.0
      atemp = a1
      btemp = 1.0+a1-b1
      call ABcoeff(atemp, btemp, Afact, Bfact)

       Cfas12(0) = 1.0
      do i = 2, 30, 2
       Cfas12(i) = (-1)**(i/2)*Afact(i)*Bfact(i)/Nfact(i)
      end do

      do i = 1, 29, 2
       Cfas12(i) = (-1)**((i+1)/2)*Afact(i)*Bfact(i)/Nfact(i)
      end do

c
c *** the asymptotic approximation for the Coefficients:
c *** asymptotic set #2, INT1
c
      a1 = 1./6.
      b1 = 2.
      atemp = b1-a1
      btemp = 1-a1
      call ABcoeff(atemp, btemp, Afact, Bfact)
       Cfas21(0) = 1.0
      do i = 2, 30, 2
       Cfas21(i) = (-1)**(i/2)*Afact(i)*Bfact(i)/Nfact(i)
      end do

      do i = 1, 29,2
       Cfas21(i) = (-1)**((i+1)/2)*Afact(i)*Bfact(i)/Nfact(i)
      end do
c
c *** the asymptotic approximation for the Coefficients:
c *** asymptotic set #2, INT2
c
      a1 = -5./6.
      b1 = 1.
      atemp = b1-a1
      btemp = 1-a1
      call ABcoeff(atemp, btemp, Afact, Bfact)

       Cfas22(0) = 1.0
      do i = 2, 30, 2
       Cfas22(i) = (-1)**(i/2)*Afact(i)*Bfact(i)/Nfact(i)
      end do

      do i = 1, 29,2
       Cfas22(i) = (-1)**(i/2)*Afact(i)*Bfact(i)/Nfact(i)
      end do
c
c *** done with initialization
c

    return
    end
```

Appendix D

Appendix D contains some of the analysis programs. The main analysis programs are "scatter3D.f," "CI12t.f," "PWF_apave.f," and "INT_apave.f." Program "scatter3D.f" calculates the 3-D scattering profile of the backscattered electric field off of 1 aerosols particle. The data generated by "scatter3D.f" is used by Mathematica to plot the profiles. Program "CI12t.f" generates the time-delayed crosscovariance of intensities between two points in the receiver aperture. Program "PWF_apave.f" evaluates the effect of aperture averaging on the path weight function, while program "INT_apave.f" examines the aperture averaging effect on the heterodyne signal power. Included in this appendix are programs "scatter3D.f" and "INT_apave.f" to give examples of analysis programs. The remaining programs can be found on a floppy disk labeled "ANALYSIS PROGRAMS" and are also listed as part of a hard copy of all of the programs used throughout this thesis project titled "CO2 LIDAR PROGRAMS."

```
c*******************************************************************************
c                          program          scatter3D.f  version 1              *
c                          10-21-92     revised 3-20-94                        *
c                          BJ Rask                                             *
c                                                                              *
c  Program that calculates the scattered intensities of an                     *
c  incident wave from a spherical aerosol particle.  Solution                  *
c  of problem is outlined in Born and Wolf.  Solution                          *
c  contains many special functions, and their derivatives,                     *
c  which will be calculated using the recurrence relations                     *
c  from Arfken.  Program outputs a three dimensional data                      *
c  set to be used by mathematica.                                              *
c*******************************************************************************
      Double Precision lambda,q,theta,rindex,a,arg,k0
      Double Precision dcos,dsin,pi
      Double Precision P(0:20),P1(0:20),DP1(0:20)
      Double Precision PSI1(0:20),DPSI1(0:20)
      Double Precision PSI2(0:20),DPSI2(0:20)
      Double Precision PSI3(0:20),DPSI3(0:20)
      Double Precision j1(0:20),Dj1(0:20)
      Double Precision j2(0:20),Dj2(0:20)
      Double Precision j3(0:20),Dj3(0:20)
      Double Complex ETA1(0:20),h1(0:20),Dh1(0:20),DETA1(0:20)
      Double Complex ETA2(0:20),h2(0:20),Dh2(0:20),DETA2(0:20)
      Double Complex ETA3(0:20),h3(0:20),Dh3(0:20),DETA3(0:20)
     Double Complex epe(180), epa(180), etot(180)
     Double Complex Be(20),Bm(20),Nume,Numm,Denome,Denomm
     Double Complex Iperp,Ipar,Itotal
     Double Complex Nume1,Nume2,Nume3,Nume4
     Double Complex Numm1,Numm2,Numm3,Numm4
     Double Complex Denome1,Denome2,Denome3,Denome4
     Double Complex Denomm1,Denomm2,Denomm3,Denomm4
     Double Complex I
     Complex et,ep,epp,b
     real Itot
     data lambda/ 10.6 /
     pi = 4.0d0*datan(1.0d0)
       I = (0.0,1.0)

       OPEN (11,file = 'Iperp.dat',status = 'unknown')
c      OPEN (12,file = 'Ipar.dat',status = 'unknown')
       OPEN (14,file = 'I.dat',status = 'unknown')
       OPEN (15,file = 'Itot.dat',status = 'unknown')
       OPEN (16,file = 'axes.dat',status = 'unknown')
c
c-------------------------------------------------------------
c  start input section
c-------------------------------------------------------------

       print*,'INPUT the refractive index of the aerosol,'
       print*,'     and the radius of the particle (um):'
       print*
       print*,'          ^  '
       print*,'          n, a'
       read*,rindex,a
            q = 2*pi*a/lambda
            k0 = 2*pi/lambda
c
       print*
      print*,'INPUT how many terms of the expansion you want'
      read*,N

       do 6 k = 1,179,1
            epa(k) = 0.0
            epe(k) = 0.0
6      continue

c-------------------------------------------------------------
c  First compute the Coefficients of the expansion used to
c  calculate the intensities.
c-------------------------------------------------------------

c  The subroutine, "subinitial," generates the first two terms
c  of each special function.  There are two different arguments
c  in the solution, and since the recursion relationships depend
c  on past terms, two separate sets of special functions are
c  needed.
c
c  start for l = 1
c
       arg = q
       call subinitialB(ETA1,DETA1,h1,Dh1,j1,Dj1,PSI1,
     >                          DPSI1,arg,theta)

       arg = q*rindex
       call subinitialB(ETA2,DETA2,h2,Dh2,j2,Dj2,PSI2,
     >                    DPSI2,arg,theta)

       Nume = rindex*DPSI1(1)*PSI2(1) - PSI1(1)*DPSI2(1)
     Denome = rindex*DETA1(1)*PSI2(1) - ETA1(1)*DPSI2(1)
     Numm = rindex*PSI1(1)*DPSI2(1) - DPSI1(1)*PSI2(1)
     Denomm = rindex*ETA1(1)*DPSI2(1) - DETA1(1)*PSI2(1)
     Be(1) = I**(2) * (3./2.) * Nume/Denome
     Bm(1) = I**(2) * (3./2.) * Numm/Denomm
c
c  Now for the rest of the Coefficients for l = 2 to N
c
       do 5 l = 2, N

c              first calculate the special functions with argument q =>

               arg = q
```

```
            call subPSI(l,arg,j1,PSI1)
            call subDPSI(l,arg,j1,Dj1,DPSI1)
            call subETA(l,arg,h1,ETA1)
            call subDETA(l,arg,h1,Dh1,DETA1)

            Nume1 = DPSI1(l)
            Nume3 = PSI1(l)
            Numm1 = PSI1(l)
            Numm3 = DPSI1(l)
            Denome1 = DETA1(l)
            Denome3 = ETA1(l)
            Denomm1 = ETA1(l)
            Denomm3 = DETA1(l)

c       Now for the special functions with argument q*rindex =>

            arg = q*rindex

            call subPSI(l,arg,j2,PSI2)
            call subDPSI(l,arg,j2,Dj2,DPSI2)
            call subETA(l,arg,h2,ETA2)
            call subDETA(l,arg,h2,Dh2,DETA2)

            Nume2 = PSI2(l)
            Nume4 = DPSI2(l)
            Numm2 = DPSI2(l)
            Numm4 = PSI2(l)
            Denome2 = PSI2(l)
            Denome4 = DPSI2(l)
            Denomm2 = DPSI2(l)
            Denomm4 = PSI2(l)

            Be(l) = (I)**(l+1) * (2*l+1)/(l*(l+1)) *
     >      ( rindex*Nume1*Nume2 - Nume3*Nume4 ) /
     >      ( rindex*Denome1*Denome2 - Denome3*Denome4 )

            Bm(l) = (I)**(l+1) * (2*l+1)/(l*(l+1)) *
     >      ( rindex*Numm1*Numm2 - Numm3*Numm4 ) /
     >      ( rindex*Denomm1*Denomm2 - Denomm3*Denomm4 )


5       continue

c-----------------------------------------------------------------
c  Start to compute the intensities of the scattered field
c  parallel and perpendicular to polarization of incoming E field.
c  The scattered intensities will be calculated as a function
c  of theta, 0 to 360 degrees.  Because of symmetry, theta
c  need only be varied from 0 to 180 degrees.
c-----------------------------------------------------------------
c
            DO 20 k = 1,179,1

                theta = (k*pi/180.)

                call subinitialP(theta,P,P1,DP1)

                DO 10  l = 1,N

                        call subP(l,theta,P)
                        call subP1(l,theta,P,P1)
                        call subDP1(l,theta,P,P1,DP1)

c
c  Now for the intensities
c

                        epe(k) = epe(k) + (-I)**l*( Be(l)/dsin(theta)*P1(l) -
     >                          Bm(l)*dsin(theta)*DP1(l) )

                        epa(k) = epa(k) + (-I)**l*( Be(l)*dsin(theta)*DP1(l) -
     >                          Bm(l)/dsin(theta)*P1(l) )

10              continue

20      continue

c
c-----------------------------------------------------------------
c  The intensities are a function of two angles, theta and phi.
c  The output file will be in cartesion coordinates, converted
c  from polar coordinates with a unit distance from the scatterer.
c  To convert from polar to cartesion, use:
c
c       x = R sin(theta) cos(phi)
c       y = R sin(theta) sin(phi)
c       z = R cos(theta)
c-----------------------------------------------------------------

            print*,'INPUT scale factor'
            read*,sc

            do 35 k = 1,171,10
c           do 35 k = 1,91,10
                ep = epa(k)
                epp = epe(k)

                do 30 j = 0,180,10
c                       do 30 j = 0,90,10
                            phi = j*pi/180.0
```

```
                        Itot = cos(phi)**2*ep*conjg(ep) + sin(phi)**2*epp*conjg(epp)
                        theta = k*pi/180.0
                        ztot = sin(theta)*cos(phi)*real(Itot)
                        ytot = sin(theta)*sin(phi)*real(Itot)
                        xtot = cos(theta)*real(Itot)
                        write(14,*) real(Itot)
                        write(15,*) xtot*sc,ytot*sc,ztot*sc
30      continue

35    continue

      do 45 k = 179,179,10
        ep = epa(k)
        epp = epe(k)

          do 40 j = 0,180,10
                        phi = j*pi/180.0
                        Itot = cos(phi)**2*ep*conjg(ep) + sin(phi)**2*epp*conjg(epp)
                        theta = k*pi/180.0
                        ztot = sin(theta)*cos(phi)*real(Itot)
                        ytot = sin(theta)*sin(phi)*real(Itot)
                        xtot = cos(theta)*real(Itot)
                        write(14,*) real(Itot)
                        write(15,*) xtot*sc,ytot*sc,ztot*sc
40          continue

45    continue

      stop
      end

c**********************************************************************
      subroutine          subinitialB(ETA,DETA,h,Dh,j,Dj,PSI,
     >                                DPSI,arg,theta)
c**********************************************************************
c
c  Subroutine that calculates the first couple of terms of each
c  function needed to compute the recursion relationships
c

      Double Precision dcos,dsin
      Double Precision PSI(0:100),DPSI(0:100)
     Double Precision j(0:100),Dj(0:100),x,arg,theta
      Double Complex h(0:100),Dh(0:100),ETA(0:100),DETA(0:100)
      Complex I
      I = (0.0,1.0)

     x = arg
       j(0) = dsin(x)/x
     j(1) = dsin(x)/(x*x) - dcos(x)/x
      PSI(0) = x*j(0)
      PSI(1) = x*j(1)
      Dj(0) = ( x*dcos(x) - dsin(x) )/(x*x)
      Dj(1) = 1.0/x * (-2*j(1) + x*j(0) )

      DPSI(0) = j(0) + x*Dj(0)
      DPSI(1) = j(1) + x*DJ(1)

        h(0) = I/x * (dcos(x) - I*dsin(x) )
c       h(0) = I/x*cexp(-I*real(x))
        h(1) = (dcos(x) - I*dsin(x) ) * (-1.0d0/x+I/(x*x))
c       h(1) = cexp(-I*real(x)) * (-1.0/x + I/(x*x) )

      ETA(0) = x*h(0)
      ETA(1) = x*h(1)

      Dh(0) = ( x*(dcos(x)-I*dsin(x)) - I*
     >          (dcos(x)-I*dsin(x)) )/(x*x)
c     Dh(0) = ( x*cexp(-I*real(x)) - I*cexp(-I*real(x)) )/(x*x)
        Dh(1) = ( -2*h(1) + x*h(0) )/x

        DETA(0) = h(0) + x*Dh(0)
        DETA(1) = h(1) + x*Dh(1)

        return
        end

c**********************************************************************
        subroutine          subinitialP(theta,P,P1,DP1)
c**********************************************************************
c
c  Subroutine that calculates the first couple of terms needed for
c  the recursion relationships
c
        Double Precision P(0:100),P1(0:100),DP1(0:100)
        Double Precision theta, dcos, dsin

      P(0) = 1.0
      P(1) = dcos(theta)
      P1(0) = 0.0
      P1(1) = dsin(theta)
      DP1(1) = dcos(theta)/dsin(theta)**2 * ( P1(1) - 2*P(1)*
     >                        dsin(theta)/dcos(theta) )

      return
      end

c**********************************************************************
        subroutine          subP(l,theta,P)
c**********************************************************************
c
c  Subroutine that calculates the Legendre ploynomial, Pl(cos(theta))
```

```
c
      Double Precision P(0:100),theta
      Double Precision dcos

      P(l) = ( (2*l-1)*dcos(theta)*P(l-1) - (l-1)*P(l-2) )/l

      return
      end

c*********************************************************************
        subroutine          subP1(l,theta,P,P1)
c*********************************************************************
c
c  Subroutine that calculates the associated Legendre Polynomial,
c         Pl'(cos(theta))
c
      Double Precision P(0:100),P1(0:100),theta
      Double Precision dcos,dsin

      P1(l) = l/dsin(theta) * ( P(l-1) - dcos(theta)*P(l) )

      return
      end

c*********************************************************************
        subroutine          subDP1(l, theta, P, P1, DP1)
c*********************************************************************
c
c  Subroutine that calculates the derivative of the Associated
c  Legendre Polynomial,
c
c        (1)'
c       Pl   (cos(theta))
c
        Double Precision P(0:100),P1(0:100),DP1(0:100),theta
        Double Precision dcos,dsin

      DP1(l) = (dcos(theta)/dsin(theta)**2)*( P1(l)-l*(l+1)*P(l)*
     >                        dsin(theta)/dcos(theta) )

        return
        end

c*********************************************************************
        subroutine          subPSI(l,arg,j,PSI)
c*********************************************************************
c
c  Subroutine that calculates the Spherical Bessel function, jl(x)
c
        Double Precision j(0:100),PSI(0:100),arg,x

        x = arg

      j(l) = (2*l-1)/x*j(l-1) - j(l-2)
      PSI(l) = x*j(l)

      return
      end

c*********************************************************************
        subroutine subDPSI(l,arg,j,Dj,DPSI)
c*********************************************************************
c
c  Subroutine that calculates the derivative of Psi = x*jl(x)
c
        Double Precision j(0:100),Dj(0:100),DPSI(0:100)
        Double Precision x,arg

      x = arg
      Dj(l) =  1.0/x * ( -(l+1)*j(l) + x*j(l-1) )
      DPSI(l) = j(l) + x*Dj(l)

      return
      end

c*********************************************************************
        subroutine          subETA(l,arg,h,ETA)
c*********************************************************************
c
c  Subroutine that calculates ETA l (x) = x* spherical hankel fcn.
c  (outward going sph. hankel)
c
      Double complex h(0:100),ETA(0:100)
      Double Precision x,arg

      x = arg
      h(l) = (2*l-1)/x * h(l-1) - h(l-2)
      ETA(l) = x * h(l)

      return
      end

c*********************************************************************
        subroutine          subDETA(l,arg,h,Dh,DETA)
c*********************************************************************
c
c  Subroutine that calculates the derivative of ETA l (x)
c
      Double complex h(0:100),Dh(0:100),DETA(0:100)
      Double Precision arg,x

      x = arg
```

```
  Dh(l) = 1.0/x * ( -(l+1)*h(l) + x*h(l+1) )
DETA(l) = h(l) + x*Dh(l)

return
end
```

```
c*******************************************************************************
c                              Program         INT_apave.f                      *
c                              3-03-93         BJ Rask                          *
c                                                                               *
c     Program that computes the effect of the aperture averaging on the         *
c     Intensity, and therefore on the SNR.  The aperture is the half circle.    *
c     Tests effect on heterodyne, coherent system.  Program computes the        *
c     aperture averaged intensity vs. detector radius.                          *
c*******************************************************************************

c  Definition of variables:
c
c               Cn2 = strength of turbulence
c               Zc = distance to center of range bin (meters)
c               rho = detector spacing, a variable over which the aperture is
c                               averaged.  P has been replaced by rho.
c               k = wave number = 2 * pi / lambda
c               FZS = Fresnel zone scale size
c               alpha = a/FZS
c               a = aperture radius of half circle
c               Ft = Transmitter focus / Zc
c               Flo = Receiver focus / Zc
c


      character*14 ans, filecov, filenorm
      integer istart, iend, istep
    real apert(0:2000), a0
    real ap1(0:2000), ap2(0:2000)

    Common /params1/ a0, Zc, nterms

c-------------------------------------------------------------
c  start program input section
c-------------------------------------------------------------
c

c     print*,'************************************************'
c     print*,'*          Program  INT_apave.f                *'
c     print*,'************************************************'
c     print*,'************************************************'
c     print*
c     print*
c     print*

c     print*,'INPUT istart, iend, istep, dfact'
c     read*,istart, iend, istep, dfact
c     read(8,*) istart, iend, istep, dfact
      nterms = 5   !number of terms of Gauss quaderature
      istart = 1   !starting radius
    iend = 20    !ending radius
    istep = 1    !step size
```

```
      dfact = 10.  !scale factor to convert to: (meters/FresnelZoneScale)

c     print*,'INPUT covariance file name'
c     read(5,*) filecov
c     read(8,*) filecov
c     print*,'INPUT normalized covariance filename'
c     read(5,*) filenorm
c     read(8,*) filenorm
      filenorm = 'Inorm.dat'
      Zc = 450

c     OPEN(10, file = filecov, status = 'unknown')
      OPEN(11, file = filenorm, status = 'unknown')
      OPEN(15,file = 'Iapave.dat',status = 'unknown')

c-----------------------------------------------------------------------
c  end of program input section
c  start to compute the aperture averaging effect.
c-----------------------------------------------------------------------
c
c     print*
c     print*,'Computing the effect of the aperture averaging'

      call initialize                          !subroutine that computes constants needed
                                               !to calculate the Log amplitude covariance
        do 10 ii = istart, iend, istep
              a0 = ii * 1.0/dfact
              call aperture_ave(apert, ii)
              ap2(ii) = apert(ii)
10    continue

c     print*
c     print*
c     print*,'------ End of INT_apave.f --------'

      close(10)
      close(11)
      close(14)
      close(15)

      stop
      end

c*******************************************************************************
      subroutine aperture_ave(apert, ii)
c*******************************************************************************
c
c  Subroutine that calculates the effect of aperture averaging
c  on the Intensity.  The effect of the aperture averaging
c  is evaluated as a function of the Fresnel Zone Scale size.
c
c           a0 = a/FZS
c           a = radius of detector aperture
c           FZS = Fresnel Zone Scale Size = (Zc/k)**(1/2)
c           Zc = Distance to center of range bin
c           k = wave number = 2pi/lambda
c           phi1 = first angular integration variable in the receiver aperture
c           P1 = first detector radius coordinate
c           phi2 = second angular integration variable in rec. apert.
c           P2 = second detector radius coordinate
c
c
      Double Precision SUMcov,SUMint, A(7),B(7)
      Double Precision ab1(10),ab2(10),ab3(10),ab4(10),ab5(10)
      Double Precision ab6(10),ab7(10)
      Double Precision w1(100),w2(10),w3(10),w4(10),w5(10)
      Double Precision w6(10),w7(10)
      Double Precision P1,P2,z
      Double Precision phi1,phi2
      Integer ifail, itrans, npts, nrand
      external D01BAZ,D01BBF
      real apert(0:2000),a0,pi,Is

      Common /params1/ a0, Zc, nterms
      Common /Ps/ P1,P2
      Common /phis/ phi1,phi2
      Common /zs/ z

      pi = 3.1415926

c *** Define the integration limits for the 5 integrations

      A(1) = pi/2.
      B(1) = 3.*pi/2.
      A(2) = 0.0
      B(2) = a0
      A(3) = pi/2.
      B(3) = 3.*pi/2.
      A(4) = 0.0
      B(4)= a0
      A(5) = (Zc-150)/Zc
      B(5) = (Zc+150)/Zc

c *** calculate the gaussian weights and abscissae for each dimension

      call D01BBF(D01BAZ,A(1),B(1),itype,nterms,w1,ab1,ifail)
      call D01BBF(D01BAZ,A(2),B(2),itype,nterms,w2,ab2,ifail)
      call D01BBF(D01BAZ,A(3),B(3),itype,nterms,w3,ab3,ifail)
      call D01BBF(D01BAZ,A(4),B(4),itype,nterms,w4,ab4,ifail)
      call D01BBF(D01BAZ,A(5),B(5),itype,nterms,w5,ab5,ifail)

c *** Now build up the 5th order summation using Gaussian
```

```
c *** quadrature for the other integrations.

      SUMint = 0.0
      do i1 = 1, nterms
       do i2 = 1, nterms
        do i3 = 1, nterms
         do i4 = 1, nterms
          do i5 = 1, nterms
            phi1 = ab1(i1)
            P1 = ab2(i2)
            phi2 = ab3(i3)
            P2 = ab4(i4)
            z = ab5(i5)
            call Integrand(Is)
            SUMint = SUMint + w1(i1)*w2(i2)*w3(i3)*w4(i4)*
     >                 w5(i5)*Is
          end do
         end do
        end do
       end do
      end do

      write(11,*) a0, SUMint/a0**4
      write(15,*) a0, SUMint
      apert(ii) = SUMint

      call flush(11)
      call flush(15)

      return
      end

c**************************************************************************
      subroutine Integrand(Is)
c**************************************************************************
c
c  Subroutine to calculate the integrand for the 5th order
c  integration for the average hetrodyne power.
c
c  Definition of variables:
c
c            Ulo = the local oscillator electirc field distribution.
c            P1 = first radial variable in detector #1
c            P2 = second radial variable in detector #1
c            phi1 = first angular variable in detector #1
c            phi2 = second angular variable in detector #1
c            Zc = distance to center of the range bin
c            k = wave number = 2pi/lambda
c            FZS = Fresnel Zone Scale Size = sqrt(Zc/k)
c            Ft = transmitter focal distance / Zc
c            Flo = receiver focal distance / Zc
c            alpha0 = transmitter beam radius / FZS
c            beta0 = receiver beam radius / FZS
c            H = fourth order mutual coherence function representing effect
c                 of turbulence on the intensity.
c                 NO joint-gaussian assumption at receiverand NO independent paths.
c            D = phase perturbation part of H
c            Cx = log-amplitude perturbation part of H
c

      Double Precision P1,P2,z
      Double Precision phi1,phi2
      real Zc,k,FZS,Ft,Flo,alpha0,H,pi,Is
      Double Precision P12sq
      Complex J,g1,g2,Uloi1,Uloi2

      Common /params1/ a0, Zc, nterms
      Common /Ps/ P1,P2
      Common /phis/ phi1,phi2
      Common /zs/ z

      Cn2 = 1.e-10
      pi = 3.1415926
      J = (0.0,1.0)
c
c *** Define the focal distances.  As it turns out, Cs1s2 can't quite
c *** be expressed completely as a function only of FSZ.  Unforturnaltely,
c *** Zc, the distance to the center of the range bin, must be chosen.
c *** Here, Zc is chosen to be 450 meters.  The receiver has a maximum
c *** focus distance of 100 meters, so Flo = 100/450.  Ft can
c *** be focussed out to about 450 meters.  Choose Ft = 450/450 = 1.0.
c *** Outgoing beam radius = 72 mm from "optics.f" program.  So
c *** alpha0 = 72mm/FZS.  Radius of LO beam = 13.1mm.
c *** So beta0 = 13.1mm/FZS.
c
      Zc = 450
      k = 2*pi/10.6e-06
      FZS = sqrt(Zc/k)
      Ft = 1.0
      Flo = 100./450.
      alpha0 = 72.e-03/FZS
      beta0 = 13.1e-03/FZS
c
c *** the following are the arguments of the phase and log amplitude
c *** perturbation terms
c
      P12sq = P1**2 + P2**2 - 2*P1*P2*cos(phi1-phi2)
      a1sq = abs(P12sq)
      a2sq = abs(.25*P1**2+.25*P2**2 +P1*P2*cos(phi1-phi2))
      a3sq = abs(.25*P1**2+9./4.*P2**2+2.5*P1*P2*cos(phi1-phi2))
      a4sq = abs(9./4.*P1**2+.25*P2**2-1.5*P1*P2*cos(phi1-phi2))
      a5sq = a2sq
```

```
c *** The first three terms are complex, of the form:
c *** exp(j*theta1)*exp(j*theta2)*exp(j*theta3)
c *** Because only the even terms will survive the integration,
c *** we want to only keep the even terms.  From analysis, only the
c *** REAL part of the above expression is even.  Therefore, ignore
c *** the odd, imaginary part.
      Uloi1 = cexp(-J*.5/Flo*float(P1)**2)
      Uloi2 = cexp(J*.5/Flo*float(P2)**2)
      g1 = cexp(J/(2.*float(z))*(float(P1**2-P2**2)))
      Ulog1 = REAL(Uloi1*Uloi2*g1)
      Ulo1 = dexp(-0.5*(P1/beta0)**2)
      Ulo2 = dexp(-0.5*(P2/beta0)**2)
c
c *** Now compute H, the 4th order mutual coherence function
c
      D1 = 2.91*3./8.*Cn2*z/Zc*k**3*FZS**(11./3.)*P12sq**(5./6.)
      D2 = 2.91*3./8.*Cn2*z/Zc*k**3*FZS**(11./3.)*a2sq**(5./6.)
      D3 = 2.91*3./8.*Cn2*z/Zc*k**3*FZS**(11./3.)*a3sq**(5./6.)
      D4 = 2.91*3./8.*Cn2*z/Zc*k**3*FZS**(11./3.)*a4sq**(5./6.)
      b = sqrt(a5sq)
      call Cx(COV, b)
      Cx1 = .132*Cn2*pi*pi*z/Zc*FZS**(11./3.)*k**3*COV
       H = exp(-1./2.*(2*D1-2*D2+D3+D4))*exp(4*Cx1)
c
c *** Now put it all together:
c
      Is = exp(-P12sq*(alpha0/2.)**2*(1/z-1/Ft)**2)*
     >     1./z**3*P1*P2*
     >     exp(-(P12sq/(2.*alpha0)**2))*Ulog1*Ulo1*Ulo2*H

      return
      end

c*******************************************************************************
      subroutine Cx(COV, b)
c*******************************************************************************
c
c  subroutine that calculates the log-amplitude
c  covariance of Intensity.  The integral in the Log-Amplitude term
c  is calculated using the algorithm found in the appendix of
c  "Two-point joint-density function of the intensity for a laser-
c  generated speckle field after propagation through the turbulent
c  atmosphere", V.S. Rao Gudimetla, J. Fred Holmes, and R. A. Elliott.
c  Equation (A13) of the above reference will be used for the calculation.
c  The hypergeometric functions were expanded and it was noticed that
c  the imaginary terms all cancelled.  The remaining form is purely
c  real, as it should be.  The remaining form also follows a pattern
c  of terms which oscillate in sign.  Howeverm, it was found that the
c  summation converges quickly (3 or 4 terms).  H(a,b) will be calculated
c  using this expanded, real form (see theory development of BJ Rask,
c  notebook on theory).  Then the integral over the normalized path
c  variable from 0 to 1 will be approximated using a Gauss quadrature
c  of 5 points.
c
      real RR(5), ww(5), w
      real TLCOV, COV
      real a1, b1, b
      data RR /.1184634,.2393143,.2844444,.2393143,.1184634/
      data ww /.0469101,.2307653,.5,.7692347,.9530899/

        sumcov = 0.0
      do i = 1, 5
            w = ww(i)
            a1 = w*(1-w)/2.
            b1 = b*w
            call LAcov(a1,b1,w,TLCOV)
            sumcov = sumcov + RR(i)*TLCOV
        end do
      COV = sumcov

      return
      end

c*******************************************************************************
      subroutine LAcov(a1,b1,w,TLCOV)
c*******************************************************************************
c
c  this function subprogram, used by the NAG routine D01BDF, computes
c  the integral in the log-amplitude turbulence term.  The integrand,
c  H(a,b) has been expanded to yield only real terms (see theory, BJ
c  Rask, notebook on theory).  The term in braces of equation (A13)
c  of reference "Two point joint density fcn. for laser-gen. speckle
c  field after prop. through turb. atmos." is an infinite sum of
c  factors which alternate in sign.  However, it was found from other
c  numerical programs that the summation converges quickly (3 or 4 terms).
c
      Double Precision SUM, X
      Double Precision fact1, fact2, fact3, fact4, fact5, fact6
      Double Precision Coeffreg(0:50),Coeffas1(0:50),Coeffas2(0:50)
      Double Precision gam1_3, gam1_6, gam11_6
      Double Precision gama, gamb, gambma
      real pi, w, a1, b1, TLCOV
      Common /coefficients/ Coeffreg, Coeffas1, Coeffas2
      Common /constants/gam1_3, gam1_6, gam11_6, gama, gamb, gambma
c-------------------------------------------------------------------------

      pi = 3.1415926
      X = b1*b1/(8.*a1*a1)
      fact1 = -.15*b1**(5./3.)/pi**(1./2.) * gam1_3**2*cos(pi/6.)/
     >        gam11_6
      fact2 = .3*gam1_6*a1**(5./3.)/2.**(1./6.)
c
```

```
c *** If X > 4pi, must use the asymptotic approx. for the summations
c
    IF (X .gt. (4.*pi) ) then

c
c *** IF HERE, THEN COMPUTING THE ASYMPTOTIC SUMMATION
c *** ================================================
c *** There are 4 summations to compute (two odd, and two even)
c *** compute the first odd summation (asymptotic):
c
      ioe = 1       !odd summation
      ia = 1        !asymptotic summation
      call summation(Coeffas1, ioe, ia, X, SUM,b)
      alpha = (pi/12.)*(5+6*(-5./6.))
      fact3 = 2.*gamb/gambma*X**(5./6.)*sin(alpha)*SUM
c
c *** compute the second odd summation (asymptotic):
c
      ioe = 1       !odd summation
      ia = 1        !asymptotic summation
      call summation(Coeffas2, ioe, ia, X, SUM,b)
      beta = X + (pi/12.)*(5+6.*(-5./6.-1))
      fact4 = 2.*gamb/gama*X**(-5./6.-1.)*sin(beta)*SUM
c
c *** compute the first even term in the summation (asymptotic):
c
      ioe = 0       !even summation
      ia = 1        !asymptotic summation
      call summation(Coeffas1, ioe, ia, X, SUM,b)
      fact5 = 2.*gamb/gambma*X**(5./6.)*cos(alpha)*SUM
c
c *** compute the second even term in the summation (asymptotic):
c
      ioe = 0       !even summation
      ia = 1        !asymptotic summation
      call summation(Coeffas2, ioe, ia, X, SUM,b)
      fact6 = 2.*gamb/gama*X**(-5./6.-1.)*cos(alpha)*SUM
c
c *** compute the LAcov:
c
      TLCOV = fact1 + fact2*(fact3+fact4+fact5+fact6)


    ELSE


c
c *** IF HERE, THEN COMPUTING THE REGULAR SUMMATION (NOT ASYMPTOTIC FORMS)
c *** ================================================================
c *** There are 2 summations to compute, 1 odd and 1 even
c
c *** compute the odd term in the summation.
c
      ioe = 1       !odd summation
      ia = 0        !regular summation
      call summation(Coeffreg, ioe, ia, X, SUM,b)
      fact3 = 2.*cos(pi/12.)*SUM
c
c *** compute the even term in the summation.
c
      ioe = 0       !even summation
      ia = 0        !regular summation
      call summation(Coeffreg, ioe, ia, X, SUM,b)
      fact4 = 2.*sin(pi/12.)*SUM
      TLCOV = fact1 + fact2*(fact3 + fact4)


    ENDIF

    return
    end

c*********************************************************************
    subroutine summation(Coeff, ioe, ia, X, SUM,b1)
c*********************************************************************
c
c  subroutine that calculates the summations.
c
    Double Precision Coeff(0:50)
    Double Precision sumeven, sumodd, SUM
    Double Precision X, error_rel
    real b1
    Integer ioe         !odd or even summation
    Integer ia          !asymptotic or regular summation

c--------------------------------------------------------------------

c
c *** Define the relative error to be 1e-4
c
    error_rel = 1.0e-04

    IF (ioe .eq. 0) then
c
c *** even summation
c
        sumeven = 1.0
        if (ia .eq. 0) then
         !use regular summation
         do i = 2, 24, 2
           sumeven = sumeven + Coeff(i)*X**(i)
         end do
```

```
        else
         !use asymptotic summation
         do i = 2, 24, 2
           sumeven = sumeven + Coeff(i)*X**(-i)
c           print*,'sumevenasy=',sumeven
         end do
        endif
       SUM = sumeven

     ELSE

c
c *** odd summation
c

       if (ia .eq. 0) then
        !use regular summation
         sumodd = Coeff(1)*X**(1)
       else
        !use asymptotic summation
         sumodd = Coeff(1)*X**(-1)
       endif

        if (ia .eq. 0) then
         !use regular summation
         do i = 3, 25, 2
           sumodd = sumodd + Coeff(i)*X**(i)
         end do
        else
         !use asymptotic summation
         do i = 3, 25, 2
           sumodd = sumodd + Coeff(i)*X**(-i)
         end do
        endif
       SUM = sumodd

     ENDIF   !odd summation or even summation IF

     return
     end

c**********************************************************************
     subroutine Nfactorial(Nfact)
c**********************************************************************
c
c  Subroutine that computes the coefficients of the factorial function.
c
     Double Precision Nfact(0:50)

     N = 50
     Nfact(0) = 1
     do 10 i = 1, N
       Nfact(i) = Nfact(i-1)*i
10   continue

     return
     end

c**********************************************************************
     subroutine ABcoeff(atemp, btemp, Afact, Bfact)
c**********************************************************************
c
c  subroutine that computes the Pochhammer constants of the confluent
c  hypergeometric series summation.
c
     Double Precision atemp, btemp
     Double Precision Afact(0:50), Bfact(0:50)

     N = 50
     Afact(0) = 1.0
     Bfact(0) = 1.0
     Afact(1) = atemp
     Bfact(1) = btemp

     do 20 i = 2, N
       Afact(i) = Afact(i-1)*(atemp + i-1)
       Bfact(i) = Bfact(i-1)*(btemp + i-1)
20   continue

     return
     end

c**********************************************************************
     subroutine initialize
c**********************************************************************
c
c  subroutine that calculates the constants needed for the TLC.
c
c
     Double Precision atemp, btemp
     Double Precision Afact(0:50), Bfact(0:50)
     Double Precision Coeffreg(0:50),Coeffas1(0:50),Coeffas2(0:50)
     Double Precision Nfact(0:50)
     Integer ifail
     Double Precision arg1_3, arg1_6, arg11_6
     Double Precision gam1_3, gam1_6, gam11_6
     Double Precision arga, argb, argbma
     Double Precision gama, gamb, gambma
     Double Precision S14AAF
     External S14AAF          !NAG routine to get the gamma fcn.
```

```
      Common /coefficients/ Coeffreg, Coeffas1, Coeffas2
      Common /constants/gam1_3, gam1_6, gam11_6, gama, gamb, gambma
c
c *** define the arguments of the gamma functions, and compute them
c
      arg1_3 = 1./3.
      arg11_6 = 11./6.
      arg1_6 = 1./6.
      gam1_3 = S14AAF(arg1_3, ifail)
      gam11_6 = S14AAF(arg11_6, ifail)
      gam1_6 = S14AAF(arg1_6, ifail)

      arga = -5./6.
      argb = 1.0
      argbma = argb - arga
      gama = S14AAF(arga, ifail)
      gamb = S14AAF(argb, ifail)
      gambma = S14AAF(argbma, ifail)
c
c *** compute the coefficients of the factorial function
c
      call Nfactorial(Nfact)
c
c *** Compute the coefficients of the summations which contained
c *** the hypergeometric function. These summations are real,
c *** and oscillate in sign, but converge quickly. First compute
c *** the coeeficients for the regular approximation to the summations,
c *** then compute the coeffients for the asymptotic approx. to the
c *** summations.

c
c *** first the regular approximation for the Coefficients:
c
      a1 = -5./6.
      b1 = 1.0
      atemp = a1
      btemp = b1
      call ABcoeff(atemp, btemp, Afact, Bfact)

       Coeffreg(0) = 1.0
      do i = 1, 29, 2
       Coeffreg(i) = (-1)**((i+1)/2)*Afact(i)/Nfact(i)**2
      end do

      do i = 2, 30, 2
       Coeffreg(i) = (-1)**(i/2)*Afact(i)/Nfact(i)**2
      end do
c
c *** now the asymptotic approximation for the Coefficients:
c *** asymptotic set #1
c
      atemp = a1
      btemp = 1.0+a1-b1
      call ABcoeff(atemp, btemp, Afact, Bfact)

       Coeffas1(0) = 1.0
      do i = 2, 30, 2
       Coeffas1(i) = (-1)**(i/2)*Afact(i)*Bfact(i)/Nfact(i)
      end do

      do i = 1, 29, 2
       Coeffas1(i) = (-1)**((i+1)/2)*Afact(i)*Bfact(i)/Nfact(i)
      end do

c
c *** now the asymptotic approximation for the Coefficients:
c *** asymptotic set #2
c
      atemp = b1-a1
      btemp = 1-a1
      call ABcoeff(atemp, btemp, Afact, Bfact)

       Coeffas2(0) = 1.0
      do i = 2, 30, 2
       Coeffas2(i) = (-1)**(i/2)*Afact(i)*Bfact(i)/Nfact(i)
      end do

      do i = 1, 29,2
       Coeffas2(i) = (-1)**(i/2)*Afact(i)*Bfact(i)/Nfact(i)
      end do

c
c *** done with initialization
c

    return
    end
```

Appendix E

Appendix E shows how equation (4.45), the low-pass filtered crosscovariance of intensities is simplified to be used in the crosswind detection scheme of Chapter5, section 5.3. Equation (4.45) is

$$C_{I_{lpf}}(\vec{\rho},\tau) = \frac{\beta^2 k}{2}(U_{T_0}\alpha_0)^4 \int d\theta_R \iint dz_1 dz_2 \frac{z_1}{(z_1 z_2)^3} \sum_m b_m$$

$$\times \left[\exp\left(4C_\chi \left(\frac{P_m z_1}{Ak}, \cos\theta_R, \vec{\rho}, -\tau \right) \right) - 1 \right] \quad . \qquad \text{(e.1)}$$

In this equation, the log-amplitude covariance function, C_χ, is

$$C_\chi(\vec{\rho},\vec{\rho},\tau) = 0.132\pi^2 k^2 C_n^2 z \int d\omega \int du \; u^{-8/3} \sin^2\left[\frac{\omega(1-\omega)u^2 z}{2k} \right]$$

$$\times J_0\left[u\left| \omega\vec{\rho} + (1-\omega)\vec{\rho} - \langle \vec{v}_\perp \rangle \tau \right| \right] . \qquad \text{(e.2)}$$

Equation (e.1) difficult because it contains three integrals, and one summation. However, the integrals can be approximated by realizing that the log-amplitude covariance function depends very weakly on the argument

$$\frac{P_m z_1}{Ak}$$

in the log-amplitude covariance function, allowing it to be dropped from equation (e.2). In addition, the argument $\cos\theta_R$ in C_χ is also a weak dependence. Taking these arguments out of equation (e.1) and using

$$\sum_m b_m = 1 \tag{e.3}$$

reduces equation (e.1) to

$$C_{I_{lpf}}(\vec{p},\tau) = \frac{\beta^2 k}{2}(U_{T_0}\alpha_0)^4 \iint dz_1 dz_2 \frac{z_1}{(z_1 z_2)^3}$$

$$\times [\exp(4C_\chi(\vec{p},-\tau)) - 1] \quad . \tag{e.4}$$

The z integrals can be performed by realizing that the z in C_χ can be approximated by setting $z = z_c$, the distance to the center of the range bin, and performing the remaining z integrations with a 300 meter range bin width,

$$C_{I_{lpf}}(\vec{p},\tau) = \frac{\beta^2 k}{2}(U_{T_0}\alpha_0)^4 150\left[\frac{(z_c+150)^2-(z_c-150)^2}{(z_c+150)^3(z_c-150)^3}\right]$$

$$\times [\exp(4C_\chi(z_c,\vec{p},-\tau)) - 1] \quad . \tag{e.5}$$

To demonstrate the validity of these approximations, equation (e.5) and equation (e.1) are plotted in figure e.1 for comparison. The difference between the two curves is so slight, that equation (e.5) can be used to approximate equation (e.1).

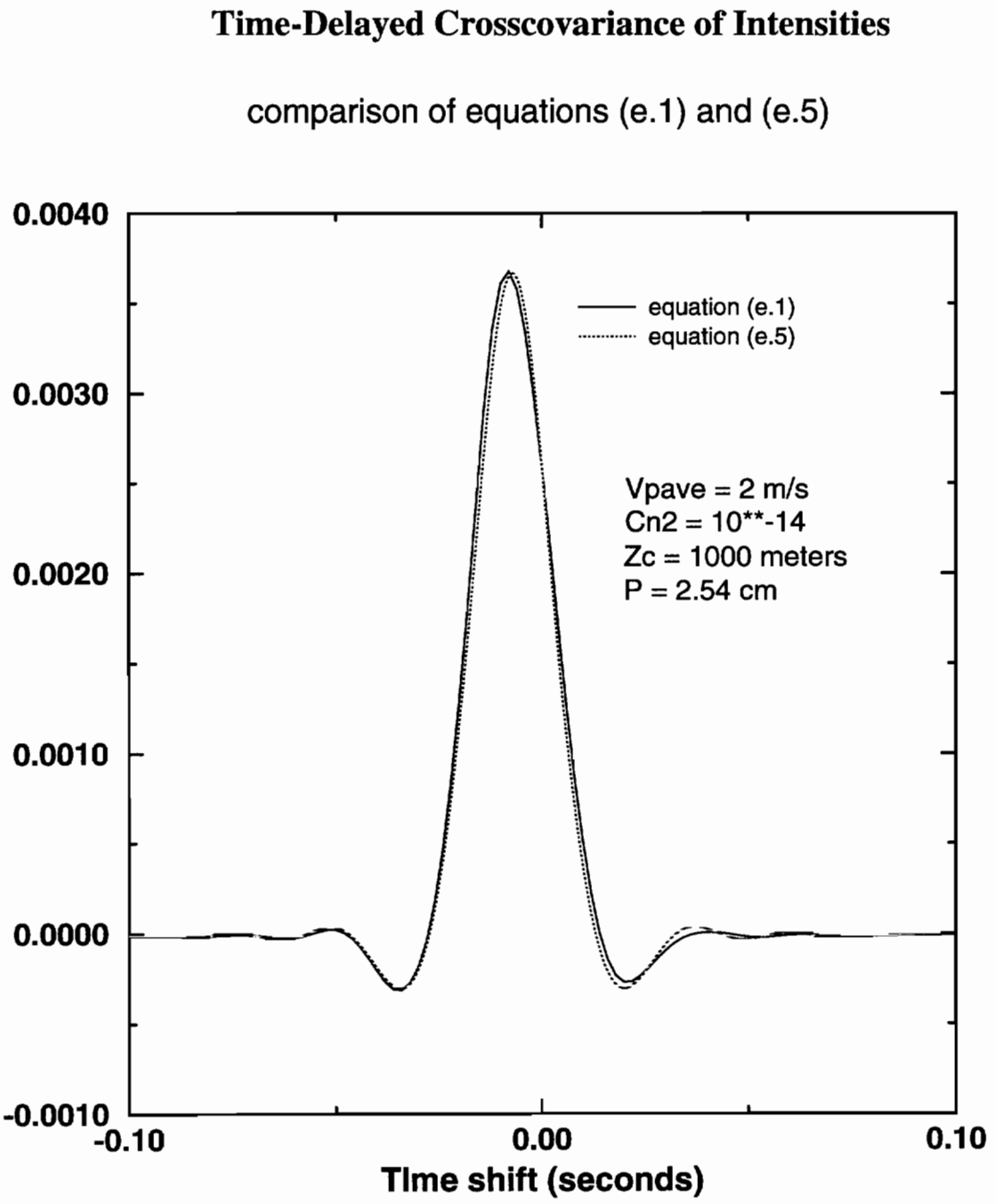


Figure e.1. Comparison of equations (e.1) and (e.5) showing the difference is so slight that equation (e.5) can be used to approximate equation (e.1).

VITA

Badih John Rask was born on January 10th, 1966 in the beautiful city of roses, Portland, Oregon. He graduated from U. S. Grant High School (Portland, OR) in 1984, with letters in football and track & field, and an interest in mathematical sciences. He enrolled in the Electrical Engineering program at Lafayette College (Easton, PA) in 1984 and graduated with a B.S.E.E. degree (Summa Cum Laude, with Honors) in 1988. His undergraduate thesis advisor, Dr. Arthur D. Gorman, whetted his appetite for deeper study in graduate school.

In September, 1989, Badih was accepted into the Oregon Graduate Institute (then the Oregon Graduate Center), and started working with Dr. J. Fred Holmes in the lidar and remote sensing field in 1990. The work with Dr. Holmes has led to 3 published papers, with 3 more submitted for publication. Badih graduated from the Oregon Graduate Institute of Science & Technology with a Ph.D. in June, 1995.

Badih loves to cross-country ski, wood-carve, play the piano, golf, and read vampire literature, and his passion for playing basketball is almost self consuming.

Published papers:

Badih J. Rask and Arthur Gorman, "On Caustics Associated with Horizontal Rays and Vertical Modes," Wave Motion, Vol. 14, 1991, p. 321-332.

J. Fred Holmes and B. John Rask, "Optimum, Optical Local Oscillator Power Levels for Coherent Detection Using Photodiodes," Applied Optics, 20 February 1995, Vol. 34, pgs. 927-933.

J. Fred Holmes and B. John Rask, "Coherent, CW, Pseudo-Random Code Modulated Lidar for Path Resolved Optical Remote Sensing," proceedings of SPIE optical conference, Orlando, Fl, April, 1994.