Development of an Eye Safe Coherent Doppler Wind Lidar System

by

David Santoro

A dissertation submitted to the Graduate Faculty in Electrical Engineering in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York

2012

This manuscript has been read and accepted for the Graduate Faculty in Engineering in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

Dr. Fred Moshary

Date — Chair of Examining Committee

Dean Ardie D. Walser

Date — Executive Officer

Dr. Samir A. Ahmed

Dr. Mark Arend

Dr. Roger Dorsinville

Dr. Barry Gross

Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

ABSTRACT
Development of an Eye Safe Coherent Doppler Wind Lidar System

by

David Santoro

Advisor: Professor Fred Moshary

An eye safe coherent Doppler wind Lidar system with the ability to measure wind velocities along a line of sight originating from the system and extending outwards was developed and tested. Multiple line of sight measurements were incorporated into calculations to ascertain the horizontal and vertical wind vectors. The wind vectors were then used in the calculation of horizontal wind speeds and directions. The system was constructed utilizing components and technologies common to the telecommunication industry to reduce the need for custom components and reduce the cost of the system. The wavelength and power levels were selected to take advantage of existing components and maintain eye safe operation allowing use of the system within public spaces without the need for constant operator and safety personal.

The system was developed in a laboratory and then moved to a research vehicle. The system is polarization maintaining fiber based to enhance coherent heterodyne detection. A balanced detector was utilized to reduce relative intensity noise of the source, and the local oscillator power level was selected to enhance the performance. A beat signal created by the backscattered laser pulse interacting with a local oscillator is analyzed in the frequency domain with the use of a field programmable gate array and the resulting accumulated spectra are further analyzed to determine wind velocities. With spectral accumulation the system is capable of achieving sub-shot noise operation. In the research vehicle the Lidar system can scan the atmosphere at three positions and calculate the vertical and horizontal wind vectors.

Wind measurements made with the system were compared to other meteorological wind instrument measurements in the area with good consistence. Backscatter intensity signals were compared to a direct detection Lidar system and exhibited good agreement. The signals from both systems were processed using a wavelet based planetary boundary layer top determination algorithm and the results for both systems were similar.

ACKNOWLEDGMENTS

To my whole family who have supported me in this endeavor, I thank you for your support and patience. I would especially like to recognize my mother who valued education so dearly even though social constraints she experienced in her youth prevented her from pursuing education beyond the high school level even though she had the ability to continue. I cannot thank her enough for taking me to extracurricular tutoring sessions year after year, struggling so we could afford to keep me in a private school, where I could receive the specialized help I needed with my learning disabilities.

I wish she could have lived long enough to see the day that her son completed his PhD, attend his graduation, partake in the joy of the day as it was the seed to quest for knowledge that she implanted in his mind as a child that has made that day possible despite the obstacles.

To my first child, my first daughter, Tesla, born February 23rd, 2012, to Kathy and I in the tempest period between the building of the system, the writing of this dissertation and recovering from surgery, I hope your mother and I can give you the nurturing that your grandmother gave me so you can pursue your dreams and desires. This will be the gift passed from the grandmother you never meet to me and onto you.

Kathy, my love, your support and understanding during this long process, during your difficult pregnancy and the raising of an infant has been amazing. You are a wonderful mother and partner. Your emotional support and understanding has made such a difference to me.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
Introduction

Statement of Problem

Wind measurements are important components to the study of atmospheric dynamics and transport, weather predictions, aviation operation, urban environmental forecasting, homeland security and wind energy equipment optimization [1-10]. The ability to measure wind velocities at multiple ranges and directions in a populated area or where aircraft may be encountered is a very desirable capability. Any technologies or equipment that would exceed electromagnetic exposure safety limits, interfere with communications or navigation, or create other hazards are unacceptable technologies in the pursuit of wind measurements. To be able to make wind measurements over a substantial area at a variety of altitudes while avoiding fixed installations of structures such as weather station towers, which only provide data at a fixed position and altitude on the earth, is desirable. A wind measurement system's cost, both the initial equipment cost and the cost of operating the system, are of great concern as well. Systems which have low initial equipment cost but require trained operators would limit the system's ability to collect data and raise the cost of operation. On the other hand, a system that could be operated autonomously but had prohibitively high equipment costs is also a limiting factor for wide-scale deployment.

Clearly there is a need for a wind profiling instrument that is low cost, safe to use in public areas, portable, and can collect data autonomously. An eye safe coherent Doppler wind Lidar system can meet these requirements.

History

The acronym Lidar (Light Detection and Ranging), was coined in 1953, seven years before the first working laser was created. The term was used to describe a way to measure cloud height or atmospheric particles with searchlight beams, a method first developed in the 1930s [11]. When the laser was invented, Lidar was one of the first technologies to incorporate it [12], and as both lasers and computers became more advanced, Lidar's range and applications advanced as well. Currently, Lidar is used in many fields: oil and gas exploration, forestry, meteorology, oceanography, geography, and law enforcement, to name just a few [13].

Lidar's Roots

Although the most important advancements in Lidar were made due to the advent of the laser [14], the optical principles upon which Lidar was created have ancient roots, beginning with the first scientific studies of optics by the Greek scholar Euclid [15] [16], who observed that light travels in straight lines (300 B.C.). The Arab or Persian scientist Alhazen, also known as Ibn al-Haytham, (Iraq, 965-1020 A.D.) made great strides in the field with his "Book of Optics," published between 1011 and 1021 A.D [17] [18]. Alhazen advanced knowledge of parabolic mirrors, magnification produced by lenses and atmospheric refraction [19]. Six hundred years later, Hans Lippershey (Netherlands), Galileo Galilei (Italy) and Johannes Kepler (Germany) developed the first telescopes, using glass lenses [20]. In 1668, Sir Isaac Newton used curved mirrors to create a far more powerful device, so powerful that his Newtonian telescope is still in use today in some Lidar applications [21]. Newton also conducted many studies on the behavior of light, including one notably dangerous study in which he inserted a bodkin (a thick needle) into his eye to investigate how pressing on the eyeball changes one's perception of light [22].

Figure 1, Newton's illustration, "An experiment to put pressure on the eye", of inserting a bodkin into his own eye to investigate changes in perception of light.[22]

Waves Versus Particle

In the 17th Century, Newton's studies of the nature of light led him to believe that light was composed of particles, or "corpuscular" in nature, something he described in his 1704 treatise, "Opticks" [23]. But before that, Newton's theory had been disputed by Dutch physicist Christiaan Huygens, who held in 1678 that light was composed of waves that vibrated in a perpendicular motion to the direction the light travels [24]. For the next 200 years, various discoveries supported the wave theory over the particle theory, until Albert Einstein's reconciled the two theories with the discovery of the photon in 1905 (the term “photon” was coined later by physical chemist Gilbert Lewis) [25]. Einstein's work on the photoelectric effect, and as a result, his proposal of the wave-particle duality of light, earned him a Nobel Prize in 1921. Einstein's exploration of the dual nature of light was a crucial stepping-stone in the invention of the laser, and thus, all modern Lidar systems.

From Searchlights to Lasers

Early use of a searchlight in meteorology for the determination of cloud heights has been reported to have been suggested as early as 1872. In May of 1897, Monthly Weather Review, in a short paragraph, mentions this fact and that "with the great increase in the power of modern search lights" more than just cloud heights were being ascertained. [26]

The first Lidar system was proposed in 1930 by Edward Hutchinson Synge, an Irish scientist who pioneered work in near-field optics with Albert Einstein, and who was the first to propose the principle of scanning--manipulating the position of a source relative to a target in an imaging apparatus, a principle that is a crucial component of Lidar and many other technologies [27]. Synge suggested "that density measurements could be made of the lower atmosphere by observing the light scattered from the beam of a search-light." [28]

Synge proposed using a bistatic system of the searchlight along with a telescope to take measurements. In 1935, French scientists began developing a monostatic, pulsed light Lidar system [29]. And in 1938, scientists in France first reported the use of a monostatic system with a pulsed light source to measure cloud height. See figure bellow for illustrations of monostatic and bistatic configurations.

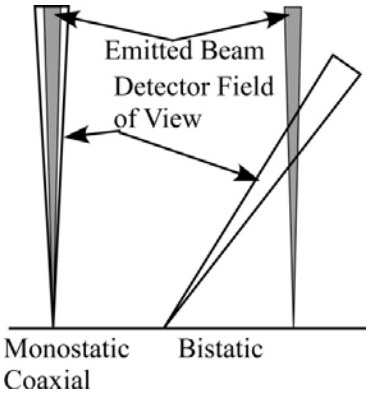


Figure 2, Monostatic coaxial and bistatic configurations.

American scientist Louis Elterman used the monostatic method, with powerful searchlights, to measure stratospheric density distribution in 1951 [28], then in 1953, was also able to measure temperature and pressure along with density using the searchlight technique [30].

That same year, the term Lidar was coined to describe this method of gathering atmospheric data, according to NASA. Middleton and Spilhaus in their 1953 book on "Meteorological Instruments" first suggested the term "Lidar" for light detection and ranging when discussing ceilometry."[11] This early development of the Lidar technique laid the groundwork for a flurry of advances following the invention of the laser.

The Advent of Modern Lidar

The race to invent the laser lasted only a short time, beginning in 1958 with Columbia University's Townes and Schawlow conceptualizing the device in their paper "Infrared and Optical Masers," and ending with Hughes Laboratories' Theodore Maiman building the first ruby laser in Malibu, California, in 1960 [31]. Once completed, the laser became "the heart of the Lidar system" as we know it today [11].

As the book "Laser Applications in Remote Sensing" points out, lasers had huge advantages over searchlights in Lidar research:

> "In General terms Lidar began with searchlights to study stratospheric aerosols and molecular density (see e.g. Elterman [p. 3 and 1966] for a review of the use of searchlights for aerosols). The invention of the ruby laser [Maiman, 1960] provided a better source since it had a short pulse length, produced a low divergence beam, was monochromatic, and had high pulse energy. These properties permitted Lidar systems to be constructed which had many advantages over the use of searchlights. It is notable that the ruby laser was incorporated into the first Lidar systems in at least three areas: aerosol Lidar [Fiocco and Grams p. 35], differential absorption Lidar (DIAL) [Schotland, p. 423] and Raman Lidar [Melfi et al., p. 296]." [32]

The early years of laser Lidar research focused on meteorological applications, with the first paper using laser data, "Detection of Scattering Layers in the Upper Atmosphere," published by Fiocco and Smullin in 1963 [33], and a subsequent paper, "Observations of the Aerosol Layer at 20 km by Optical Radar" published in 1964 by Fiocco and Grams.[34]

A decade later "all basic Lidar techniques had been suggested and demonstrated," according to Claus Weitcamp in "Lidar: Range-Resolved Optical Remote Sensing of the Atmosphere." At that point, Weitcamp writes, the first Lidar textbook, edited by E.D. Hinckley, appeared in 1976.

Coherent Doppler Lidar

As laser technology improved, so did the accuracy, efficiency and variety of Lidar applications. This can be seen in the evolution of Coherent Doppler Lidar (CDL), which NASA first demonstrated in 1967. Used to measure wind fields, these CO_2 Lidars were used in airborne measurements in the 1970s and 1980s. A ground-based pulsed Doppler Lidar was also developed by NOAA in the 1980s, and was used to measure winds in complex terrain. The CO_2 lasers operated near 10 microns wavelength but in the early 1990s, 2-micron solid state laser technology was introduced, which dramatically increased the use of Coherent Doppler Lidar systems. As a NASA research puts it: "The solid state laser technology offered several key advantages over the CO_2 laser that drastically increased the utilization of CDLs. Solid state laser technology allowed for the development of more compact and efficient systems that can operate autonomously. Operating at a shorter wavelength, solid state laser technology eliminated the need for consumables and cryogenically cooled detectors while providing more accurate wind velocity measurements with higher spatial resolution." [35]

Commercial Applications

As Lidar systems grew more powerful, more portable and safer during the 1990s and beyond, Lidar began to realize its commercial potential. Eye-safe systems for use in aviation, aerosol detection, range finding, and target designation were developed. [12, 36] One of the first commercial uses of Lidar in the United States was for mapping vegetation in power line corridors [37]. Doppler Lidar has seen commercial applications in selection of wind energy sites. Optimization of the placement of a multi-million dollar wind turbines can have significant economic impact on the productivity of a wind farm project. Doppler Lidar systems have been

used to optimize the control of wind turbines.[4] It has also seen application in urban meteorology, which strives to quantify the impacts of urban heat island effects on regional weather, climate, air quality, public, health and water resources and management. [2, 38] Doppler Lidar also has applications in weather prediction/modeling, detection of clear air turbulence in aviation and applications to runway wind shear at busy airports. [39-43] Doppler Lidar has even been used by ornithologists to characterize an atmospheric thermal providing lift for soaring raptors.[44]

Structure of Thesis

The structure of the remaining chapters of this thesis is as follows. Chapter 2 presents background information which includes the motivation for this work, the applications of wind Lidar and addresses the traditional wind measurement techniques and their limitations. In Chapter 3 various Lidar principles are reviewed as well as the principles of non-coherent Lidar and Doppler wind Lidar. Chapter 3 also describes the design goals and components of the system developed. A system overview is given that covers the characteristics and performance of the optical, mechanical and electrical aspects of the system. Chapter 4 addresses the analysis of the system. This includes noise, signal-to-noise ratio, optical design, as well as signal detection and processing. This chapter also presents the processes of data collection, analysis and software used to accomplish these tasks. Chapter 5 summarizes the interpretation of the processed signals and explores the collection and calculations performed on the data. The resulting wind measurements from the processed data are also compared to other instruments. Assessment of the system is found in chapter 6. Future work, improvements, component improvements and suggestions are presented in chapter 7. Chapter 8 is a summary of the work and chapter 9 is the conclusion. The appendix and the bibliography are presented in chapters 10 and 11 respectively.

CHAPTER 2
Background

Motivation for Work

There is a need for an instrument that can autonomously measure wind velocities while being low cost, portable, reliable, safe, and able to make measurements in three dimensions (3D) over a substantial area. Current wind-measuring instruments have limitations or do not meet all of these criteria. A fiber optic based Doppler wind Lidar instrument can meet these needs. Some of the traditional wind measurement techniques, their characteristics and limitations when operating within the planetary boundary layer follow.

Planetary Boundary Layer

The planetary boundary layer (PBL), also known as the atmospheric boundary layer (ABL), is the lower layer of the atmosphere that is directly affected by the condition at the surface of the planet. This atmosphere layer changes throughout the day and can vary from tens of meters to several kilometers. The PBL is the conductor between the Earth and the free atmosphere above the PBL. It acts to transport energy, momentum, and substances such as aerosols between these two regions. The convective mixing layer or convective boundary layer (CBL) is the result of thermal stratification of the PBL due to positive buoyancy heat flux at the ground level (i.e. Earth surface warmer than air above it). This can lead to strong convective turbulence. The stable boundary layer (SBL) is found at night and occurs when negative buoyancy flux at the ground level (i.e. Earth surface cooler than air above it) damps the turbulence. [45] The distribution of aerosols at various altitudes will change with the change in these layers. See Figure 3 for an illustration of these layers for various altitudes and time of day.

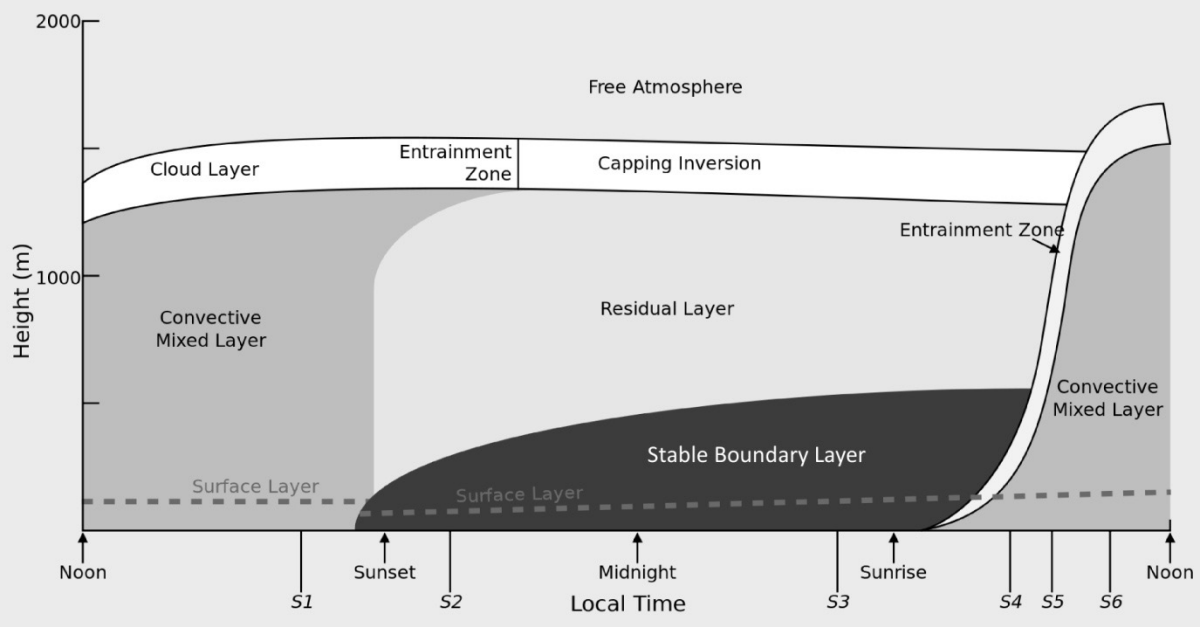


Figure 3, Planetary boundary layer (PBL) schematic representation of layers with respect to altitude and time of day. Modified from [46]

Weather Balloons

Small packages of expendable electronic radiosondes, suspended bellow larger buoyant balloons, are released by approximately 870 land based upper-air observation stations in more than 150 countries twice daily at 0000 Coordinated Universal Time (UTC) and 1200 UTC, 365 days a year by NOAA's National Weather Service. There are also 22 upper-air stations on board special ships. More frequent releases are performed during severe weather events. These "weather balloons" can reach an altitude of 40 km (25 miles) while rising at close to 5 m/s and measuring atmospheric pressure, air temperature, and relative humidity. The data is sent by radio link to the ground while the position of the balloon, and thus, wind direction and speed, are determined by an on-board global positioning system (GPS) or by radio tracking antennas on the ground. These "weather balloon" soundings of the atmosphere can last two hours and the balloon can travel horizontally 300 km (about 180 miles) from the release point. Once the balloon bursts due to its expansion with the decreased atmospheric pressure, a small parachute slows the descent of the

radiosonde. Only about 20% or the nearly 70,000 radiosondes released each year are returned, reconditioned and reused by the NOAA National Weather Service. [47] [48] [49]

While balloon-based wind measurements are made over a significant range of altitudes, the measurements are made only twice daily for a particular longitude, latitude and altitude, require trained operators 7 days a week and have significant costs in the consumables used to make the measurements.

Meteorological towers

Meteorological towers do offer the ability to make continuous measurements. These towers are often free-standing platforms for a variety of meteorological instruments and may have multiples of an instrument along the tower's length. Instruments found on a tower may measure parameters such as horizontal wind speed, horizontal wind direction, temperature, barometric pressure, solar radiation, humidity and precipitation. Wind measurements from a meteorological tower can be continually recorded but are limited to a relatively low altitude above ground level and are not easily portable. In a complex terrain, changes at the surface conditions may require multiple towers be installed to characterize the wind. Conditions may be very inhomogeneous and thus measurements from one location cannot be assumed to apply over the larger area. Cup and propeller (windmill), anemometers with wind vanes measure direction and speed of the wind. These instruments cannot measure the vertical component of a wind or updraft. They also must be specially constructed to have a low starting threshold to be able to measure very slow speed winds and thus be of light construction, yet be able to withstand high speed winds and wind gusts. Three-axis sonic anemometers use time of flight measurements of sonic pulses to measure wind vectors but have some disadvantages. They have lower accuracy due to precipitation and

must be calibrated to account of the distortion of wind flow due to the transducer support structure. See Figure 4 for images of typical wind speed anemometers.

Figure 4, Image of cup anemometer (left), propeller anemometer (center), and wind vane (right).[50-52]

Wind Profilers

Wind measurement instruments known as wind profilers are able to make measurements along a range of altitudes. These instruments include sodars (sonic detection and ranging), and radar (radio detection and ranging) wind profilers. They rely on small scale turbulent fluctuations in the atmosphere, creating eddies that scatter the probing signal. These classes of instruments tend to be large and not very portable but can be run in an autonomous manner. Many are installed permanently but portable instruments do exist (see Figure 5).

Figure 5, Image of a portable sodar wind profiler (left) and perminant radar wind profiler installation (right).[53, 54]

Sodar wind profilers' measurements rely on the scattering of sound waves by atmospheric turbulence. There are monostatic or bistatic configurations. The monostatic configuration only measures the scattering caused by thermally-induced turbulence. With the bistatic configuration the radial velocity and thermal structure can be accessed by measurement the intensity and frequency shift of the return signal by means of multiple antennas. The backscattered signal is not exclusively measured along the outgoing trajectory but at angles other than 180 degrees. The vertical range of sodar is approximately 2 km. By using multiple antennas in a multiple axis configuration, a single instrument can calculate vertical and horizontal components of the wind. [49]

Radar wind profilers work in a similar manner to sodar wind profiler with the substitution of electromagnetic radio frequency signals instead of acoustic signals. With a large antenna size and a judicious selection of operating frequencies, a range of 20 km can be achieved. Smaller radar wind profilers have ranges on the order of a kilometer. Like sodar, radar wind profilers are not very portable.

Doppler Wind Lidar

Eye safe Doppler wind Lidar systems are available from a number of companies around the world. Companies such as the ZephIR product line manufactured by Natural Power, Windscube by Leosphere and Galion LiDAR by Halo Photonics manufacture and sell wind Lidar systems which are very costly. A cost of $200,000 is not uncommon and a $1,000,000 system is on the market. These systems claim maximum ranges from 200 m to 10,000 m. With increased range come increased costs.

CHAPTER 3
Lidar

Lidar's principles of operation are similar to radar, sonar or ultrasound but with a change in the frequency of electromagnetic radiation used to probe the area of interest. Light at terahertz frequencies is substituted for radio waves at megahertz to gigahertz frequencies or sound waves at kilohertz frequencies. An electromagnetic wave is sent into the area of interest and the return signal is collected, analyzed and interpreted to ascertain some characteristic about the that area remotely.

In the case of Lidar the electromagnetic wave is generally a pulse of laser light and the return signal is groups of photons scattered back from an interaction with some constituent of the atmosphere. These photons are collected with the help of some form of optical telescope. The time difference between the pulse being sent into the atmosphere and the time the back scattered signal is collected reveals the location of the scattering event while measured parameters of the back scattered signal may be interpreted to reveal characteristics of the atmosphere at the location of scattering. A continuous wave laser source may be substituted for the pulsed source if timing information can be encoded into the continuous beam of light by some scheme of modulation, but pulsed sources are more common.

A Lidar system generally consists of a laser based source that can be used to generate short intense pulses of light at known wavelength(s), optics to expand, collimate and direct the beam, optics to collect the scatter signal, optics to perform some analysis of the light, a detector(s) to convert the optical signal into an electrical signal, and a computer to log the data and perform some analysis.

The physical arrangement of the source and telescope can vary. In a coaxial or monostatic system the light is being emitted along the optical axis of the receiver while in a biaxial or bistatic system the source and the receiver's optical axes are separated by at least the radius of the receiver. The laser beam's diameter, divergence, and shape along with the telescope's imaging properties, and the distance separating their optical axes affects the degree at which the signal can be detected at short ranges. The function that describes the geometric effects on the Lidar is known as the laser beam receiver field-of-view overlap function. This function equals zero at the Lidar system and become equal to one when the laser beam is completely imaged onto the detector. Large telescopes may have overlap functions that affect the return signal for a few kilometers.

Scattering

The atmosphere is highly complex and dynamically linked to the earth and solar radiation. Large solid particles (aerosols) and molecular constituents are mainly found in the troposphere (approximately 0 km to 15 km) and the lower stratosphere (approximately 15 km to 20 km). Approximately 80% of the atmospheric components lie within the troposphere. Only 1% of the remaining components are found outside of the troposphere and the stratosphere (i.e. above 50 km). Light traveling in the atmosphere can be refracted, absorbed and/or scattered. Elastic scatter results in the scattered photon’s wavelength remaining virtually unchanged while its path is probably altered. Inelastic scattering of a photon results in a wavelength change as well as its path probably being altered. When the scatterer is much smaller than the wavelength of light, such as a molecule, Rayleigh scattering occurs with intensity proportional to λ^{-4} and an equal amount of forward and backscatter is produced by this elastic scattering. If the scatterer is on the order of a wavelength, such as with aerosol, Mie scattering occurs with intensity proportional to

λ^{-2} and predominantly forward scatter is seen by this elastic scattering. Mie scattering is used to approximate aerosol scattering although aerosols are not often spherical as Mie theory assumes. Backscattered light from the atmosphere exhibits spectral broadening due to two main phenomena, random and systematic motion of the scatterers. Systematic motion of scatterers shifts the entire spectral distribution in the frequency space while the random motion of scatterers changes the shape of the distribution. Small particles exhibit Rayleigh scattering and are more affected by random motion than larger particles. This results in Doppler broadening which is significant and symmetric with respect to that seen with Mie scatterers which is more narrow and directional. A heuristic depiction of the back scattered spectrum is illustration in Figure 6.

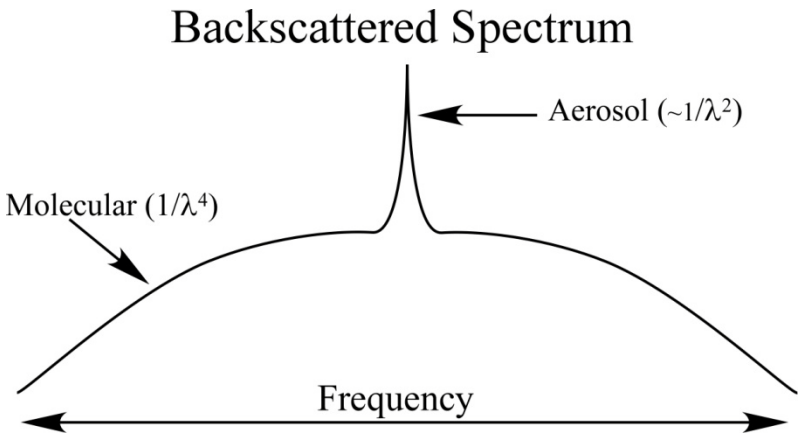


Figure 6, Backscattered spectrum due to Doppler broadening by molecules and particles. Rayleigh, molecular, scattering intensity is proportional to λ^{-4} while Mie, aerosol, scattering is proportional to λ^{-2}.

The broadening of the spectrum of light after a scattering event can be explained by assuming a Maxwellian velocity distribution for the molecules and particles in the atmosphere. The light scattered will have a wider frequency spectrum than before the interaction and have a Gaussian profile. The width of the spectrum will correspond to a characteristic frequency shift, $\Delta\nu = \nu - \nu'$, which is proportional to

$$\Delta v \propto \frac{v}{c}\sqrt{\frac{2kT}{m}} \tag{3.1.1}$$

where k is Boltzmann constant, T is the absolute temperature of the scatterers, and m is the mass of the scatterers. [55, p. 408] Since a non-ideal laser has a linewidth the intensity distribution is the result of the convolution of the laser's intensity profile and a Gaussian profile. A laser source with a wide spectrum would perform more poorly than a very narrow line width laser because of this. Since aerosols are significantly heavier than molecules they will produce a narrower spectral profile after a scattering interaction with light.

Even though the predominate molecular scatter seen in the atmosphere is the elastic Rayleigh scattering there exists another molecular scatter mode. Raman scattering is an inelastic scattering involving the change in the vibration-rotational energy level of a molecule. The scattered light experiences a frequency shift corresponding to the difference between the quantum energy state of the initial and final molecular state. This means the frequency shifted light observed is specific to the molecule that is scattering the light and may be a positive frequency shift (blue shifted or anti-Strokes Raman Scattering) or a negative frequency shift (red shifted or Strokes Raman Scattering). The intensity of the Raman scatter depends on the cross section of the molecule and that depends upon the polarizability of the molecule.

Equations of Lidar

The return signal power, P, from range, R, at wavelength, λ, in its simplest form describing a Lidar system is

$$P(R,\lambda) = K \cdot G(R) \cdot \beta(R) \cdot T(R) \tag{3.1.2}$$

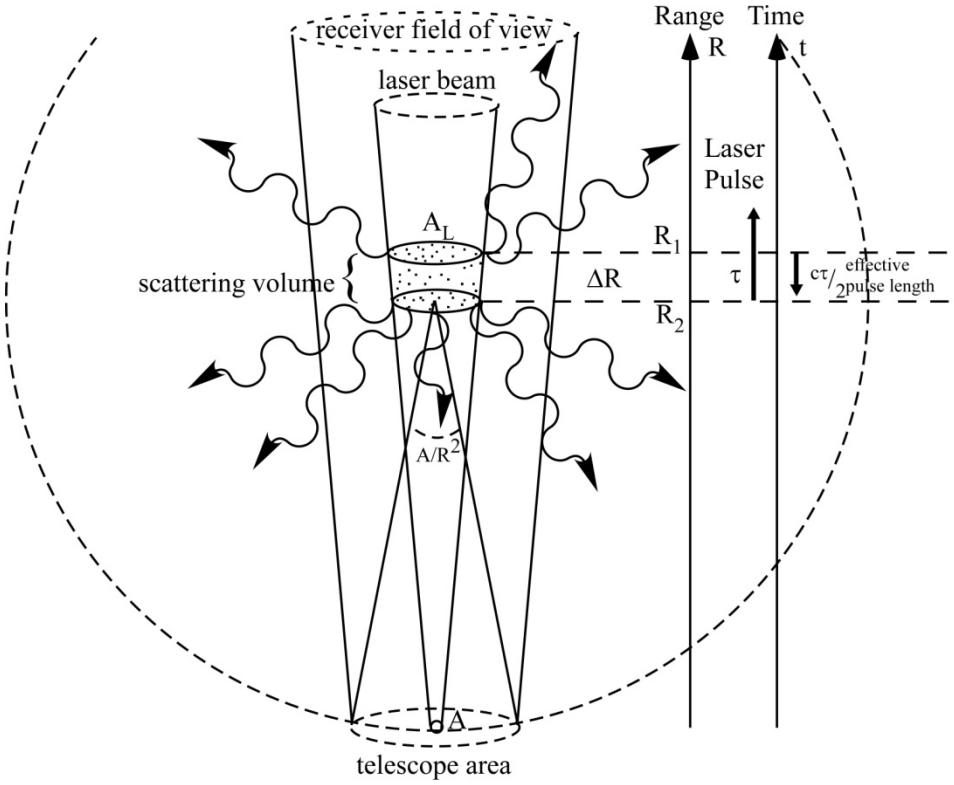
where K, a system constant, summarizes the performance of the Lidar system, G(R) is implemented to account for any range dependent measurement geometry, β(R) is a backscatter coefficient that describes how well that atmosphere at range R scatters light back along the direction from which it came, and T(R) is the transmission term that describes how much light is lost from the beam as it travels to range R and back to the Lidar system. The performance parameter K is equal to,

$$K = P_0 \frac{c\tau}{2} A\eta \tag{3.1.3}$$

where P_0 is the average power of a single pulse from the laser of temporal length τ. The speed of light is c making cτ the length of the volume illuminated by the laser pulse at particular time. The effective spatial pulse length is cτ/2 as there is a folding of the laser beam when it is backscattered. At a time t after the leading edge of the pulse left the laser, backscattered light from the leading edge comes from a distance R1 = ct/2 and the backscattered light from the trailing edge of the pulse comes from R2 = c(t-τ)/2. Thus ΔR = R1-R2 = cτ/2. The area of the optics collecting the signal is A and η is an overall system efficiency incorporating optical efficiencies of optical components and detector(s). The geometric factor is equal to,

$$G(R) = \frac{O(R)}{R^2} \tag{3.1.4}$$

where O(R) is the laser beam receiver-field-of-view overlap function and $1/R^2$ is a range dependence that accounts for the fact that the receiver optics can be considered a portion of a sphere of radius R that surrounds the scattering volume.



(a)

Figure 7, Illustration of Lidar geometry of coaxial system telescope, scattering volume, and overlap of beam and field of view. [56]

If an isotropic scattering event occurs a distance R from the telescope the area A of the collecting optics would collect a portion of the scattered light described by

$$\frac{I_c}{I_s} = \frac{A}{4\pi R^2} \tag{3.1.5}$$

where I_s is the total amount of light scattered and I_c is the amount collected. A/R^2 is the perception angle of the Lidar signal scattered at range R. The 4π term is eliminated by the terms used to define the backscatter coefficient β.

The backscatter coefficient is a specific case of the scattering coefficient when the angle $\theta = 180$ degrees. This is the main parameter that determines the strength of the Lidar return signal by

describing the amount of light scattered back to the Lidar for a particular wavelength at a range. The backscatter coefficient can be written to represent many kinds of a scatters as,

$$\beta(R,\lambda) = \sum_j N_j(R)\left(\frac{d\sigma_{j,sca}(\pi,\lambda)}{d\Omega}\right) \tag{3.1.6}$$

where N_j is the concentration of scattering particles of kind j in the volume illuminated by the laser pulse and $d\sigma_{j,sca}(\pi,\lambda)/d\Omega$ is the particles' differential scattering cross section at wavelength λ for the backwards direction. The concentration is given in inverse cubic meters (m^{-3}), and the differential cross section is in m^2/sr and the backscatter coefficient in units of $1/(m \cdot sr)$.

If there were to be only one type of scattering constituent in the scattering volume the backscatter coefficient and an isotropic scattering cross section σ_{sca} is $4\pi\beta = N\sigma_{sca}$. If the laser beam cross section were A_L then the scattering volume would be $V = A_L\Delta R = A_L c\tau/2$. The scattering cross sectional area of all particles in the volume V is $A_s = N\sigma_{sca}V$. Then the intensity of scattered light from the illuminated volume V is proportional to the scattering cross sectional area A_s. The relative intensity of scattered light is then,

$$\frac{I_s}{I_O} = \frac{A_s}{A_L} = \frac{N\sigma_{sca}c\tau}{2} = \frac{4\pi\beta c\tau}{2} \tag{3.1.7}$$

Using (3.1.5) and (3.1.7) we can get the ratio of collected light to emitted light intensity as,

$$\frac{I_s}{I_O}\cdot\frac{I_c}{I_s} = \frac{A}{4\pi R^2}\frac{4\pi\beta c\tau}{2} = \underbrace{\frac{A}{R^2}}_{\substack{Perception\\ Angle}}\ \underbrace{\frac{\beta c\tau}{2}}_{\substack{Size,\ Backscatter\\ Properties}} \tag{3.1.8}$$

The backscatter of laser light in the atmosphere is the result of scattering from molecules and particle (aerosols) and the backscatter coefficient can be broken down into these component contributions and written as

$$\beta(R,\lambda) = \beta_{mol}(R,\lambda) + \beta_{aer}(R,\lambda) \tag{3.1.9}$$

The molecular scattering is mainly the result of oxygen and nitrogen molecules in the atmosphere and has an altitude dependence as the air density decreases with altitude. Thus the molecular backscatter from high altitudes is less than that from lower altitudes. The backscattering from particles is highly variable over the atmosphere and changes with time. The majority of particles are found in the troposphere bellow 15 km on the high side. The distribution of the particles can be very variable during the day/night cycles. For particles (aerosols) the Mie backscatter cross section is on the order of $10^{-8}\cdots10^{-10}\ m^2/sr$ and for molecules the Rayleigh backscatter cross section is on the order of $10^{-27}\ m^2/sr$.

To account for the light that is lost during the laser beam's trip to the scattering volume and back from the Lidar system the term T(R) is utilized and may take a value between 0 and 1. It is given by,

$$T(R,\lambda) = e^{\left(-2\int_0^R \alpha(r,\lambda)dr\right)} \tag{3.1.10}$$

This term results from a specific form of the Beer–Lambert–Bouguer law for Lidar. The integral takes into account the change in the scattering and absorption behavior of the atmosphere along the path from the Lidar to the range R as the atmosphere is not a homogeneous volume. The factor of 2 doubles the value of the integral to account for the path from range R to the Lidar.

The behavior is assumed not to have changed in the period of a round trip. The light extinction is the sum of all transmission losses and $\alpha(R,\lambda)$ is the extinction coefficient. The concentration and extinction cross section, $\sigma_{j,ext}$, for each kind of scatterer j is used to define the extinction coefficient as,

$$\alpha(R,\lambda) = \sum_j N_j(R)\sigma_{j,ext}(\lambda) \tag{3.1.11}$$

Absorption and scattering by molecules and particles are the causes of extinction. The sum of the contribution from each combination can be written as,

$$\alpha(R,\lambda) = \alpha_{mol,sca}(R,\lambda) + \alpha_{mol,abs}(R,\lambda) + \alpha_{aer,sca}(R,\lambda) + \alpha_{aer,abs}(R,\lambda) \tag{3.1.12}$$

The indices "sca" stands for scattering and "abs" stands for absorption. The extinction cross section is the sum of the scattering cross section, σ_{sca} and the absorption cross section, σ_{abs} .

$$\sigma_{ext}(\lambda) = \sigma_{sca}(\lambda) + \sigma_{abs}(\lambda) \tag{3.1.13}$$

The backscatter coefficient and the extinction coefficient are wavelength depended and is change with the particles size, refractive index, and shape.

To summarize the Lidar equation (3.1.2) in detail

$$P(R,\lambda) = \underbrace{P_0 \frac{c\tau}{2} A\eta}_{K} \cdot \underbrace{\frac{O(R)}{R^2}}_{G(R)} \cdot \beta(R,\lambda) \cdot \underbrace{e^{\left(-2\int_0^R \alpha(r,\lambda)dr\right)}}_{T(R,\lambda)} \tag{3.1.14}$$

Direct Detection Lidar at CCNY

A non-eye-safe direct detection Lidar system operating at the City College of New York's (CCNY) optical remote sensing laboratory was used for comparative analysis to the eye-safe coherent Doppler wind Lidar system. Photographs of the direct detection vertically pointing multi-wavelength elastic-Raman scattering Lidar at CCNY are shown in Figure 8. The system is neither portable nor eye-safe as it is not compact or light, emits three wavelengths of light (1064nm, 532nm, 355-nm) at non-eye-safe intensities for the purpose of measuring aerosols, cloud and water vapor with a range of 500 m to 15 km. A schematic representation of the system components is seen in Figure 9 and the specifications of the system are shown in Table 1. [57]

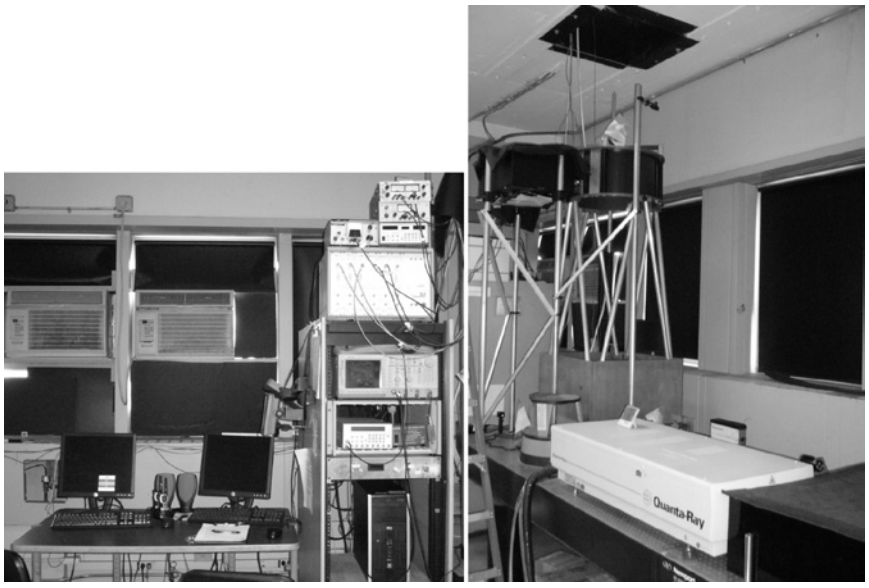


Figure 8, Direct detection Lidar system at The City College of New York. Detection, control, data storage equipment (left) and telescope, laser, optics (right) are shown.

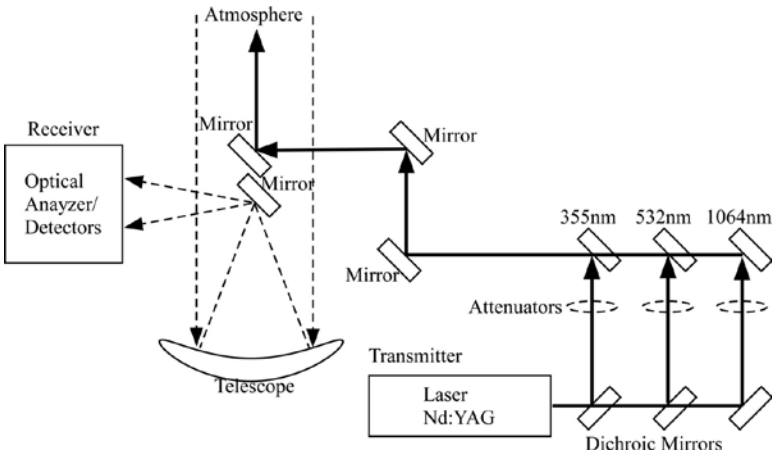


Figure 9, City College of New York direct detection Lidar schematic of system layout.

The system sends a beam consisting of three wavelengths of light (1064nm, 532nm, 355-nm) into the atmosphere but monitors the additional wavelengths of 407nm and 387nm. Although the laser emits a beam consisting of three wavelengths the beam is broken down by wavelengths, passed through attenuators, and recombined into a beam for transmission into the atmosphere. This is done so that the strength of each wavelength component may be adjusted and that the alignment of the different wavelength beams can be fine tuned beyond the manufacturer only specifications. The system uses a variety of different detectors and optics to directly measure the backscattered signals.

The system simultaneously monitors near infrared, visible, and ultraviolet wavelengths to help identify the class of aerosols detected. If there is a dramatic decrease in the backscatter properties at longer wavelengths the particles causing the signal are likely finer in their dimensions and if the inverse is true the particles are likely more coarse.

The 1064nm wavelength is utilized to determine the planetary boundary height as it is less sensitive to molecular backscatter and attenuation than the other wavelengths. The signal seen at 1064nm is mainly due to aerosols in the atmosphere and since most of the aerosols are confined

in the planetary boundary layer the change in the 1064nm signal can indicate the transition to the free atmosphere. The Raman scattering seen at 407 nm is the result of inelastic interaction with water molecules while inelastic interaction with nitrogen is seen at 387 nm. The system's output power levels make the system non-eye safe. With the location of the system being close to two very busy international airports and a large number of helicopters in the skies over New York City, a lookout and a radar system are employed to block the beam from entering the atmosphere should an aircraft approach. A typical data set can be seen for this system in the Figure 10.

Table 1, The direct detection vertically pointing multi-wavelength elastic-Raman scattering Lidar at the City College of New York's (CCNY) remote sensing laboratory

Laser	Q-switched Spectral-Physics Quanta Ray Pro230 Nd:YAG with variable output power up to 475 mJ at 532 nm, 950 mJ at 1064 nm and 300 mJ at 355 nm. Repetition rate of 30 Hz with 8-12 ns pulse duration at 1064 nm, <0.5 mrad divergence.
Telescope	Diameter 20 inch Newtonian Reflector, F number 3.5
Detectors	APD (silicon enhanced avalanche photodiode) for the 1064 nm (Infrared) channel. PMT (Hamamatsu photomultiplier tubes) for the 532 nm (visible), 407 nm (Raman), 386 nm (Raman) and 355 nm (UV) channels.
Digitizing System	Lidar Transient Recorder TR 40-160 (LICEL) with 12-bit, 40 MHz analog to digital converter for 10 MHz and 200 MHz signals, 64-level fast discriminator for signals > 200 MHz.
Range Resolution	500 m to 15 km
Data Acquisition	Acquisition system is configured by using the TR 40-160 Transient Recorder modules for all channels in a five-channel rack comprising power supplies and interface ports to a PC computer equipped with National Instruments digital I/O card, DIO-32F.

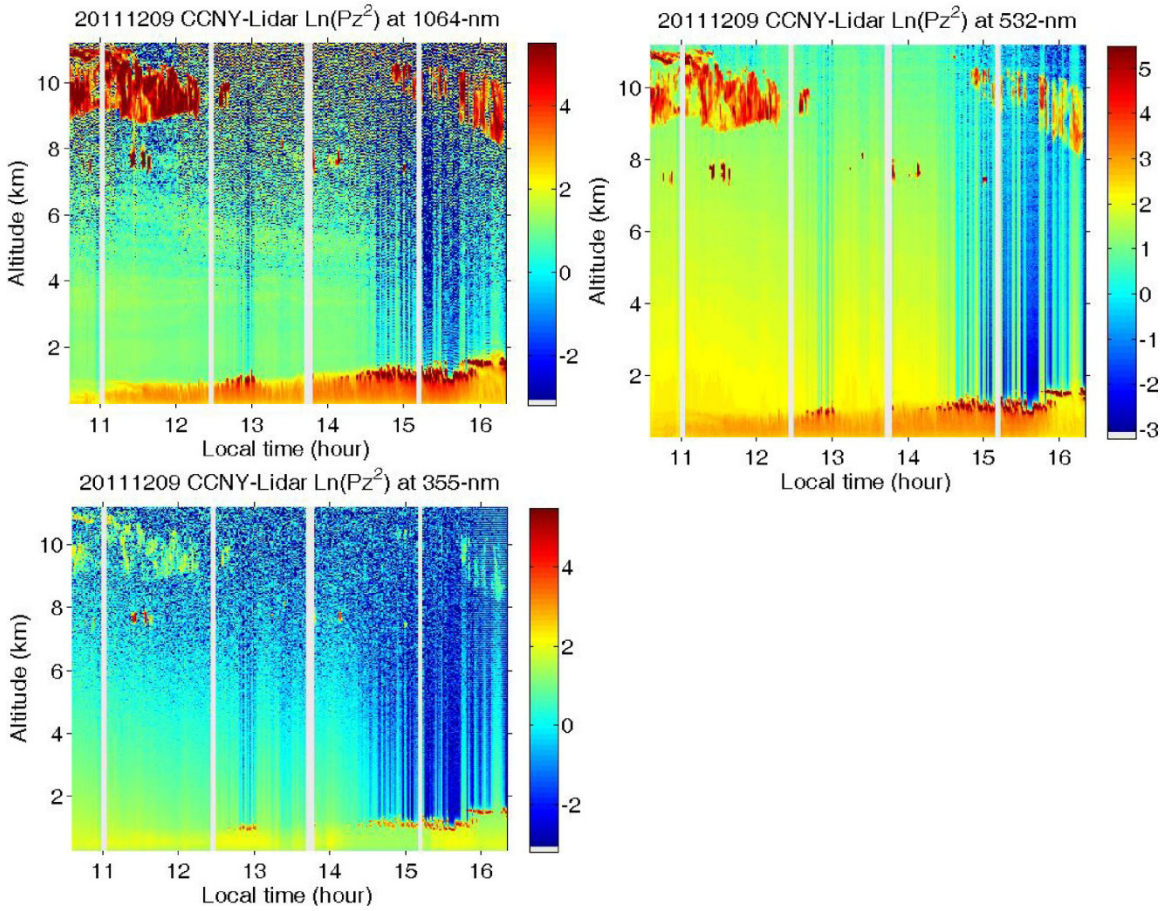


Figure 10, CCNY direct detection Lidar: height cross section of the range-corrected Lidar returns for wavelengths 1064 nm (top, left), 532 nm (top, right), and 355 nm(bottom, left).

Lidar Wind Measurements

Lidar can be used to measure wind speed through tracking and correlating measurements over time and by measuring frequency shifts by coherent and incoherent methods. The Lidar methods can have higher temporal and spatial resolution than sodar and radar based instruments while allowing remote measurements to be made. Wind speed instruments that measure a specific local point in space may have high resolution 0.01 m/s [58] as compared with Lidar instruments but they are only single point instruments. Making measurement of a wide area at an significant altitude would be prohibitively expensive.

Lidar can determine the speed of wind by tracking atmospheric structures in time and by measuring frequency shifts of the laser beam probing the atmosphere. By tracking the differences in particulate distributions in the atmosphere over time the transport of the particulate distributions can be found and the wind speed and direction found. This method can require one or a number of Lidar beams to probe the atmosphere or a beam to scan the atmosphere. The structure of the atmosphere is discerned from the contrast seen from region to region due to differences in particulate concentration distributions. This is a key limitation for the tracking method as without contrast, tracking cannot be performed. This method has lower accuracy than the frequency shift methods.[55]

When light is backscattered by an object the frequency of the light can be shifted by an amount that is dependent upon the speed of the object, given that the source is not moving. This Doppler shift, once measured, can be used to ascertain the speed of the scatterer and by inference the speed of the wind. There are a few different techniques for drawing the degree of the frequency shift from a backscattered signal resulting in higher temporal and spatial resolution than seen from sodar or radar systems. The wind Lidar systems can implement coherent (heterodyne) detection or incoherent (direct) detection techniques.

Doppler Shift

The relative shift in frequencies, $\Delta f/f$, where Δf is the frequency shift and f is the original signal frequency, is proportional to the ratio of the object's velocity to the speed of light, v/c. With the wind speeds of interest being from 0.1 m/s to 100 m/s and the speed of light being $3x10^8$ m/s, the task of discerning the speed of wind from weak backscattered signals from hundreds or thousands of meters away with a temporal resolution to allow range resolutions on the order of

10s of meters amounts to being able to resolve a frequency shift of 30 parts in a billion (30ppb) to 30 parts in a million (30ppm) from a very noisy signal.

If light were emitted from an object traveling at velocity v along a line of sight with an initial frequency of $f_0 = c/\lambda_0$, where c is the speed of light and λ_0 is the wavelength, the observed frequency would be

$$f = f_0 + \frac{f_0 v}{c} \tag{3.3.1}$$

Scatterers in the atmosphere do not emit their own light but scatter light 180 degrees back to the Lidar system as they move. This makes the frequency of the light detected by the Lidar system which has been scattered by an object traveling along the laser beams axis

$$f = f_0 + 2\frac{f_0 v}{c} \tag{3.3.2}$$

If the object were not traveling on the optical axis of the beam but at an angle the change in frequency would have an angle dependence,

$$f = f_0 + 2\frac{f_0 v}{c}\cos(\theta) = f_0 + 2\frac{v}{\lambda}\cos(\theta) \tag{3.3.3}$$

Thus an object traveling in a plane orthogonal to the axis of the beam would have no effect on the frequency as the cos(90) = 0. If the frequency shift is positive the object is moving towards the Lidar and if the shift is negative the opposite is true.

Doppler Shift of Backscatter Spectrum

When wind, the movement of a mass of air, consisting of a mixture of aerosols and molecules, backscatters light from a Lidar system the broadened spectrum is shifted. As described earlier, the constituents of the atmosphere are in constant motion and the velocity distribution is related to the mass of the components. Low mass components, molecules, will have a broader velocity distribution than the heaver components, aerosols. When a mass of air moves along the line of sight of the Lidar system, the spectrum of the laser source is Doppler broadened due to the small scale motion of the component of the air, and a Doppler shift of the center frequency of the laser source is also observed. Figure 11 shows an illustration of a backscatter broadened spectrum when the air mass is stationary and when the air mass is moving towards the Lidar system, creating a Doppler shifted version of the Doppler broadened spectrum of the laser source. Lidar techniques to determine the degree of the Doppler shift using the spectral peak of aerosol backscatter and/or the broad spectrum of molecular backscatter are able to determine wind speed. Gas molecules are ubiquitous in the atmosphere although their density can vary with altitude. The means there is a scatterer nearly everywhere for a Lidar that uses Rayleigh scattering for its measurements. Aerosols are mostly found in the planetary boundary layer and at very low concentration in the free atmosphere. This means that a Lidar system using Mie scattering to make measurements may have difficulty at high altitudes. Aerosols produce a stronger, more easily discernible peak in the backscatter spectrum, making attractive subjects for a wind Lidar systems. A molecular scatter Lidar system has the benefit of a Lidar target at every altitude.

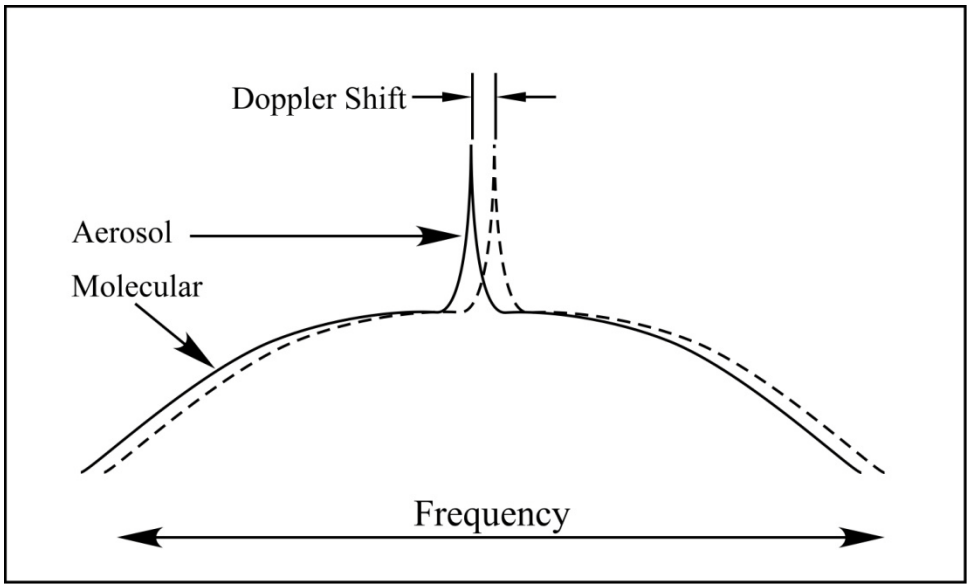


Figure 11, Backscattered spectrum due to Doppler broadening by molecules and particles (solid line). A Doppler shift of the broadened spectrum due to Doppler shift resulting from movement of a volume of air. (dashed line)

Incoherent and Coherent Wind Lidar

Incoherent

Incoherent or direct detection wind Lidar systems send a pulsed laser beam into the atmosphere and use optical techniques to discern the frequency shift of the backscattered signal. The fringe technique also known as the multi-channel technique, relies on the signal being dispersed on to many detectors so the signal can be sampled at many points in the frequency space. The return signal is compared with a reference signal, the Doppler shift is found and the wind speed determined. With the incoherent edge technique the backscattered signal is filtered with a high spectral resolution filter and the intensity changes measured are related to the frequency shift. By centering the high resolution filter on the steepest part of the backscatter Doppler broadened spectrum and comparing the return signal with a reference the frequency shift may be calculated. A variation on this technique is the double edge technique where two filters are used. This has

been reported to have improved sensitivity and measurement accuracy over the single edge technique with not more complexity. [59]

An example of this is the direct detection double edge based Doppler Lidar system named Goddard Lidar Observatory for Winds (GLOW), designed to be able to operate on molecular backscattered light at 355 nm and at 1064 nm for aerosol backscattered signals. At 1064 the molecular scatter is minimal and at 355 nm the λ^{-4} dependence of molecular scatter can be taken advantage of. This has the advantage of allowing operation in the free atmosphere where aerosol concentrations are low. [60]

Coherent

Coherent Doppler Lidar was first demonstrated in 1967. Early coherent system used CO_2 gas lasers and operated at 10 μm. In the 1990's solid state laser technology helped to advance coherent Lidar by offering a 2 μm sources that were much more compact than the CO_2 gas lasers. Moving away from 10 μm to 2 μm eliminated the need for cryogenically cooled detectors, tanks of consumable gas, and improved the accuracy of wind velocity measurements. [35] With progress in fiber optic based telecommunications came improvements in solid sate lasers and the advent of fiber based lasers, fiber based optical amplifiers with narrow line widths, stability and sufficient optical power for use in coherent Doppler Lidar.

Coherent Lidar systems use heterodyne detection where a weak return signal is combined with strong known local reference signal. The combining of the two signals superimposes the optical waves which are then sent to a square law detector, a photodetector. Constructive and destructive interference of the waves creates an electrical signal that is the sum of the two signals' frequencies, the high frequency component, and another which is the a result of the difference

between the frequencies, the beat component. In optical heterodyne detection the beat component is used to determine the frequency shift of the backscattered signal. The high frequency component is far beyond the frequency cutoff of the detector and is ignored.

By splitting the outgoing laser source into two portions, one can be transmitted into the atmosphere as a pulse and its collected backscattered signal combined with the local oscillator, the portion of the beam which was not altered by transmission into the atmosphere. In order that the frequency difference not be near zero Hertz when there is not wind, to allow for detection of positive and negative frequency shifts, and to have greater spatial resolution, the outgoing laser pulse is often altered with a frequency shift. See Figure 12 for a schematic representation of a coherent Doppler wind Lidar. Coherent Lidar systems tend to be more expensive than direct detection systems due to performance requirements of the laser source (line width, stability) but they are able to achieve shot noise limited performance with noisy detectors that that do not require cryogenic cooling.[61]

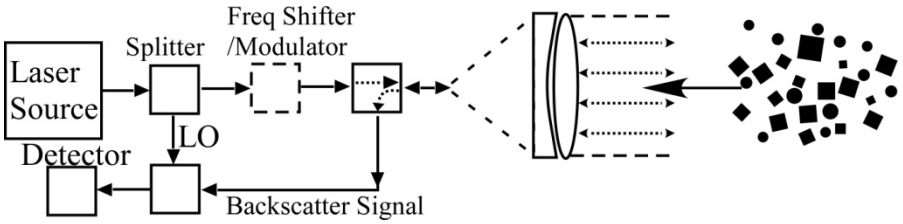


Figure 12, Schematic representation of coherent (heterodyne) Doppler Wind Lidar system.

If Δf is the frequency shift, f_0 is the frequency of the laser pulse to the atmosphere, (f_0+ Δf) is the frequency of the backscattered light, and f_{LO} the frequency of the local oscillator, the detector's current, excluding the DC component due to the local oscillator, is,

$$i_{AC} = \rho\left\{\sqrt{2P_{LO}P(R)}\cos\left[2\pi\left(f_{LO}-\left(f_0+\Delta f\right)\right)\right] + \sqrt{2P_{LO}P(R)}\cos\left[2\pi\left(f_{LO}+\left(f_0+\Delta f\right)\right)\right]\right\} \tag{3.3.4}$$

where ρ is detector sensitivity, P_{LO} is the power of the local oscillator, P(R) is the power of the backscattered signal at range R. Since the second term oscillates at a frequency (f_{LO}+(f_0+ Δf) which is far beyond the cutoff frequency of the detector it is ignored. The current beat signal from the detector is then,

$$i_{DET} = \rho\sqrt{2P_{LO}P(R)}\cos\left[2\pi\left(f_{LO} - \left(f_0 + \Delta f\right)\right)\right] \tag{3.3.5}$$

Fiber Based Coherent Doppler Wind Lidar

The near infrared wavelengths in the range of 1500 nm to 1800 nm are generally considered more eye safe by the organizations that set laser safety standards. [62, 63] A wavelength in this range would make a good selection for use in an eye safe coherent Doppler wind Lidar system as it would allow more optical power to be sent into the atmosphere than a visible, UV, or far infrared wavelength while still remaining eye safe. By being eye safe, Class 1, a Lidar system may be operated without human supervision or other safety protocols needing to be implemented. This is an attractive characteristic for continuous monitoring of the atmospheric winds as it allows autonomous data collection and reduces the manpower requirements of collecting data. In an urban environment such as New York City the safety of air traffic from two major airports and assorted commercial and pleasure fixed wing and rotary aircraft must be maintained.

Being able to send more optical power into the atmosphere means a greater measurement distance should be able to be reached. The backscatter intensity from aerosols is also greater for this region than the visible or UV wavelength ranges, possibly providing a greater operating range of the system. The levels of transmission of these wavelengths through the atmosphere

where water vapor and carbon dioxide are major absorbers/scatters is also high. A high level of transmission increases the possible operating range of a Lidar system.

The fiber optic telecommunication industry has developed around silicon based fibers which have less attenuation per kilometer in the near infrared region, specifically in the 1510 nm to 1600 nm region and the area around 1300 nm. [64] Optical sources, amplifiers, detectors, and other devices, both passive and active, have been refined over years of development and offer a rich choice of mature, "off the shelf" components for a coherent Doppler wind Lidar system. In addition to a mature technology base from the fiber optic telecommunication industry fiber optics also have advantages over free space optics in a system that will be portable and autonomously run in a variety of environments. Fiber optics are less sensitive to environmental changes in temperature and humidity while still able to supply optical power on the level that is eye safe. Fiber optics would also be advantageous in a system that was to be portable as they would need little or no adjustment before beginning operation and are less vulnerable to vibrations. This is generally not true for a geometric optics based system.

Fiber optics are produced in a variety of designs to transmit light down their length with little attenuation. This is accomplished by the selection of the index of refraction of the core and the cladding which surrounds it. The difference in indexes at the boundary between the core and the cladding can reflect light impinging upon it if the angle at which it approaches the boundary is above a critical angle. This phenomenon of total internal reflection keeps the light from exiting the core as it progresses. Generally fiber optics are classified as single mode and multimode types with the diameter of the core determining the mode type. For coherent detection the polarization of the light should be maintained as it travels through a fiber optics. This is possible

with single mode polarization preserving fiber optics which has commonly been designed to operate near the 1550 nm range.

The coherent Doppler wind Lidar system developed uses with single mode polarization preserving fiber optics and a wavelength of 1545.2 nm. The justification for the selection and a more detailed analysis is located in the design section that follows.

Design Goals

The goal of the work is to increase efficacy with respect to wind measurements and maintain safe operation while minimizing complexity and component cost. To make wind measurements at range, the signal to noise ratio should be as large as possible while not surpassing the maximum permissible exposure of persons in the vicinity of the measurements. Ocular exposure is one of the primary concerns and is often the main limiting factor in creating a system that may be operated in public areas. Increasing the output power may allow for the measurement of wind at a greater range but can make the system non-eye safe and increase the cost of the components.

If the system was being designed to operate in a controlled area where eye safety was not a concern and a low cost, high output power laser source was available, an increase in the signal to noise ratio at range could be effected. But, this may also increase the cost of the optical components aside from the laser. All of the optical components would need to operate at higher optical power levels in order to handle the increase in the source's power levels. This may raise the cost of the components and thus the overall cost of the system.

By judicious selection of components and the system's design tradeoffs, an increase in the signal to noise ratio, i.e. greater measurement range, can be achieved without increased complexity,

cost, or safety compromises. The coupling of light from the system to the atmosphere and the coupling of back reflected light into the system is a critical design component. Poor coupling efficiency at this point of the system's operation can lead to a much-reduced return signal, and thus, a much-reduced signal to noise ratio which reduces or possibly eliminates measurement of wind velocity at range.

System Overview

The system was developed in a laboratory located on the fifth floor of City College of New York's Grove School of Engineering, Steinman Hall (latitude 40° 49' 16.68" N, longitude 73° 56' 54.86" W). Laboratory windows were modified to allow for access to the northerly sky from inside the laboratory. After primary development, the system was moved to a vehicle for further development of portability. The system is currently located in a vehicle (custom 1998 25' Econoline Winnebago) on the campus of The City College of New York (CCNY) at 140th Street and Convent Avenue. A hatch allows the Lidar to make measurements vertically and at an angle. A rollup gate located on the rear of the vehicle allows for the Lidar to make horizontal measurements as well as at various angles (see Figure 13). The vehicle is equipped with a marine shore line to bring electrical power to experiment when the vehicle is at CCNY and also has two 3-phase generators attached to the undercarriage to allow for the running of the experiment when the shore line is unavailable. The system is secured to an optical table bolted to the bed of the vehicle to allow the user to have access to individual components during the development stage of the project (Figure 14). Figure 15 is a schematic representation of the system components and their interconnections. In this section each component will be described and relevant performance/characteristics will be presented.

Figure 13, Research vehicle containing Lidar system. Images show location of roof hatch and rear rollup gate.

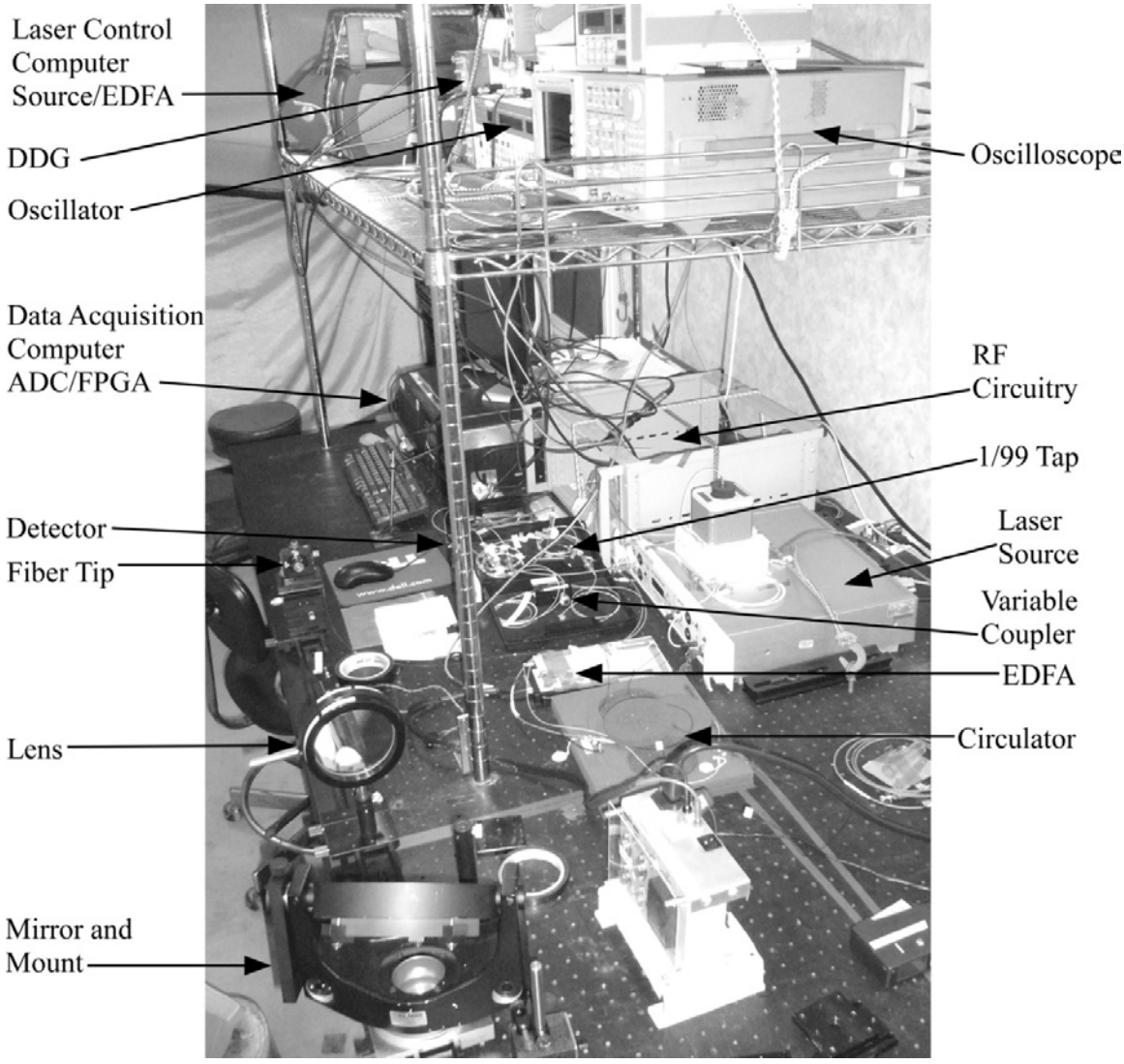


Figure 14, Doppler Lidar system components secured to optical table located in research vehicle.

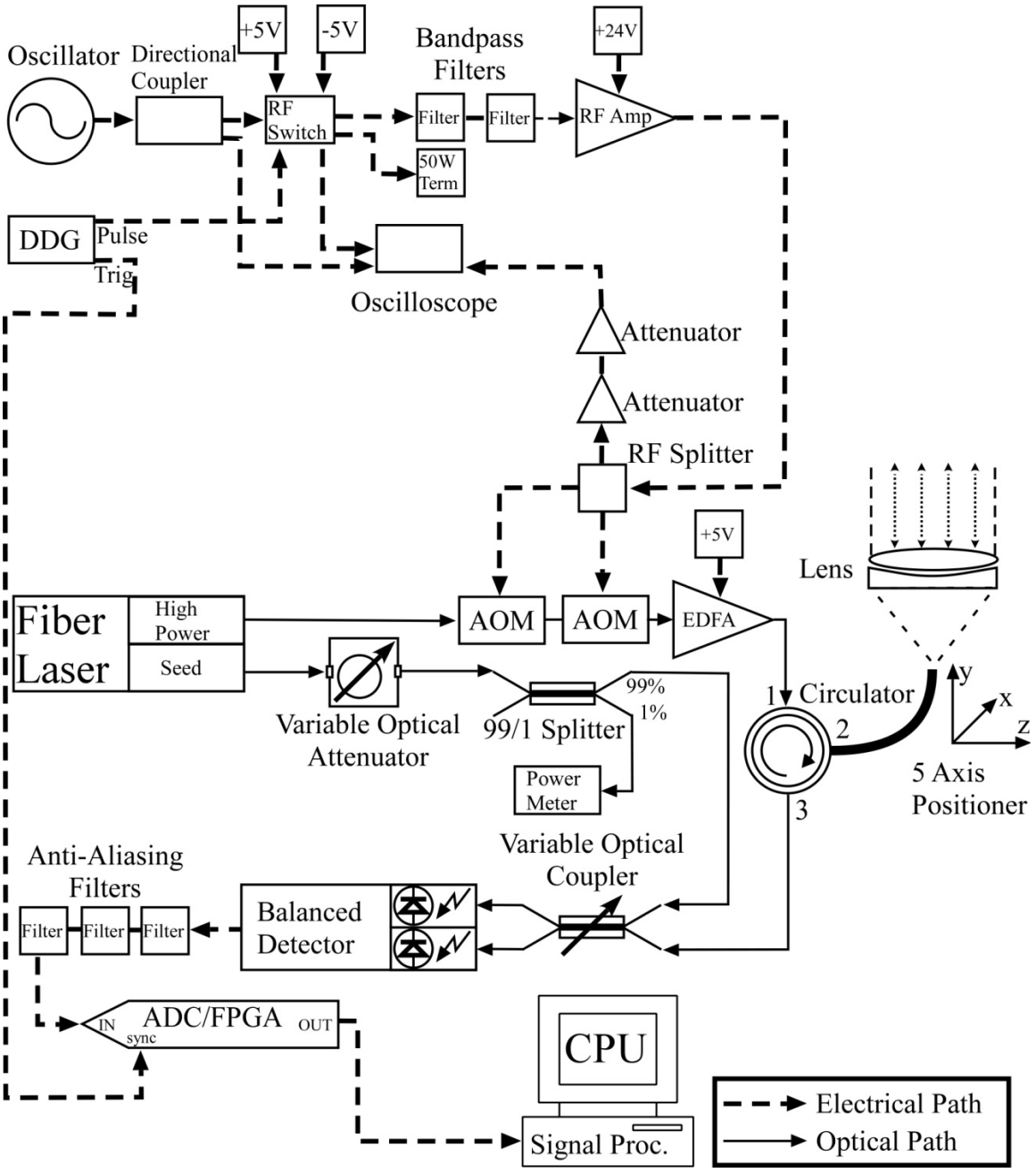


Figure 15, Doppler Wind Lidar system overview diagram. Optical connections are indicated by solid lines while electrical connections are indicated by dashed lines.

System Operation

Figure 15 illustrates the interconnection of the components in the system and the type of signals conveyed. The digital delay generator (DDG) is the unit which synchronizes the operation of the system. By generating a trigger the analog to digital converter (ADC) is initiated and by

generating a time delayed pulse of 300 ns which is sent to the RF switch the optical pulse is initiated. The DDG repetition rate is 20 kHz.

The oscillator generates a continuous sinusoidal wave at 42 MHz which is sampled by the directional coupler and fed to the RF switch. The majority of the time the RF switch conducts the 42 MHz sinusoid to a terminator. While the pulse from the DDG is high the RF switch conducts the sinusoidal waveform to the main driving circuitry. The circuitry driving the acousto-optical modulators (AOMs) must amplify the sinusoidal signal so there is sufficient driving power. The output of the RF switch is bandpass filtered before being fed into the RF amplifier to reduce unwanted frequency components. The output of the RF amplifier is split into three signals so that two AOMs may be driven and the driving signal may be monitored after being attenuated. This allows observation of the signal at a point just prior to being fed into the AOMs.

While the RF driving signal is applied to the AOMs, the frequency of the light entering the AOMs will be shifted by the frequency of the RF driving signal and passed to the output. When the RF driving signal is absent the light entering the AOMs will attenuate. The cyclic application of the driving signal creates optical pulses of frequency shifted light from the continuous wave fiber laser source. The optical pulses are amplified by a erbium doped fiber amplifier (EDFA) whose gain may be adjusted depending upon the desired output power. The amplified optical pulses are input to an optical circulator that conveys them to an input/output port of the circulator that is coupled to a lens which collimates the light for atmospheric transmission. After the short outgoing optical pulse is transmitted the lens collects the light that has been backscattered from the atmosphere and conveys it to the third port of the circulator where it is output to a variable coupler. The variable coupler combines the atmospheric signal from the circulator with light from the fiber laser source which has not been frequency shifted. This un-shifted light is known

as the local oscillator (LO). The combined signals are then split by the variable coupler and conveyed to a balanced detector. At the detector the superposition of the LO and the backscattered signal creates constructive and destructive interference resulting in a beat signal that has a frequency equal to the difference between the LO and the atmospheric signal. If the backscattered signal resulted from an interaction with a scatterer that was in motion the frequency of the beat signal will be shifted from the original frequency shift imposed by the AOMs on the outgoing light.

The frequency shift of the atmospheric signal away from the originally imposed frequency shift on the outgoing light is used to determine the speed and direction of the scatterer. To accomplish this, the detector's output is filtered before being fed into an ADC so as to avoid aliasing. The DDG trigger is used to synchronize the ADC collection of data so that some data is collected prior to the outgoing pulse being transmitted to the atmosphere, which may be used for background and noise reduction in the signal processing of the return signals.

The ADC's onboard field programmable gate array (FPGA) is used to divide the return signal into a sequence of samples representing consecutive range gates and to calculate the signal spectrum of each range gate using the Fast Fourier transforms (FFTs). The ADC's memory is used to accumulate multiple return signal spectrums before they are transferred to the host computer for additional signal processing, graphing, and storage.

The host computer interprets the spectrums transferred from the ADC to determine a sub FFT bin frequency and determine the line of sight wind velocity from the measured frequency shift. When the system is operated in a scanning mode the system is configured so three different predetermined line of sight measurements are made in a cyclic manner. The line of sight wind

velocities are converted into vertical velocities and horizontal wind speeds and directions. These data points can then be averaged and displayed as desired.

Fiber Optic

Fiber optic wave guides allow the delivery of light down a narrow cable by utilizing materials with dissimilar indexes of propagation in such a way that the phenomena of total internal reflection causes light to be confined near the center of the cable that may bend or twist with very little attenuation of optical power as it travels. They allow a design to be more resistant to mechanical vibrations and bumps than free space geometric optics. With free space propagation of beams the alignment of components is a significant concern. Moving a free space propagating type system may make it being necessary to realign the optics before measurements can be taken. A fiber optic-based construction will also be more resistant to environmentally harsh conditions. Humidity generally cannot condense on the optically active surfaces of a system that is cooled as it could in a free space type system. Temperature changes will have very little effect on the alignment and performance of a fiber based system while possibly having significant performance degrading effects on a free space based system.

Fiber optic based systems do present their own challenges. Care must be taken not to crush or stress the fibers excessively as they are often constructed of thin glass strands only 125 μm in diameter. Outer jackets and a variety of protective coatings are used to protect the fibers. Fiber optic waveguides can have significant amounts of power confined to a very small cross sectional area which may lead to unwanted nonlinear effects that can change an expected behavior or even burn the fiber, damaging it beyond use. In some applications the nonlinear effects are actually desired. [65-68]

The system developed here utilizes single mode as opposed to multimode fiber optics. The fibers utilized are also polarization maintaining as opposed to the more common fiber optics which do not preserve the polarization of the light. Multimode polarization maintaining fibers have been created in the laboratory but are generally exotic, unavailable and have properties that are undesirable in this system. [69] The selection of the single mode polarization maintaining fiber optic waveguides was made so as to enhance the performance of the system and to maximize the sensitivity of the detection techniques utilized.

Single Mode

Fiber optic wave guides can be designed to be single mode or multimode. Multimode fibers can carry more light and are easier to couple into but are more sensitive to bending and any stress applied to the fiber. The type of mode a fiber will be depends upon the indexes of refraction used for the core and the cladding, the way in which the transition from one index to the next is made, and the size of the core and cladding. Many standard fibers have a cladding diameter of 125 μm. The core of a fiber will be on the order of 8 or 9 μm for single mode fibers and 50 to 100 μm for multimode fibers. The change in the index of refraction of the center of the fiber (core) to the outer portion (cladding) can be made in an abrupt step or in a gradual transition. These are known as step index fibers and graded index or gradient index fibers, respectively. Figure 16 shows a schematic representation of the different types of index fibers.

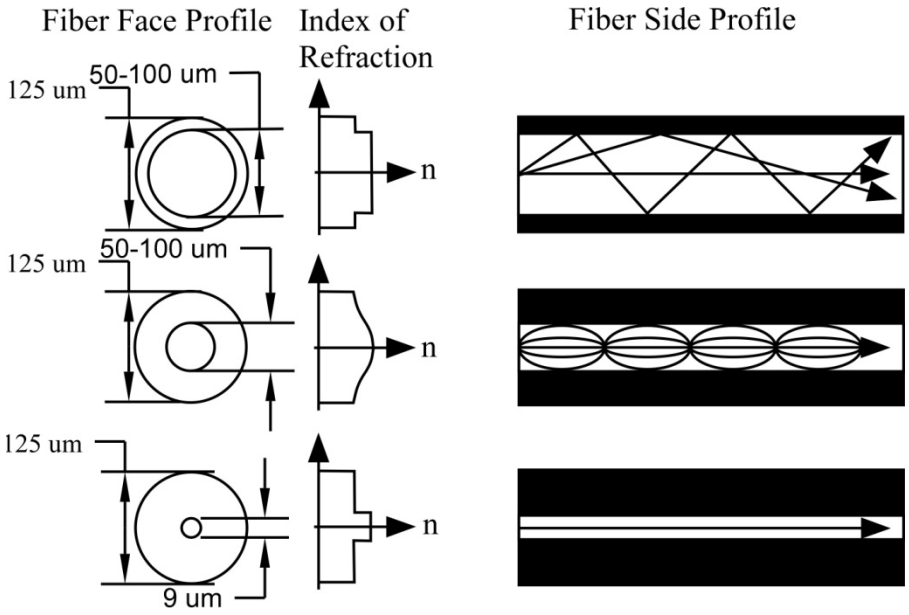


Figure 16, Schematic representation of step index and gradient index fibers, multimode step index (top), multimode gradient index (center), singlemode step index (bottom).

The index of refraction of the core and the cladding are chosen so light entering the fiber will experience total internal reflection and propagate down the fiber with the index of the core greater than that of the cladding. This can be understood by way of Snell's law, which describes the relationship between the angles of incidence and refraction when waves travel through a boundary between two layers of different indexes of refraction. In Snell's law,

$$n_0 \sin(\theta_0) = n_1 \sin(\theta_1) \tag{3.3.6}$$

where n_0 is the refractive index of the first surface, and n_1 is that of the second material. When light is incident upon a medium with a lower index of refraction than the one it is currently traveling in, the light is bent away from the normal to the interface. This makes the exit angle greater than the incident angle. At some point the incident angle will result in the exit angle exceeding >90 degrees and the light will experience total internal reflection. The angle at which this occurs is known as the critical angle. Light incident upon the interface at greater than the critical angle will experience total internal reflection.

Launching light into a fiber also involves Snell's law, the critical angle, and the indexes of refraction of the fiber and the surrounding material at the face of the fiber. Using these parameters, the angle relative to the axis of the fiber at which light striking the face of the fiber will be transmitted down the fiber by total internal reflection is known as the angle of acceptance. The shape that a line projected from the surface of the fiber sweeps out when rotated about the axis of the fiber is known as the cone of acceptance.

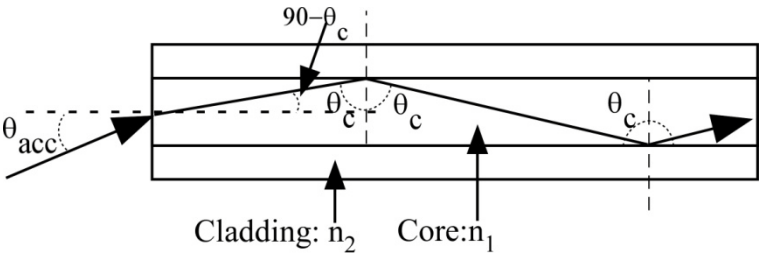


Figure 17, Profile of fiber optic's core/cladding relationship to the critical angle and the angle of acceptance.

If θ_2 is equal to the critical angle θ_c, θ_1 =90- θ_c and the index of refraction just outside of the fiber is n_0,

$$\sin(\theta_c) = \frac{n_2}{n_1} = \frac{n_{clad}}{n_{cor}} \tag{3.3.7}$$

which leads to the angle of acceptance of the fiber,

$$\theta_{acc} = \sin^{-1}\left[\frac{\sqrt{n_1^2 - n_2^2}}{n_0}\right] \tag{3.3.8}$$

Light entering the fiber at less than θ_{acc} will be transmitted down the fiber but light entering at an angle greater than that will be transmitted into the cladding and after a few reflections most of

the light will be lost. This angle of acceptance is an important parameter in the coupling of light into a fiber, and manufactures commonly state a parameter know as the numerical aperture (NA),

$$NA = n_0 \sin(\theta_{acc}) = \sqrt{n_{core}^2 - n_{cladding}^2} \tag{3.3.9}$$

For a step index multimode fiber this numerical aperture is a useful parameter. For a single mode fiber using the indexes of refraction is not a good predictor of the far field behavior of light leaving a fiber. Utilizing what is known as the mode field diameter (MFD) of the light in the fiber to predict the far field behavior is a better technique. By treating the output of the fiber as a Gaussian beam and finding the angle from the center of the beam where the intensity to $1/e^2$ (13.5%) of the maximal value the effective numerical aperture can be calculated as, [70]

$$NA_{eff} = \frac{2\lambda}{\pi MFD} \tag{3.3.10}$$

When it comes to making measurements of a beam diameter the $1/e^2$ is a common definition of the edge of a beams diameter but other parameters are used and there may not be a clear definition across manufacturers. The variations on the definition of the edge of a beam include $1/e^2$ (13.5%), 1/e (36.7%), 5% (-13dB), 1% (-20 dB), and the full width at half max (FWHM). [70-73]

The mode field diameter of a fiber is a special case of the mode field distance in a cylindrical dielectric waveguide. The field distribution across the fiber extends beyond the core to the cladding. In the core the field distribution is harmonic but in the cladding the field is due to the evanescent wave and decays exponentially. Thus the mode filed diameter is larger than the diameter of the core.

A fiber optic will propagate modes depending upon its size, shape and the wavelength and polarization of the light. The single mode fiber will only allow the transverse electro-magnetic mode TEM00 to propagate to the end of the fiber. This mode is Gaussian in shape. Multimode fibers will allow many modes to propagate and will create an output beam that is very inconsistent in intensity as seen in Figure 18.

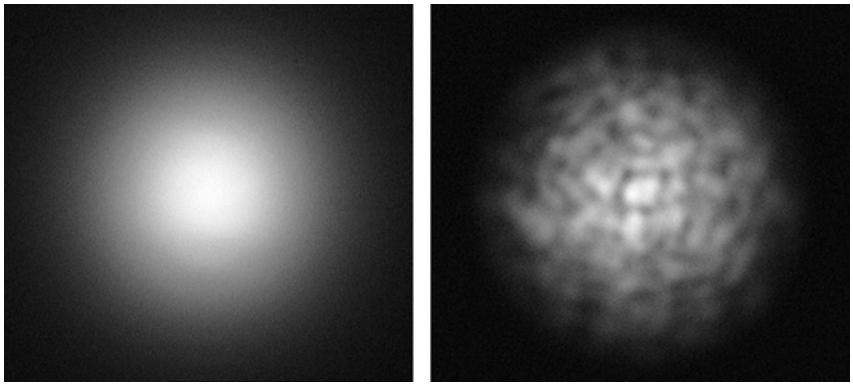


Figure 18, Images of the output beam profile of a single mode fiber (left) and a multimode (right). [74]

Movement or application of pressure to a multimode fiber can prevent some modes from propagating, making the output appear as a combination of any of a number of patterns. An example of this can be seen in Figure 19. With extreme bending of a multimode fiber the output beam profile becomes characteristic of a single mode fiber. The small radius created by bending the multimode fiber acts to destroy the higher order modes by coupling them to the cladding where they are scattered.

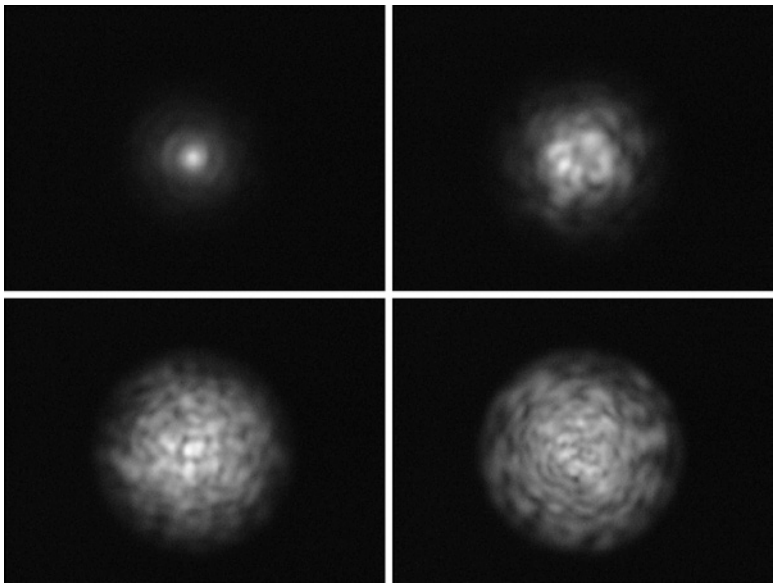


Figure 19, A variety of beam profiles of a multi-mode fiber, depending on the light in-coupling characteristics and the bending of the fiber. [75]

Measure of Mode Field Diameter

It is difficult to make a measurement of the mode field diameter (MFD) at a very close range, near field, at the tip of the fiber before it begins to expand. Instead, the beam intensity at a distance from the peak is measured at distance much greater than the radius squared divided by the beam wavelength and the mode field diameter is calculated. [76] This techniques is useful as the manufacturers of optical components often do not specify the fiber's characteristics in great detail as this information may be proprietary. They will often specify the operating wavelength, the mode, and the profile of the index of refraction for the fiber. It is uncommon for information about the size of the core or the fibers' indexes of refractions to be given.

Polarization Maintaining Fiber

If linearly polarized light were to be launched down an ideal single mode fiber with perfect rotational symmetry (circularly symmetric) it could maintain any polarization along the length of the fiber. The two degenerate orthogonal polarization modes in a single mode fiber have

identical propagation characteristics which would allow this to occur. But, if rotationally asymmetric stress to the fiber changes the refractive indices in an orthogonal polarization manner the propagation characteristics change and the orientation of the light would be continuously rotated as it traveled. The external stress can be as simple as a difference in temperature, a bend, or pressure on the fiber.

Polarization maintaining fibers preserved the polarization of the light as it travels down the fiber by strong birefringence being built into the fiber. [64, 77-81] These fibers do not polarize the light traveling down their length. Polarization maintaining fibers are not the same and should not be confused with single-polarization fibers which transmit light with a particular polarization but attenuates, through scattering or absorption, light with other polarizations. Polarization maintaining fibers' preservation of polarization is accomplished by applying stress to the structure of the fiber optic waveguide. Varieties of polarization maintaining fibers are commonly available while other exotic polarization maintaining fibers exist mainly in the laboratory. A few of these more common polarization maintaining fiber cross-sections are shown in Figure 20. The incorporation of stress applying parts (SAP) creates a fast and a slow optical axis. The SAPs induce a difference in the speed of light traveling in the fiber for two perpendicular polarizations. This birefringence intrinsic to the fiber is greater than the external asymmetric stress that may be applied. Linearly polarized light traveling down the fiber along one of these axes will remain aligned with the axis as it travels. If it were not for the strong intrinsic birefringence, coherent mode coupling between two orthogonal polarizations traveling down a fiber would be caused by a small fluctuation of the refractive index resulting in optical amplitude moving back and forth between the two modes. Since the birefringence changes the propagation constants of the two axes unequally, the relative phase rapidly changes and coupling is made inefficient.

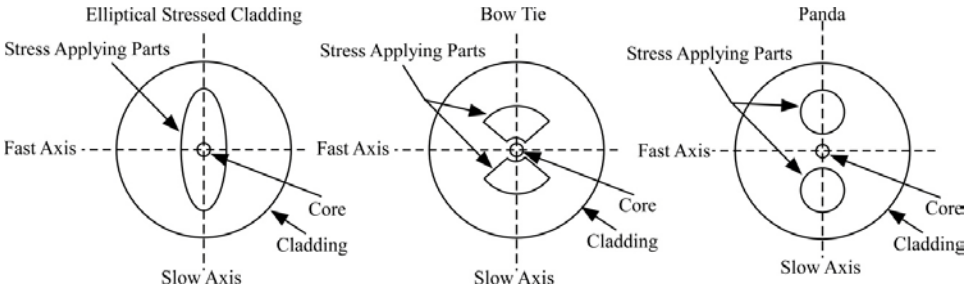


Figure 20, Cross-sections of some common polarization maintaining fiber optic waveguides illustrating variations in the cores, claddings, and stress applied parts.[77-80]

If the polarization of the light propagating down the fiber is not aligned with or perpendicular to the stress direction (fast or slow axes) of the fiber there will be cross talk. This can result in the polarization of light exiting the fiber to vary between linearly and elliptically polarizations. The polarization cross talk is defined as the ratio of the amount of light coupled to the unintended axis (P_{min}) when the source is aligned orthogonally to the unintended axis and the amount of light on the intended axis (P_{max}). This is commonly expressed in dB.

$$Polarization\_Cross-Talk = 10\log\left(P_{\min}/P_{\max}\right) \tag{3.3.11}$$

A measure of the degree or strength of a polarizer is its polarization extinction ratio (PER) which is defined as the ratio of the optical power transmitted when a polarized source is aligned with the blocked axis to when it is aligned with the pass axis of the polarizer.[82, 83]

$$Polarization-Extinction-Ratio(PER) = 10\log\left(P_{blocked\_axis}/P_{pass\_axis}\right) \tag{3.3.12}$$

It is important to launch the light into a polarization maintaining fiber along the slow or fast axis as accurately as possible. In the system under development the slow axis is used. This is the common axis to use in industry. Any deviation in the alignment of the polarization axis of the light from that of the desired axis will affect the extinction ratio of the light exiting the fiber.

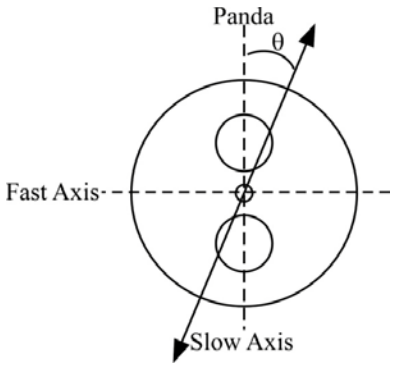


Figure 21, Panda type polarization maintaining fiber optic cross section showing slow and fast axes and misalignment of polarization axis of light launched into the fiber.

Generally the extinction ratio is expressed in dB and is the ratio of the maximum and minimum transmission.

$$Extinxtion\_Ratio(ER) = 10\log(P_{\max}/P_{\min}) \tag{3.3.13}$$

The Figure 21 shows an example of the misalignment of light being launched into the fiber and the slow axis. The maximum extinction ration achievable is, [82, 83]

$$Extinction\_Ratio(ER) \le -10\log\left(\tan^2(\theta)\right) \tag{3.3.14}$$

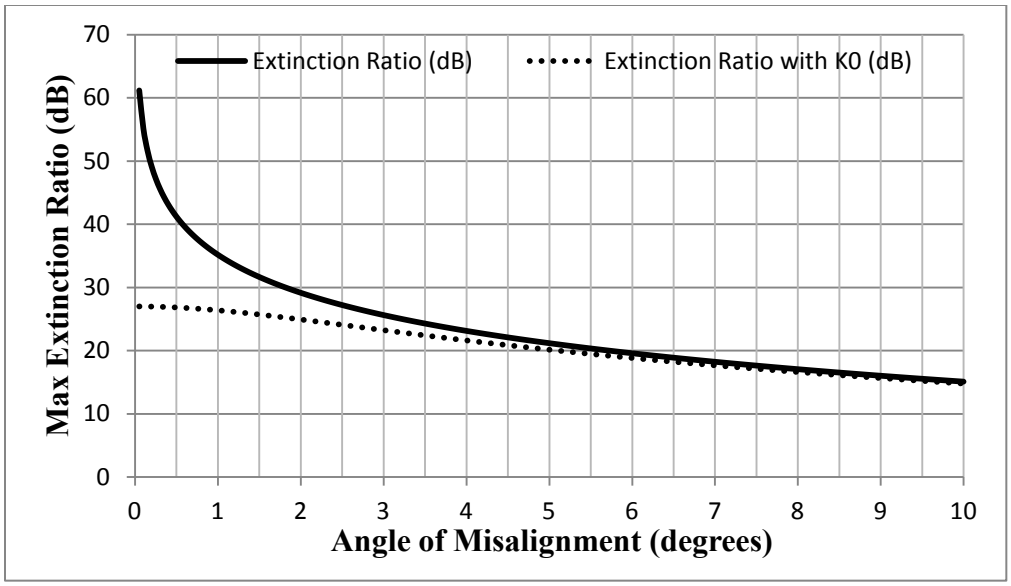


Figure 22, Angular alignment mismatch between polarized light and fiber axis and the maximum extinction ratio with and without correction factor.

Figure 22 show how quickly the maximum extinction ratio decreases with the increase in angular mismatch between the polarized and fiber axis with ideal polarization. This graph would seem to imply that the extinction ratio would be infinite if there was perfect alignment. Manufacturers of polarization maintaining fibers use a correction factor, K_0, based on experimental measurements to compensate for this error: $ER = -10\log(\tan^2(\theta) + K_0)$. [84] When the light from the source is not ideally polarized it will have an unwanted light leaking through incoherently and acting as a background signal that accumulates at the points where the polarization fibers are connected. If the extinction ratio of the source were ER_0, and the extinction ratio across the connector due to angular misalignment were ER_{con} then the final extinction ration ER_{final} would be,

$$ER_{final} = -10\log(10^{-ER_0/10} + 10^{-ER_{con}/10}) \tag{3.3.15}$$

If a source had ER_0 = 17 dB and the connector had ER_{con} = 25 dB then the final extinction ratio would be ER_{final} = 16.35 dB.

Fiber Optic Connectors

There is a variety of fiber optic connector finishes. The finish is the manner in which the tip of the fiber optic is polished flat, at an angle, or with a radius. FC type connectors can be finished flat, PC (physical contact), super PC, ultra PC, angled PC, and angled flat. The amount of back reflection from a fiber to fiber connection can be from -11 dB to -60 dB based on the finish of the connectors. See Table 2 for a comparison of finish profiles. FC/APC connectors were used in the system as they have low back reflection relative to the finishes, are commonly available and utilized by manufacturers of fiber optic based devices.

Table 2, Fiber optic finish profiles, common applications and expected return loss (back reflection). [70]

Finish Polish Type	Profile	Application	Return Loss
Flat		Multimode Fibers	-11 dB
Super PC		PC profile with enhanced polishing Standard Single Mode and PM Fibers	-40 dB
Ultra PC		PC profile with enhanced polishing: SM and PM systems needing low return loss when parts are connected	-50 dB
Angled PC (APC)		Systems needing lowest return losses at all times	-60 dB
Angle Flat (APC)		Fiber to free space delivery systems needing both low return losses and good repeatability when mated to lenses	-60 dB

The standard FC/APC connector is polished to either 8 or 9 degrees with a radius of curvature between 5 and 12 mm. The angle at the interface reduces the back reflection at the fiber/air interface. The radius at the tip help to align the cores of two similar connectors when brought together. See Figure 23 for an illustration of the APC polish profile radius and angle.

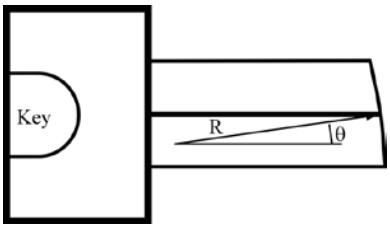


Figure 23, Profile of APC fiber tip showing angle normal to the surface at the core of the fiber and radius of curvature of interface surface.

Fiber Optic Connector Key Dimensions

Fiber optic connectors have a variety of form factors and alignment methods to assure proper joining of fibers. The FC fiber optic connectors have keyways to assure proper orientation of the fibers and connectors. They can be of the form narrow key width or wide key width. [78, 85] Typical specifications for the key and the keyway are shown in the table below. When using polarization maintaining fibers the significant of the key tolerances can become an important detail to consider.

Table 3, FC fiber optic connector typical key specifications.

Type of Key	**Key Width**	**Keyway Width**
Narrow	1.98mm - 2.02 mm	2.03mm - 2.07mm
Wide	2.09mm - 2.14mm	2.15mm - 2.20mm

In order to align the axis of polarization of the light traveling from one fiber's slow or fast axis to the next fiber's desired axis in polarization maintaining fibers the axes of the two fibers should be tightly aligned. Small amounts of error can lead to degradation of performance.

Light Emitted from Fiber Tip

When light exits a fiber optic it will spread out due to diffraction. With a mode field diameter on the order of micrometers an emerging beam will have a significant divergence. With a 10 micrometer mode filed diameter the emerging beam diameter will reach 100 mm within

approximately half a meter. When the tip of the fiber is angled the beam exiting the fiber will tilt off of the axis of the fiber. An illustration of this is shown in Figure 24.

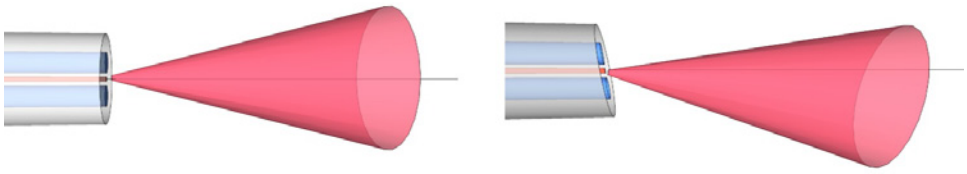


Figure 24, Illustration of beam divergence while exiting the end of a single mode polarization maintaining fiber optic which has a flat face (left) and an angled flat face (right).

The tilt of axis of a beam leaving a fiber with a tilted face can be calculated. [82, 86] The typical value for the cladding of an optical fiber (pure silica) is 1.44968 [87] and the core index of refraction calculated from the effective numerical aperture is then, $NA_{eff} = \sqrt{n_{core}^2 - n_{cladding}^2} \rightarrow n_{core} = 1.45228$. The optical axis of a beam leaving an angled fiber tip is tilted at an angle of,

$$\theta_0 = \sin^{-1}\left(\frac{n_{core}}{n_0}\sin\theta_p\right) - \theta_p \tag{3.3.16}$$

where θ_0 is the angle of the beam, θp is the angle of the face of the fiber, n_{core} is the index of the core and n_0 is the index of the medium the light is entering (see Figure 25). In this case $n_0 = 1$ for air. If the angle of the tip of the fiber were 10 degrees then the angle that the optical axis of the beam forms with the mechanical axis of the fiber is $\theta_0 = 4.60$ degrees. Figure 26 shows an angled fiber tilted so as to align the beam axis with the horizontal plane. The tilt advances one edge of the fiber along the optical axis while retarding the other edge of the fiber.

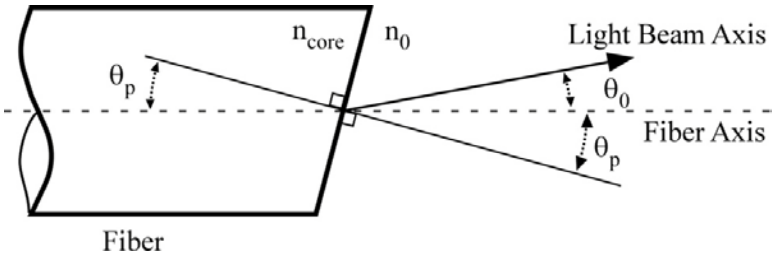


Figure 25, Schematic of angle fiber tip and angle of propagation of beam leaving an angled face.

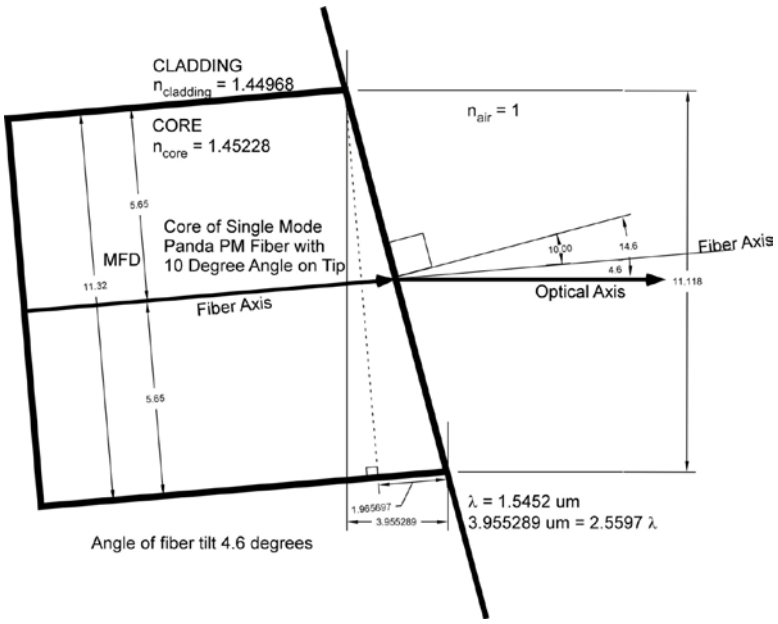


Figure 26, Tilt of fiber optic tip with end surface polished to 10 degrees so as to make the exiting beam axis horizontal.

Wavelength Selection

In order to take advantage of the available technology in the telecommunications field the wavelength selected for the system has to be available from telecommunication equipment manufacturers and have characteristics that would make the wavelength a good choice for transmission into the atmosphere. Most optically based telecommunications equipment is not designed for free space propagation through the atmosphere but for propagation through fiber optic cables where the propagation characteristics such as attenuation, dispersion, polarization are more easily measured or supplied by fiber optic manufacturers. The behavior of common

telecommunication wavelengths in the atmosphere and their safe use in public spaces must also be considered.

Telecommunication Wavelength Bands

International Telecommunication Union (ITU) is the United Nations specialized agency for information and communication technologies. This group makes recommendations on communications and issues standards. For the fiber optic communication industry they have recommended different bands of wavelengths [88-90]. See Table 4 for the single mode spectral bands recommended by the ITU. These bands set the range of possible wavelengths that manufactures would commonly use in producing equipment. The ITU also prescribes multiple channels within these bands.

Table 4, International Telecommunication Union recommendations for single mode spectral bands.

Band	Description	Range (nm)
O-Band	Original	1260-1360
E-Band	Extended	1360-1460
S-Band	Short Wavelength	1460-1530
C-Band	Conventional	1530-1565
L-Band	Long Wavelength	1565-1625
U-Band	Ulra-Long Wavelength	1625-1675

Atmospheric Propagation

If the wavelength selected were commonly available from manufacturers and safe to use in public areas but had very high absorption in the atmosphere or did not interact with the atmospheric aerosols it would be a poor selection. Figure 27 illustrates the total absorption of the atmosphere and a number of major components of the atmosphere that are optically active. In the wavelength bands of the telecommunication industry water vapor is the major component of absorption.[91] The water vapor absorption for the region near 1545nm was plotted using Matlab

and Hitran data to produce Figure 28. The wavelength of 1545.2 nm was chosen for the system as there was good transmission at that wavelength and it has some good characteristics for safe use in public areas.

This wavelength is also near some absorption peaks. In the future the neighboring peaks could be used to help adjust the system. By tuning the laser from 1545.2 nm to the absorption peak at 1544.4 nm the change in the return signal might be useful in calibrating the system as there would be a change in performance relative to a known absorption change.

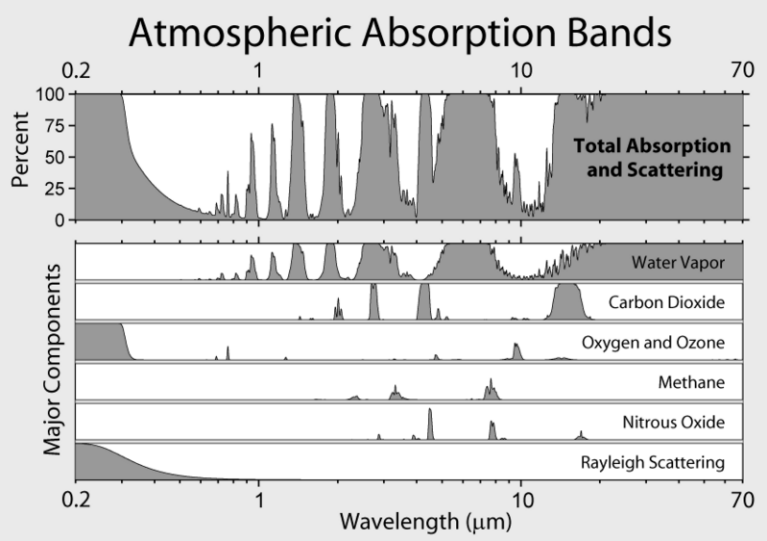


Figure 27, Atmospheric absorption and scattering by major components. [92]

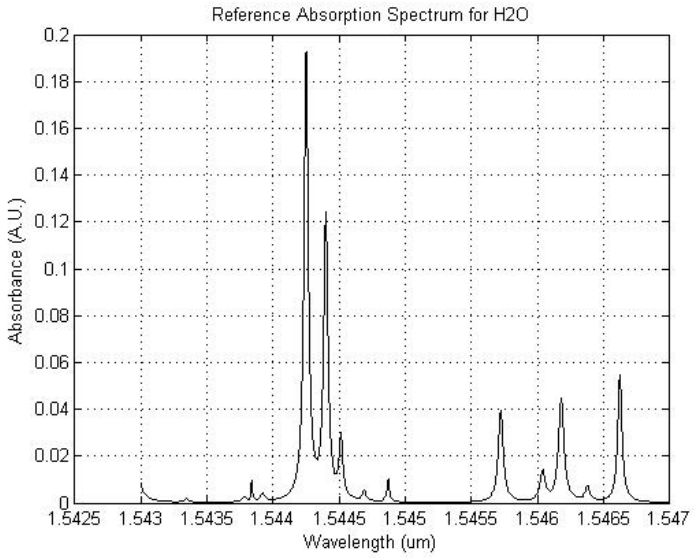


Figure 28, Absorption spectrum of H2O near 1545nm.

Laser Eye Safety

In order to operate a laser safely the American National Standards Institute (ANSI) has published a number of standards. The Safe Use of Lasers Outdoors (ANSI Z136.6) and National Standard for Safe Use of Lasers (ANSI Z136.1) are the two most applicable references for the system developed here. [62, 63] The primary concern with the use of a laser is the potential damage it can do to a person's vision but there are other dangers.

The non-ocular hazards are chemical, electrical, tissue damage, explosive and fire. Some lasers use solvents or compressed gases that may be toxic, carcinogenic or explosive. A laser beam's interaction with a material may produce fumes which are dangerous or cause the material to ignite. Many lasers use very high voltages that can present an electrocution hazard should a fault develop or a safety interlock be bypassed. These high voltage supplies also contain capacitors that can explode. Some lasers are capable of burning a person's skin. Aside from causing permanent damage to a person's vision, other ocular hazards include flash blindness similar to the afterglow seen with flash photography. This can be very hazardous to pilots of all manner of

vehicles, not just aircraft. [93-100]. Figure 29 illustrates the effects of various laser beam intensities can have on an aircraft pilot of a Boeing 727 as it approaches an airport to land.

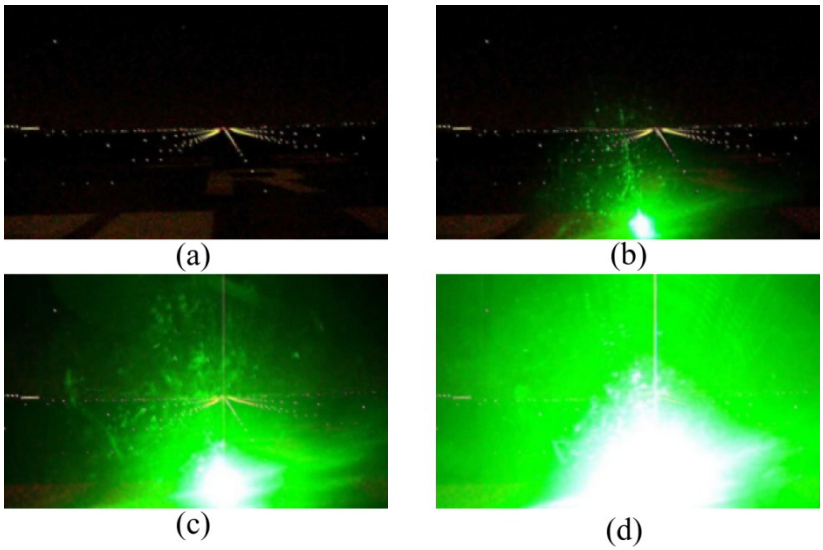


Figure 29, Laser illumination of Boeing 727 aircraft cockpit. (a) No laser exposure, (b) 0.5 $\mu W/cm^2$ or 5 mW laser pointer at 3,700 feet (1,128 m), (c) 5.0 $\mu W/cm^2$ or 5 mW pointer at 1,200 feet (366 m), (d) 50 $\mu W/cm^2$ or 5 mW pointer at 350 feet (107). [101]

Ultraviolet and infrared light does not penetrate to the retina very effectively but can damage the cornea or the lens. Visible light by definition, reaches the back of the eye where the retina is located and can possibly damage it. As the lens of the eye focuses the light on the retina the irradiance can be significant. A 2 mW laser, typical in a laser pointer, can be focused down to a spot of only 16 μm in diameters where the irradiance would be 1000 w/cm^2. The fact that the light is visible will generally cause a person to avert their stare should a visible laser beam hit their eye. The length of exposure is part of the calculation of determining the laser hazard. Light just past the visible light spectrum would not cause the person to look away but may still be transmitted to the retina and therefore has lower limits. The process of determining safe levels of exposure is a complicated undertaking. Characteristics that must be considered include the

wavelength, output energy and power, duration of exposure, size of irradiated area, and pulse or continuous wave operation.

The ANSI standards classify lasers based upon the power levels and uses. A laser's class determines the degree of precautions necessary. A laser found in a Blu-ray optical disc player has a wavelength of 405nm and can be of 250 mW in power. The ANSI standard would classify this as being 25,000 times the exposure limit as it is visible and powerful, but since it is contained within an enclosure that prevents accidental exposure. Since the Doppler wind Lidar system here is designed for outdoor use, attention must be given to the safe use of the system in public areas. It is not possible to contain the beam in a manner that would assure someone could not accidently observe the beam and still allow it to make wind measurements. The ANSI standards call for knowledgeable and trained personnel known as a Laser Safety Officer (LSO) to evaluate the hazards of systems. [102]

The maximum permissible exposure (MPE) is the level of laser radiation that will not have an adverse biological effect or hazard. The nominal hazard zone (NHZ) is the area in which the level of radiation, either direct, reflected, or scattered, exceeds the MPE. Beyond the NHZ the levels are bellow the MPE. The accessible emission limit (AEL) is the maximum emission level permitted within a class of laser. The classes of lasers consist of Class 1, Class 2, Class 3A, Class 3B and Class 4. Class 4 is the higher power, more hazardous class of lasers. Class 1 systems can be used for unattended operation in unsupervised areas without additional control measures. For the ANSI standards, when performing calculations to determine the class of a laser the beam's diameter, the edges of the beam, is defined as the level of intensity that is 1/e that of the peak.

When considering an exposure time of 10s the greatest MPE levels are for wavelengths of 1500 nm to 1800 nm. The MPE for this range is 1.0 J/cm^2, 0.1 W/cm^2. The AEL for 1400 nm to 10,000 nm is: Class 1: 9.6 mW, Class 3: > 9.6mW to < 500mW, Class 4: > 500mW for continuous wave of repetitive pulse. For a sing pulse the classes are: Class 1: 79 μJ to 7.9 mJ, Class 3b: > 7.9 mJ to 125mJ, Class 4: > 125mJ.

To determine the class three rules must be examined. The rules are,

> "Rule 1: Single Pulse MPE
>
> The exposure from any single pulse in a train of pulses shall not exceed the MPE for a single pulse of that pulse duration. (Rule 1 protects against thermal injury from any single pulse having greater than average energy.)"
>
> "Rule 2: Average Power MPE for Thermal and Photochemical Hazard.
>
> The exposure from any group of pulses (or sub-group of pulses in a train) delivered in time T shall not exceed the MPR for time T. That is, the average power for any duration shall not exceed the MPE of a single pulse of train duration T and the total energy of all pulses within the train. Depending upon the complexity of the pulse train, the calculation of several potential MPEs (for different pulse groupings) may be required. The calculation of Rule 2 usually provides a lower MPE value for lasers with a high duty cycle than applying Rule 3 below. (Rule 2 protects against cumulative injury from photochemical damage mechanisms and also against heat buildup from average power for thermal injury.)"

"Rule 3: Multiple-pulse MPE for Thermal Hazards.

The exposure from any single pulse within a group of pulses (each separated by at least tmin) shall not exceed the single pulse MPE (for t> tmin) multiplied by a multiple-pulse correction factor Cp. The multiple-pulse correction factor is n-0.25, where n is the number of pulses (See Figure 13 for a graphical representation of n-0.25). Rule 3 applies only to MPEs for thermal injury, where all pulses delivered in less than tmin are treated as a single pulse. (Rule 3 protects against sub-threshold pulse-cumulative thermal injury.)

NOTE: For repetition rates which are so high that multiple pulses occur in a time frame less than tmin, pulse energies delivered within those time frames are summed directly. It is assumed that the energy within tmin act as if it were delivered in a single pulse.

The methods of applying the three rules for determining the MPEs for repetitive laser exposures for specific spectral bands and determination of tmin are given in sections 8.2.3.1 to 8.2.3.3 especially. See section 8.2.3.3 for proper apertures to use with each rule."

The system under development has the following characteristics, Wavelength = 1545 nm, Pulse repetition rate = 20 KHz, Pulse Period = 50 us, Pulse length = 200 ns/pulse, Max Pulse Energy = 14.7 uJ/pulse, Average power = 14.7 uJ/pulse * 20000 pulses/sec = 294 mW, Peak Power = 14.7 uJ/pulse / 200 ns/pulse = 73.5 W, lens diameters 10.0 cm.

The beam area is $\pi(5\text{ cm})^2 = 78.5\text{ cm}^2$. The diameter of the eye aperture is 3.5 mm, making the area 0.0961 cm^2. So the MPE multiplied by the area = 1 W/cm^2*0.0961 cm^2 =9.6 mW.

Rule 1

For a Gaussian beam the fraction of total power or energy transmitted though a measurement aperture, Dm, is

$$\frac{\phi_d}{\phi_0} = \frac{Q_d}{Q_0} = 1 - e^{-\left(D_m / D_L\right)^2} \tag{3.3.17}$$

where D_L is the beam diameters. For a 100 mm lens and a pulse energy of 14.7 uJ/pulse,

$$\phi_d = \phi_0 \left(1 - e^{-\left(D_m / D_L\right)^2} \right) \tag{3.3.18}$$

ϕ_d = 17.9 nJ. This is well below the upper limit of a Class 3b so it is a Class 1.

Rule2

For a 10 second exposure the number of pulse is n = 200,000. Therefore

$$\phi_d = n\phi_0 \left(1 - e^{-\left(D_m / D_L\right)^2} \right) \tag{3.3.19}$$

ϕ_d = 3.59mJ or an average power of 0.359 mW. Since this does not exceed the AEL of 9.6 mW it is Class 1.

Rule 3

A correction factor, $Cp = n^{-0.25}$, is used to calculate Rule 3. For a 10 s exposure n = 200,000 and Cp = 0.04728. So the AEL for Class 1 is 7.9 mJ*Cp = 373.5 μJ/pulse. The system is Class 1 according to rule 3.

The calculations show that the wavelength of 1545.2 nm should be a good choice for the Lidar systems as it appears to have good transmission through the atmosphere is available from manufacturers, and a relatively good low laser hazard when operating in a public area.

Components

Fiber Laser Source

NP Photonics Rock high power single frequency narrow line width fiber laser source Model RFLSA-500-1-1545.2-PM-S-NSI, S/N J10-000780-379 (see Figure 30).

Seed laser has output to a FC/APC connector (narrow key), while high power output to a FC/APC fiber optic pigtail (narrow key). Both employ polarization maintaining fiber optic cables. Polarization extinction ratio (PER) >17 dB, side mode suppression ratio (SMSR) > 50 dB, signal to ASE noise ratio (integrated) > 45 dB, signal to ASE noise ratio (50 pm, bandwidth) > 55 dB. The output of the high power is specified to be 500mW while the seed output is 159mW. This supplies sufficient optical power for the local oscillator from the seed source and sufficient optical power to be fed into the modulating components and optical amplifier downstream of the high power output. The wavelength is tunable between 1530nm to 1565nm. The spectral line width is < 5kHz. The relative intensity noise (RIN) was measured by the manufacturer to peak at relaxation oscillation frequency of 2.3 MHz at a level of -113 dB/Hz.

See Figure 31 and Figure 32 for RIN spectrums supplied by manufacturer. The RIN is a measure of the laser's power instability.

Figure 30, NP Photonics Rock high power single frequency narrow line width fiber laser source.

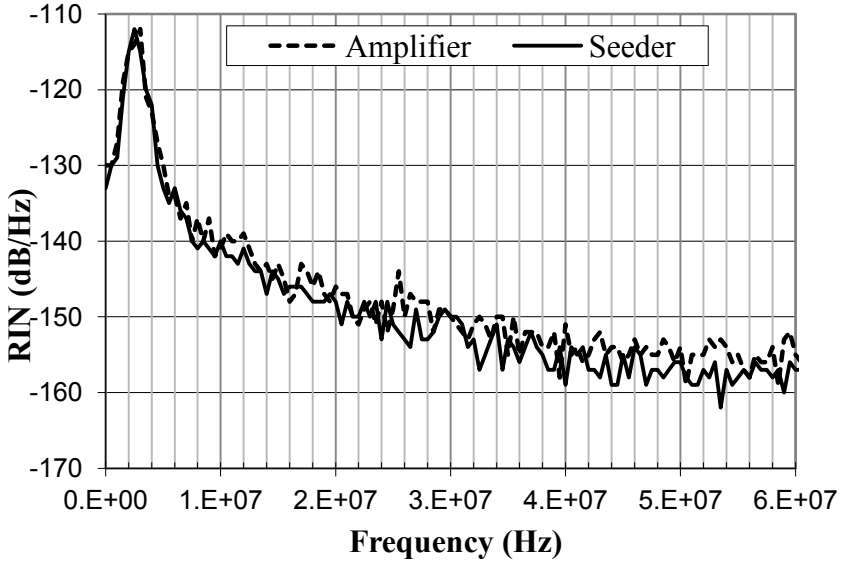


Figure 31, Close up of relative intensity noise (RIN) spectrum for NP Photonics Rock high power single frequency narrow line width fiber laser source.

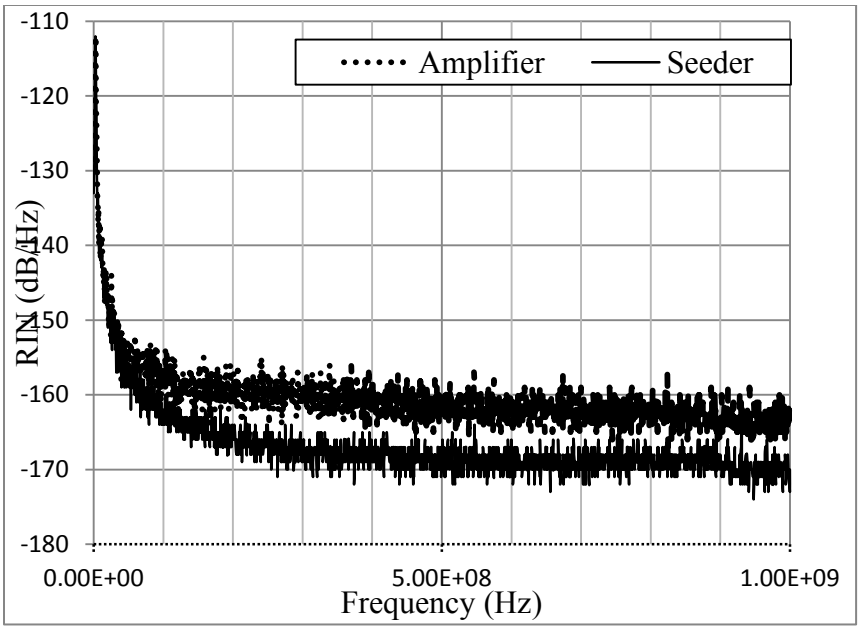


Figure 32, Wide view of relative intensity noise (RIN) spectrum for NP Photonics Rock high power single frequency narrow line width fiber laser source.

The temporal coherence length, or longitudinal coherence refers to the relative phase of two waves at a distance along the propagation direction. If the two waves were coherent, exactly in phase, at the beginning of the propagation and remained at least partially in phase for a distance, that distance is the coherence length. When the waves become 180 degrees out of phase they are no longer coherent. The coherence length can be found by [103, 104]

$$I_c = \lambda\left(\frac{\lambda}{\Delta\lambda}\right) = \frac{\lambda^2}{\Delta\lambda} \tag{3.3.20}$$

where λ is the average wavelength and $\Delta\lambda$ is the difference in wavelengths. The manufacturer specifies a linewidth of < 5 kHz. With an average wavelength of 1545.2 nm the coherence length is calculated to be I_c = 59,691 meters.

For coherent detection, which is used in this system, to be effective coherence is necessary over a length equal to twice the range of the farthest measurement. The current system uses a pulse

repetition rate of 20 kHz which makes the round trip distance 15 km and the maximum range of a measurement 7.5 km. This is well within the coherence length of the source. The Lidar system can be operated at a repetition rate of 10 kHz for a maximum range of 15 km. The current system uses the 20 kHz repetition rate.

Variable Attenuator

OZ Optics Variable Optical Attenuator (see Figure 33) ,P/N BB-100-11-1550-8/125-P-60-3A3A-3-1-ER=25, S/N 115903-02, 24 mm housing, fiber: 1550 nm, 8 µm core/125 µm cladding, polarization maintaining, extinction ratio: 25 dB, back reflection: -60 dB, connectors: angled NTT FC/PC, fiber jacket: 3 mm OD Kevlar reinforced PVC, input and output port fiber length: 1m.

The unit is connected to the seed output of the laser source and allows the power level of the local oscillator to be adjusted to a desired level. The level of optical power sent to the variable couple can be monitored by the 1/99% tap. The mechanical nature of this attenuator allows it to handle the power levels coming from the laser seed.

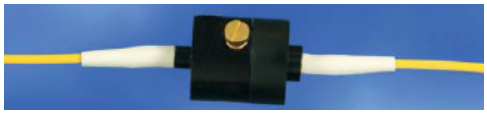


Figure 33, OZ Optics variable attenuator with polarization maintaining fiber optic cable and FC/APC connectors

1/99 Polarization Maintaining Splitter

Canadian Instrumentation and Research, 1/99 % polarization maintaining, single mode tap coupler/splitter, model: 954PS, S/N: 6379-1.

PortA to PortX = PM to PM, 99% (Datasheet measurement = 99.05% (-0.04 dB))

PortA to PortY = PM to SM, 1% (Datasheet measurement = 0.95% (-20.22 dB))

Connectorized with FC/APC connectors on ports A, X, and Y

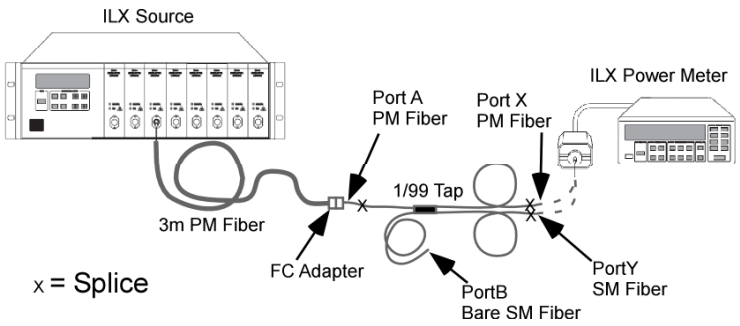


Figure 34, Test setup of 1/99% polarization maintaining single mode fiber optic tap coupler/splitter.

Setup 1: Output of 3mPM Fiber = 14.979 mW (11.750 dBm)

Setup 2: Output of 1/99% Tap PortX = 13.800 mW (11.399 dBm)

PortY = 131.23 μW (-8.819 dBm)

The percentage of power sent to each output port was calculated as follows,

$$Port\% = \frac{Port(mW)}{PortX(mW) + PortY(mW)}(100) \tag{3.3.21}$$

Table 5, Summary of high and low power measurements of fiber optic 1/99% polarization maintaining, single mode tap coupler/splitter.

Input PortA		Output PortX		Output PortY		Z = PortX + PortY		% of Total PortX	% of Total PortY
(mW)	(dBm)	(mW)	(dBm)	(mW)	(dBm)	(mW)	(dBm)		
14.979	11.7548	13.800	11.3988	0.13123	-8.8197	13.93123	11.43989	99.058	0.942

Figure 34 show the setup used to test the splitting ratio of the 1/99% polarization maintaining, single mode tap coupler/splitter. Table 5 shows the splitting ratio to be similar to the manufacturer supplied measurements.

Power Meter

Figure 35, Fiber optic power meter by ILX, model FPM-8210 with integration sphere.

ILX Lightwave FPM-8210 Fiber Optic Power Meter with measurement head (integration sphere), +20 dBm to -70 dBm, S/N 82101420, Head S/N 821A013E.

This is an optical power meter with an integration sphere designed for the measurement of optical power exiting from a fiber optic.

Variable Optical Coupler

Canadian Instrumentation Variable Ratio Coupler, Model 905P, S/N 6379-2 (see Figure 36) with fiber type: PM 15-U25A Corning PANDA

Figure 36, Canadian Instrumentation polarization maintaining single mode variable fiber optic coupler

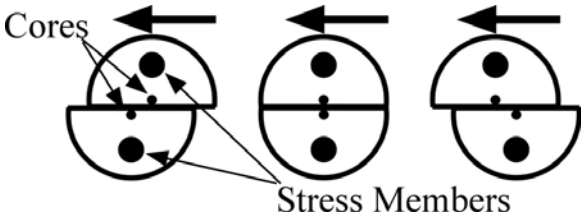


Figure 37, Variable coupler's polished fiber movement from minimally coupled to maximally coupled to minimally coupled (read from left to right).

The variable coupler adjusts the proportion of the input from one port to the two output ports by means of a micrometer that moves a PM fiber that has been polished down to near the core across another PM fiber that has also been polished down to near the core. The adjustment of the output coupling ratio utilizes evanescent waves. The micrometer is marked with English increments of 0.001”. The range of settings spans from approximately 0.050” to 0.235”. The variable coupler allows the local oscillator power incident on to the detectors to be balanced.

The manufacturer of the coupler states that the micrometer is not an absolute indicator of the coupling ratio and that due to backlash and hysteresis the ratio of the output ports should be measured. If, while turning the micrometer clockwise the desired stopping point is missed and the micrometer is then turned counter clockwise, the value measured will not be the same as if the micrometer had not overshot the desired value. This presented a problem for setting the ratio of the coupler. If a desired setting is overshot the micrometer must be turned in the other direction far past the setting point desired so that the micrometer can be adjusted again to the

desired setting from the original direction. In this test the micrometer was turned counter clockwise (from 0.050" toward larger values) for all measurements.

Most of the optical power presented to a port is guided to the two output ports in a proportion that is determined by the micrometer setting. A small portion of the optical power is not guided to the output ports but reflected to the second input port. The test configuration is shown in Figure 38 and has the NP Photonics high power output not active, nor the RF source, RF amplifier, or Keopsys Fiber amplifier.

The NP Photonics seed output was connected to the OZ Optics optical attenuator which was then connected to the Canadian Instrumentation 1/99 tap coupler. The 99% port of this coupler was connected to the Canadian Instrumentation variable ratio coupler and its outputs were both connected to the Terahertz Technologies balanced detector. This detector output was connected to a Tektronix oscilloscope. The oscilloscope was used to measure the voltage output of the balanced detector as the variable coupler's micrometer was varied.

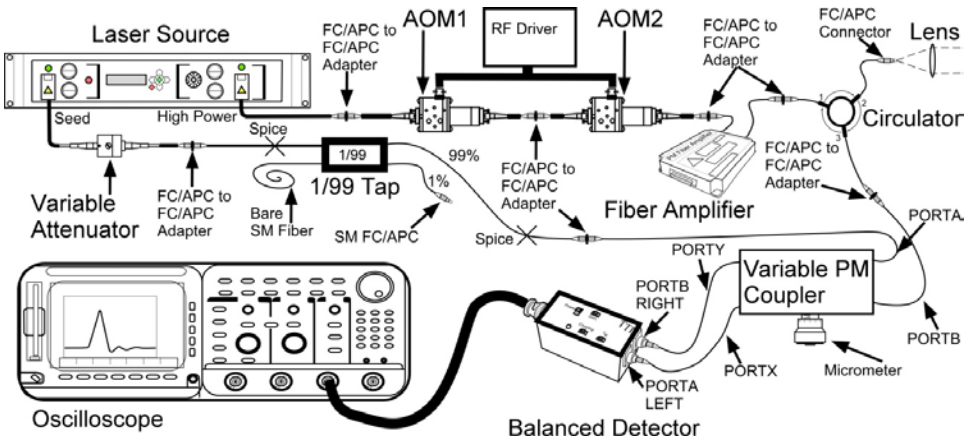


Figure 38, Variable coupler test configuration. Inactive components include high power output of laser source, RF AOM driver circuitry, and fiber amplifier.

The balanced detector was set to a gain of 1x, DC coupling, and transimpedance resistance Tr of 1.4k. The Variable Coupler's micrometer was turned counter clockwise from a minimal setting

of 0.050” to a maximal setting of 0.235”. The voltage readings were made on the Oscilloscope with the mean measurement reading on channel 3 with input impedance of 50Ω. The display vertical resolution was set to 500 mV/div while the horizontal resolution was set to 100 μs/div.

Figure 39 illustrates a typical measurement made with the oscilloscope around the region of 0 mV output from the Balanced Detector. The oscillation appears to be approximately 2 MHz in frequency and the oscilloscope measured a peak to peak voltage of 3.16 mV. Note that figure’s horizontal and vertical scale was not the scale used to make most of the measurements.

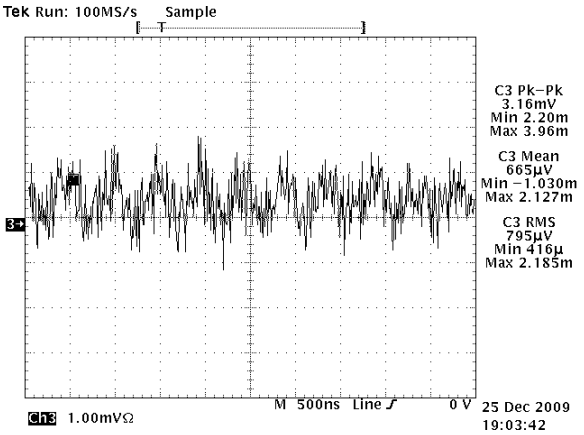


Figure 39, Typical output waveform seen on oscilloscope when balanced detector output was approximately zero volts after adjusting the variable ratio coupler's micrometer. The oscillation seen in the noise is approximately 2 MHz.

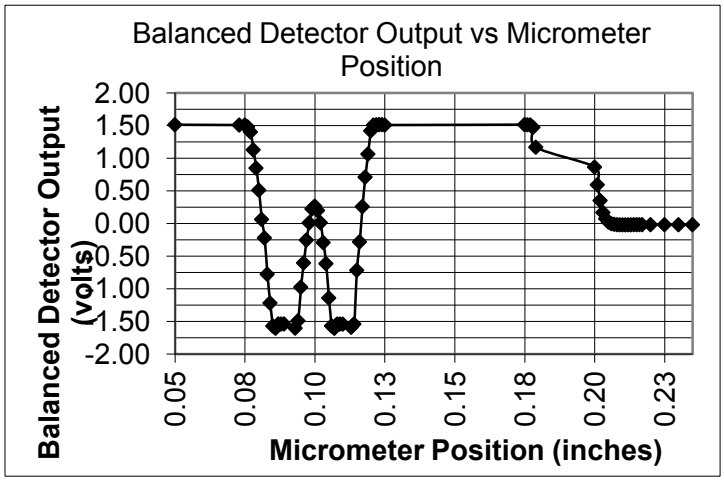


Figure 40, Balanced detector voltage output versus variable coupler's micrometer setting.

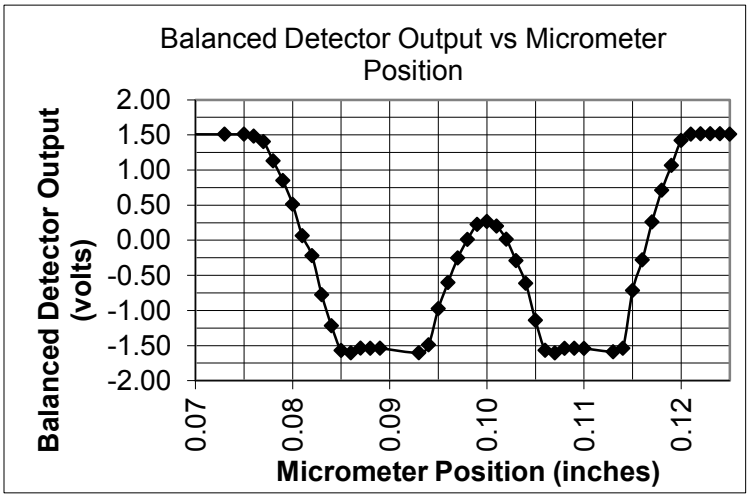


Figure 41, Balanced detector voltage output versus variable coupler's micrometer setting. Close up view of area with output variation.

Table 6, Coupler Input/Output measurements with balanced detector output approximately 0 mV

Canadian Instrumentation Variable Coupler									
Input		**Output**		**Output**		**Total Output**		**Loss**	**Insertion**
PortA	**PortA**	**PortX**	**PortX**	**PortY**	**PortY**	**PortX + PortY**			
(mW)	**(dBm)**	**(mW)**	**(dBm)**	**(mW)**	**(dBm)**	**(mW)**	**(dBm)**	**(mW)**	**(dB)**
6.3095	7.9999	1.702	2.310	1.608	2.065	3.310	5.199	2.998	2.759

Figure 40 and Figure 41 are a graph of micrometer settings versus the voltage output from the balanced detector. Table 6 shows the input and output optical powers measured and calculated with the use of the ILX Power Meter for the variable coupler. The 1/99 Tap's single mode 1% port (PortY) was measured with the ILX Power Meter and found to be -12.012 dBm (0.063095 mW). This would indicate that 6.3 mW was input to the variable coupler.

The 2 MHz oscillation seen on the output of the Balanced Detector (Figure 39) may originate from the NP Photonics Laser Source as there is noise specified at this frequency by the manufacturer. The anti aliasing bandpass filter set, which will be placed on the input to the analog to digital converter, should eliminate this 2 MHz signal from the data collected.

If the micrometer were to change the coupling in a monotonic fashion one would expect the output of the detector to transition from a negative voltage output to a positive voltage output as the micrometer was repositioned. The peak seen near the 0.100" position is the result of the input to the detector becoming unbalanced. This is being caused by overcoupling of the input fiber. The manufacturer explains the overcoupling as:

> "The tunable coupler operates by varying the core separation between fibers; as the fiber cores approach, coupling increases. If the coupling increases to 100% before fiber core separation is minimized, the phenomenon called overcoupling occurs. Overcoupling causes light to couple back into the original fiber, hence reducing the coupling ratio. It is possible to have several overcoupling cycles for long interaction length and small separation."[105]

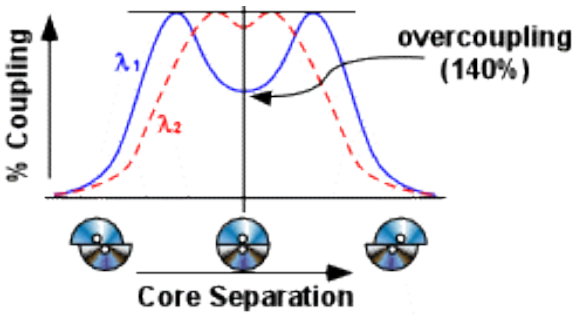


Figure 42, Example of coupling and overcoupling as a function of core seperation and wavelength (from manufacturer). [105]

The back reflection of the variable coupler was measured by inputting optical power on input port A and measuring the optical power seen on input port B. The isolation varied with the position of the micrometer. The minimal isolation was 60.8 dB and was measured near the micrometer position of 0.100" which is close the position of 50/50 splitting ratio. The maximum isolation was approximately 68.5 dB and was seen when the splitting ratio of the variable coupler was unbalanced.

Balanced Detector

Terahertz Technologies TIA-527 Balanced Optical/Electrical Converter (see Figure 43), model: TIA-5271-FC, S/N: 08006, maximum optical input without damage: 10 mW, maximum optical input with linear output: 1.2 mW, detector: InGaAs 900-1700nm, transimpedance T_R= 1.4K/14K selectable, post amplifier gain = 1.0/10.0 selectable, bandwidth (-3 dB) = DC-125MHz at 1.4 K transimpedance, output impedance = 50 ohms BNC, fiber optic connector: FC , numerical aperture = 0.29, maximum output voltage = 4 V pk-pk (no load) 2 V pk-pk, 50 ohm load (AC coupling), Noise Level 3.6 pW/Hz^(1/2), output offset voltage < +/- 0.1 V at max gain, size: 2.5x1.2x1.5”, operating temperature: 0-40 deg C, weight: 5 oz., power = 9v lithium battery (30Hrs Operation) or external power supply.

Figure 43. Terahertz Technologies Inc. fiber coupled balanced detector model TIA-527.

Acousto-Optic Modulator

AOM1: Neos technologies (a Gooch & Housego Company) Model: N23042-1-1.55-LTD-FO-PM-FCAPC-HPCAP, S/N:116003, operating frequency: 42MHz, throughput loss (excludes connector loss): 2.0 dB, 3.00 dB max, static contrast ratio: 55dB, return loss (back reflection) <= -40dB, rise time: 59ns, RF power level: < 2.0W,fiber optic connector type: FC/APC, polarization: linear, fiber model number: LPC-03-1550-8/125-P-0.44-2AS-40-3A-3-1-HPC S/N:105178-01 input, 105178-02 output.

AOM2: Neos technologies (a Gooch & Housego Company) Product = 23042-1-1.55-LTD-FO-PM-FCAPC,-CAP-PE, S/N:135461, operating frequency: 42MHz, throughput loss (excludes connector loss): 2.0 dB, 3.00 dB max, static contrast ratio: 55dB, rise time: 49ns, RF power level: < 2.0W,fiber optic connector type: FC/APC, polarization: linear, fiber model number: LPC-03-1550-8/125-P-0.44-2AS-40-3A-3-1-HPC S/N:1196113-01 (Non HPC) input, 1196114-01 output

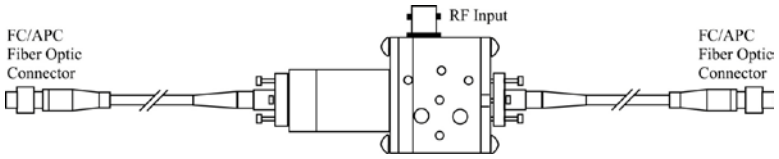


Figure 44, Acousto-optic modulator with polarization maintaining (PM) single mode (SM) fiber optic input/output with FC/APC connectors.

An acousto-optic modulator (AOM) can be constructed to amplitude modulate the beam intensity passing through the AOM in addition to shifting the frequency. Acousto-optic interactions occur in an optical medium as a result of an acoustic wave generating a refractive index wave that behaves like a sinusoidal grating and a laser source interacting.[106-108] Tandem AOMs are used instead of a single AOM in order to increase the contrast ratio. Each AOM will act to block the light from passing through when the RF signal is not present by at least 55dB. In tandem the AOMs would effectively have a contrast ratio of 110 dB. There would be the additional loss due to each device's insertion loss, excess loss, and loss due to connectors. AOM1 contract ratio was measured to be 63 dB and to have a throughput loss of 3.7 dB which includes connector loss. Assuming AOM2 has the same characteristics as AOM1 the tandem combination of AOMs would have a contrast ratio of 126 dB with a throughput loss of 7.4 dB. The penalty for using two AOMs to increase the contrast ratio, i.e. reduce the light leaking past the modulation portion of the system, is having to increase the optical input power to the AOMs by more than 2.34 times

to have the same the same optical output from the use of a single AOM. With the high power laser source having a 500 mW output level there is sufficient power to allow for the use of two AOMs. The reduction in light leaking through to the optical amplifier from the AOM section is worth the penalty as the optical amplifier will amplify the light leaking through to it causing a condition where there is a continuous wave source being sent out to the atmosphere. This would create a constant return signal that could interfere with detection of the return signal. Additionally the leaking light would constantly act to deplete the amplifier's available power for the amplification of pulses.

A LabView program was written to test the AOMs. The program recorded the output of the AOMs when driven with a range of frequencies from 37 MHz to 48 MHz in 100 kHz step sizes. The program made multiple measurements and averaged the results. AOM1 was found to have a peak throughput at 41.8 MHz and AOM2 and had a peak throughput at 41.12 MHz (see Figure 45). Even thought the units were specified for 42.0 MHz the small deviation is not significant.

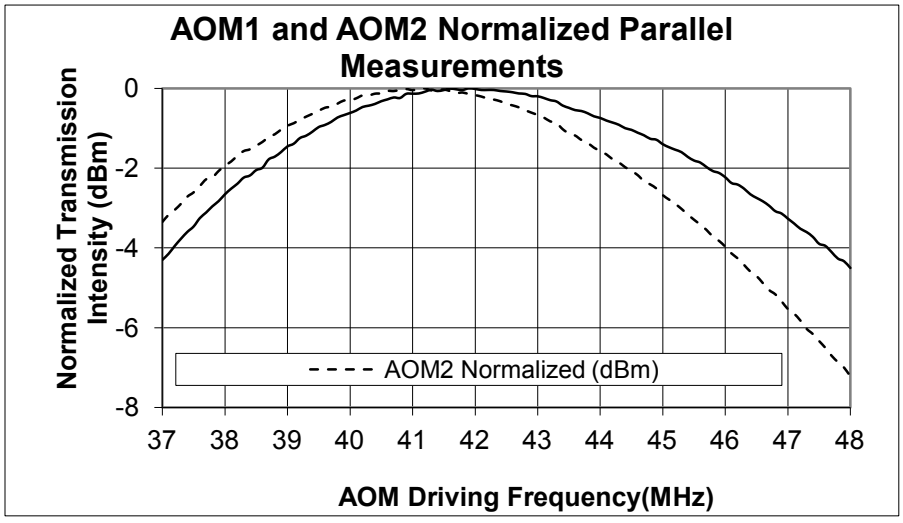


Figure 45, Normalized throughput vs. driving frequency for AOM1 and AOM2.

Erbium Doped Fiber Amplifier (EDFA)

Figure 46, Keopsys pulse Erbium Doped Fiber Amplifier (EDFA)

Keopsys polarization maintaining pulse fiber optic amplifier, Model: KPS-CUS-OEM-PFA-1550-014-200-020-PM-FA, S/N:8090496. Operation pulse repetition frequency = 20 kHz, 14.7 μJ/pulse.

The Erbium Doped Fiber Amplifier (EDFA) from Keopsys was selected as it is a fast pulse width amplifier with a high repetition rate and can operate at 1545.2 nm with up to 14.7 μJ/pulse. The amplification of the unit is controllable by adjustments to the driving current of the first and second stage of the unit. The unit is also equipped with fiber port which allows for the monitoring of the stimulated Brillouin scattering (SBS). The SBS can occur when an optical fiber amplifier amplifies a narrow band optical signal in a fiber optic where the mode field area is typically small. Nonlinear behavior can occur and the SBS can result in a reflection of most of the incoming optical power.[109, 110] By monitoring the SBS port during the system's operation adjustment can be made to the manner the amplifier is driven so as to avoid damage.

In order to reach higher power outputs from the Keopsys amplifier without having excess SBS the output fiber of the Keopsys amplifier was spliced to the fiber circulator. This eliminated an interface between FC/APC couplers which eliminates some loss and the reduction in fiber length reduces the SBS. The SBS vs. output power was measured and the length of fiber between the

circulator and the Keopsys amplifier was reduced. The measurements were repeated and a final splice was done. The measurements were again repeated. By reducing the fiber length the SBS measured for a certain power output was reduced.

The Keopsys has two stages. The first stage is set to 170 ma while the 2nd stage is varied. As the 2nd stage current is increased the output power will increase as will the SBS. At some point the increase in SBS will become more significant with an increase in 2nd stage driving current. This forms a "knee" where the SBS is become significant. As the fiber length between the circulator and the Keopsys amplifier was shortened the "knee" was observed at higher 2nd stage driving currents.

Figure 47 shows a screen capture of an oscilloscope display with the trigger pulse from the digital delay generator (2nd from top- channel 2), the AOM RF driving signal (top- channel 3), and the optical output pulse of the Keopsys amplifier measured with a detector (3rd from the top- channel 4). The optical output of the Keopsys amplifier is approximately 200 ns at the full width at half maximum (FWHM) even though the RF AOM driving signal is approximately 300 ns. The lower wave form may be disregarded.

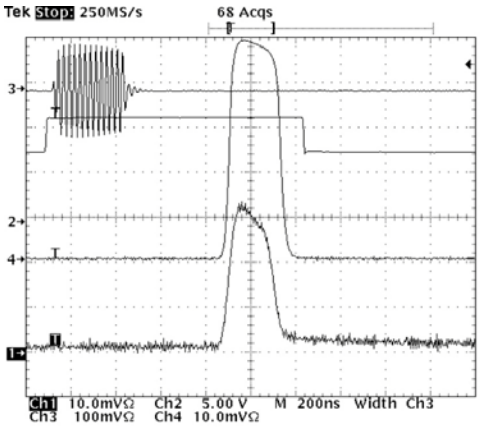


Figure 47, Optical Output of Keopsys EDFA amplifier. Waveforms from top to bottom: RF Driving signal for AOMs, DDG trigger, Optical output of Keopsys amplifier.

Circulator

Micro-Optics Inc., Polarization Maintaining Circulator, P/N: PMFC-3P-1-P-2-1-0-2-HXT-ECF(HP-2W), S/N 29907, Port 1 = Blue, Port 2 = White, Port 3 = Yellow, wavelength: 1550nm +/- 20 nm, optical power (excluding connector damage): <= 2.0 W CW, 1 m Panda 1550 PMF (slow axis).on each port, FC/APC connectors on all ports with slow axis aligned to connector key.

Insertion loss (port 1 to port 2, w/o connectors): <= 0.80 dB, Insertion loss (port 2 to port 3, w/o connectors): <= 0.85 dB, polarization extinction ratio (port 2 w/o connector): >= 27 dB, polarization extinction ratio (port 3 w/o connector): >= 30 dB, directivity or cross-talk (1-3 & 3-1): >= 65 dB, return loss (all ports, w/o connectors): >= 50 dB, polarization extinction ratio (port 2 with connector) >= 27 dB, polarization extinction ratio (port 3 with connector) >=30 dB.

The polarization maintaining circulator is a 3 port fiber optic device that conveys light entering on Port 1 to Port 2, light entering on Port 2 to Port 3, and light entering on Port 3 to Port 1. Light is prevented from traveling in the opposite direction, i.e. from Port 3 to Port 2 etc., by the nature of the design of the circulator. The optical isolation is significant but not absolute, so some small amount of a signal can travel in the opposite direction.

The FC/APC fiber optic connector interface surface is set to 8 degrees angled to increase coupling to the next length of fiber having an FC/APC connector by the use of an FC coupler. The back reflection from an FC/APC to FC/APC junction is on the order of < -60 dB (return loss of >60 dB). When the tip of the fiber is not in physical contact with another fiber but the atmosphere, the back reflection is greater due to the mismatch of the indices of refraction of the core and the air.

In order to reduce the back reflection of the outgoing pulse by the tip, the FC/APC connector was polished from the standard angle of 8 degrees to 10 degrees by Micro Optics of Hackettstown, NJ. Micro Optics reported the following:

Table 7, Test setup configuration by Micro Optics for the polishing of the tip of port 2

PORT No.	**CONFIGURATION**
1	Laser (Power = 1 mW, wavelength = 1552 nm, FWHM = 4 nm)
2	Open to the atmosphere
3	Detector (noise floor = -75 dB)

Table 8, Test results by Micro Optics of back reflected light at port 3 for varying angles at the connector of port 2.

PORT 2 Polished to Angle	**PORT 3 Back Reflection Measurement (dB)**
FC/APC (8°, PC Polished, Original)	-47
8° Flat Polish	-61
9° Flat Polish	-62
10° Flat Polish	-63

Polishing to a flat fiber-air interface from the original FC/APC reduced the back reflection by 16 dB. This helps reduce the intensity of the reflection of the pulse seen at the detector when the outgoing pulse is sent into the atmosphere. This helps to reduce any recovery time of the detector from a large impulses.

The mode field diameter is a useful parameter in the examination of coupling light out of and back into the system. Since the manufacturer of the circulator does not provide this information it was measured. It is difficult to measure the mode field diameter of a fiber in the near field, so the angle of the beam's divergence was measured in the far field and used to calculate the mode field diameter.

The light traveling down the fiber will have a transverse profile approximating a Gaussian profile. A Gaussian beam will expand as it propagates from a point where the beam is minimal in size. This minimal beam diameter is known as the beam waist. As the beam moves farther from

the beam waist, the beam diameter increases. The smaller the waist, the greater the divergence of the Gaussian beam.

The diameter of the beam is measured at a point where the intensity is a prescribed fraction of the maximum intensity. A very common intensity value is $1/e^2$ or 13.533% of the maximal intensity. Some other values used to measure beam diameters are 1% of the maximal intensity or 1/e, 36.787%, of the maximal intensity. In some cases the full width at half maximal (FWHM) value is used but this is less common.

The optical circulator is constructed from single mode polarization maintaining fiber. For comparison, a similar type of fiber from Fujikura Inc. has a cladding diameter is 125 μm while the core has a mode field diameter of 10.5 μm +/- 1 μm. The mode field diameter is considered the beam waist at the end of the fiber. Since the beam waist is on the scale of micrometers, the divergence will be significant over a relatively short range. With the tolerance of +/- 1 μm in the mode filed diameter size, the effect on the size of the beam at 50 cm varied from 10.35 cm if the mode filed diameter was 9.5 μm, and 8.55 cm if it was 11.5 μm. This is more than a 20% difference from the minimal size to the maximal size.

Figure 48 shows the setup used to make the measurements of the beam diameter emanating from the FC connector of port 2 of the optical circulator. Port 2 of the optical circulator was affixed to the fiber chuck and inserted into the fiber positioned (Figure 49). The fiber positioned was affixed to an XY translation stage and spacers to raise the fiber and to allow for some adjustments to the position. The measurements were made after sunset and with the room as dark

as possible so the detector could be set to a range that would allow the drop off in intensity to be determined accurately.

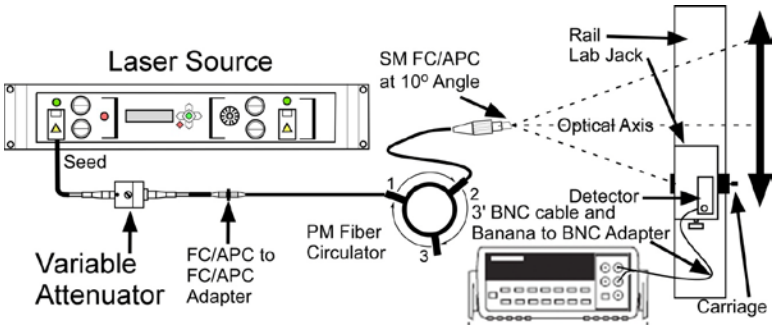


Figure 48, Setup to measure beam intensity at various lateral offsets from optical axis of optical circulator port 2.

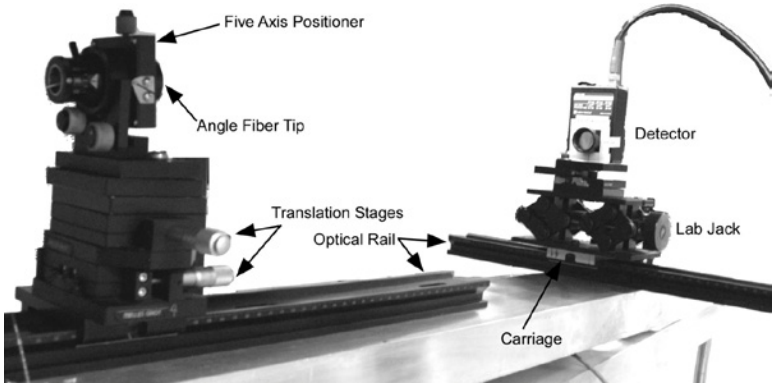


Figure 49, Photograph of setup to make measurement of beam diameter at a distance from the fiber tip.

The variable optical attenuator was adjusted so as not to saturate the detector. The height of the lab jack was adjusted so that the center of the optical detector was at the same height as the tip of the fiber in the fiber positioner. Adjustments were made to the tilt of the fiber positioner so the maximum readings were seen at the detector's output. This establishes the optical axis. The lab jack height was adjusted to confirm the maximal reading as was the lateral position of the lab jack on the rail. A measurement was made with the beam blocked to establish the baseline of the detector. The maximal value minus the baseline value was multiplied by $1/e^2$ and the baseline was added to determine the detector's output level at the $1/e^2$ level of the beam.

$$Voltage(1/e^2 V_{peak}) = \frac{1}{e^2}(V_{\max} - V_{Baseline}) + V_{Baseline} \tag{3.3.22}$$

The detector was moved along the rail, perpendicular to the optical axis, as the output of the detector was monitored. The position of the carriage on the rails was noted when the voltage was equal to the calculated $1/e^2$ value and the same was repeated for the other side of the optical axis. Table 9 shows the measurements made and the distances used in the calculation of the $1/e^2$ beam diameter.

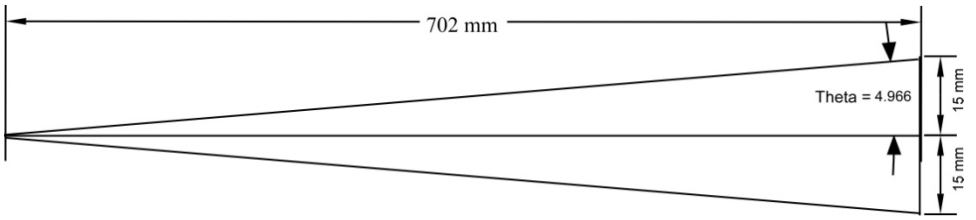


Figure 50, Approximate dimensions of measurements made for beam diameter calculations.

Table 9, Measurements of $1/e^2$ beam diameter.

	V_{Max} (V)	**$V_{Baseline}$ (V)**	**Position V_{Max} (mm)**	**$1/e^2$ Voltage (V)**	**Position Left (mm)**	**Position Right (mm)**	**$1/e^2$ Diameter (mm)**	**Separation Distance (mm)**
Run1	5.403	0.932	170	1.537	107	229	122	702
Run2	4.740	0.306	168	0.606	109	231	122	702

The $1/e^2$ radius of 61mm at a distance of 702mm results in half angle of θ = 4.988 which was calculated as $\arctan\left(\frac{61mm}{702mm}\right) = \theta$. The numerical aperture can be calculated using the half angle and equation (3.3.9) as $NA = n\sin(\theta)$ where n is the index of refraction of air equals 1. The NA = 0.08699.

The radius of a Gaussian beam at a range z, w(z), can be calculated based upon w_o, the beam waist radius at the face of the fiber optic, λ the wavelength, and η the index of refraction of the medium in which the beam is traveling.

$$w(z) = w_0\left[1+\left(\frac{\lambda z}{\eta\pi w_0^2}\right)^2\right]^{1/2} \tag{3.3.23}$$

The Rayleigh range is [103]

$$z_R = \frac{\eta\pi w_0^2}{\lambda} \tag{3.3.24}$$

Equation (3.3.23) can be written as,

$$w(z) = w_0\left(1+\frac{z^2}{z_R^2}\right)^{1/2} \tag{3.3.25}$$

$$\left(\frac{w(z)}{w_0}\right)^2 = 1+\left(\frac{\lambda z}{\eta\pi}\right)^2\left(\frac{1}{w_0^4}\right) \tag{3.3.26}$$

Let $A = \left(\lambda z/\eta\pi\right)^2$, $B = -w^2(z)$ and re-writing the equation as

$$\frac{-B}{w_0^2} = 1+\frac{A}{w_0^4} \rightarrow -Bw_0^2 = w_0^4 + A \tag{3.3.27}$$

$$0 = w_0^4 + Bw_0^2 + A \tag{3.3.28}$$

This is a 4th order polynomial known as a biquadratic. In order to solve an equation of the form $Q(x) = a_4x^4 + a_2x^2 + a_0$ the equation can be manipulated into the familiar quadratic equation

$q(y) = ay^2 + by + c$ which has the solutions $y_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$. The solutions to the biquadric equation is then; $x_1 = +\sqrt{y_+}$, $x_2 = -\sqrt{y_+}$, $x_3 = +\sqrt{y_-}$, $x_4 = -\sqrt{y_-}$.

To do this let $f = w_0^2$ which results in equation (3.3.28) becoming,

$$0 = f^2 + Bf + A \tag{3.3.29}$$

with A = 1.19218x10^{-13} and B = -0.003721

$$w_{0_1} = +\sqrt{\frac{-B + \sqrt{B^2 - 4A}}{2}} = 0.060999 \tag{3.3.30}$$

$$w_{0_2} = -\sqrt{\frac{-B + \sqrt{B^2 - 4A}}{2}} = -0.060999 \tag{3.3.31}$$

$$w_{0_3} = +\sqrt{\frac{-B - \sqrt{B^2 - 4A}}{2}} = 5.660x10^{-6} \tag{3.3.32}$$

$$w_{0_4} = -\sqrt{\frac{-B - \sqrt{B^2 - 4A}}{2}} = -5.660x10^{-6} \tag{3.3.33}$$

The solutions w_{01}, w_{02} and w_{04} are not physically viable solutions. The solution w_{03} would mean the MFD is 11.32 μm. This is within the specifications of the manufacturer (10.5 um +/- 1μm). Using equation (3.3.10) and the calculated field mode diameter the effective numerical aperture can be calculated as $NA_{eff} = 2\lambda/\pi MFD = 2(1.542\times10^{-6})/\pi(11.32\times10^{-6}) = 0.08689$. This is in good agreement with the numerical aperture calculated using the measured half angle.

Lens

CVI Melles Griot Aplanat Lens, P/N: LAP-500.0-100.0-PM, Clear aperture = 100.0 mm, focal length at 633nm = 500.0 mm.

The lens is employed for atmospheric transmission and reception of light traveling to and fro. It collimates the light diverging from the tip of the fiber so that it propagates with little divergence in the atmosphere and then guides the light back reflected from the atmosphere to the tip of the fiber. To efficiently couple light into a single mode fiber optic requires attention be paid to many parameters. The angle at which the light is focused down to a point is an important one but not the only one. Aberrations resulting from lens manufacturing anomalies or design compromises can negatively affect the ability of a lens to focus the light to the smallest possible spot size.

To maximize the coupling of light a high quality lens with anti-reflective coatings should be used. Aspheric lenses perform well in the elimination of some primary aberrations but are difficult to manufacture in larger sizes and therefore expensive. This lens selected is known as an Aplanat lens. It consist of two optical elements, each of which is designed to cancel the monochromatic wave front errors, spherical aberration and coma, created by the other lens. This reduces the aberrations to increase the coupling into the fiber optic circulator.

Five Axis Positioner

Siskiyou connectorized fiber translator for 5 axis (P/N CFT-5), X and Y axis travel = 2.1 mm, Z axis travel = 6.3 mm, pitch and yaw = 5 degrees. X, Y and Z axis minimum controllable motion = 1 μm, Pitch and yaw minimum controllable motion = 5 arc seconds. Connectorized fiber chuck (CFC-NTT) with FC connector holder

Figure 51, Siskiyou 5-axis positioner

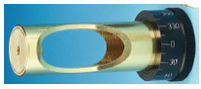


Figure 52, FC connectorized fiber optic chuck.

The 5-axis positioner is capable of movement of the inserted chuck in the X, Y, and Z axis as well as pitch and yaw. The FC chuck allows the connectorized end of the fiber from the circulator to be secured with a standard FC connector and the chuck inserted into the 5-axis positioner. The chuck can be rotated before being secured so as to align the slope of the angled tip of the fiber at any desired angle. This is necessary as the light exiting the tip of the fiber will follow a path that is titled at an angle from the axis of the chuck.

Translation Stages, Optical Rail and Custom Base

An optical rail and carriages are used to securely position the output/input fiber, lens and mirror relative to one another. The 5-axis positioner sits atop two stacked translation stages that are secured to a rail carriage and are used to adjust the X axis and Z axis in a finer manner than possible with the positioner alone. The translation stage which adjust the Z axis position is a dual micrometer with dial gradations of 0.5 micrometers. This allows for very fine adjustments to be

made. The 5-axis positioner is secured to the translation stages through multiple bases to raise the height of the fiber so it is at the same height as the centers of the 6 inch mirror and the lens.

To compensate for the tilt of the beam leaving the angled fiber tip a special base was machined so as to have a 5 degree angle between the upper and lower faces of the base. A 5 inch sine vise and gauge blocks were used to create a precision 5 degree angle on the vise and an aluminum base was machined using a vertical milling machine. The 5-axis positioner does have the ability to tilt the fiber tip but the amount of pitch necessary to have the beam travel down the optical rails axis was at the limit of the positioner's adjustability.

Front Surface Mirror

Esco coated front surface aluminum mirror (P/N D460100), 6 (+/- 0.005) inch (152.4 mm) diameter by 1.5 (+/- 0.005) inch (38.1 mm) thick, surface flatness: $\lambda/10$, surface quality: 40/20 scratch-dig, coated AL-SIO, Pyrex glass.

Mirror Mount: Newport Klinger Motorized Gimbaled Mirror Mount SL-A Type.

The mirror installed in a large mirror mount allows the beam to be steered. The mirror is angled so the normal to the surface is at 45 degrees from the vertical axis. The mirror can then be rotated about the vertical axis to steer the beam into the atmosphere. The clear aperture diameter of the lens is 100 mm (approximately 4 inches) so a mirror of larger dimensions was necessary in order to present a unobstructed reflective surface to the impinging beam. The mirror when tilted at 45 degrees from vertical presents a surface of 107.7 mm in the vertical dimension.

A large mirror mount with tilt and yaw capabilities was obtained but unable to be used without modification as the opening for the mirror was 1 to 2 mm too small. The mount was

disassembled and opening in the ring that holds the large mirror was carefully and precisely cut on a lathe to accommodate the mirror's diameter.

In order to align the mirror with the fiber tip and lens a custom optical rail carriage was carved from a solid block of aluminum based upon the dimensions of smaller commercial carriages that are available. Special attention was paid to centering the mirror mount so that the center of the mirror would align with the center of the optical rail. See Figure 53 for pictures of custom optical rail carriage and see Figure 54 for pictures of mirror mount secured to custom optical rail carriage.

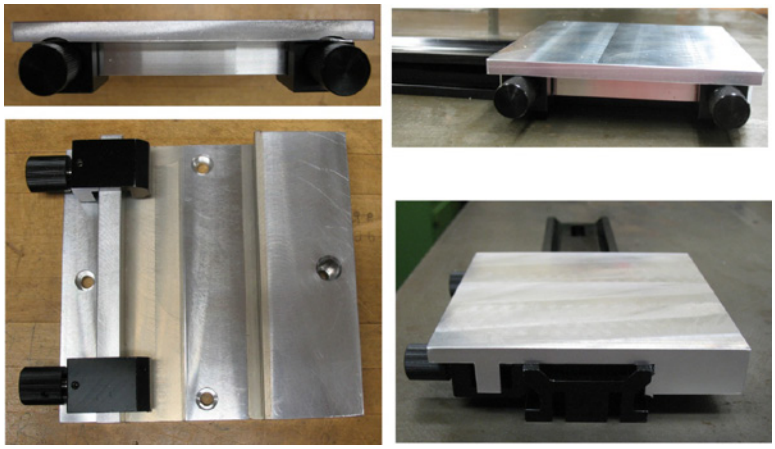


Figure 53, Pictures of custom optical rail carriage cut from a solid aluminum block. Clockwise from upper left: Side view of carriage showing knobs of clamping blocks, Side view of carriage secured to optical rail, Top view of carriage secured to optical rail, Bottom view of carriage showing holes used to secure mirror mount to carriage and clamping blocks.

Figure 54, Picture of modified mirror mount secured to custom optical rail carriage (left) and same secured to an optical rail.

Oscillator

Figure 55, Hewlett Packard 8657B signal generator with 0.1 to 2000 MHz sinusoidal output frequency range.

HP 8657B 0.1-2000MHz Signal Generator, Resolution=1 Hz, ±1 dB level accuracy, N type connector with adapter to SMA output. Weight 45 lbs, 5.25x16.75x22.6”, 200VA Max

This unit supplied the oscillating source that was used to drive the acousto-optical modulators. Its resolution, adjustable power output and computer controllability allowed the development of the system to proceed without limiting the test performed. This unit will be replaced with a smaller dedicated oscillator during further development. The possibility of combining all of the acousto-optical modulators driving components into one component also exists in the future. A spectrum analyzer was used to examine the spectrum of the output when the unit was tuned to 42.0 MHz. A wide spectrum and a close in spectrum are shown in the following figures.

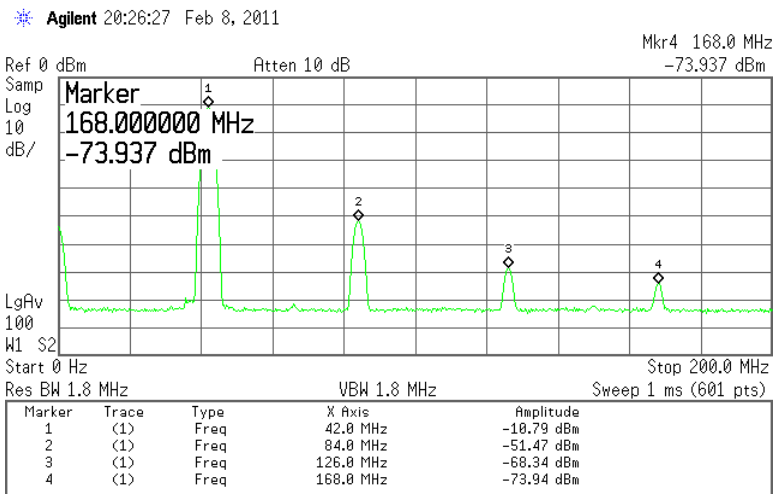


Figure 56, Hewlett Packard 8657B signal generator output wide view spectrum. Frequency set to 42.0 MHz.

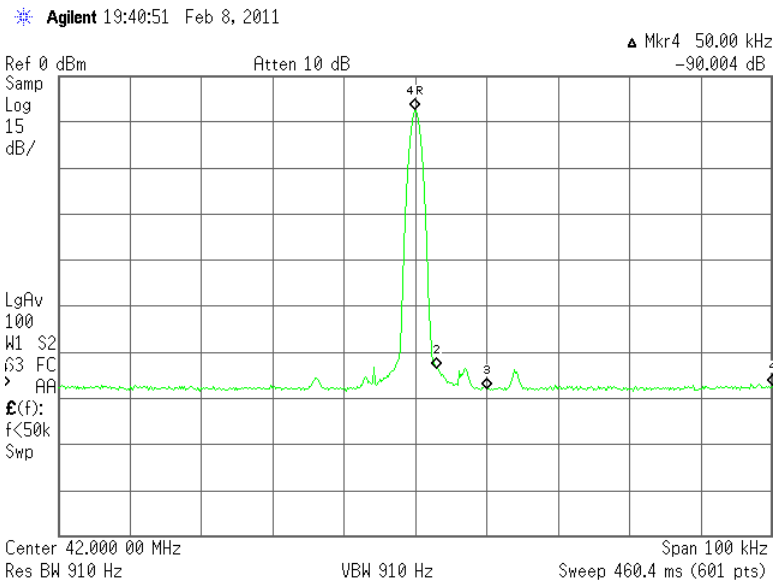


Figure 57, Hewlett Packard 8657B signal generator output close in view spectrum. Frequency set to 42.0 MHz.

Directional Coupler

Figure 58, Mini-Circuits directional coupler.

Mini-Circuits ZMDC-10-1 Directional Coupler, Connectors SMA, RF Input, RF Output, Coupled RF, Specs at 48 MHz: Mainline Loss (dB) In-Out = 0.8, Coupling (dB) In-Cpl = 11.40, Directivity (dB) = 60.96, Return Loss (dB) in=49.46 Out=34.66 Cpl=24.10

The directional coupler allows the signal from the oscillator to be monitored. If the oscillating signal were to be lost the system could be damaged by the absences of an input signal for the EDFA optical amplifier.

RF Switch

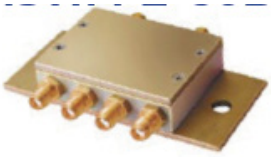


Figure 59, Mini-Circuits radio frequency switch with TTL control logic

Mini-Circuits ZASWA-2-50DR Switch TTL DC-5GHz, Power= +5vdc @60ma, -5vdc @60ma, Control Current 5ma High, 0.2ma Low TTL=0-0.8v Low, 2-5v High Rise/Fall Time=5-15ns, 3.25x1.5x0.62”

The RF switch controls the application of the driving signal from the oscillating to the AOMs. The RF switch is driven by the digital delay generator.

Terminator

Figure 60, Mini-Circuits terminator, 50 ohms.

Mini-Circuits ANNE-50+ Terminator 50ohm DC-18GHz, SMA male, Return Loss DC-4GHz 30dB min., Power Rating=1W@50deg C, 0.58x0.37” diameter.

The terminators are used to assure proper termination of signals and help to reduce signal reflections in the system.

Bandpass Filter

Figure 61, Mini-Circuits filter

Mini-Circuits SIF-40+ Bandpass Filter 35-49 MHz 50 ohm, Center freq = 42 MHz, 1.98x0.67" diam., Insertion loss @40.4MHz 0.23 dB

The bandpass filter helps to reduce extraneous signal from being input to the RF amplifier. This is an attempt to make sure the signal driving the AOMs from the RF amplifier is a good clean signal. Bandpass filters on the output of the RF amplifier would be useful but they would have to handle much more power. The available bandpass filters were not designed for the power levels coming from the RF amplifier.

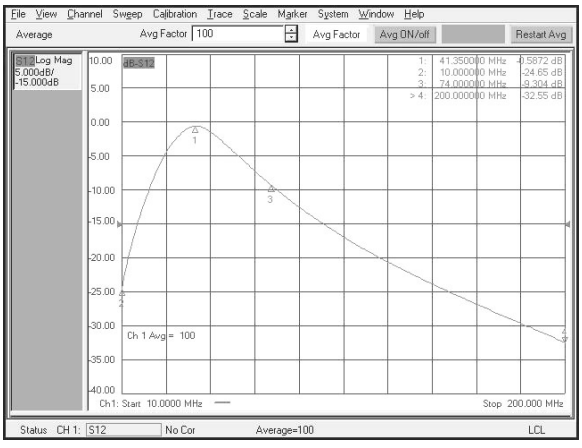


Figure 62, Bandpass filter response of Mini-circuits SIF-40+ filter.

RF Amplifier

Figure 63, Mini-Circuits ZHL-5W-1 RF power amplifier.

Mini-Circuits ZHL-5W-1, 5 Watt 5-500 MHz 50ohm Amplifier Output power=+37dBm min, gain=40 dB min, flatness max=+-1.7dB, noise figure 4 dB typ., IP3 +40 dBm type, 24vdc @ 3.3A, dimensions 8.35x7x3.13”

The RF amplifier drives the AOMs with the signal generated by the previous components. It must be able to supply up to 2W of driving power to each AOM after the signal passes through a 3 way splitter.

RF Splitter

Figure 64, Three way RF splitter

Mini-Circuits ZA3CS-400-3W-S high power combiner/splitter 3 way, 20400 MHz, 0 degree, 2 to 400 MHz, SMA connectors, low insertion loss = 0.5 dB typ., isolation = 25 dB typ., very low amplitude, 0.15 dB typ. and phase unbalance, 0.2 deg. typ. Power input = 10W as splitter, 2x2x0.75”

The 3 way splitter is used to deliver the driving signals to the AOMs and to allow for the monitoring of the driving signal through the third port.

Attenuator

Figure 65, Mini-Circuits VAT-10W2+ Fixed Attenuator 2W 10dB DC-6GHz

Mini-Circuits VAT-10W2+ Fixed Attenuator 2W 10dB DC-6GHz , SAM male/female, 1.43x0.74” diameter, Attenuation=10+-0.3dB,Flatness DC-3GHz=0.10dB typ., Input Power=2W max, VSWR(:1) DC-3GHz=1.15 typ.

Mini-Circuits VAT-3W2+ Fixed Attenuator 2W 3dB DC-6GHz, SMA male/female, 1.43x0.74” diameter, Attenuation=3+-0.3dB, Flatness DC-3GHz=0.20dB typ., Input Power=2W max, VSWR(:1) DC-3GHz=1.05 typ.

The attenuators are used to reduce the voltage of the signal that is driving the AOMs so that its power levels are sufficiently low as to not damage the monitoring instrumentation such as an oscilloscope or a spectrum analyzer.

Digital Delay Generator (DDG)

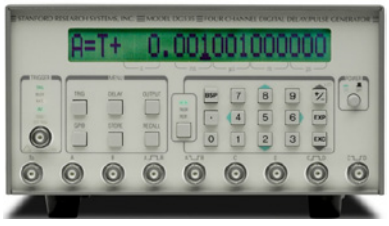


Figure 66, Digital delay generator from Stanford Research Systems used as a time keeper for system.

Stanford Research Systems, Model DG535 Digital Delay Generator. Timebase Standard: 25 ppm crystal oscillator, Accuracy: 1500 ps + timebase error × delay, RMS jitter <100 ps + (10^{-8} ×

delay), Trigger delay (typ.) 85 ns (ext. trigger to T0 output), Rise time 2 to 3 ns (typ.), Slew rate 1 V/ns, channel-to-channel jitter is typically 50 ps.

The DDG can provide four precisely-timed logic transitions or two independent pulse outputs. The unit is used to generate two pulses at precise times at 20 kHz. One pulse is used to time the Lidar system by initiating the collection of data and the second pulse is set to a precise duration and a precise delay from the first pulse to trigger the generation of an RF pulse to drive the acousto optical modulators.

Anti-Aliasing Filters

Figure 67, Mini-Circuits Anti-aliasing RF filter.

One low pass RF filter followed by two high pass RF filters. Mini-circuits high pass filter (SHP-25+) (S/N R8634410817), 27.5 to 800 MHz, SMA connectors, stop band loss > 40 dB = 13 MHz, > 20 dB = 19 MHz, nominal cutoff frequency (3 dB) = 25 MHz. (see Figure 67)

Mini-circuits low pass filter (SLP-150+) (S/N R8947500850), DC to 140 MHz, SMA connectors, stop band loss > 40 dB = 300-600 MHz, > 20 dB = 210-300 MHz, nominal pass band frequency = DC to 140 MHz, nominal cutoff frequency (3 dB) = 155 MHz

These filters reduce signals at frequencies beyond half the sampling frequency of the analog to digital converter so as they are not aliased down to lower frequencies and may be interpreted as actual signal at a frequency of interest. Figure 68 shows the frequency response of the anti-aliasing filters. The response is not flat but has less than a 0.7 dB difference in the frequency

span of interest. Figure 69 shows a linear and log magnitude for the filters for the frequency range of 10 to 400 MHz.

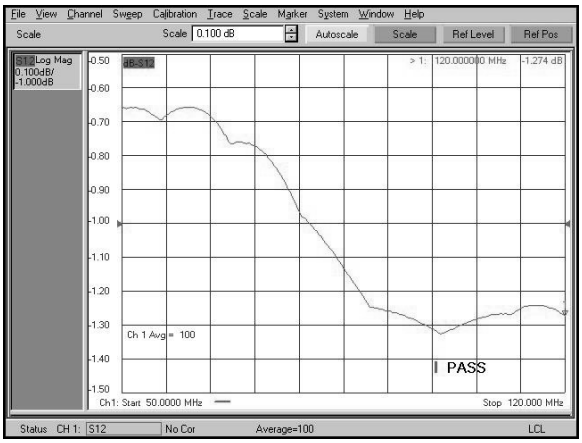


Figure 68, Frequency response of anti-aliasing filters (one SLP-150+ and two SLP-150+) over 10 to 120 MHz.

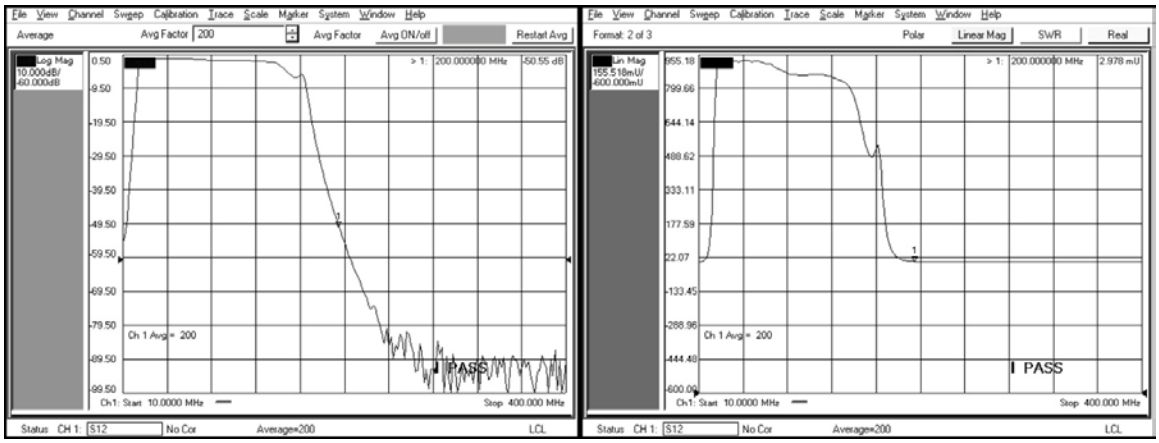


Figure 69, Frequency response of anti-aliasing filters (one SLP-150+ and two SLP-150+) over 10 to 400 MHz. Logarithmic (left) and linear (right) magnitude scales.

Oscilloscope

Figure 70, Tektronix TDS754C four channel oscilloscope.

Tektronix TDS754C 4 channel, 500 MHz, 2GS/s, Insta-Vu Color digitizing oscilloscopes, S/N B011832.

The four channel oscilloscope was used to monitor signal and make measurements during the development of the system and operation of the system. The unit has the ability to record data and images of the display to a 3.25 inch floppy drive.

Analog to Digital Converter with Field Programmable Gate Array Card

Figure 71, Data acquisition card X5-400M with two 14 bit ADC channels with a 400 MHz sampling rate.

Innovative Integrations X5-400M RevE., S/N E135, installed in PCIx 8 lane XMC adapter card (S/N A148). Card contains two 400 mega samples per second (MSPS), 14 bit, analog to digital converters, 50 Ohm input impedance (ADS5474 Texas Instruments), with on board field programmable gate array (FPGA, Xilinx Vertex 5 SX95T with 512 MB of DDR2 DRAM and 4MB QDR-II memory).

The X5-400M data acquisition (DAQ) card was chosen in part for its high speed acquisition and resolution. The card is capable of providing over 1 GB/s sustained transfer rates to the host through the 8 lane PCI express interface.

Nyquist sampling rate theory states that the minimal sampling frequency to avoid aliasing is twice the highest frequency of the signal being sampled. In this case that would be a sampling frequency of 228 MHz as the signals of interest are between 54 MHz and 114 MHz. The

bandwidth of interest is 60MHz. This is the minimal theoretical limit for a periodic signal. The signal from the atmosphere will have a great deal of noise. The higher sampling rate will add to the effective number of bits of the system and improve the ability to detect a signal which is buried in noise.

The X5-400M can be customized using VHDL and Matlab with the use of a FrameWork Logic toolset from the manufacturer. The manufacturer also supplies C++ libraries and drivers for Windows and Linux.

The analog to digital converter (ADC) chips utilized are the Texas Instruments ADS5474 with 400 mega samples per second (MSPS), 14 bit resolution, 11.2 effective number of bits, with an signal to noise ratio of 69.8 dB full scale. [111] The manufacturer of the DAQ card, Innovative Integration, specifies the performance of the card at a lower level than the performance stated by Texas Instruments. This is most likely due to complications of laying out a high speed circuit board and its effect on performance. The ENOB is 10.1 bits, spurious free dynamic range (SFDR) is 67.9 dB, harmonic distortion is -65.1 dB, and a SNR of 73.6 dB from a 64K FFT.

The input channels to the X5-400M were terminated with 50 Ohm loads, data was collected at 400 MSPS for both channels and a FFT was performed with 131072 data points (2^{17}). The results if the FFT is shown in Figure 72. Note the suppression of the noise level to approximately -110 dB is partially due to the large number of points used in the FFT.

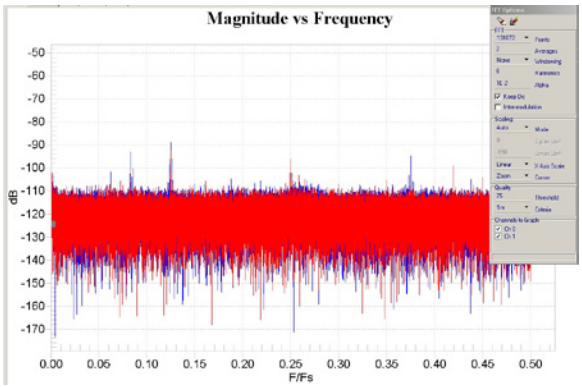


Figure 72, FFT spectrum of channels 0 and 1 of X5-400M DAQ card with inputs terminated by 50 Ohm loads. Note: the x axis is in sampling frequency (400 MHz).

The X5-400M contains a digital to analog converter (DAC) which can be used to create a sinusoidal waveform. A 20 MHz sinusoidal waveform was created and the output was observed on an oscilloscope and a spectrum analyzer. See Figure 73. The quality of the signal was not very pure and appeared to have distortions and random spikes. Note the harmonics are only approximately 20 dB bellow the signal. In comparison to the oscillator source seen in Figure 56 where the harmonics are 40 dB down from the signal, the DAC has harmonics 100 times stronger, relatively speaking, than the harmonics of the oscillator. The DAC was not chosen to be the source of the oscillator signal in part due to the level of performance. An additional reason was that the card would need to use an external high precision clock if it were to generate DAC output and ADC at 400 MHz. With the on board clock the DAQ card would only be able to sample at 160 MHz with the ADC if the DAC was implemented. The input amplitude variation, from the manufacturer, of the X5-400M ADC is shown in Figure 74.

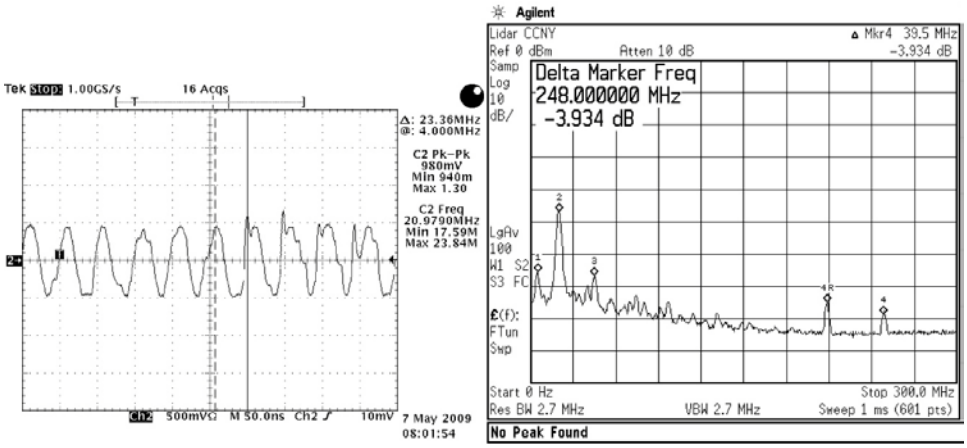


Figure 73, Oscilloscope, time domain (left), and spectrum analyzer (frequency domain) (right), displays of X5-400M digital to analog converter for 20 MHz sinusoidal waveform.

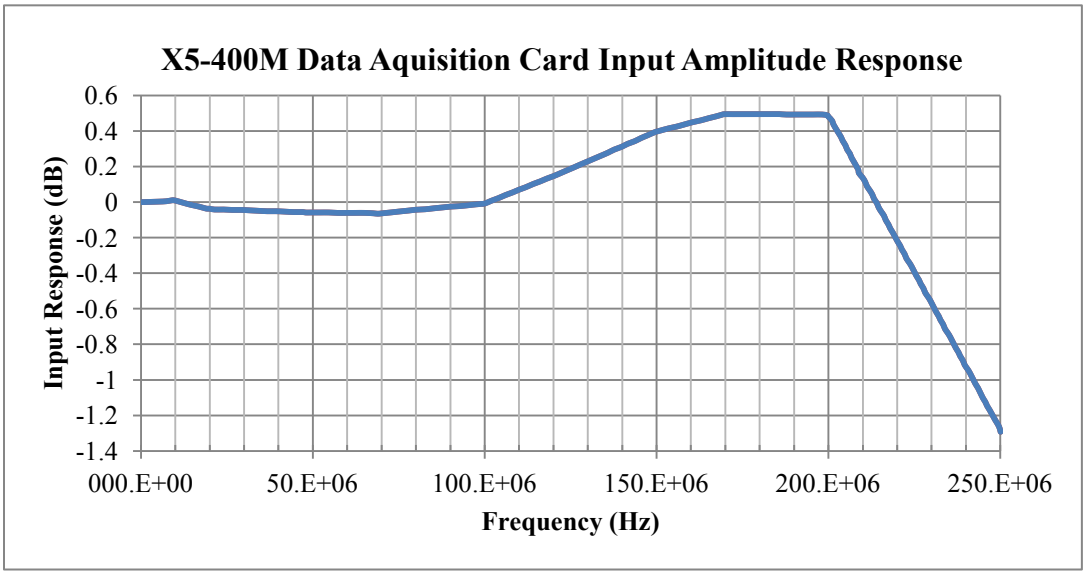


Figure 74, X5-400M analog to digital converter amplitude variation reported by the manufacturer.

Signal Processing/Data Collection Computer

TYAN Tempest i5400XT dual Xeon server board with INTEL 5400 chipset, 8GB DDR2 667MHZ RAM, 2X PCI-E X16 GEN2, 1X PCI-E X8, 2X PCI-X, 1X PCI slots, Adaptec 5805 RAID card with 4 hard drives (500 GB) in stripped (RAID 1) configuration for data and separate 500 GB hard drive for operating system (Windows XP 64bit). Custom applications were run on the computer to control data acquisition, processing and storage.

A 64 bit operating system was selected so that higher transfer rates of data could be achieved and more memory could be accessed by the system than possible with a 32 bit operating system. Since the data acquisition card has two channels and samples at 400 mega-samples per second with 14 bits of resolution, it can generate 1.6 GB/s of data. In this system only one channel is utilized, making the data rate 800 MB/s. The data acquisition card has on board memory that allows the data to be pooled before it is transferred across the computer's data bus. The optimal pooling of data into packets and frames is depends upon the memory allocated to the busmaster of the computer. Once the data has been transferred to the computer it must be stored. The RAID controller card was configured as a RAID1 where the data was stripped across four different hard drives. Stripping involves a portion of the data being written to each of the drives. Spreading the process of writing the data across multiple drives reduces latency and allows the data to be written more quickly. If one of the hard drives should fail then all of the data would be lost. Since the data written at this high rate would only be needed until it was processed the risk is not significant. Each hard drive is a SATA configuration with 3Gb/s, or 375 GB/S, transfer rate. Since the data acquisition card is generating 800 MB/s, three hard drives theoretically should be able to handle the data rate if configured properly. A fourth hard drive was implemented to allow for overhead. The PCI Express X8 bandwidth for upstream and downstream is 2 GB/s. Since the process of taking data will most often be a process of sequential writes and very few reads the throughput is improved.

The RAID controller card utilized can manage eight hard drives as two channels of four hard drives each. The four data hard drives were placed on the same channel to maximize the speed of transfer. Since the initial acquisition of the computer system, advances in hard drive technologies

have increased the transfer rates possible. Currently there are faster RAID card, 6 Gb/s SATA hard drives and other technologies that have increased throughput.

Spectrum Analyzer

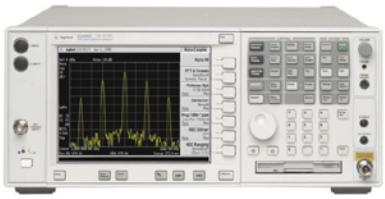


Figure 75, Agilent spectrum analyzer.

Agilent E4446A 3 Hz – 40 GHz, Options 1DS- preamp, D7J digital demodulation hardware spectrum analyzer (S/N US44020325) with Picosecond Pulse Labs' DC blocking capacitor Model 5509-205-224, 145120 1705 and Agilent 2.4 to 3.5 mm female to female adapter 16 V max 0.22 µF.(P/N 1250-2277)

This unit was used during the development of the system. It is not part of the working system but was very useful during the laboratory phase of development.

FC Adapter

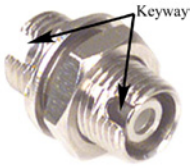


Figure 76, FC to FC bulkhead fiber optic adapter. Arrows indicate the keyway that allignes the angles tips of the FC connector.

OZ Optics FC Adapter with narrow key (2.06 mm), P/N PMPC-03, Lot# M030883, Barcode 11

Narrow key adapters were used in order to maintain the extinction ratio where two segments of fiber with FC/APC connectors are joined. The keyway assures the angles of the connectors are

aligned. Adapters were used instead of a fusion splice so that the system could be disassembled for transport and measurements could be made at points along the optical path.

Network Analyzer

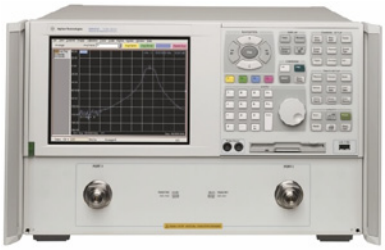


Figure 77, Agilent network analyzer model E8363B, 10 MHz to 40 GHz.

E8363B PNA Network Analyzer, 10 MHz to 40 GHz, general-purpose network analysis, 110 dB of dynamic range <0.006 dB trace noise, <26 μsec/point measurement speed, 32 channels, 16,001 points.

The network analyzer is able to measure mixer conversion loss, return loss, isolation, and absolute group delay, amplifier gain compression, harmonics, etc. The unit was used to measure the characteristics of some of the RF components used in the development of the system.

Operations

For the system to operate the equipment must be turned on, warmed up and initiated in the proper sequence. A number of computer control and acquisition programs must be running in order to acquire data. The programs make adjustments to the laser source, laser amplifier, data acquisition, and data processing.

There are individual software packages to run the NP Photonics laser source and the Keopsys optical amplifier. These two software packages are currently run on a computer separate from the data acquisition computer for safety reasons. The laser source and optical amplifier control programs are run on a computer that does not have developmental software running on it. If the software being developed were to halt the computer operations due to a programming or operation error, the laser source and optical amplifier could be damaged. Once software development is finalized, and the software is stable all of the software packages should be able to be run from a single computer.

The data acquisition (DAQ) card is supplied with an executable example program known as "SNAP.exe." It allows for the configuration of the DAQ card and transfers data to the host computer. A modified version of SNAP was created to collect data from the DAQ card and transfer it to the hard drive of the computer as a text formatted file in a specific manner. The modified SNAP program also creates a file that is continuously appended as the system runs. The DAQ card's field programmable gate array (FPGA) has been programmed and configured to collect 8192 data points, and perform a fast Fourier transform (FFT) in sets of 128 data points resulting in 64 spectrums of 64 points each which represent the spectrum of each range bin. This process creates a 64 point spectrum representing the spectral content of a particular range. The FFT spectra results of 500 ms of data collection, 10,000 pulses, are accumulated and then transferred to the host computer and appended to a data file. The accumulation of the spectrums is the summing of the spectrums from pulse to pulse. The 64 spectrums resulting from the accumulation of 10,000 pulses are stored until 60 sets, 30 seconds, of data is collection. This data is sent to the computer as a text file. Figure 78, top, shows the data acquisition card's FPGA text file output for 12/05/11 at 14:31:48. The graphs show the 30 second output, or 60 500 ms

spectral accumulations, of the ADC output text file. The large vertical spikes are the signals resulting from the back reflection of the outgoing pulse from the fiber optic tip of the circulator and to a lesser degree the leakage from port 1 to port 3 of the fiber topic circulator. The variations in the size of these large spikes are seen as a fluctuation in the system over time. The size of the signals was seen to grow and shrink over time.

These may be due to variations in the gain of the optical amplifier, changes in polarization, changes is the phase of the back reflected signal due to environmental perturbation of the fibers. When the fiber optic connecting the circulator to the telescope was bent as to have a small radius the signal was greatly reduced but still exhibited a fluctuation in intensity. The circulator has some crosstalk between the laser source port and the detector port. This crosstalk and the fact that some very small portion of the outgoing pulse may still be back reflected from the fiber tip may account for the presence of a signal. The fluctuation in the intensity therefore may be due to small perturbations in the birefringence of the polarization maintaining fiber as discussed in an earlier section and phase changes caused by environmental perturbations.

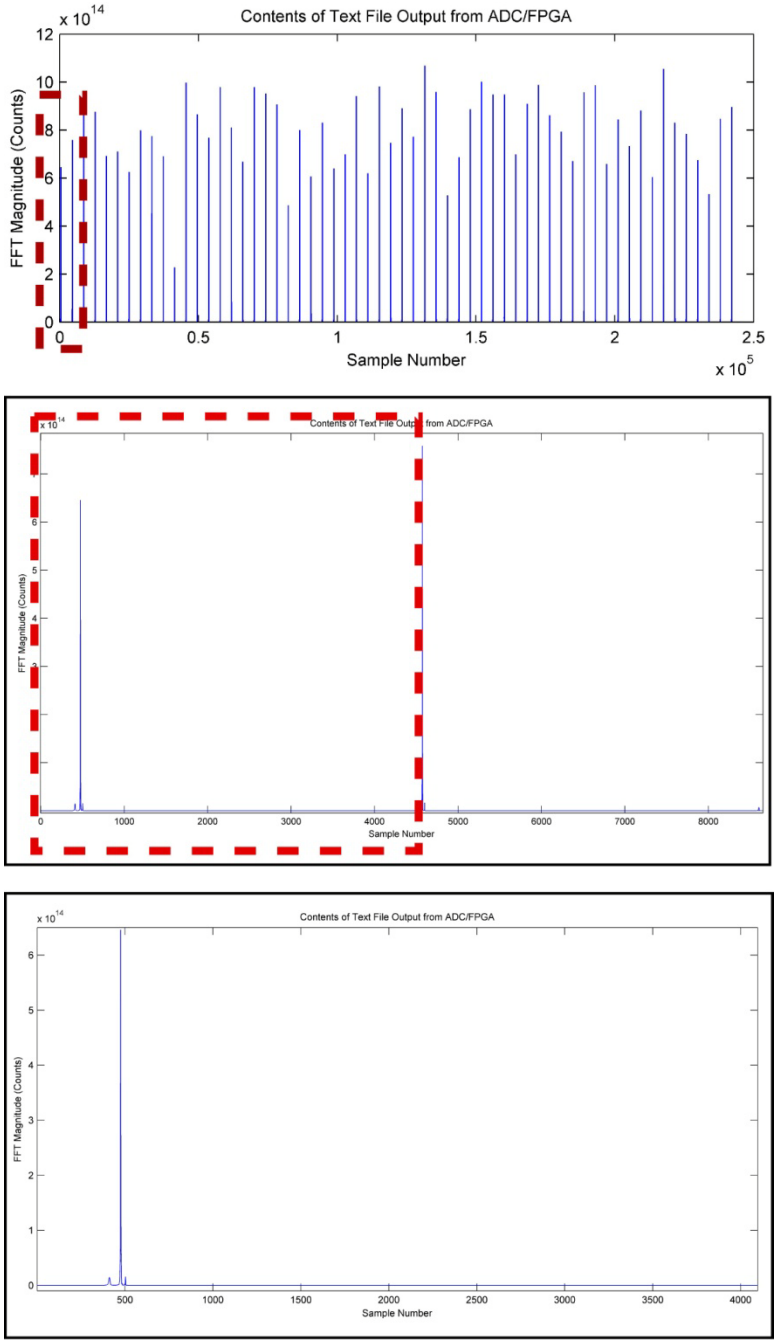


Figure 78, Data acquisition card's FPGA text file output for 12/05/11 at 14:31:48. (Top) The graph plots the 30 second output, or 60, 500 ms spectral accumulations, of the ADC output text file. (Center) The graph plots the first two 500 ms accumulations. (Lower) The graph plots the 1st 500 ms accumulation.

The center plot of Figure 78 is an expanded view of the first two 500 ms accumulations from the ADC/FPGA. The lower plot is an even closer view of the spectrums, focused on the first set of spectrums from the first 500 ms accumulation. The large peak is, again, the spectrum of the back

reflected outgoing pulse. The spectrums of the atmospheric return signals are too small to be seen on this scale. The text file contains the accumulation of the spectrums of the back scattered return signals for 10,000 outgoing pulses. For wind velocity estimation the frequency shift of the back reflected return signals is more significant than the intensity or variation in intensity. Figure 79 shows progressively closer views of the spectrums from the first 500 ms accumulation. The upper plot shows a close up of the first 29 spectrums from the first 500ms accumulation. The vertical scale is magnified to better show the details of the individual peaks that span two spectrums which represent the back reflection of the outgoing pulse. The outgoing pulse back reflection and the detectors response to a large input signal spans two spectrums or range gates. The first six spectrums, which are too small to be seen in the center graph of the figure and precede the large spikes that are seen, where taken so as to allow for a system noise spectral density measurement to be made for each 500 ms accumulation if desired. Since the system does not have a flat spectral response when there is no return signal (system spectral noise density) the return signal spectrum must have the system's response taken in to account. The center plot of Figure 79 shows a close up view of the 9th through 29th spectrums while the lower plot shows a close up of the 29th spectrum. The 29th spectrum corresponds to the 21st atmospheric return signal due to the fact that the first eight spectrums are the result of the outgoing pulse or earlier. The frequency range of interest is also indicated. The range of interest spans 46.87 MHz to 125.0 MHz. The Lidar system was designed to measure frequency shifts +/- 30 MHz from the original frequency shift imposed upon the outgoing pulse by the AOMs, 84 MHz. The span would be from 54 to 114 MHz but in order to locate the frequency shift with sub FFT frequency bin resolution the points bellow and above the desired frequency span are necessary. With the FFT input data size of 128, the FFT frequency bin resolution is

$f_{bin} = f_s / N_{FFT} = 400 \times 10^6 / 128 = 3.125 \times 10^6$ MHz, or 4.8 m/s. Without sub bin frequency estimation the wind velocity estimates would be coarse. These spectral plots are not flat due to the response of the system components, such as the detectors, ADC, and filters.

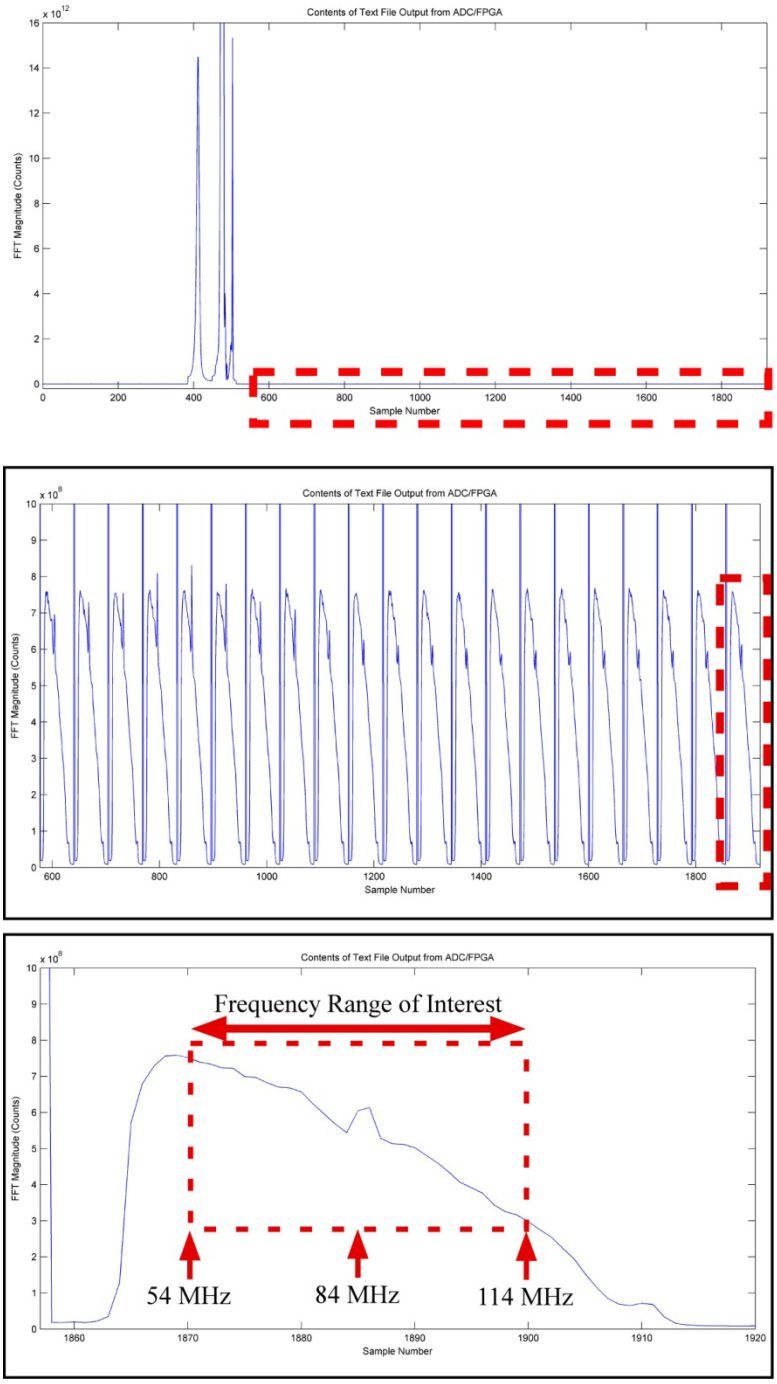


Figure 79, Close-up views from data acquisition card's FPGA text file output for 12/05/11 at 14:31:48. (Top) Close up view of first 500 ms accumulation first 30 spectrums. (Center) Close up view of 21 post outgoing pulse spectrums. (Lower) Close up view of spectrum from 29th gate (21st post outgoing pulse) spectrum with frequency range from 46.87 MHz to 125.0 MHz marked as frequency range of interest.

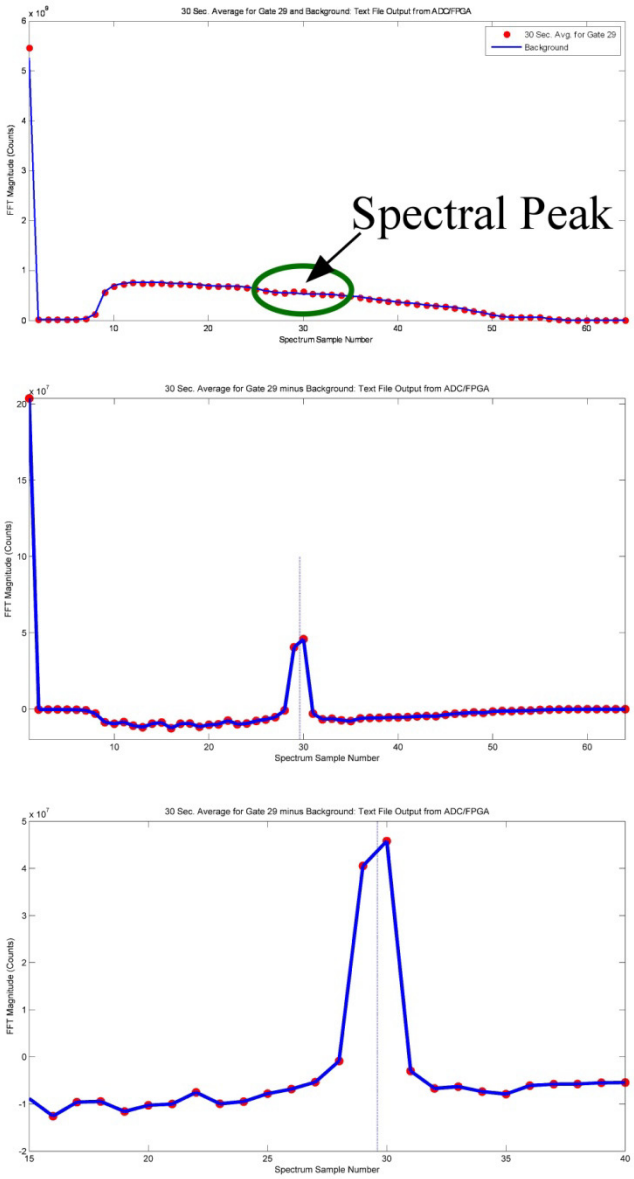


Figure 80, Various plots of the 30 second average of return signal from the 29th gate spectrum (21st atmospheric spectrums) and system noise spectral density spectrum. (Top) Plot of spectrum of return signal (points) and system noise spectral density spectrum (solid line). (Center) Plot of spectrum of return signal with system noise spectral density compensated for 0 Hz to 200 MHz. (Lower) Plot of spectral region of interest for return signal with system noise spectral density subtracted. Vertical line indicates the sub FFT frequency bin resolution estimate of the return signal center frequency.

The plots of Figure 80 show the 30 second average for the 29th gate, or 21st atmospheric return spectrum. The top plot shows spectrum of return signal (points) and system noise spectral density spectrum (solid line) on the same graph. The center plot shows the complete spectrum of

the return signal with the system noise spectral density compensated for while the lower plot shows the same but only for the spectrum region of interest. The lower plot also shows a vertical line that represents the sub FFT frequency bin resolution estimate of the center frequency of the return signal. With the FFT the spectrum of a pure sinusoidal would be a spectrum with as single point representing the frequency if the frequency is a multiple of the FFT frequency resolution and the number of points input to the FFT were to represent the sinusoidal without discontinuities. When the sinusoidal's frequency is not an integer multiple of the FFT frequency bin resolution the neighboring bins will also indicate spectral content. This artifact of the FFT could be reduced and the FFT frequency resolution increased by using more than 128 points but this would reduce the spatial range resolution of the system which is undesirable.

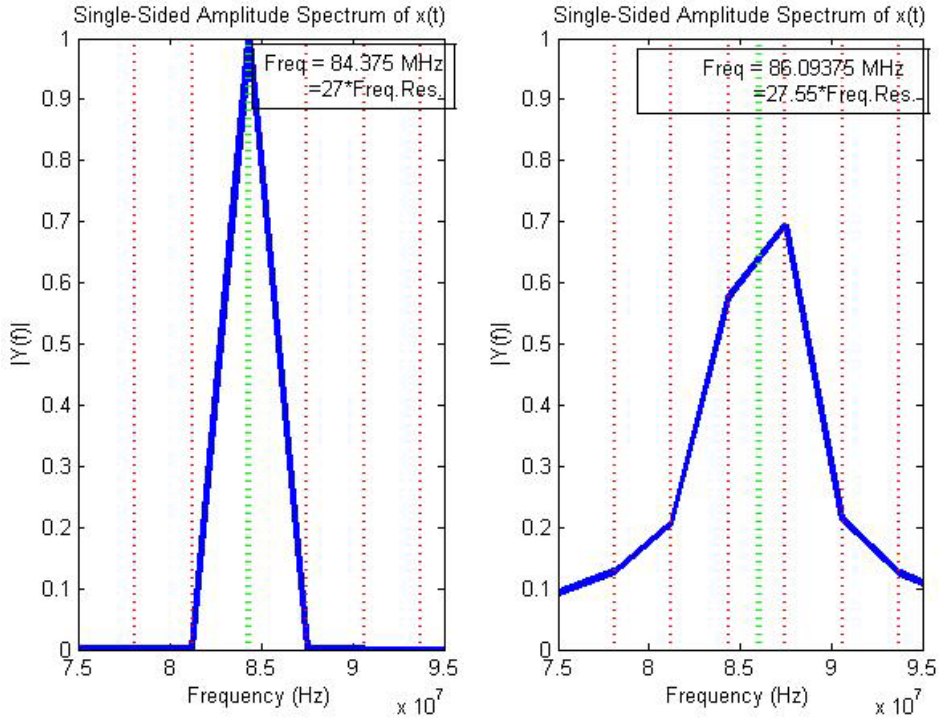


Figure 81, Examples of FFT of signal sampled at 400 MHz with a frequency of (left) 84.375 MHz, an exact multiple of the FFT frequency bin resolution and (right) 86.09375 MHz which is equal to the frequency of the signal on the left plus 0.55*FFT frequency bin resolution. The narrow red dotted vertical lines indicate the FFT frequency bins and the thick green vertical line indicates the exact frequency of the input signal.

The importance of estimating the frequency of the atmospheric return signal with sub-FFT frequency bin resolution is illustrated in Figure 81. The plot on the left illustrates how a signal sampled at 400 MHz and having a frequency, 84.375 MHz, which is exactly a multiple of the FFT frequency bin resolution, 3.125 MHz, will form a spectrum that is unambiguous as to the frequency of the signal. The thick vertical green dotted line indicates the frequency of the input to the FFT. If the number of points input to the FFT were to be increased from 128 to 256 the peak would not shift at all as the FFT bin resolution would be half of the current resolution and the signal frequency would still be an exact multiple of the FFT bin frequency resolution. The frequency difference between the true input signal and the neighboring bins would become smaller. Note that the triangular shape of the FFT does not indicate that the signal has a bandwidth of twice the FFT frequency bin resolution as the peak is a single point. The triangular shape is a result of using lines to represent the spectrum. The plot on the right of Figure 81 illustrates the results of an FFT where the input frequency is not an integer multiple of the FFT frequency bin resolution. The thick vertical green dotted line indicates the frequency of the input to the FFT. In order to achieve a finer frequency resolution than the FFT frequency bin resolution, without increasing the number of data points feed into the FFT, an estimation technique is used to estimate the frequency more precisely. This is discussed latter in this section.

They system may be operated in vertical measurement mode or scanning measurement mode. The vertical measurement mode makes measurements with laser pulse orientated normal to the earth's surface and the data is used to determine the vertical wind velocities. In the scanning measurement mode the laser beam is steered with the use of a mirror and the measurements along three line of sight pointing directions are used to calculate the horizontal and vertical wind

velocities. Before measurements can be made in either mode of operation the system's components are allowed to warm-up. The output levels, frequency settings, and other parameters are configured. Just before atmospheric data is to be collected some data is collected without the optical output active. This data is used to calculate the systems' spectral noise density and will be used to flatten or equalize the atmospheric return signal spectrums.

Vertical Measurement Mode

If the system is operating in a vertical measurement mode, the file on the host computer that has been created from the 30 seconds of data collection by the modified SNAP program is re-named with the date and time in the file name by a Matlab program, "test30s06.m". The Matlab program accomplishes this by monitoring the directory on the computer's hard drive where the modified SNAP program saves the data text files. Once the file has been closed by the modified SNAP programs it carries out the renaming procedure. At the beginning of vertical mode operations the modified SNAP program is instructed to take a predefined total number of measurements before self terminating. These files are later processed to determine the vertical wind vectors.

Scanning Measurement Mode

If the system is operating in a scanning mode, where the beam is steered to three locations in the sky by the rotation of a mirror in the optical setup, the procedure is slightly different than the vertical measurement mode. The equipment and lasers are warmed up and configured as previously described. A Matlab program known as "takelidar15.m" ushers the operator through the system configuration, recording measurements without optical output, collecting atmospheric data and rotating the steering mirror through the use of dialog boxes and messages. The measurements made without any active optical output provide data for creation of a spectrum of

the system's spectral noise density. This spectrum will allow the atmospheric signal spectrum to be equalized thus allowing the determination of the spectral peak.

The modified SNAP program is run as it would be in the vertical measurement mode, but only for 30 seconds periods before halting and transferring the data to a directory on the computer's hard drive. The Matlab program monitors the directory holding the text data file, reads the data, renames the data file, processes the data, and displays graphical representations of the signals and wind velocities to the operator as feedback during the data collection process.

Instructional dialog boxes and messages provided by the Matlab program guide the operator in great detail while the system is operating. Instructions are given in a step by step manner from the beginning of operations with the configuration of the system and the taking of data for the calculation of the systems spectral noise density to the end of operations with the termination of operations.

The Matlab program starts by having the operator confirm the DAQ card is properly configured, enter the cardinal orientation of the research vehicle, the location of the vehicle, the spectral frequency range of interest, the directory where data will be saved, the number of spectra that will be considered as not atmospheric in the data files, and comments or "meta" data. The first data file transferred from the DAQ card is re-named with the date and time as well as a suffix that designates it as a background measurement (system spectral noise density). The operator is instructed to bring the laser and optical amplifier up to operating parameters. Then they are instructed to position the steering mirror and to activate the modified SNAP program. When the modified SNAP program has finished writing the data file of the last 60 accumulations of 10,000 pulses each to the hard drive of the host computer, the Matlab program recognizes this, renames

the file with the date and time. The spectrums that have been recorded in the text file are processed to generate a line of sight (LOS) wind velocity for each range bin. These are saved to the computer as text files with the same date and time stamp as the accumulated spectrums but with the addition of a suffix "_BeamXVel" where "X" indicates the beam number. The beam number indicating whether the beam was steered vertically, or the mirror was turned a positive or negative rotation to steer the beam. The Matlab program prompts the operator to move the steering mirror to the next position in the sequence of measurements. The sequence is vertical (Beam 0), positive rotation (Beam 1), negative rotation (Beam 2) and then the cycle repeats.

The Matlab program allows the operator to record comments with each set of measurements allowing them to indicate conditions of the system or the environment. The Matlab program also records the operating conditions, parameters of the system, its own file name and other information that may be useful when looking at the data after collection.

The line of sight velocities are calculated from the spectral peak seen in each range gates and its shift from the original 84 MHz frequency shift imposed by the AOMs. In order to locate the peak, the background spectrum or spectral noise density must be taken into account. If the spectrum was not equalized a value other than the true peak may be mistakenly used to determine the wind's velocity. The measurements made prior to the start of atmospheric data collection may be used for this purpose, but the data collected just prior to the outgoing laser pulse reflection may also be used. The digital delay generator can be configured to start the data collection prior to the optical pulse reaching the optical circulator and creating a back reflection which will be detected and processed. The data prior to the back reflection or pre-trigger data can be processed and used for spectral equalization.

Once the signal spectrum is equalized the frequency of the peak is estimated and not assumed to be the largest point in the spectrum as the spectral resolution resulting from a 128 point FFT is course for velocity estimation. Interpolation is performed using [112]

$$k_{peak} = k + \delta \tag{3.3.34}$$

$$f_{tone} = \frac{k_{peak} f_s}{N} \tag{3.3.35}$$

where f_s is the sampling rate in Hz, N is the number of points used in the FFT, and δ is the frequency offset from the bin frequency

$$\delta = \frac{\left(|X_{k+1}| - |X_{k-1}|\right)}{\left(4|X_k| - 2|X_{k-1}| - 2|X_{k+1}|\right)} \tag{3.3.36}$$

Once the beam frequency shifts are calculated the beam velocities can be calculated. From the Beam velocities the vertical and horizontal wind components can be found. From these, the wind speed and direction can be found. The line of sight beam velocities are determined from the frequency shift, Δf, using

$$\Delta f = f_{tone} - f_0 \tag{3.3.37}$$

where f_0 is the original frequency shift of the outgoing light. The velocity of the wind is calculated as

$$velocity = \frac{c}{2}\frac{\Delta f}{f_0} = \frac{\lambda_0 \Delta f}{2} \tag{3.3.38}$$

CHAPTER 4
Analysis

In the development of the Doppler wind Lidar system the selection of components, wavelength of operation, means of detection, and the general design of the system have been made to maximize performance and minimize cost and system size, while remaining eye safe. There is a balance to be struck, leveraging desired characteristics of the system against the destroyer of performance, "noise", while achieving multiple goals. Noise is the phenomenon that makes long range wind measurements difficult.

Noise is randomness. Random fluctuations in a source or a measurement that are unwanted are noise. The term noise is often loosely applied to deterministic signals that are present, but these are not noise. Spurious signals, interference, and intermodulation distortions are not noise as they lack randomness. In this system there are physical sources, electrical sources and optical sources of noise. All arise from randomness in such things as the electrons flowing in a detector, the non-uniform movement of wind, spontaneous photon generation, and many more. The following will describe signals, noise, how they are represented, the types of noise, and conventions of representing and calculating noise.[113-115]

Signals and Noise Representation

An ideal oscillator would generate a pure sinusoid

$$v(t) = V_0 \cos(\omega_0 t + \varphi) \tag{4.1.1}$$

where the peak amplitude is equal to the root mean square voltage multiplied by the square root of two, ($V_0 = \sqrt{2} V_{rms}$), $\omega_0 = 2\pi v_0$ is the angular frequency, and φ is a constant. The pure

sinusoid can be represented as a complex number in the Cartesian coordinate system as a phasor after factoring out the $\omega_0 t$. The phasor, $\boldsymbol{V} = A + jB$, where the absolute value $|\boldsymbol{V}|$ is the rms value of v(t), and phase arg $\boldsymbol{V}$ is to φ. A pure sinusoidal signal may also be represented as

$$\begin{aligned} \boldsymbol{V} &= \frac{V_0}{\sqrt{2}} e^{j\varphi} \\ \boldsymbol{V} &= \frac{V_0}{\sqrt{2}} \left(\cos\varphi + j \sin\varphi \right) \end{aligned} \tag{4.1.2}$$

The pre-envelope or analytical signal z(t) is a useful tool as it treats the negative frequencies of a spectrum as superfluous. While the phasor is restricted to time invariant amplitude, phase, and frequency, the analytical signal is not. The analytical signal uses the Hilbert transform which in the case of v(t) is equivalent to a shift by 90 degrees and is represented by $\hat{v}(t)$. The analytical signal can be written as

$$z(t) = v(t) + j\hat{v}(t) \tag{4.1.3}$$

The analytical signal of v(t) is obtained by doubling the positive frequency components of the spectrum and eliminating the negative frequencies of the spectrum. Total power can be calculated with only positive frequencies and a phase shift by θ can applied to v(t) by multiplying z(t) by $e^{j\theta}$. It can also be obtained by

$$v(t) = \frac{V_0}{\sqrt{2}} \cos(\omega_0 t + \varphi) \Rightarrow \begin{cases} z(t) = V(t) e^{j[\omega_0 t + \varphi(t)]} \\ z(t) = V(t) e^{j\varphi(t)} e^{j\omega_0 t} \\ z(t) = V(t)(\cos\varphi + j\sin\varphi) e^{j\omega_0 t} \end{cases} \tag{4.1.4}$$

The complex envelope of z(t) is used when the amplitude and phase vary with time and obtained by removing the complex oscillation ($e^{j\omega_0 t}$) from analytical signal.

$$\boldsymbol{V} = \frac{V_0}{\sqrt{2}} e^{j\varphi} \qquad \Leftrightarrow \qquad \hat{v}(t) = V(t) e^{j\varphi(t)} \tag{4.1.5}$$

Real world oscillating signals have fluctuations in their amplitude and their phase. A sinusoidal signal with a high signal to noise ratio is often used as a clock or local oscillator. To make (4.1.1) more realistic small perturbations can be represented in the signal as

$$v(t) = V_0 \left[1 + \alpha(t)\right] \cos(\omega_0 t + \varphi(t)) \qquad |\alpha(t)| \ll 1, \qquad |\varphi(t)| \ll 1 \tag{4.1.6}$$

Amplitude fluctuations in oscillators are small, on the order of 10^{-3} to 10^{-6}, for the fractional amplitude $|\alpha(t)|$. The phase fluctuation $|\varphi(t)|$ is the main source of clock error from an oscillator. Phase which is slowly varying is referred to as drift.

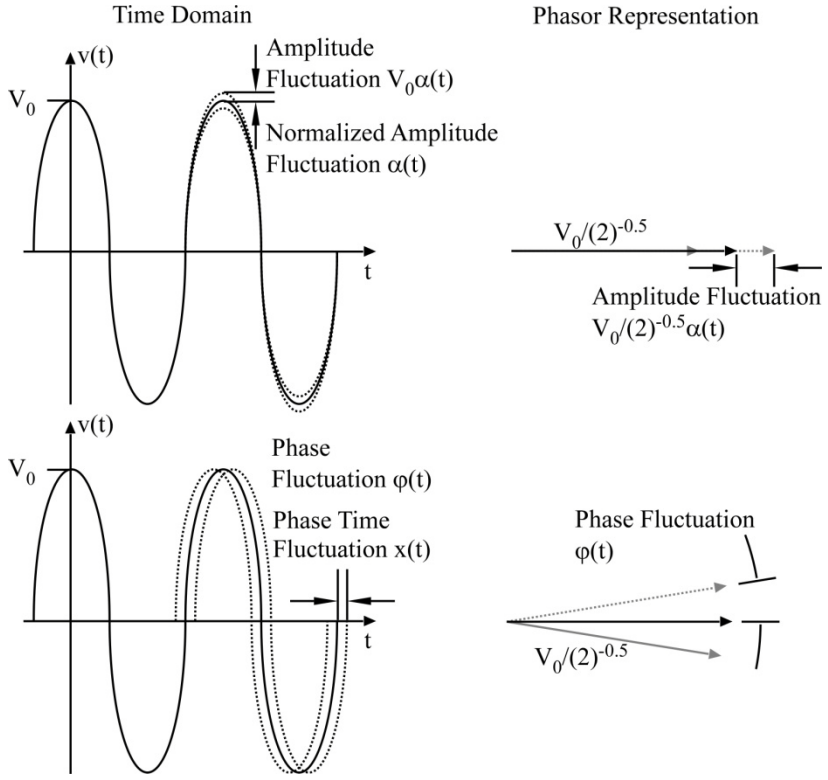


Figure 82, Time domain and phasor representation of amplitude and phase fluctuations.

Rewritten, the oscillating signal in Cartesian coordinates gives

$$v(t) = V_0 \cos \omega_0 t + v_x(t) \cos \omega_0 t - v_y(t) \sin \omega_0 t \tag{4.1.7}$$

The in-phase component of the noise is $v_x(t)$ and the quadrature component is $v_y(t)$. Equations (4.1.6) and (4.1.7) are related by the fractional amplitude and phase

$$\alpha(t) = \sqrt{\left[1 + \frac{v_x(t)}{V_0}\right]^2 + \left[\frac{v_y(t)}{V_0}\right]^2} - 1 \tag{4.1.8}$$

$$\varphi(t) = \arctan \frac{v_y(t)}{V_0 + v_x(t)} \tag{4.1.9}$$

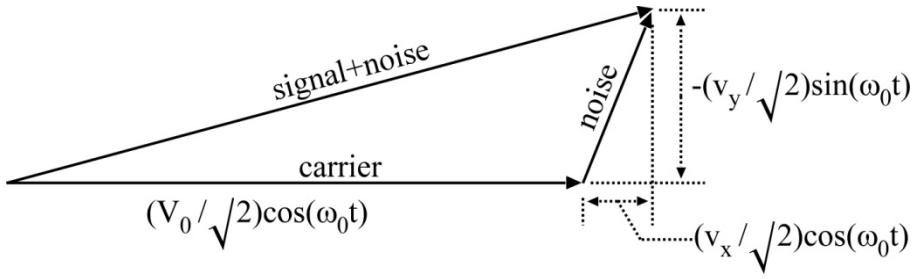


Figure 83, Noisy sinusoidal signal represented as a phasor. The noisy signal can be viewed as the sum of the noise and the signal (carrier).

A phasor representation of a noisy sinusoidal signal is show in Figure 83. The noisy signal is the sum of the signal or carrier and the noise. As the phase and amplitude vary with time the due to noise, the result would be a line with width that is proportional to the bandwidth widening of the signal caused by the noise. An ideal pure sinusoid would a single thin line. The signal power is not affected by random phase $\varphi(t)$. The instantaneous power is $P(t) = dWork / dt = v(t)i(t) = v^2(t) / R_0$. If the power is averaged over a time T_m which is longer than the period of the sinusoid and shorter than the time scale of the amplitude fluctuation, T_α, the power, in a low noise arrangement will be

$$\overline{P(t)} = \frac{V_0^2}{2R_0}\left[1 + 2\alpha(t)\right], \quad for \quad \left|\alpha(t)\right| \ll 1, \frac{2\pi}{\omega_0} \ll T_m \ll T_\alpha \tag{4.1.10}$$

Noise Statistics

A random or stochastic process is described by random variable(s) that describe a system's behavior over time. The random process has an infinite number of realizations. These form an ensemble. Through experiments a number of realizations of the random process can be determined. The simple mean, time average, and expectation are three types of "averages" used to measure random processes.

If all of a random process' statistical properties are unchanging or invariant with arbitrary shifts in time then it is said to be strict sense stationary. A weak or wide sense stationary process when the 1st moment (expectation) and 2nd moment (variance) are unchanging with shifts in time. If a system is stationary then repeatability is possible with enough experiments.

When dealing with signals that have statistical properties that are cyclic with time, cyclostationary processes can describe them. A random process is cyclostationary of period T if all of its statistical properties are invariant with a time shift of nT with n equal to an integer $\left(\mathbf{x}(t) \to \mathbf{x}(t+nT)\right)$. A cyclostationary process may be viewed as many stationary processes woven together. The maximum temperature recorded on a given day at New York city's Central Park weather station can be seen as a cyclic process. The peak temperature on a specific winter's day is statistically different than a peak temperature on a specific summer's day. But the statistics of the specific winter's day are statistically the same year to year.

A process is ergodic if the statics of the process can be estimated from a sufficiently long realization. If a process is not stationary or cyclostationary the statistics of the process could not be estimated from a single realization and therefore could not be ergodic. Reproducibility is related to the ergodicity. Measurements made of a process by different operators at different times should have the "same results" as each measurement or experiment is just one realization of the ensemble. The reproducibility is expressed in the spread of values of the different realizations. If the statistics of a coin toss were done with a fair coin in multiple locations the statistics of each realization are expected to be close if a sufficient number of tosses were made.

Normally measurements made of a system relate the input and output so that an analysis of the transfer function or system function can be made. In order be able to make this analysis the system is assumed to be linear or at least locally linear so that,

$$L\{av_i\} = aL\{v_i\} \tag{4.1.11}$$

$$L\{v_1 + v_2\} = L\{v_1\} + L\{v_2\} \tag{4.1.12}$$

and time invariant,

$$v_0(t) = L\{v_i(t)\} \Leftrightarrow v_0(t+t') = L\{v_i(t+t')\} \tag{4.1.13}$$

A system that is linear and time invariant can completely be described by its impulse response, which is the response to a Dirac delta function, δ(t), and is denoted as h(t). The convolution of the system response, or transfer function, with an input signal $v_i(t)$, is equal to the system's output signal $v_o(t)$. In the time domain the convolution is

$$\begin{aligned} v_0(t) &= h(t) * v_i(t) \\ &= \int_{-\infty}^{+\infty} v_i(\tau) h(t-\tau) d\tau \end{aligned} \tag{4.1.14}$$

In the frequency domain the output is equal to the input multiplied by the transfer function. The upper case letters denote the Fourier transforms of the lower case time domain functions into the frequency domain,

$$V_o(j\omega) = H(j\omega) V_i(j\omega) \tag{4.1.15}$$

In a linear time invariant system the response to $e^{j\omega}$ is an important property. The exponential $e^{j\omega}$ is an eigenfunction making the response to $v_i(t) = e^{j\omega t}$ be $v_o(t) = Ce^{j\omega t}$. The complex constant C is equal to H(jω),

$$v_i(t) = e^{j\omega t} \Leftrightarrow v_o(t) = H(j\omega) e^{j\omega t} \tag{4.1.16}$$

The frequency response of a system is often plotted as a Bode plot were the logarithmic frequency axis is argH(jω) and the transfer function is plotted as $20\log_{10}\left|H(j\omega)\right|$ in decibels (dB). [116-118] When the input to the system is a random process the input and output Fourier transforms are replaced by the power spectra $\left(V_i(j\omega) \to S_i(\omega), V_o(j\omega) \to S_o(\omega)\right)$ and (4.1.15) becomes

$$S_o(\omega) = \left|H(j\omega)\right|^2 S_i(\omega) \tag{4.1.17}$$

With a causal system, where the output is zero prior to an input, the system transfer function is causal (h(t)=0 for t<0). The Laplace transform is useful when dealing with causal systems. The Fourier transform

$$X(j\omega) = \int_{-\infty}^{+\infty} x(t)e^{-j\omega t}dt \tag{4.1.18}$$

becomes

$$X(j\omega) = \int_{0}^{+\infty} x(t)e^{-j\omega t}dt \quad for\, x(t) = 0, \forall t < 0 \tag{4.1.19}$$

By setting $s = \sigma + j\omega$ the Laplace transform is found which allows the frequency to be complex,

$$X(s) = \mathcal{L}\{x(t)\} = \int_{0}^{+\infty} x(t)e^{-st}dt \tag{4.1.20}$$

If a system is observed over a period T which is sufficiently long enough to include almost all of the energy of h(t) the Fourier and Laplace transform can be evaluated over 0 to T. If a system is time variant but the fluctuation is slow as compared to the observation time T the system can be treated as time invariant over 0 to T. A slowly varying system can be described in terms of roots that fluctuate in the complex plane if Laplace transforms are used.

The spectrum of a clock is broadened by noise with interesting noise phenomena clustered around the carrier frequency. Low frequency processes of amplitude and the phase or frequency fluctuations describe the noise seen in an oscillator.

Phase Noise

Phase noise is often used to describe the rapid, short term, random fluctuations of the phase of a waveform. The one-sided power spectral density of the random phase fluctuations, $\varphi(t)$, define the phase noise $S_{\varphi}(f)$ and has units of rad^2/Hz. Phase time is defined as $x(t) = \varphi(t)/2\pi v_0$ which is used find the phase time power spectral density, which has units of s^{2}Hz or s^3,

$$S_x(f) = \frac{1}{(2\pi\nu_0)^2} S_\varphi(f) \tag{4.1.21}$$

Manufacturers do not tend to use $S_x(f)$ but half of the $S_\varphi(f)$. This is defined as

$$\mathcal{L}(f) = \frac{1}{2} S_\varphi(f) \tag{4.1.22}$$

This is often given in units of decibels bellow the carrier in 1 Hz bandwidth, dBc/Hz where $10\log_{10}(S_\varphi(f)/2)$ or $10\log_{10}(S_\varphi(f)) - 3dB$ is used.

The power law function

$$S_\varphi(f) = \sum_{i\leq -4}^{0} b_i f^i \tag{4.1.23}$$

is useful for describing oscillator phase noise. The index and the noise process they coincide with are shown in Table 10. Oscillators have all of the terms seen in the table while two port devices have white noise and flicker noise. It is convention to give the coefficients in units of rad^2/Hz.

Table 10, Frequently encountered phase noise processes. Note: The coefficient is usually given in units of rad^2/Hz.[114]

Law	Slope	Noise Process Description	Units of bi
$b_0 f^0$	0	White phase noise	rad^2/Hz
$b_{-1} f^{-1}$	-1	Flicker phase noise	rad^2
$b_{-2} f^{-2}$	-2	White frequency noise (or random walk of phase)	rad^2 Hz
$b_{-3} f^{-3}$	-3	Flicker frequency noise	rad^2 Hz^2
$b_{-4} f^{-4}$	-4	Random walk of frequency	rad^2 Hz^3

Frequency Noise

Randomness in frequency fluctuations, $(\Delta\omega)(t)$ or $(\Delta v)(t)$, instead of the phase fluctuation description is sometimes used. Because $\Delta v = (1/2\pi)\Delta\omega$ and $S_{\Delta v} = (1/4\pi^2)S_{\Delta\omega}$

$$S_{\Delta v}(f) = f^2 S_{\varphi}(f) \tag{4.1.24}$$

With the fractional frequency fluctuation defined as $y(t) = (\Delta v)(t)/v_0$

$$S_y(f) = \frac{f^2}{v_0^2} S_{\varphi}(f) \tag{4.1.25}$$

$S_{\Delta v}(f)$ is usually used when referring to lasers and $S_y(f)$ when referring to radio frequency discussions. In the literature the coefficients of $S_y(f)$ are denoted by h_i. Since the power law (4.1.23) also applies

$$S_y(f) = \sum_{i=-2}^{2} h_i f^i \tag{4.1.26}$$

with h_i in units of Hz^{-1}, as convention instead of proper units,

$$h_i = \frac{1}{v_0} b_{i-2} \tag{4.1.27}$$

Amplitude Noise

Unwanted fluctuations in a signal's amplitude can be a concern in systems using amplitude modulation. In an oscillator, the variations in amplitude may result from nonlinearities in a quartz crystal that are inherent to the lattice structure, which can affect the resonant frequency.

The power law can describe the amplitude noise of a clock signal and a two port device. The dominant noise (f^0) is white noise and flicker noise (f^{-1}).

Noise Sources in Semiconductors and Amplifiers

Noise is present in electronic circuits. Noise can be additive, as white (uniform) noise seen in thermal noise, or it can be parametric, such as low frequency non-uniform flicker noise resulting from impurities in conductive channels of solid state electronics. Even though flicker noise is low frequency it depends on the type of device, and the low frequency noise can be mixed up to the higher frequencies of an oscillator.

Thermal Noise

Thermal noise in a resistor was first observed in 1928 by Johnson [119] after being predicted by Einstein in 1906. Thermal noise is a result of thermal agitation of the electrons inside the conducting medium. This noise is known also known as Johnson noise and Nyquist noise. “The mean square potential fluctuation over the conductor is proportional to the electrical resistance and the absolute temperature of the conductor."[119] This noise is nearly constant across the frequency range and exhibits a Gaussian distribution.

The power spectral density of a blackbody radiation

$$S(v) = \frac{hv}{e^{hv/kT} - 1} \qquad (4.1.28)$$
$$\approx kT \quad for\ hv \ll kT$$

where T is temperature in Kelvin, k is Boltzmann's constant (1.38 x 10^{-23} J/K), h is Plank's constant (6.6260 x 10^{-34} J s). The blackbody radiation is white noise at low frequencies below the cutoff frequency kT/h.

Resistor noise is the result of coupling the blackbody radiation of the resistor into a circuit. The power spectrum density is usually designated as N=kT. The spectrum of the random voltage available at the ends of a resistor in V^2/Hz if the resistor is impedance matched to a load is,

$$S_v(v) = kTR \tag{4.1.29}$$

where R is the resistance (Ω). If the resistor is open then

$$S_v(v) = 4kTR \tag{4.1.30}$$

The IEEE standard specifies the temperature standard, T_0,as 290 °K. Thermal noise in a resistor is typically modeled as a voltage noise source in series with an ideal resistor or as a current noise source in parallel with an ideal resistor as illustrated in Figure 84.

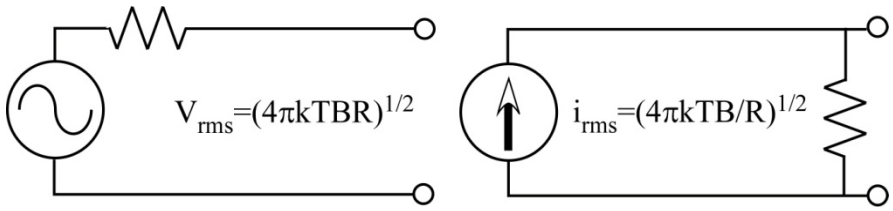


Figure 84, Schematic representation of models of thermal noise in a resistor.

The open circuit rms voltage and the short-circuit rms current at the terminals of a resistor are

$$V_{rms} = \sqrt{4kTBR} \tag{4.1.31}$$

$$i_{rms} = \sqrt{4kTBG} = \sqrt{\frac{4kTB}{R}} \tag{4.1.32}$$

where B is the bandwidth and G is the inverse of resistance, conductance. Although less common it is possible to define the sources without the resistors by a noise resistance or a noise conductance,

$$G = \frac{\overline{|i|^2}}{4kT_0B} \qquad \boxed{\text{Noise Conductance}} \tag{4.1.33}$$

$$R = \frac{\overline{|v|^2}}{4kT_0B} \qquad \boxed{\text{Noise Resistance}} \tag{4.1.34}$$

Shot Noise

The conduction of electricity across a barrier generates a pulse of current due to the fact that the flow of electricity is actually a stream of individual electrons. Each electron has a charge q (1.602×10^{-19} C) and creates a pulse. The fluctuation is a noise process and the mean square value of the noise is

$$\overline{|i|^2} = 2qIB \tag{4.1.35}$$

where I is the junction current. Shot noise is a white noise with a flat spectrum at frequencies where the transit time of the junction is short compared to the inverse frequency. The spectral density decreases at higher frequencies. The spectrum of the shot noise is

$$S_i(v) = 2q\bar{i} \qquad (\text{A}^2/\text{Hz}) \tag{4.1.36}$$

where $\bar{i}$ is the average current. The power spectrum of the random current i(t) flowing into a resistor is

$$S_i(v) = 2q\bar{i}R \quad \text{(W/Hz)} \tag{4.1.37}$$

Flicker Noise (1/f Noise)

Flicker noise, also known as excess noise and contact noise, is present in a wide variety of phenomena. It has been found in Nile flooding, lake turbulence, all solid state devices and some resistors (carbon composite type). It is present at low frequencies and has been observed to follow the spectrum scaling of $1/f^x$ with x equal to 0.8 to 1.2.

Burst Noise

Burst noise is also known as popcorn noise, impulse noise, bi-stable noise, or random telegraph signal noise. It consists of a sudden step in voltage at random intervals. It is seen in certain types of transistors, point contact diodes, and some semiconductor op-amps. The theory to explain this phenomenon does not seem to explain all of the places it is seen. The cause is theorized to be the trapping and release of charged carriers at thin film interface or at defect sites in semiconductor crystals. Burst noise has a $1/f^2$ spectrum with a low frequency cutoff.

Noise Temperature and Noise Figure

The noise temperature and noise figure are often used to characterize an amplifier or receiver. An amplifier's noise spectrum is defined in terms of the input and equivalent noise temperature T_a

$$N_a = kT_a \tag{4.1.38}$$

where k is Boltzmann's constant (1.38×10^{-23} J/K). The input to the amplifier is terminated with a load that itself is a source of noise, making the total equivalent noise of the amplifier

$$\begin{aligned} N_e &= N_a + N_r \\ &= k(T_a + T_r) \end{aligned} \tag{4.1.39}$$

where the index r refers to the input termination. An older method of characterizing noise of a two port device is the noise figure, which is given in decibels. A related term, noise factor is the same quantity but as a ratio. The noise figure is the signal to noise ratio of the input divided by the signal to noise ratio of the output. The noise figure is,

$$F = \frac{SNR_{input}}{SNR_{output}} \tag{4.1.40}$$

A common practice definition of the noise figure is to use the standard temperature and the total equivalent noise,

$$F = \frac{N_e}{kT_0} \tag{4.1.41}$$

By having the input termination of a two port device at standard temperature, $T_r = T_0$, the noise figure can be written as

$$F = \frac{T_0 + T_a}{T_0} = 1 + \frac{T_a}{T_0} \tag{4.1.42}$$

The amplifier's noise spectrum can be re-written by solving (4.1.42) for T_a,

$$T_a = (F - 1)T_0 \tag{4.1.43}$$

and then substituting into (4.1.38)

$$N_a = (F - 1)kT_0 \tag{4.1.44}$$

By using noise figure calculations the resistor thermal noise can be replaced by the equivalent noise

$$N_e = FkT_0 \tag{4.1.45}$$

When amplifiers are cascaded, the noise feed into the sequence of amplifiers is also amplified. The Friis formula uses the noise figure of each stage (F_i) and the power gain of each stage A_i^2 to calculate the noise figure of the amplifier series. The Friis formula is

$$F = F_1 + \frac{F_2 - 1}{A_1^2} + \frac{F_3 - 1}{A_1^2 A_2^2} + \frac{F_4 - 1}{A_1^2 A_2^2 A_3^2} + ... \tag{4.1.46}$$

From the formula it can be seen that the noise figure of the first amplifier has the greatest influence on the final calculation of noise for the cascaded amplifiers. In a low noise amplifier the noise figure is affected by the technology used, the bandwidth, and the frequency. A typical low noise amplifier has a noise figure from 0.5 dB to 2 dB while power and low cost amplifiers have higher noise figures.

Noise in Photodetector

When a photodiode is reverse biased and a photon is detected an electron hole pair is created. Using the Shockley–Ramo theorem allows the shape of the current pulses at the detector ends to be calculated such that the charge transferred is q. The photo current created by light with optical power P(t) at frequency *v* is

$$i(t) = q\eta \underbrace{\frac{P(t)}{hv}}_{\substack{\text{Number of} \\ \text{photons per} \\ \text{second}}} \tag{4.1.47}$$

where η is the detector's quantum efficiency, h is Plank's constant (6.6260 x 10^{-34} J s) and q is electron charge (1.602x10^{-19} Coulombs). The responsivity of the detector is defined as

$$p = \frac{q\eta}{hv} \qquad \text{(A/W)} \tag{4.1.48}$$

The shot noise power spectral density of a reversed biased photodetector can be found by substituting the time average current of (4.1.47) into (4.1.36)

$$S_i(v) = 2q^2\eta\frac{\overline{P}}{hv} \qquad (\text{A}^2/\text{Hz}) \tag{4.1.49}$$

If a laser beam with optical power of P=1 mW at wavelength 1545.2 nm were to strike a detector which had an responsivity of 0.8 A/W, quantum efficiency is then η=0.642, the photocurrent is i=0.799 mA, the power spectral density of the shot noise S_i = 2.563 x 10^{-22} A^2/Hz, and $\sqrt{s_i} = 16.01\,\text{pA}/\sqrt{\text{Hz}}$. When the photo current flows into a 50 ohms resistor the noise is 1.281x10^{-20} W/Hz or 6.40x10^{-19} V^2/Hz or $800\mu\text{V}/\sqrt{\text{Hz}}$.

Signal-to-Noise of Analog to Digital Converter

The analog to digital converter (ADC) on the data acquisition card is specified at 400 mega-samples a second with 14 bits of resolution. Each digitized sample of the input voltage requires two bytes to represent the voltage. Even though 14 bits are specified, the device does not actually have use of all 14 bits. Since the theoretical maximum signal-to-noise ratio (SNR) achievable by the ADC depends on the number of bits, the reduction in the number of bits from 14 affects the theoretical maximum SNR. The theoretical maximum SNR is also affected by how the signal is sampled.

An ADC samples the signal and converts the voltage to a number. [120] This requires quantization as the span of the input to the ADC must be represented by the number of bits available. The number of quantization levels is 2^N where N is the number of bits. For a 14 bit ADC there are 16383 (=2^N-1) levels to describe the span of the input voltage. For a system with input from -1 V to +1 V and 14 bits each bit would represent q = 122 μV. When the signal is between quantization levels there is an error between what the ADC outputs and what the true analog signal voltage is. The error has an approximate saw tooth form oscillating between a positive half quantization level and a negative half quantization level. The error is

$$e(t) = st, \qquad -q/2s < t < +q/2s \quad \text{where s is the slope} \tag{4.1.50}$$

$$e(t) = st, \qquad -q/2s < t < +q/2s \quad \text{where s is the slope} \tag{4.1.51}$$

The mean square error of e(t) is

$$\overline{e^2(t)} = \frac{s}{q}\int_{+q/2s}^{-q/2s} (st)^2\, dt = \frac{q^2}{12} \tag{4.1.52}$$

Making the root mean square (rms) quantization error

$$e_{rms}(t) = \sqrt{\overline{e^2(t)}} = \frac{q}{\sqrt{12}} \tag{4.1.53}$$

The noise is assumed to be Gaussian and uniformly spread over the spectrum of 0 Hz to half the sampling rate (f_s/2). For a full scale (FS) sinusoidal input the theoretical SNR can be calculated. The full scale input is

$$v(t) = \frac{q2^N}{2}\sin(2\pi ft) = A_0 \sin(2\pi ft) \tag{4.1.54}$$

The rms value of the input signal level A_0 is $A_{0,rms} = q2^N / 2\sqrt{2}$. The theoretical rms SNR for the ideal ADC over the bandwidth of 0 Hz to $f_s/2$ is

$$SNR = 20\log_{10}\frac{A_{0,rms}}{\sqrt{\overline{e^2(t)}}} = 20\log_{10}\frac{q2^N/2\sqrt{2}}{q/\sqrt{12}} \qquad (4.1.55)$$
$$SNR = 6.02N + 1.76dB$$

If the signal of interest has a bandwidth less than the full bandwidth, and digital filtering is used to filter out noise components outside of the bandwidth of interest, then the theoretical SNR is increased. A correction factor known as a process gain is included in the SNR equation. When a signal is sampled at greater than twice the bandwidth of the signal, the sampling is referred to as oversampling. Sampling at four times the signal bandwidth is equivalent to increasing the number of bits by 1. The oversampled theoretical SNR is

$$SNR = 6.02N + 1.76dB + \underbrace{10\log_{10}\frac{f_s}{2B}}_{\text{Process Gain}} \qquad (4.1.56)$$

A 14 bit ADC, sampling a signal with a bandwidth of 60 MHz (54-114 MHz), at 400 MHz would have a theoretical SNR of

$$SNR = 6.02N + 1.76dB + 10\log_{10}(f_s / 2B) = 6.02(14) + 1.76 + 5.22 = 91.3\ dB \qquad (4.1.57)$$

When a fast Fourier transform is performed on a signal there is a gain in SNR due to the FFT being performed on a number (M) of samples. The additional reduction in the noise floor is

$$10\log_{10}(M/2) \qquad \text{in dB} \qquad (4.1.58)$$

If 128 sample points of a full scale sinusoidal signal were collected by a 14 bit ADC then the noise floor when an FFT was performed with the 128 points would be

$$SNR = 6.02N + 1.76dB + 10\log_{10}(M/2) = 6.02(14) + 1.76 + 18.0 = 104.1\ dBc \qquad (4.1.59)$$

An ADC total harmonic distortion (THD) is the ratio of the rms value of the fundamental signal to mean value of the root sum square of its harmonics. The first five harmonics are generally used as they are more significant. An ADC total harmonic distortion plus noise (THD+N) is the THD plus the noise components excluding the DC component. The signal to noise and distortion ratio, SINAD, is the rms signal amplitude of the mean value of the root sum square of all spectral components including harmonics but for the DC. The effect of distortion on the ADC is to effectively reduce the number of bits. The effective number of bits (ENOB) is calculated

$$ENOB = \frac{SINAD - 1.76\ dB}{6.02} \qquad (4.1.60)$$

The analog to digital converter used on the data acquisition card is the ADS5474, 400 mega samples per second (MSPS), 14 bit resolution, 11.2 effective number of bits, with a signal to noise ratio of 69.8 dB full scale. [111] The specifications of the converter are also specified by the manufacturer of the data acquisition card, Innovative Integration. They are a lower than the performance stated by Texas Instruments. The ENOB is 10.1 bits, spurious free dynamic range (SFDR) is 67.9 dB, harmonic distortion is -65.1 dB, and a SNR of 73.6 dB from a 64K FFT. [121] The SINAD can be calculated from the stated ENOB,

$$\begin{aligned} SINAD &= ENOB(6.02) + 1.76\ dB \\ &= 10.1(6.02) + 1.76 = 62.562\ dB \end{aligned} \qquad (4.1.61)$$

The non-ideal behavior of the ADC and the data acquisition card has significantly affected the expected performance from a 14 bit ADC. The FFT spectrum of 131072 points taken with the inputs terminated with 50 Ohm loads is shown in the Analog to Digital Converter with Field Programmable Gate Array Card section of the components section. The spectrum is approximately 111 dB bellow the full scale input. This would seem to contradict the manufacturers of the ADC chip and the DAQ card but if the number of points included in the FFT calculation is taken into account the graph is in reasonable agreement with the DAQ card specifications. With the additional suppression of the noise floor due to the number of samples in the FFT, (4.1.58), SNR of the DAQ card is

$$\begin{aligned} SNR &= 6.02N + 1.76dB + 10\log_{10}(M/2) \\ &= 6.02(10.1) + 1.76 + 48.2 = 112.4\ dB \end{aligned} \tag{4.1.62}$$

This is close agreement with the measured noise floor when the input is terminated by a 50 Ohm load.

The ENOB and the SINAD are related by [122, 123]

$$\begin{aligned} ENOB &= \log_2(SINAD) - \frac{1}{2}\log_2(1.5) - \log_2\left(\frac{A}{V_{full}/2}\right) \\ &= \log_2(SINAD) - \log_2(\sqrt{1.5}) - \log_2\left(\frac{2A}{V_{full}}\right) \\ &= \log_2\left(\frac{SINAD}{\sqrt{1.5}\,2A/V_{full}}\right) \end{aligned} \tag{4.1.63}$$

where A is the amplitude of the sinusoidal input to the ADC. This can be rewritten as

$$
\begin{aligned}
ENOB &= \log_2(SINAD) - \log_2(\sqrt{1.5}) - \log_2\left(\frac{2A}{V_{full}}\right) \\
&= \log_2\left(\frac{SINAD}{\sqrt{1.5}\,2A/V_{full}}\right)
\end{aligned}
\tag{4.1.64}
$$

$$
\begin{aligned}
SINAD &= \sqrt{1.5}\,\frac{2A}{V_{full}}2^{ENOB} \\
&= 20\log_{10}\left(\sqrt{1.5}\,\frac{2A}{V_{full}}2^{ENOB}\right) \quad \text{in dB} \\
&= 20\log_{10}\left(\sqrt{1.5}\;2^{ENOB}\right) + 20\log_{10}\left(\frac{2A}{V_{full}}\right) \qquad \text{in dB}
\end{aligned}
\tag{4.1.65}
$$

The relationship between ENOB and SINAD is easily seen in (4.1.65). If the input signal amplitude covers the full scale of the ADC voltage span then the second term in (4.1.65) is 0 and there is no reduction in the SINAD. If this is not true then the SINAD decreases proportionally with the amplitude to full scale ratio as ($2A/V_{full}$) is < 1 which results in a dB reduction.

Oscillator

The figures in the component section for the oscillator show the close in (100 kHz span) and wide view (200 MHz span) spectrums of the oscillator output. The peak output frequency is 90 dB above the noise floor in the figure. Because the resolution bandwidth is 910 Hz, the noise per Hz is actually lower. The peak output frequency is approximately 83 dB above signals seen to the side of the main peak. The other figure shows the harmonics of the 42 MHz signal as being 40 dB, 57 dB, and 63 dB down from the main peak for the first, second, and third harmonics.

As the oscillator drives the AOMs there will be a component of the harmonics driving the AOMs. The expected behavior is that the first AOM frequency shifts the laser light by 42 MHz

and the second AOM does the same for a total frequency shift of 84 MHz. If there are multiple frequencies driving the AOM one might expect that the light exiting would be shifted by all of these frequencies too. If the light had a 42 MHz shift, an 84 MHz shift down some 40 dB from the 42 MHz, and a 126 MHz shift down some 57 dB from the 42 MHz when this light entered the second AOM, a complex combination of the varied frequencies may be expected. This is not a concern as the physical configuration of the fiber coupled AOMs does not create a sum and difference of the RF signal frequency and the light's frequency. The AOMs only create a positive frequency shift. From the AOM transmitted intensity versus RF driving frequencies from the component section we can see that a change of only 5 MHz in the AOM driving frequency can reduce the transmission by 5 dB. The RF signal's harmonic components, which are far from the designed driving frequency, should not create a significant amount of light at other than the desired frequency shift. This does not address frequency fluctuations close in to the main peak.

Optical Analysis

Previous sections have discussed the fiber optic aspects of the system. In the following sections the remaining optical design of the system will be discussed save the coupling into and out of the fiber optic tip of the circulator. This will section include the telescope's exchange of light with the atmosphere, the type and configuration of the telescope, and its adjustment.

Free Space Propagation

The design of the system's telescope for the transmission and collection of light is purposefully simple as to minimize deleterious effects that would reduce the optical throughput. The more complicated the optical system the more surfaces must be penetrated, which inevitably leads to

lower transmission. Lower transmission leads to lower levels of signals which leads to shorter maximum range measurements.

In some circumstances additional complexity offers an advantage that is justified. The use of a large mirror to steer the beam, so line of sight measurements can be made quickly without moving the telescope, reduces the throughput as the mirror has less than perfect reflection and adds wave front distortions, which can affect coupling into the fiber optic tip of the telescope. But if the telescope had to be moved to change the direction of measurement there would be repeated stresses applied to the fiber and the system's pulses may have to be halted during the movement. It is not advisable to move a fiber optic while there is a high power density traveling in the core. Another issue with moving the whole telescope is that the distance and alignment of the fiber optic tip and lens may change with the dynamic stresses. This may lead to the beam wandering away from the expected azimuth and zenith pointing direction with varying degrees of severity depending upon the zenith angle and balance of the telescope about the pivot point. It may also lead to the beam becoming divergent or focused. If the whole telescope were to be pointed, the support structure would need to be made very rigid to maintain the alignment and separation distance of the fiber tip and the lens. Rigidity comes with the addition of weight and or complexity. Changes in the distance between the fiber tip and the lens as small as an eighth of a millimeter can have dramatic effects on the system's performance. Temperature changes can also cause a change in the distance by the expansion and contraction of the telescopes structure.

Optical telescopes can be broken down into three main groups, refractive, reflective and a compound (catadioptric) telescope consisting of refractive and reflective components. Refractive telescopes use lenses to focus the light while reflective telescopes use curved mirrors. Reflective telescopes often have multiple mirrors where at least one is curved so as to focus the light. This

often allows them to have a reduced length. The second mirror is used to transfer the light off of the optical axis so it may be observed. This second mirror lays on the optical axis and forms an obfuscation that blocks some of the area used to collect the light. This would also mean that light sent into the atmosphere from such a telescope would have an annular profile. There are off axis reflective telescopes that use an off axis parabolic mirror to shuttle the light collected off of the main optical axis thus avoiding the losses due to a secondary mirror. These types of mirrors are difficult to manufacture and are expensive. For a Doppler Lidar application the surface variations from the desired curvature would have to be very low in order not to create distortion that would degrade the system. A four inch off axis mirror would cost approximately $4000. The combination telescopes also suffer from losses due to an obscuration.

The refractive telescope was chosen for the design for its simplicity and lack of an obscuration. The refractive telescope could be as simple as a simple convex lens and the tip of the fiber optic port from the circulator. Light exiting the fiber with a mode field diameter on the order of 10 μm will expand by diffraction when it exits the fiber and be collimated by the lens.

Fiber to Free Space Propagation

In the fiber optic circulator portion of the design section, the expansion of the light exiting the fiber optic tip of the circulator was measured and the mode field diameter of the light in the fiber optic was found to be 11.32 μm.

Assuming the light traveling in the single mode polarization maintaining fiber has a Gaussian profile, the diameter of the beam can be calculated with the same equations used to find the field mode diameter based on the beam's diameter measured at a known distance.

The radius of a Gaussian beam at a range z, w(z), can be calculated based upon w_o, the beam waist radius at the face of the fiber optic,

$$w(z) = w_0 \left[1 + \left(\frac{\lambda z}{\eta \pi w_0^2} \right)^2 \right]^{1/2} \tag{4.1.66}$$

where λ is the wavelength, and η the index of refraction of the medium in which the beam is traveling. The Rayleigh range is the distance from the waist of the beam to where the beam's cross sectional area has doubled. It is equal to

$$z_R = \frac{\eta \pi w_0^2}{\lambda} \tag{4.1.67}$$

The focal length of the Aplanat lens at 1545.2 nm was calculated from the manufacturer's prescription with the optical design program Zemax to be 525.86 mm. The diameter of the beam at the lens was found to be 91.4 mm. The diameter versus range is shown in Figure 85. A collimated beam that is passed through an aperture of 91.4 mm will continue to diverge but at a much lower rate as seem in Figure 86. This supports the decision to implement a larger beam diameter as the divergence of a collimated beam is much reduced compared to a smaller beam, which helps to preserve the overlap of the outgoing pulse and the telescope's field of view.

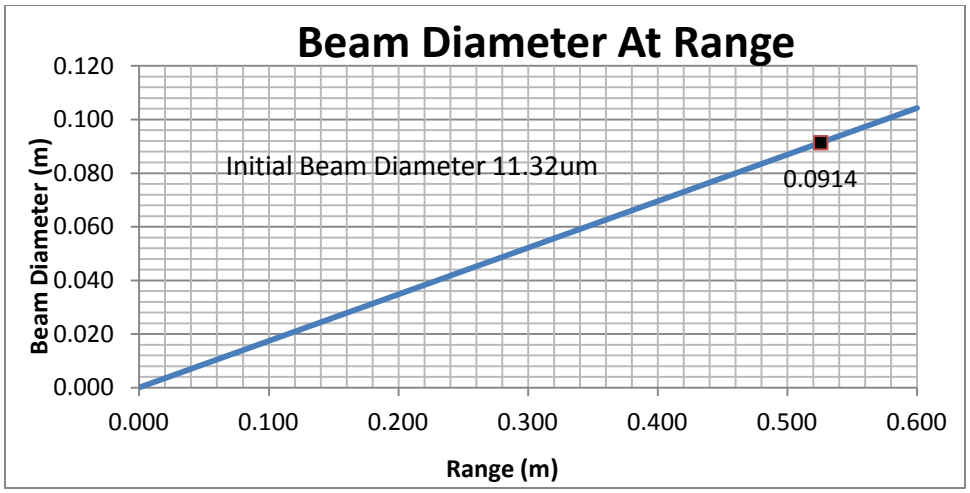


Figure 85, Diameter at range of beam exiting the fiber optic tip of circulator.

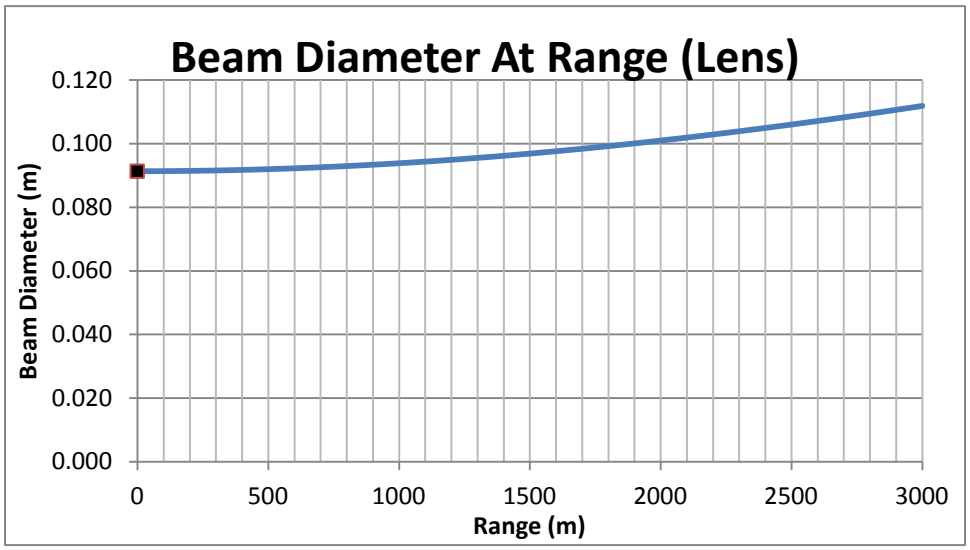


Figure 86, Diameter at range of beam exiting the lens.

Free Space to Fiber Propagation

Once the outgoing optical pulse is backscattered by the atmosphere the light reaching the system must be collected and conveyed into the system. Light must be collected by the lens and coupled into the optical circulator through the fiber optic tip. Coupling light into a fiber requires the light be focused onto the core of the fiber optic, which has a mode field diameter of 11.3 μm. The angle at which the light approaches the fiber will also be critical for efficient light coupling.

A lens parameter known as the f-number (f#), also known as focal ratio or relative aperture, describes the ratio of the focal length (f) to the diameter of the lens (D) or the clear aperture (CA) if the housing of the lens is considered.

$$f\# = \frac{f}{D} = \frac{f}{CA} \tag{4.1.68}$$

See Figure 87 for illustration of dimensions. The numerical aperture (NA) characterizes an optical system's ability to accept or emit light over a range of angles. The NA is defined as the sine of the angle, θ, that a ray of light traveling along the outer edge of a lens makes with the optical axis multiplied by the index of refraction of the medium the lens is located in, n.

$$NA = n\sin(\theta) \tag{4.1.69}$$

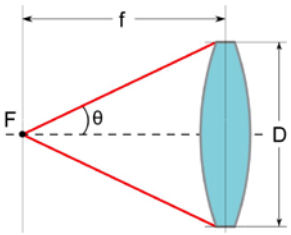


Figure 87, Lens dimensions and angles. [124]

The numerical aperture of a fiber optic is defined differently. Since there is no focal point to determine an angle θ another angle is used. The NA of a fiber optic is sine of the largest angle between the optical axis and the ray of light accepted into and conveyed in the fiber optic. If the indexes of refraction of the core and the cladding are known the NA can also be calculated.

$$NA = n_0 \sin(\theta_{acc}) = \frac{1}{n_0}\sqrt{n_{core}^2 - n_{cladding}^2} \tag{4.1.70}$$

When single mode fibers are used, the core is small and the wave nature of the light is more appropriate than the ray model of light. From the design section, the effective numerical aperture was defined as

$$NA_{eff} = \frac{2\lambda}{\pi MFD} \tag{4.1.71}$$

where MFD is the mode field diameter of the light traveling in the fiber optic.

The numerical aperture determines the size limit that light can be focused down to, "waist size", by the lens. A high numerical aperture is required to collimate a laser beam from a small source aperture. A low numerical aperture can cause distortions, and aberrations.

In order to bring the backscattered light from the atmosphere, which has been collected by the objective lens of the telescope, into the system for analysis, the light must be coupled into the optical fiber tip of the circulator located in the telescope. Ideally, all the light collected by the lens would be focused to a point on the surface of the fiber optic tip just over the core, and the light would be guided through the fiber. This is not possible due to a number of factors such as optical aberrations, the degree of coherence of the backscattered light after traveling through the atmosphere and diffraction limits.

The Airy disk is the smallest possible spot size that can be focused down to with an ideal lens and is limited by the diffraction of light. The bright center of the airy disk is surrounded by concentric rings. The Airy pattern consists of the central peak and the concentric rings surrounding it. The disk in the center of the Airy pattern contains most of the energy as can be seen in Figure 88. The diameter of the center of the Air disk can be calculated as

$$d = 2.44\lambda f\# = \frac{2.44\lambda f}{D} \tag{4.1.72}$$

If the diameter of the center of the Airy disk is less than the fiber optic's mode field diameter then one would expect light to be guided into the fiber optic. This diameter, or spot size, for the lens in the telescope is 19.82 μm. This is larger than the mode field diameter of the fiber optic tip. If the diameter of the lens were to remain at 100 mm and the wavelength at 1545.2 nm so as to maintain laser eye safety and avoid severe spreading of the beam by diffraction and by turbulence, see following section, the focal length of the lens would have to be altered. For an airy disk diameter to match the mode field diameter of the fiber optic the focal length of the lens would have to be 0.300 m, or 43% shorter than three times the current focal length. This would mean the telescope could be more compact, making the system more portable. This could reduce the need for mirrors to fold the beam within an enclosure. This would be beneficial as each mirror used reduces optical throughput, add wavefront distortions, increase the complexity of the system and multiply the degrees of freedom in the alignment of the telescope. The drawback is that the diameter of the beam when it reaches the lens would be reduced to approximately 0.052 m, from 0.0914 m (Figure 85). This would reduce the area over which the beam's energy is distributed by 2/3 thus increasing the intensity of the beam. To maintain eye safe operation the pulse energy and/or pulses per second would have to be altered. Also a smaller beam would have a greater divergence as it propagates.

The addition or multiple mirrors and the associated mounting hardware would act to reduce the benefits of a fiber optic based system needing few adjustments, being resistant to vibrations and misalignment due to transportation.

The area of the Airy disk central peak is 3.06 times larger than the mode field diameter of the fiber optic. This does not mean that only one third of the optical power is guided into the fiber as the intensity is not uniform across the central peak as seen in Figure 88.

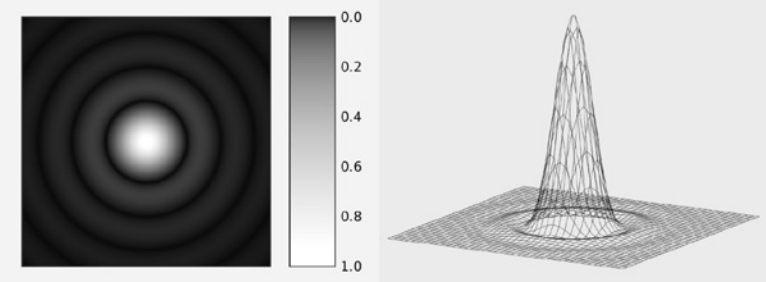


Figure 88, Airy disk representations in two (left) and three dimensions (right). [125, 126]

The energy encircled within the first null is 83.8 % of the total optical power. Within the first and second nulls the encircled energy is 91.0 % and within the first, second, and third nulls 93.8 %. The encircled power within an angle, P(θ), can be calculated as a percentage of the total power, P_0, as

$$P(\theta) = P_0\left[1 - J_0^2(ka\sin\theta) - J_1^2(ka\sin\theta)\right] \tag{4.1.73}$$

where $k = 2\pi/\lambda$ is the wavenumber, a is the radius of the aperture and J_n are Bessel functions. The angle θ is formed at the center of the aperture between the optical axis and a point on the target at a radius from the center of the Airy pattern. Figure 89 shows the geometry for the Airy disk calculations (left) and the relative encircled power and relative intensity (right) where the x-axis is $x = ka\sin\theta$.

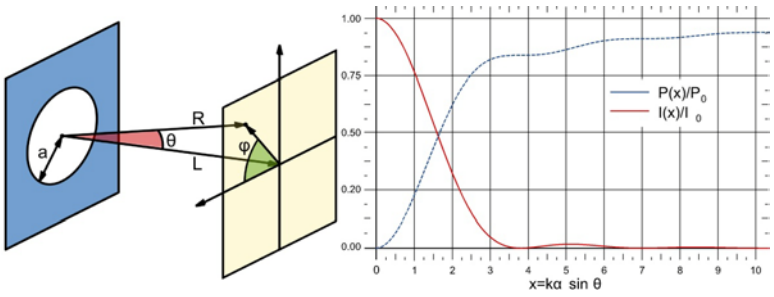


Figure 89, Airy disk geometry (left) for calculations and (right) relative encircled power, $P(x)/P_0$ and relative intensity, $I(x)/I_0$ where x=kasin(θ). [127, 128]

The energy encircled within the diameter of the mode field diameter is 61.7 % of the total possible power with uniform illumination of the aperture. The encircled power, $P(x)/P_0$ and relative intensity, $I(x)/I_0$ in the figure show that the quantities continue to change with increasing radius but the percentage increase is very low beyond the first ring.

If the aperture is illuminated by a Gaussian beam, the diameter of the Airy disk changes based upon the truncation ration that describes the relationship of the Gaussian beam to the aperture. [129-131] The diameter of the spot size with the inclusion of truncation becomes [130]

$$d = K\lambda f\# = \frac{K\lambda f}{D} \tag{4.1.74}$$

where K is a constant depending on the truncation ratio, T,

$$T = \frac{D_{beam}}{D_t} \tag{4.1.75}$$

where D_{beam} is the $1/e^2$ diameter of the Gaussian beam and D_t is the clear aperture diameter of the lens. The constant K is calculated as

$$K_{1/e^2} = 1.6449 + \frac{0.6460}{(T-0.2816)^{1.821}} - \frac{0.5320}{(T-0.2816)^{1.891}} \tag{4.1.76}$$

With D_{beam} = 0.0914 m, and D_t = 0.100 m, K = 1.865 which make the diameter of the spot d = 15.15 μm, which is significantly closer to the 11.32 μm diameter of the mode filed diameter of the fiber optic than was previous calculated with uniform illumination, 19.82 μm. The spot size is only 1.8 times larger by area than the mode field diameter. For a truncated Gaussian beam the total power loss due to the truncation, P_L ,can be calculated as

$$P_L = e^{-2\left(D_t / D_{beam}\right)} \tag{4.1.77}$$

With the current configuration the power loss due to the truncation is P_L = 9.125 % or 0.42 dB.

The effects of optical aberrations can increase the focused spot size. This is one reason for choosing an Aplanat lens to help reduce the aberrations. The backscattered light from the atmosphere may not retain a pure Gaussian profile upon its arrival at the system due to atmospheric anomalies. The returning light may be more uniform than Gaussian in its profile or it may have changes in its spatial coherence that can affect the system's ability to couple the signal into the system for detection.

For optimal coupling into the fiber there should be perfect alignment of the focal spot with the fiber core. There should be no lateral misalignment from the center and there should not be any tilt to the fiber from the optical axis. Also the incoming light should be at the proper focus to form the smallest spot possible. When there is no miss-adjustment the coupling efficiency is described by the equation,

$$\eta = \frac{2}{\chi^2}\left[1 - e^{-\chi^2}\right]^2 \tag{4.1.78}$$

where $\chi = a / w_A$ is the ratio of the aperture radius (a) to the radius of the back propagated fiber mode (w_A). A maximum coupling efficiency of 81% is obtained when the ratio $\chi = 1.121$.[132] With a clear aperture of 0.100 m and a beam diameter of 0.0914 m the ratio is 1.094.

The theoretical maximum of 81% is with perfect alignment and without aberrations. This is not the case in any physical system. These calculations do not take into account the fact that the tip of the fiber is angled to reduce the back reflection of an outgoing pulse and as a result the fiber optical axis is tilted slightly as compared to that of the lens.

Atmospheric Turbulence

The turbulence of the atmosphere can also have an effect on the beam diameter and the transverse, or spatial coherence, of a propagating beam. Random variations in the index of refraction of the atmosphere can cause fluctuations in the intensity of a beam which is known as scintillation. This can cause the beam to wander from the optical axis and alter the spreading of the beam at a rate not predicted by diffraction. Since this Doppler Lidar system is coaxial and the duration of the optical pulses' round trip time is much smaller than the rate of change of the atmosphere, the path of the beam to and fro will not change significantly. Speckle patterns are intensity patterns resulting from the interference of many waves of the same frequency exhibiting random phase and intensity differences. A coherent wavefront exhibits a more dynamic speckle pattern where the diameter of coherent regions are smaller and changing the greater the turbulence.

The variations in the refractive index of the atmosphere are the result of small density fluctuations, changes in temperature and moisture content, and fluctuations in velocities. In 1941

Komogorov and Obukhov studied isotropic velocity fields and developed velocity structure fluctuations. [133-135] The application of these works to Lidar today often invoke Komogorov's name and a unit, C_n^2 (units of $m^{-2/3}$), the index of refraction structure constant or structure parameter, to quantify the strength of the fluctuations of the refractive indexes. Weak turbulence has structure parameter value on the order of 10^{-17} $m^{-2/3}$ or less while strong turbulence has a value on the order of 10^{-13} $m^{-2/3}$ or more. The structure parameter value can also vary diurnally and with altitude. The structure parameter value tends to be larger during the day than at night and larger at lower altitudes than at higher altitudes. [133, 135, 136] As an illustrative example from August 27, 2008 in Campbell California the structure parameter measured from an elevation of 409 m above ground level to the ground level are shown bellow. [137, 138]

Table 11, Structure parameter for Campbell California measured on August 27, 2008.

Time	**9:00**	**13:00**	**14:00**	**18:00**
C_n^2	1.25 x10^{-14}	8.90 x10^{-15}	1.10 x10^{-14}	2.05x10^{-15}

The Submarine Laser Communications (SLC) model developed at an Air Force base in Rome, NY for day time is given as [139, 140]

$$C_n^2 = \begin{cases} 0 & 0 \text{ m} < h < 19 \text{ m} \\ 4.008\times10^{-13}h^{-1.054} & 19 \text{ m} < h < 230 \text{ m} \\ 1.300\times10^{-15} & 230 \text{ m} < h < 850 \text{ m} \\ 6.352\times10^{-7}h^{-2.966} & 850 \text{ m} < h < 7000 \text{ m} \\ 6.209\times10^{-16}h^{-0.6229} & 7000 \text{ m} < h < 20000 \text{ m} \end{cases} \quad (4.1.79)$$

Other SLC models of the structure parameter from Hawaii give different characteristics. Since these models were developed with data from an island in the Pacific Ocean it is not surprising they are slightly different from the above model. [135, 141]

$$C_n^2 = \begin{cases} 1.7\times10^{-14} & 0\text{ m} < h < 18.5\text{ m} \\ 3.13\times10^{-13}h^{-1.05} & 18.5\text{ m} < h < 240\text{ m} \\ 1.300\times10^{-15} & 240\text{ m} < h < 880\text{ m} \\ 8.87\times10^{-7}h^{-3} & 880\text{ m} < h < 7200\text{ m} \\ 2.0\times10^{-16}h^{-0.5} & 7200\text{ m} < h < 20000\text{ m} \end{cases} \quad (4.1.80)$$

For the night hours the model is

$$C_n^2 = \begin{cases} 8.4\times10^{-15} & 0\text{ m} < h < 18.5\text{ m} \\ 2.87\times10^{-12}h^{-2} & 18.5\text{ m} < h < 240\text{ m} \\ 2.5\times10^{-16} & 240\text{ m} < h < 880\text{ m} \\ 8.87\times10^{-7}h^{-3} & 880\text{ m} < h < 7200\text{ m} \\ 2.0\times10^{-16}h^{-0.5} & 7200\text{ m} < h < 20000\text{ m} \end{cases} \quad (4.1.81)$$

Figure 90 shows the simulated profile of a 2 cm laser beam of wavelength 3.5 μm and its expansion in a vacuum and in turbulence. The simulation is not for 1.5452 μm but it illustrates the effect of turbulence on a beam propagating through strong turbulence. The distortion of the wavefront will affect the focus of the beam and thus affect the coupling into a fiber optic.

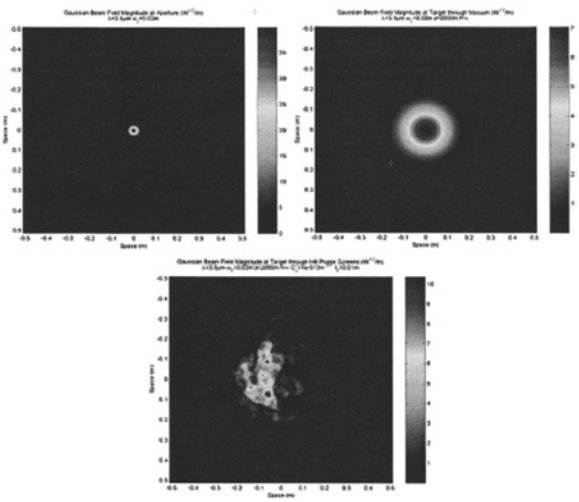


Figure 90, Beam profile at source, 2 cm (top left), beam profile after propagation in a vacuum (top right), beam profile after propagation in atmosphere with turbulence (bottom center), for wavelength 3.5 μm, and structure parameter of 10^{-13} $m^{-2/3}$ [133]

The effects of turbulence on a beam propagating 2 km in the atmosphere with weak and strong turbulence can be calculated to illustrate the benefits of using a beam with a larger aperture. In this case the beam diameters will be 91.4 mm and 9.14 mm, the wavelength is 1545.2 nm. Utilizing atmospheric propagation formulas the size and area of the beams will be calculated. [134, 135]

The beam curvature parameter at the source, Θ_0, describes the effects of focus on the amplitude. Here L is the range and F_0 is the phase front radius of curvature at the source which is > 0 for a converging beam and infinity for a collimated beam.

$$\Theta_0 = 1 - \frac{L}{F_0} \tag{4.1.82}$$

The Fresnel ratio, Λ_0, of the beam at the source defines the amplitude changes due to diffraction. Here the wavenumber of the beam is $k = 2\pi/\lambda$, and W_0 is the beam radius at the source.

$$\Lambda_0 = \frac{2L}{kW_0^2} \tag{4.1.83}$$

From these the beam curvature parameter at the range, Θ, can be calculated as

$$\Theta = \frac{\Theta_0}{\Theta_0^2 + \Lambda_0^2} \tag{4.1.84}$$

The Fresnel ratio at range, Λ, is

$$\Lambda = \frac{\Lambda_0}{\Theta_0^2 + \Lambda_0^2} \tag{4.1.85}$$

Rytov variance, σ^2, physically represents the irradiance fluctuations

$$\begin{aligned} \sigma_1^2 &= 1.23 C_n^2 k^{7/8} L^{11/6} \quad \text{for plane wave} \\ \sigma_2^2 &= 0.5 C_n^2 k^{7/8} L^{11/6} \quad \text{for spherical wave} \end{aligned} \tag{4.1.86}$$

The beam radius in free space at range, W, is calculated as

$$W = W_0 \sqrt{\Theta_0^2 + \Lambda_0^2} \tag{4.1.87}$$

The effective beam radius in random media at range is

$$W_e = W \sqrt{1 + 1.63 \sigma_1^{12/5} \Lambda} \tag{4.1.88}$$

Table 12, Beam diameter at range (2000 m) calculations for various source beam diameters and degrees of turbulence normalized to source beam diameter.

	Beam Diameter at Range in Units of Original Diameter					
Beam Diameter at Source	C_n^2 **0**	C_n^2 **1E-16**	C_n^2 **1E-15**	C_n^2 **1E-14**	C_n^2 **1E-13**	C_n^2 **1E-12**
0.914 cm	47.11	47.11	47.15	47.65	55.04	122.68
9.140 cm	1.11	1.11	1.12	1.32	3.05	11.38

The Table 12 shows the effects of turbulence on the diameter of the beam at a range of 2000 m for a small beam diameter and a beam diameter ten times greater. The tabulated values are a ratio of the effective diameter at range to the original beam diameter so the effects of turbulence may be compared. A smaller diameter beam is expected to diffract at a greater rate than a larger beam diameter as seen in Figure 85 and Figure 86. Table 12 shows that the degree of turbulence as described by the structure parameter has an a greater effect on the smaller beam than the larger beam and the beam becomes 2.6 times larger than expected from diffraction alone at very strong turbulence levels. The larger beam expands to a lesser degree than the smaller beam, but its expansion beyond diffraction affects is more severely affected by the very strong turbulence than the smaller beam. The smaller beam's divergence is greater initially so the turbulence has a lower proportional effect.

As the system uses the same optics for transmission and collection there should not be a concern of pointing jitter or beam wander. In a bi-axial system the aiming of the receiver and its alignment would be a concern. Beam wander, where the beam randomly deflects as it propagates as a result of atmospheric turbulence, is also not a concern due to the choice of telescope design.

Signals in the System

In this section examples of the driving and received signals will be presented. This will provide a view of the types of signals being used to run the components and the kind of signals that are collected. Figure 91 shows a basic schematic of the Doppler Lidar system for illustration of the types of waveforms seen in the system.

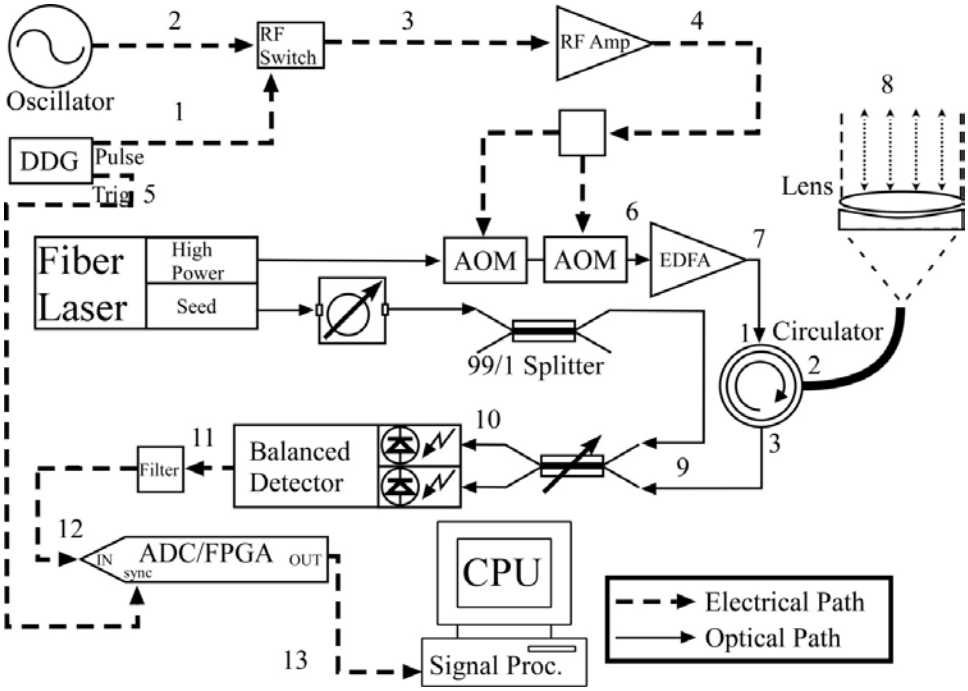


Figure 91, Schematic of Doppler Lidar system with reduced details to illustrate the signals in the system.

The waveforms seen at locations indicated in Figure 91, the trigger pulse to the RF switch (1), the RF driving pulse to the AOMs (4), and a detected back reflection from the fiber optic tip of the circulator (11), are shown in Figure 92. The data acquisition card, X5-400M, capture of back reflection from fiber optic tip and signal (left), and close up view of back reflection from tip (right) are shown in Figure 93. The waveforms seen in Figure 94 are the trigger pulse to data acquisition card starting acquisition (5), the RF driving pulse for AOMS (4), and the optical

output pulse from fiber optic tip of circulator (8). The spectrum of the detector's output before and after the anti-aliasing filters is shown in Figure 95.

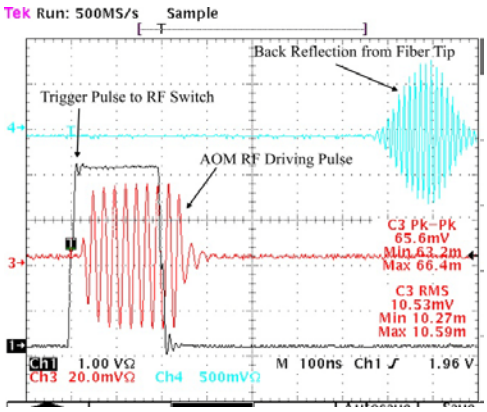


Figure 92, Illustration of waveforms seen in system. Trigger pulse to RF switch, AOM RF driving pulse and detected back reflection from fiber tip.

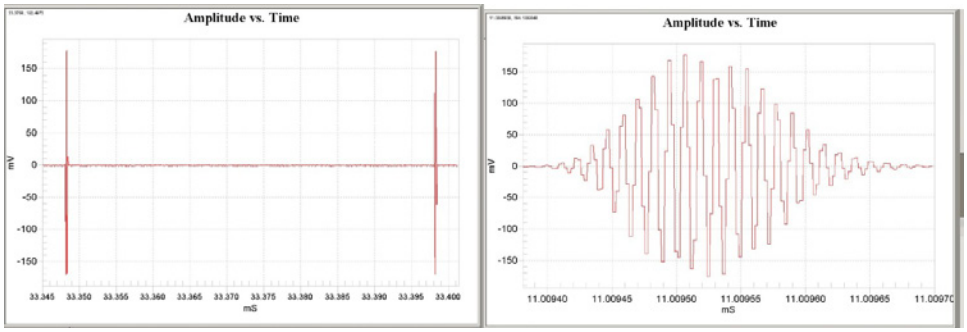


Figure 93, Data acquisition card, X5-400M, back reflection from fiber optic tip and signal (left), and close up view of back reflection from tip (right).

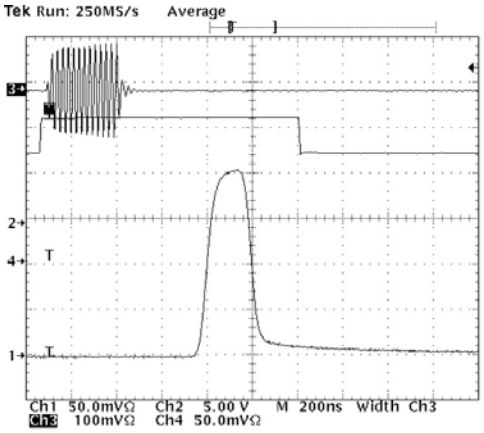


Figure 94, Waveforms of Doppler Lidar system, (middle) Trigger pulse to data acquisition card starting acquisition (5 in schematic), (top) RF driving pulse for AOMS (4 in schematic), (bottom) Optical output pulse from fiber optic tip of circulator (8 in schematic).

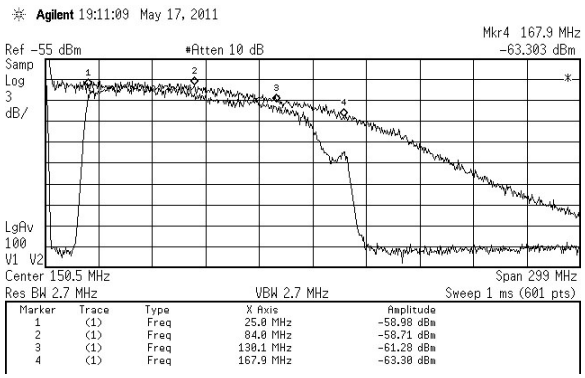


Figure 95, Spectrum of detector output with anti-aliasing filters (12 in schematic) (lower waveform) and without anti-aliasing filters (11 in schematic) (upper waveform).

Detection Analysis

With direct detection the signal is directly measured with an optical detector. The thermal and shot noise of the system can be reduced to a degree with judicious selection of components and cooling. With judicious selection of components, heterodyne detection with a balanced detector can reach a lower level of noise.

The relative intensity noise, RIN, of the laser source is the power noise normalized to the average power level. If P(t) is the laser power described by

$$P(t) = \overline{P} + \delta P(t) \tag{4.1.89}$$

and δP(t) is the zero mean power fluctuation of the beam the RIN is $\delta P(t)/\overline{P}$. When the RIN is limited by shot noise the level of RIN is not constant. The power spectral density of shot noise limited RIN in dBc/Hz is equal to

$$S(f) = \frac{2h\nu}{\overline{P}} \tag{4.1.90}$$

In heterodyne detection the local oscillator (LO) is mixed with the signal. In order to avoid thermal noise dominating the shot noise and the detection efficiency suffering, the power of the LO can be increased so shot noise dominates the thermal noise on the detector's impedance load. The RIN can be decreased with the use of a balanced detection system. Once the LO is mixed with the signal the power is divided between two detectors. The detectors are arranged in a configuration where the common signal to both detectors will cancel and the differences will be amplified by a transimpedance amplifier (TIA).

The relationship between local power and efficiency on power penalty is derived from the heterodyne detection. It is dependent upon the RIN of the laser source and the thermal noise of the receiver. The efficiency on power penalty is [142]

$$\eta_{pp} = \left(1 + \frac{2kTF_N h\nu}{\eta_q e^2 P_L R_L} + \frac{\eta_q P_L R_{in} R_B}{2h\nu} \right)^{-1} \tag{4.1.91}$$

where H is Plank's constant (6.63×10^{-34} Js), k is Boltzmann's constant (1.38×10^{-23} J/K, e is the electron charge (1.6×10^{-19} C), T is the temperature (K), F_N is the noise figure, ν is the laser frequency (Hz), η_q is the quantum efficiency of the balanced receiver, P_L is the local oscillator power (W), R_L is the transimpedance gain of the receiver (Ohms), R_{in} is the RIN level of the laser source (Hz-1), and R_B is the RIN suppression ratio obtained by the balanced receiver. When plotted, this equation indicates that approximately 10 mW of LO power would be beneficial. The balanced detector in use will be damaged at that power level and the manufacturer states that the detector becomes nonlinear at 1.2 mW of optical power.

The system efficiency can be found from [142]

$$\eta_D(L) = \frac{\eta_S \eta_F}{\left(1 + \left(1 - \frac{L}{F}\right)^2 \left[\frac{\pi\left(A_C D\right)^2}{4\lambda L}\right]^2 + \left(\frac{A_C D}{2S_0(L)}\right)^2\right)} \tag{4.1.92}$$

where F is the focal range of the telescope (m), η_F is the far field system efficiency in the case of negligible atmospheric refractive turbulence, A_C is a correction factor for the aperture diameter. A_C and η_F are used to take into account truncation of the beam. L is the range, D is the effective diameter of the telescope (m), $S_0(L)$ is the transverse coherence length (m). η_S is the a system efficiency and is

$$\eta_S = \eta_I \eta_{RE} \eta_{PP} \eta_q \tag{4.1.93}$$

where η_I is the factor of insertion loss of the optical components except for the telescope. η_{RE} and η_{PP} are to account for reflective and absorptive losses of the telescope lenses. The signal to noise at range can be calculated as [142]

$$SNR(L) = \frac{\eta_D(L)\lambda E \beta K^{2L/1000} \pi D^2}{8hBL^2} \tag{4.1.94}$$

where E is the outgoing pulse energy (J), K is the one way atmospheric transmittance (km^{-1}).

The system was operated with 1 mW of optical power on each port of the balanced detector. This allowed some benefit from raising the LO power without damaging the detector or becoming nonlinear. A balanced detector capable of handling 10 mW of power was not found and the system operated with the current detector.

Beam Geometry and Wind Vector Calculations

Three line of sight measurements are used to calculate the horizontal and vertical wind vector components. To take a measurement a mirror is rotated about the Z axis to aim the beam. The Z axis is normal to the surface of the earth, while the normal to the face of the mirror is fixed at a 45 degree angle with respect to the Z axis. The mirror rotated about the Z axis to one of three predefined positions where the measurement is taken. The three rotational positions are prescribed as 0 degree rotation, $+\theta$ degrees and $-\theta$ degrees with respect to the optical axis defined by the path from the tip of the fiber optic cable from the optical circulator to the center of the lens. The optical rail in the XY plane parallels the optical axis. A counterclockwise rotation about the Z axis of the mirror is defined as a positive angle rotation while a clockwise rotation is defined as a negative angle rotation. The derivation of the calculations that follow assumes the normal to the face of the mirror is 45 degrees from the Z axis and that the positive and negative rotations are equal in magnitude but may be at any angle that is physically realizable given the dimensions of the mirror. The greater the angle of rotation about the Z axis the larger the mirror must be in its horizontal dimension to present an unobstructed reflective surface to the impinging beam. The calculation of the wind vectors is much simplified if the angle between the projections of the beam, for positive and negative rotation on to the horizontal plane, are orthogonal. Due to the physical limitations of the system on the optical table relative to the hatch opening on the ceiling of the vehicle this orthogonality cannot be realized. Thus the calculations must allow for non-orthogonal beam configurations.

Three rotational positions are incorporated into the calculations to arrive at the horizontal and vertical wind velocity components. Defining the coordinate system in the frame of the vehicle where, as view from above, the +X coordinate is towards the front of the vehicle, the +Y

coordinate is towards the left, driver's side (United States), and the +Z coordinate is normal to the surface of the optical table. For these calculations the vehicle is assumed to have been leveled with the assistance of mechanical jacks at the four corners of the vehicle so that the normal to the surface of the optical table and the normal to the surface of the Earth are coaxial. See Figure 96 for a schematic representation of the optical table, optical rail, beam steering optical components, hatch opening of the research vehicle and the vehicle's coordinate system.

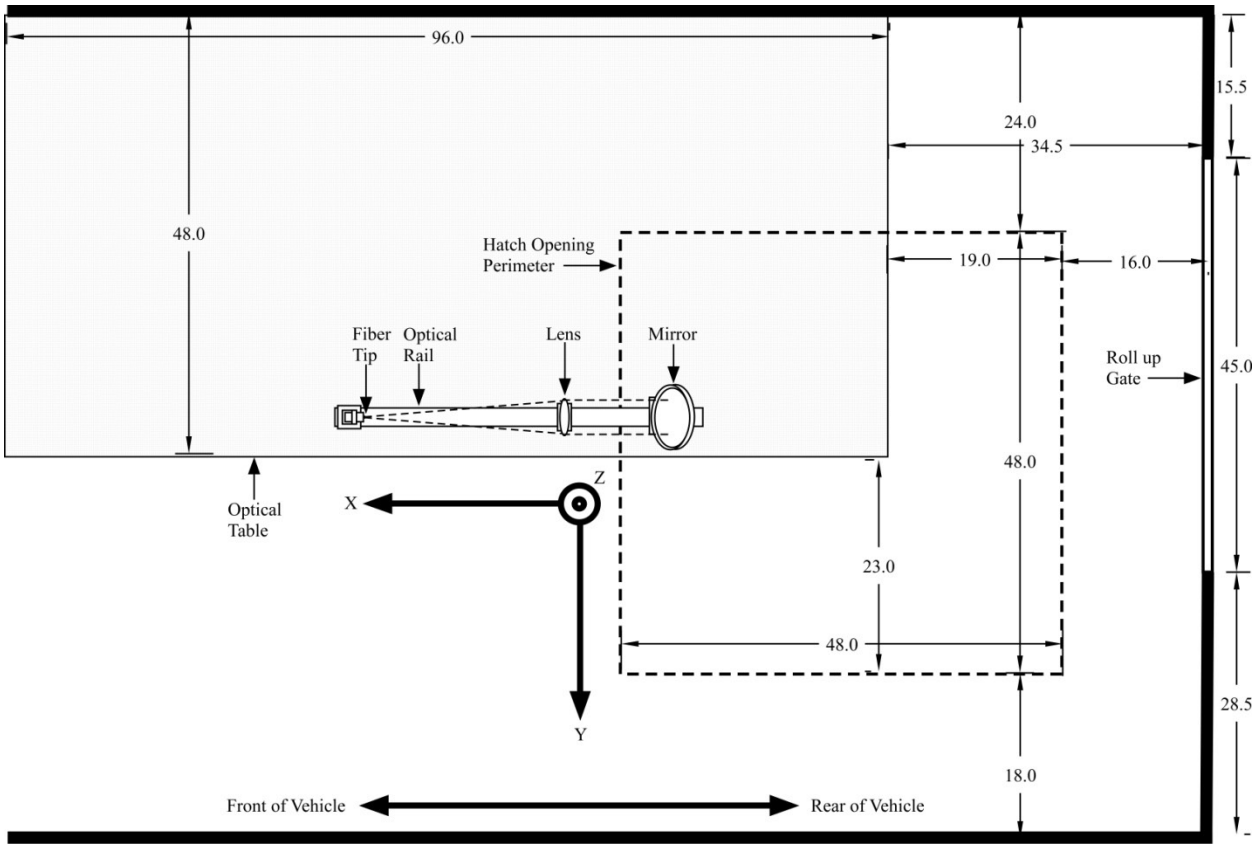


Figure 96, Schematic representation of optical table, beam-steering optical components, and hatch opening of research vehicle and the vehicle coordinate system.

The direction of propagation of the outgoing source beam (input beam), leaving the fiber optic tip and passing through the lens is $-\hat{x}$ along the optical rail. See Figure 97 and Figure 98 for an illustration of the change in the geometry of the mirror when it is tilted to 45 degrees.

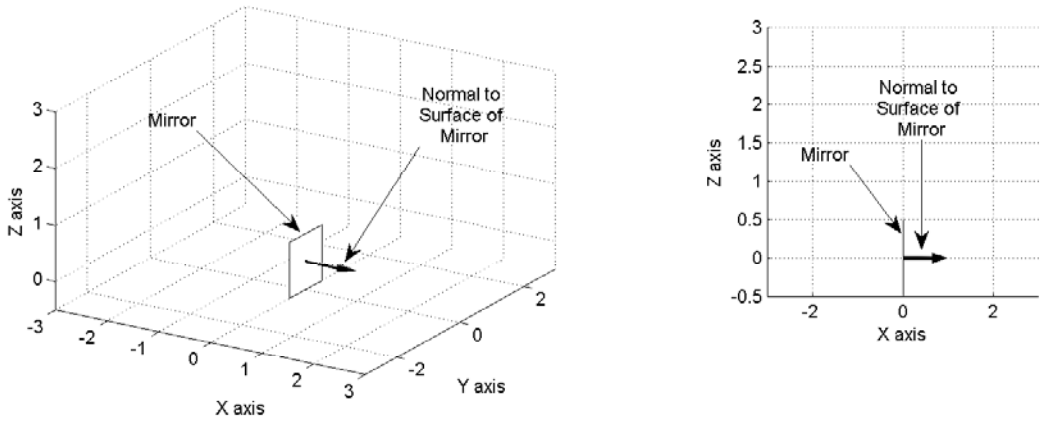


Figure 97, Illustration of vehicle coordinate system with un-rotated and un-tilted mirror located at center.

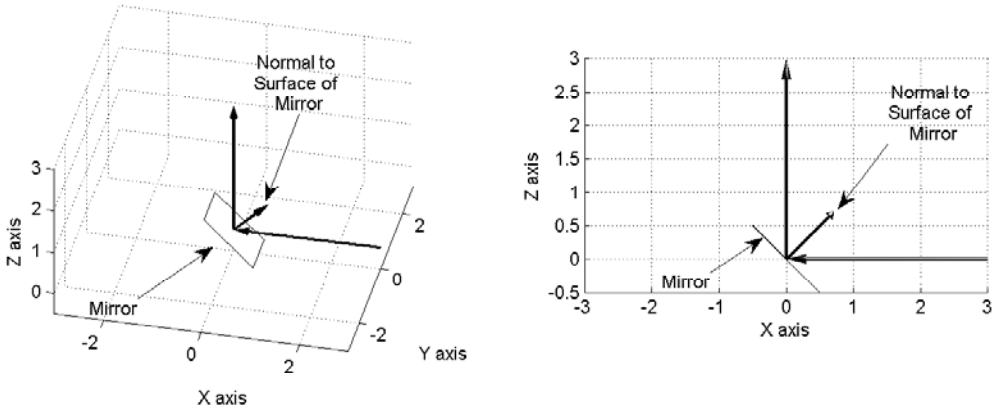


Figure 98, Illustration of mirror tilted 45 degrees from the Z axis with the normal to the surface indicated as well as the incoming beam and outgoing reflected beam.

With no rotation of the mirror about the Z axis, the normal to the mirror at an angle of 45 degrees in XZ plane is,

$$\hat{n} = \hat{x}\left(\frac{1}{\sqrt{2}}\right) + \hat{y}(0) + \hat{z}\left(\frac{1}{\sqrt{2}}\right) \tag{4.1.95}$$

After a rotation of +θ (counter clockwise by θ degrees) (see Figure 99), the normal is

$$\hat{n} = \hat{x}\left(\frac{\cos\theta}{\sqrt{2}}\right) + \hat{y}\left(\frac{\sin(\theta)}{\sqrt{2}}\right) + \hat{z}\left(\frac{1}{\sqrt{2}}\right) \tag{4.1.96}$$

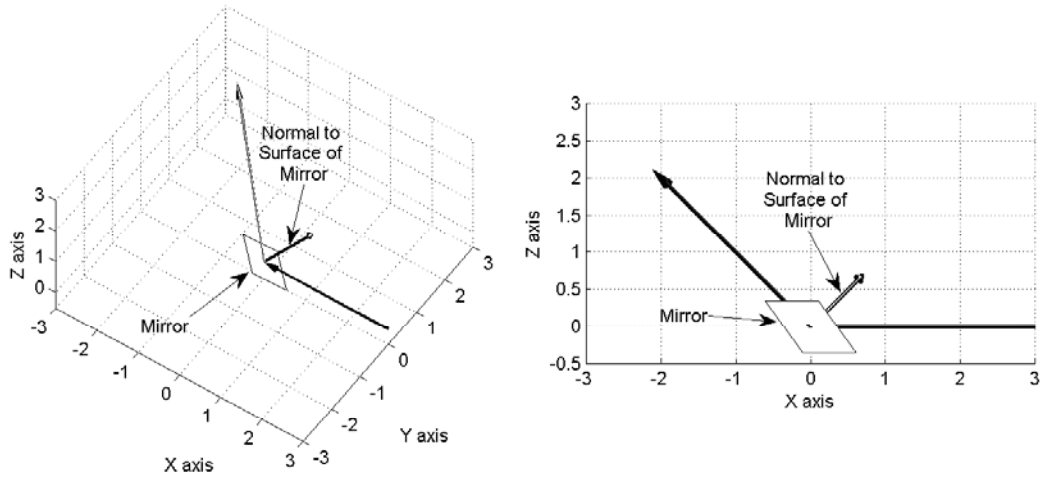


Figure 99, Illustration of incoming beam and reflected beam when mirror has been rotated by a positive angle. The negative rotation angle will change to polarity of the y component of the vector.

Now a coordinate change to x', y', z' is performed with

$$\begin{aligned}\hat{x}' &= \hat{n} \\ &= \hat{x}\left(\frac{\cos(\theta)}{\sqrt{2}}\right) + \hat{y}\left(\frac{\sin(\theta)}{\sqrt{2}}\right) + \hat{z}\left(\frac{1}{\sqrt{2}}\right)\end{aligned} \tag{4.1.97}$$

The output beam to the atmosphere, by Snell's law, will be along the unit vector

$$\hat{o} = \hat{x}'\left(\frac{\cos(\theta)}{\sqrt{2}}\right) + \hat{y}'\left(\frac{\sin(\theta)}{\sqrt{2}}\right) + \hat{z}'\left(\frac{1}{\sqrt{2}}\right) \tag{4.1.98}$$

Therefore to find the two unit vectors $\hat{y}'$, $\hat{z}'$ to form the right handed coordinate system with $\hat{x}'$

$$\hat{z}' = -\hat{x}\left(\frac{\cos(\theta)}{\sqrt{2}}\right) - \hat{y}\left(\frac{\sin(\theta)}{\sqrt{2}}\right) + \hat{z}\left(\frac{1}{\sqrt{2}}\right) \tag{4.1.99}$$

And since

$$\begin{aligned}\hat{y}' &= \hat{z}' \times \hat{x}' \\ &= -\hat{z}\left(\frac{\cos(\theta)\sin(\theta)}{2}\right) + \hat{y}\left(\frac{\cos(\theta)}{2}\right) \\ &+\hat{z}\left(\frac{\cos(\theta)\sin(\theta)}{2}\right) - \hat{x}\left(\frac{\sin(\theta)}{2}\right) \\ &+\hat{y}\left(\frac{\cos(\theta)}{2}\right) - \hat{x}\left(\frac{\sin(\theta)}{2}\right)\end{aligned} \tag{4.1.100}$$

$$\hat{y}' = \hat{y}\big(\cos(\theta)\big) - \hat{x}\big(\sin(\theta)\big) \tag{4.1.101}$$

Going back to the xyz coordinates in the vehicle's frame of reference, the output beam will then be along the unit vector:

$$\begin{aligned}\hat{o} &= \hat{x}\left(\frac{\cos^2(\theta)}{2}\right) + \hat{y}\left(\frac{\cos(\theta)\sin(\theta)}{2}\right) + \hat{z}\left(\frac{\cos(\theta)}{2}\right) \\ &-\hat{x}\left(\frac{\sin^2(\theta)}{\sqrt{2}}\right) + \hat{y}\left(\frac{\cos(\theta)\sin(\theta)}{\sqrt{2}}\right) + \hat{z}\big(0\big) \\ &-\hat{x}\left(\frac{\cos(\theta)}{2}\right) - \hat{y}\left(\frac{\sin(\theta)}{2}\right) + \hat{z}\left(\frac{1}{2}\right)\end{aligned} \tag{4.1.102}$$

$$\begin{aligned}\hat{o} &= \hat{x}\left(\frac{\cos^2(\theta)}{2} - \frac{\cos(\theta)}{2} - \frac{\sin^2(\theta)}{\sqrt{2}}\right) \\ &+\hat{y}\left(\frac{\cos(\theta)\sin(\theta)}{2} + \frac{\cos(\theta)\sin(\theta)}{\sqrt{2}} - \frac{\sin(\theta)}{2}\right) \\ &+\hat{z}\left(\frac{1}{2} + \frac{\cos(\theta)}{2}\right)\end{aligned} \tag{4.1.103}$$

After a rotation of -θ (clockwise rotation of θ degrees) from the 0 degree position, the second output beam will be in the direction of,

$$\begin{aligned}\hat{q} = \hat{x}&\left(\frac{\cos^2(\theta)}{2} - \frac{\cos(\theta)}{2} - \frac{\sin^2(\theta)}{\sqrt{2}}\right)\\ -\hat{y}&\left(\frac{\cos(\theta)\sin(\theta)}{2} + \frac{\cos(\theta)\sin(\theta)}{\sqrt{2}} - \frac{\sin(\theta)}{2}\right)\\ +\hat{z}&\left(\frac{1}{2} + \frac{\cos(\theta)}{2}\right)\end{aligned} \tag{4.1.104}$$

For an arbitrary wind vector in the frame of the truck

$$\vec{V} = A\hat{x} + B\hat{y} + C\hat{z} \tag{4.1.105}$$

We make measurements M_1, M_2, and M_3 along directions $\hat{z}$, $\hat{o}$, and $\hat{q}$ respectively

$$M_1 = \vec{V} \bullet \hat{z} \tag{4.1.106}$$

$$M_2 = \vec{V} \bullet \hat{o} \tag{4.1.107}$$

$$M_3 = \vec{V} \bullet \hat{q} \tag{4.1.108}$$

Since the vertical measurement will be directly along the Z axis

$$M_1 = C \tag{4.1.109}$$

$$M_2 = A\left(\frac{\cos^2(\theta)}{2} - \frac{\cos(\theta)}{2} - \frac{\sin^2(\theta)}{\sqrt{2}}\right) + B\left(\frac{\cos(\theta)\sin(\theta)}{2} + \frac{\cos(\theta)\sin(\theta)}{\sqrt{2}} - \frac{\sin(\theta)}{2}\right) + C\left(\frac{1}{2} + \frac{\cos(\theta)}{2}\right) \tag{4.1.110}$$

$$M_3 = A\left(\frac{\cos^2(\theta)}{2} - \frac{\cos(\theta)}{2} - \frac{\sin^2(\theta)}{\sqrt{2}}\right) - B\left(\frac{\cos(\theta)\sin(\theta)}{2} + \frac{\cos(\theta)\sin(\theta)}{\sqrt{2}} - \frac{\sin(\theta)}{2}\right) + C\left(\frac{1}{2} + \frac{\cos(\theta)}{2}\right) \tag{4.1.111}$$

The solution is then,

$$C = M_1 \tag{4.1.112}$$

$$M_2 - M_3 = 2B\left(\frac{\cos(\theta)\sin(\theta)}{2} + \frac{\cos(\theta)\sin(\theta)}{\sqrt{2}}\right) - \frac{\sin(\theta)}{2} \tag{4.1.113}$$

$$B = \frac{\left(M_2 - M_3\right)}{\cos(\theta)\sin(\theta)\left(1+\sqrt{2}\right) - \sin(\theta)} \tag{4.1.114}$$

$$M_2 + M_3 = 2A\left(\frac{\cos^2(\theta)}{2} - \frac{\cos(\theta)}{2} - \frac{\sin^2(\theta)}{\sqrt{2}}\right) + 2C\left(\frac{1}{2} + \frac{\cos(\theta)}{2}\right) \tag{4.1.115}$$

Substituting (4.1.112) into (4.1.115) yields,

$$M_2 + M_3 = A\left(\cos^2(\theta) - \cos(\theta) - \sqrt{2}\sin^2(\theta)\right) + M_1\left(1 + \cos(\theta)\right) \tag{4.1.116}$$

Solving (4.1.116) for variable A, and restating the solutions for B (4.1.114) and C (4.1.109) for convenience results in,

$$A = \frac{M_2 + M_3 - M_1\left(1 + \cos(\theta)\right)}{\cos^2(\theta) - \cos(\theta) - \sqrt{2}\sin^2(\theta)} \tag{4.1.117}$$

$$B = \frac{M_2 - M_3}{\cos(\theta)\sin(\theta)\left(1 + \sqrt{2}\right) - \sin(\theta)} \tag{4.1.118}$$

$$C = M_1 \tag{4.1.119}$$

Now that the three components of the wind vector in the vehicle coordinate system have been calculated, a final rotation to align to the Earth coordinate system so that the X axis of the vehicle rotates into the East direction and the Y axis is the North direction; while the Z direction is the zenith or updraft component.

The angle of rotation, ϕ, is the clockwise rotation required to move the X axis in the vehicle to the coordinate system of the earth. Since the vehicle has been leveled to align the normal to the surface of the optical table with that of the Earth, the vertical wind vector is

$$V_z = C \tag{4.1.120}$$

While the horizontal win components consist of the East and North wind vectors,

$$V_{east} = A\cos(\phi) - B\sin(\phi) \tag{4.1.121}$$

$$V_{north} = A\sin(\phi) + B\cos s(\phi) \tag{4.1.122}$$

Now the wind vectors are in the coordinate system of the Earth.

CHAPTER 5
Results

The coherent Doppler wind Lidar system was operated over multiple days at various times of the day and evening in scanning as well as a staring, usually vertical, measurement modes from the research vehicle located on the campus of the City College of New York [Lat: 40.8213400 (40° 49' 16.8234"), Lng: -73.9482960 (-73° 56' 53.865"), altitude of approximately 38 m above sea level]. On some of the days that the Doppler Lidar system was operated the CCNY direct detection Lidar described earlier was also recording measurements. Other instruments' measurements and atmospheric models were also collected for comparison to the coherent Doppler Lidar.

The time of day the data was recorded is presented in a number of ways. Coordinated Universal Time, (UTC), sometimes referred to as Universal Time Coordinated, is the successor to Greenwich Mean Time and is a widely used time standard that does not recognize daylight savings time or summertime. In the United States, UTC is five hours ahead of the Eastern Standard Time (EST) but only four hours ahead Eastern Summertime (March 13, 2011 to November 6, 2011), when daylight savings time is observed. The UTC time zone is often designated with a "Z" and referred to as "Zulu time."

Data collected with the coherent Doppler wind Lidar system was time stamped with the date and local time utilizing the International Organization for Standardization (ISO) standard ISO 8601. The year, "yyyy", is followed by the month, "mm", the day, "dd", then a capital t, "T", the 24 clock hour, "HH", the minutes of the hour, "MM", and the seconds of the minutes, "SS", which has the form yyyymmddTHHMMSS. The local time of 6:10:07 AM on the date of February 23, 2012 would be 20120223T061007 as seen in the equation bellow.

$$\underbrace{2012}_{\text{year}}\ \underbrace{02}_{\text{month}}\ \underbrace{23}_{\text{day}}T\ \underbrace{06}_{\text{hour}}\ \underbrace{10}_{\text{minutes}}\ \underbrace{07}_{\text{seconds}} \tag{5.1.1}$$

During data collection approximately 9.2 GB of spectral data was collected in text format. This represented approximately 17.5 hours of active collection and excludes any time spent setting up, aligning, or shutdown the system. Approximately 2900 wind velocities along a line of sight were calculated. The table bellow shows a summary of Doppler Lidar data collected.

Table 13, Summary of coherent Doppler wind Lidar data taken from research van.

Date	Local Start Time	Local End Time	UTC Start Time	UTC End Time	Data: Mode of Operation	Run Time hh:mm	Number of Beam Velocities	Text File Size MB
12/13/11	9:04	12:04	14:04Z	17:04Z	Vertical	3:00	371	947
	12:21	15:21	17:21Z	20:21Z	Verical	3:00	371	947
	15:28	17:28	20:28Z	22:28Z	Vertical	2:00	242	620
12/12/11	12:33	13:19	17:33Z	18:19Z	Scanning	0:46	37	96
	14:51	16:17	19:51Z	21:17Z	Scanning	1:26	78	222
12/11/11	16:14	21:14	21:14Z	02:14Z	Scanning	5:00	300	744
12/09/11	10:43	12:57	15:43Z	17:57Z	Scanning	2:14	112	292
	13:01	14:01	18:01Z	19:01Z	Verical	1:00	75	391
	15:19	16:36	20:19Z	21:36Z	Scanning	1:17	75	391
	16:41	17:41	21:41Z	22:41Z	Vertical	1:00	120	461
12/08/11	15:41	18:32	20:41Z	23:32Z	Scanning	2:51	148	386
	18:52	19:22	23:52Z	00:22Z	Vertical	0:30	60	230
12/05/11	12:22	16:33	17:22Z	21:33Z	Scanning	4:11	178	465
	16:39	16:48	21:39Z	21:48Z	Scanning	0:09	5	15
12/04/11	18:09	22:02	23:09Z	03:02Z	Scanning	3:53	165	441
12/02/11	15:16	16:08	20:16Z	21:08Z	Scanning	0:52	19	52
	16:22	16:52	21:22Z	21:52Z	Verical	0:30	60	230
	16:56	17:26	21:56Z	22:26Z	- 29 deg	0:30	60	230
	17:30	18:00	22:30Z	23:00Z	+ 29 deg	0:30	60	230
12/01/11	13:15	14:15	18:15Z	19:15Z	Vertical	1:00	120	461
	14:49	15:59	19:49Z	20:59Z	+ 29 deg	1:10	120	461
	16:22	17:22	21:22Z	22:22Z	- 29 deg	1:00	120	461
	12:16	13:10	17:16Z	18:10Z	Scanning-test	0:54	15	41
	18:45	19:19	23:45Z	00:19Z	Scanning-test	0:34	7	26
11/30/11	18:30	19:39	23:30Z	00:39Z	Scanning	1:09	27	73
11/28/11	14:27	15:27	19:27Z	20:27Z	Vertical	1:00	60	313

Wind data is often reported in a graphical form, such as a weather map with wind barb symbols to represent the direction that the wind is going towards and its velocity in knots. The number of knots is indicated by lines, half lines and flags on the end of the wind barb. Figure 100 illustrates the convention used with wind barbs. The feathers or barbs point to the direction from which the wind is coming from and the small end of the barb is pointing to the direction that the wind is heading towards. The number of knots is indicated by the length of the barbs, where a half barb represents 5 knots, a full barb represents 10 knots and a triangular flag represents 50 knots. One knot is equal to 1.15 mph, 1.85 km/h, 0.51 m/s, or 1.6 ft/s. Wind direction can alternatively be represented as cardinal directions in degrees on a compass where north is 0º or 360º, east is 90º, south is 180º, and west is 270º. When cardinal directions are used the direction of the wind, in degrees, describes the direction from which the wind is coming from. An easterly wind (90 º) would describe wind coming from the East and going towards the West.

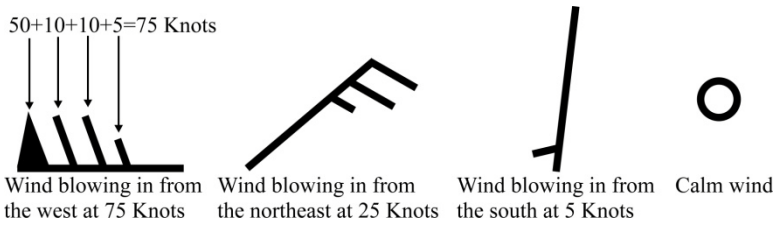


Figure 100, Wind barb symbols illustrating direction and speed in knots.

Table 14, Conversion table for knots and common units.

	m/s	**km/h**	**mph**	**knot**	**ft/s**
1 m/s =	1	3.6	2.236936	1.943844	3.280840
1 km/h =	0.277778	1	0.621371	0.539957	0.911344
1 mph =	0.444704	1.609344	1	0.868976	1.466667
1 knot =	0.514444	1.852	1.150779	1	1.687810
1 ft/s =	0.3048	1.09728	0.681818	0.592484	1

Instrumentation and Models

One of the instruments used in the comparison was the direct detection Lidar at CCNY system described earlier in the Lidar section. The CCNY direct detection Lidar was operated during some of the same periods and overlapped the hours that the Doppler wind Lidar was recording measurements. As a result of its high output power, the direct detection Lidar is not eye safe but is generally able to record measurements from a greater distance then the coherent Doppler wind Lidar. Its effectiveness at less than 500 m is limited due to the configuration of the telescope and outgoing beam.

The North American Mesoscale Model (NAM) is a numerically based short term weather prediction model from the National Centers for Environmental Prediction. The data from this models is over a large area, 12 km square, and is generally for periods of time on the order of three hours. The altitude is given as a pressure altitude in units of millibars (mb). Under standard conditions a 1013.25 mb absolute pressure altitude is near sea level at a standard temperature of 15º C. To convert between altitude pressure and altitude in meters one may use the equation bellow.

$$h_{alt} = \left(1 - \left(\frac{p_{sta}}{1013.25}\right)^{0.190284}\right) \times 44307.69396 \qquad (5.1.2)$$

Table 15, Pressure altitude (mb) to altitude (m)

Pressure Altitude (mb)	Altitude (m)	Pressure Altitude (mb)	Altitude (m)	Pressure Altitude (mb)	Altitude (m)
1013.25	0.0	875	1219.7	725	2734.3
1000	110.8	850	1456.7	700	3010.9
975	323.2	825	1699.4	675	3295.7
950	540.1	800	1948.2	650	3589.2
925	761.7	775	2203.3	625	3891.9
900	988.1	750	2465.2	600	4204.7

Data from a Vaisala CLC1 ceilometer located on the roof (i.e. 7th floor) of CCNY's Steinman Hall was obtained for periods that the Doppler wind Lidar was in operation. The Doppler Lidar system was located across the street from Steinman Hall, which also housed the direct Lidar system on the 5th floor. The ceilometer operates in a manner similar to Lidar. The device sends out a low power 910 nm pulse and measures the return signal from a cloud base as the backscatter from a cloud is high compared to the rest of the atmosphere. The device is designed to measure cloud base height and vertical visibility.

Data

Examples of comparison of coherent Doppler wind Lidar with other instruments for 08/17/2011 covering same time periods are shown in the next few figures. The signal strength seen on the coherent Doppler Lidar and the direct detection Lidar at CCNY system have a similar pattern in intensities distribution with time. Figure 101, Figure 102 and Figure 103 show agreement on cloud heights and overall patterns at the points 0 minutes/14:30, 75 minutes/15:50, and 100 minutes/6:15. The large return signals seen at the upper altitudes are due to clouds.

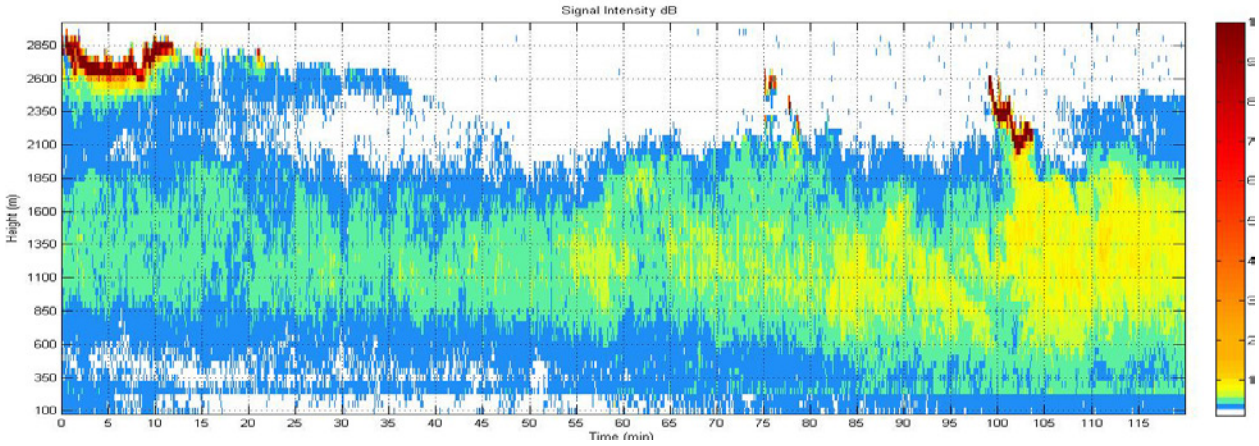


Figure 101, Coherent Doppler Lidar vertical signal strength versus height, at various times 08/17/2011.

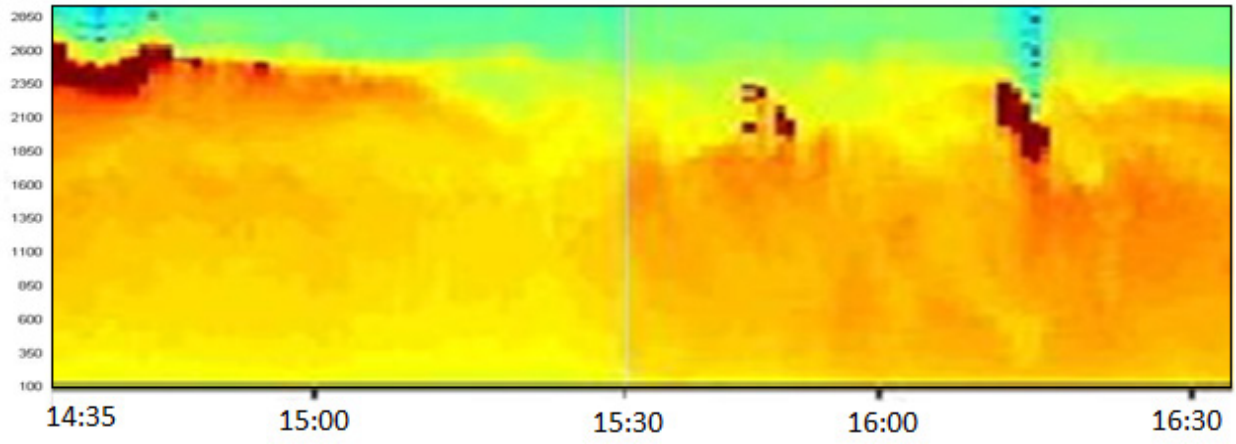


Figure 102, Signal strength of CCNY direct detection Lidar versus height at versus time, 08/17/2011.

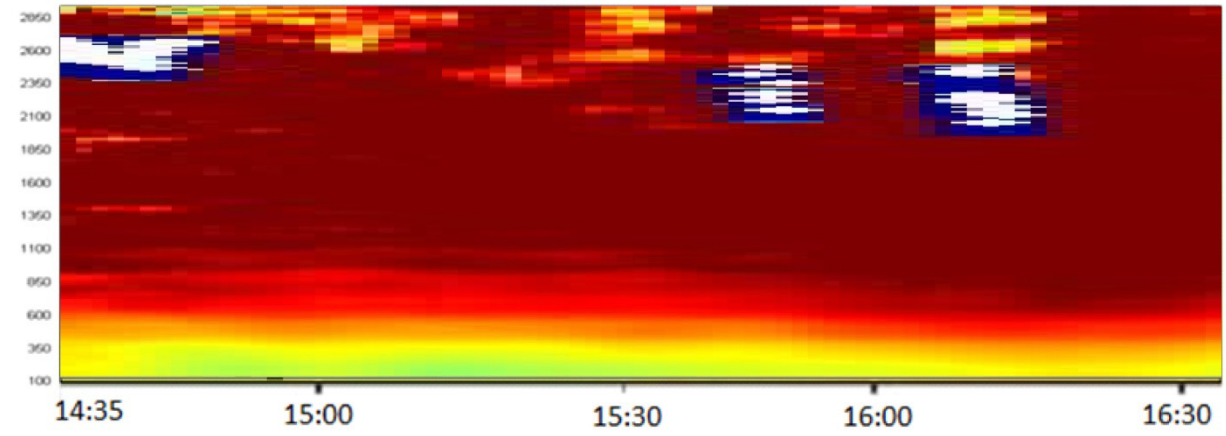


Figure 103, Ceilometer attenuated backscatter signal profile with cloud base at CCNY versus height for various times, 08/17/2011.

An example of the comparison of coherent Doppler wind Lidar intensity, Figure 104, and vertical wind speed measurements, Figure 105, for date 08/18/2011 are shown bellow. The return signal envelop or boarder that defines the upper limit of the altitude from which a return signal was detected is nearly identical for both systems. Even small undulations appear to be mirrored between the systems. The direct detection Lidar does not perform well at less than 500 m so the intensity difference seen between the two systems at low altitudes may be the result of this.

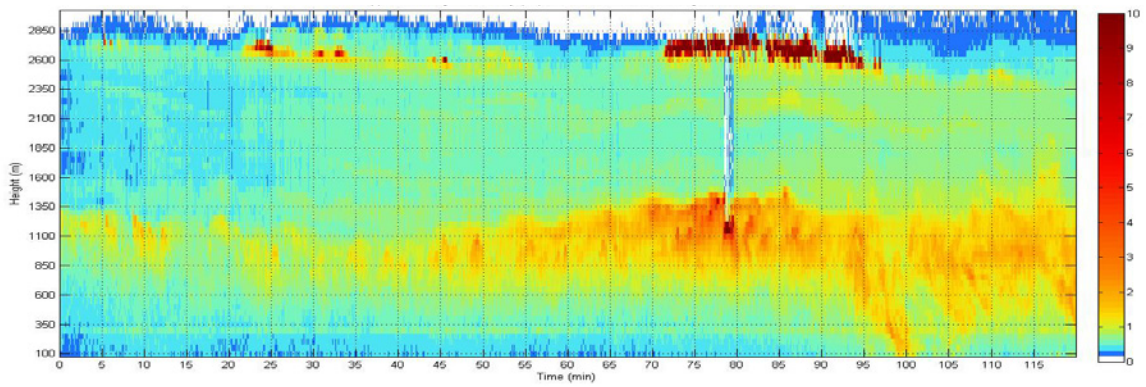


Figure 104, Doppler Lidar atmospheric backscattered signal intensity versus height measured on 8/18/2011.

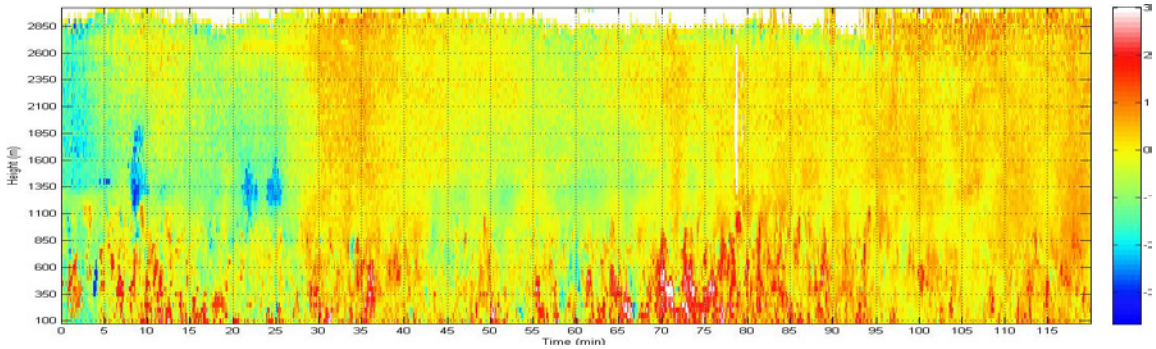


Figure 105, Coherent Doppler Lidar vertical wind speed versus height measured on 8/18/2011

12/04/2011 Data

On December 4th, 2011 the Doppler Lidar system was operated for approximately 3 hours and 53 minutes. The range corrected signal intensity and the vertical wind velocities are show in Figure 106 and Figure 107. A plot from the ceilometer for the same date is shown in Figure 108.

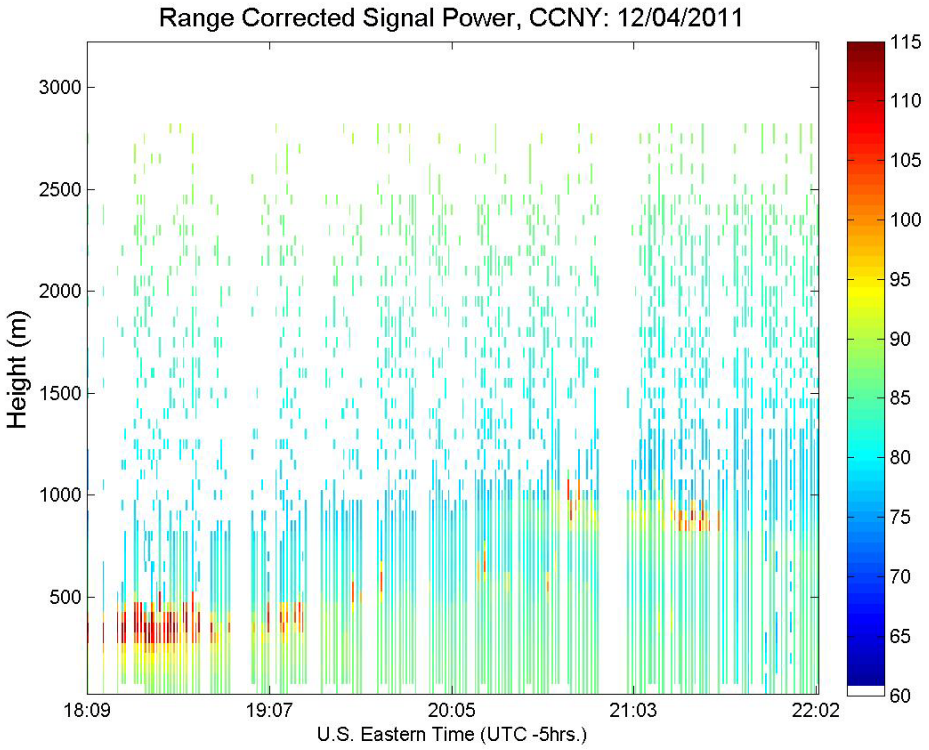


Figure 106, Range corrected signal intensity on 12/04/11.

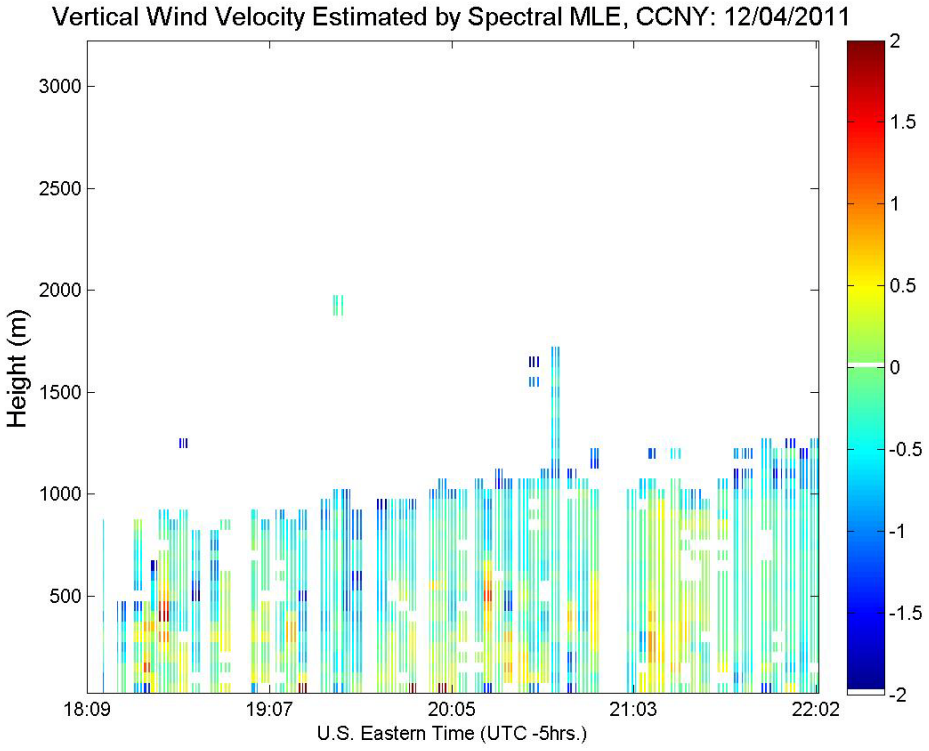


Figure 107, Vertical wind velocity on 12/04/11

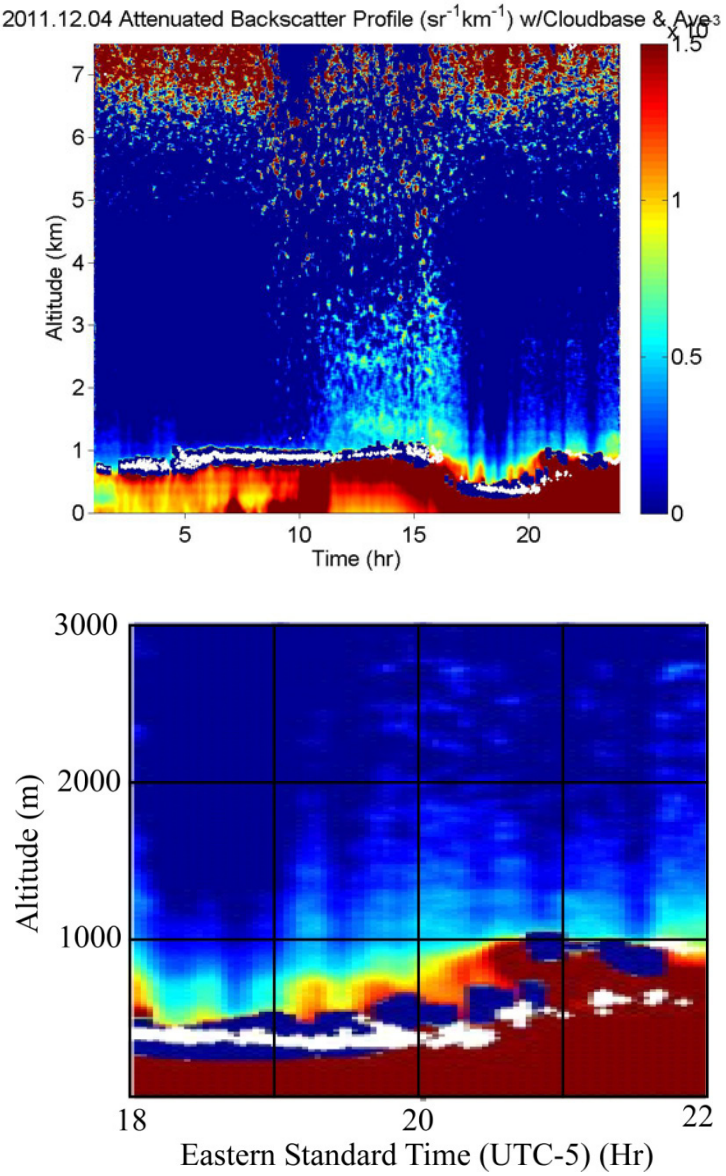


Figure 108, Ceilometer profile on 12/04/11. (Top) Full day plot. (Bottom) 18:00 to 22:00 plot.

The range corrected signal intensity of Figure 106 shows a similar profile as compared to the same time period of the ceilometer. At 20:00 an increase in altitude of strong scatters is seen to occur. A cloud layer at approximately 500 m climbs to a greater altitude. In both the Doppler Lidar and the ceilometer profiles, at approximately 21:30, there are corresponding decreases in the strong scatters' altitudes. The wind barb plot for this data is shown in Figure 109. When compared to the windgram plot (Figure 110) of 12/05/11 for 00Z and 03Z (19:00 and 22:00 local

time) this appears to track. In both plots, 1t 19:00 and 22;00 local time the direction that the wind is going towards is north at low altitudes and north east at the altitude increases. This is seen in both figures. The velocity is also seen to track between the two plots. In Figure 111 and Figure 112 the NAM model wind speed and direction are plotted for 12/05/11 00Z and 03Z. The wind direction is the angular compass direction from where the wind is coming from and not to where the wind is going. There are some variations but the overall agreement is good.

In Figure 113 and Figure 114 are shown the measured hourly average wind speed and directions from the radar wind profiler located at Rutgers University approximately 18 km from the Doppler Lidar system. When compared to the wind barbs generated from the Doppler Lidar system seen in Figure 109 a similar trends is seen between both data sets. There is a trend towards higher velocities at greater altitudes and the direction tends to be stable over the range of altitudes measured by the Doppler Lidar.

Figure 115 and Figure 116 show the wind profiles for 18:00 and 22;00 local time from the Stevens Institute of Technology located approximately 10.3 km from the City College of New York. As with the other wind measurements sited above there is a general agreement between the wind profiler and the Doppler Lidar wind barb plots.

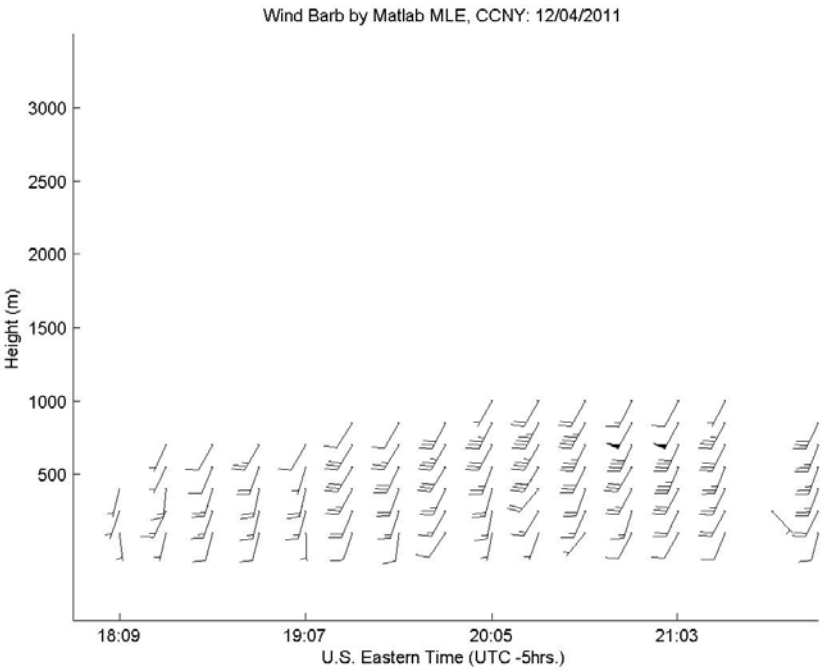


Figure 109, Wind barb plots for data taken on 12/04/11.

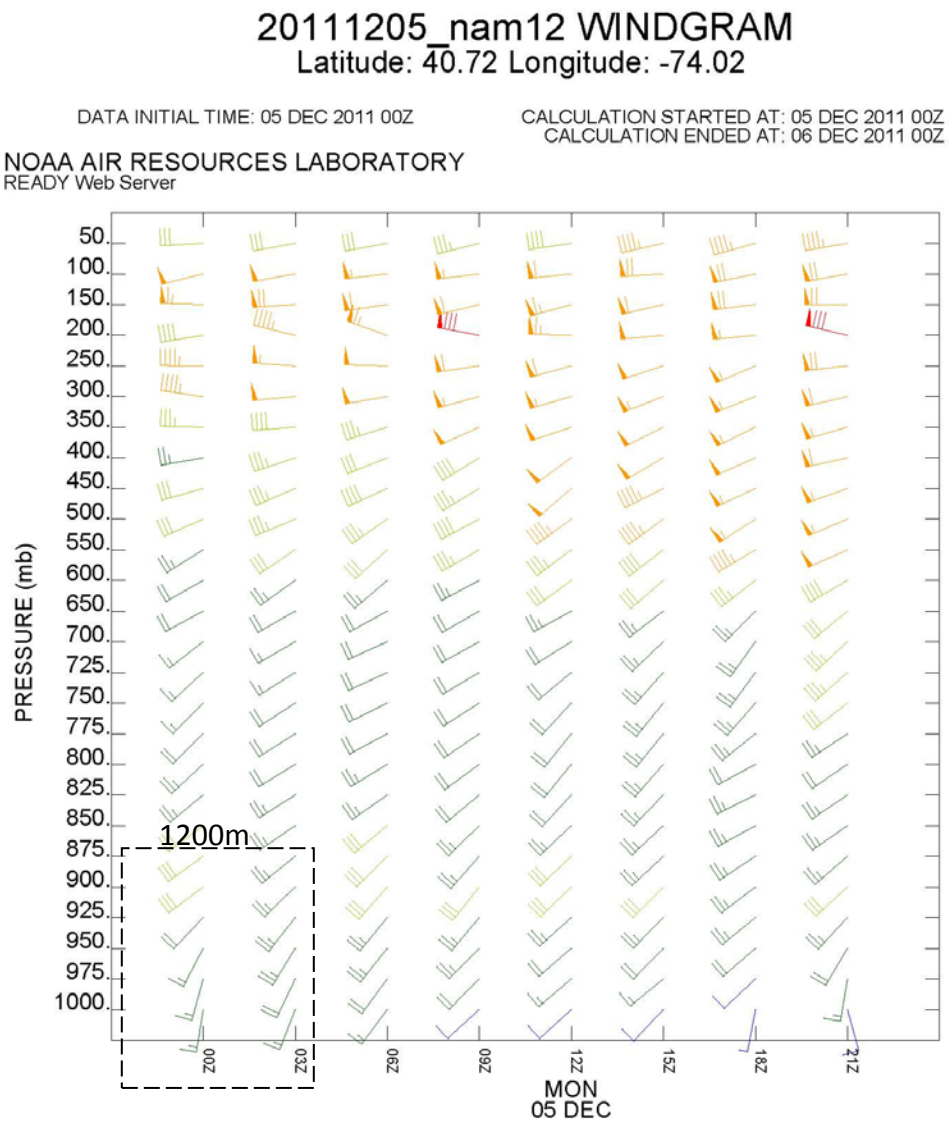


Figure 110, Windgram plot with wind barbs for 12/05/11 starting at 00Z time (19:00 local time).

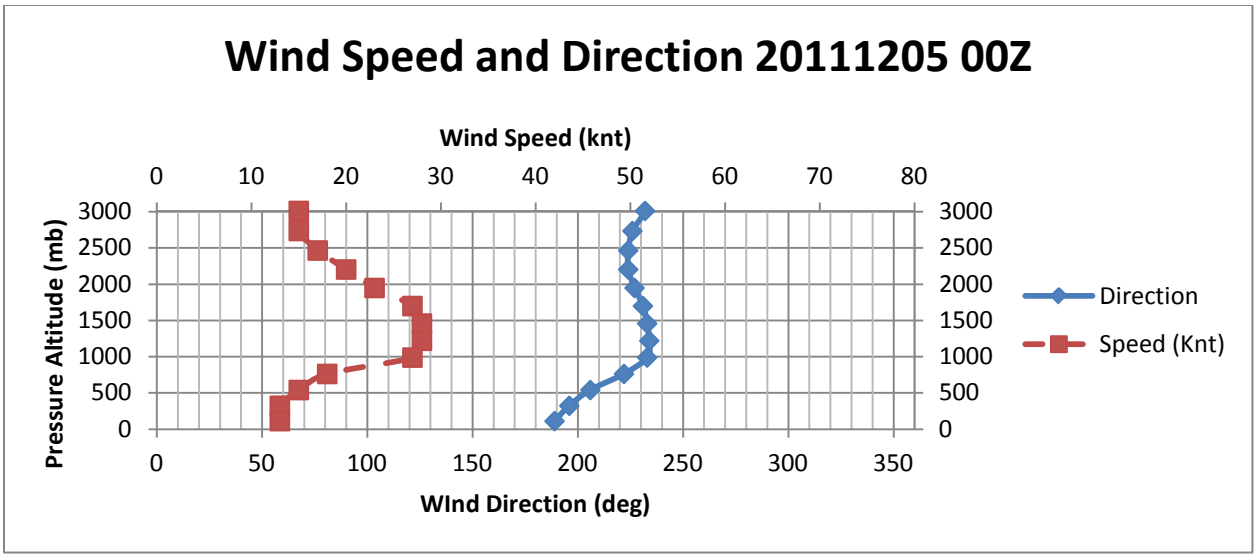


Figure 111, NAM 12km model wind speed and direction, 12/05/11 for 00Z.

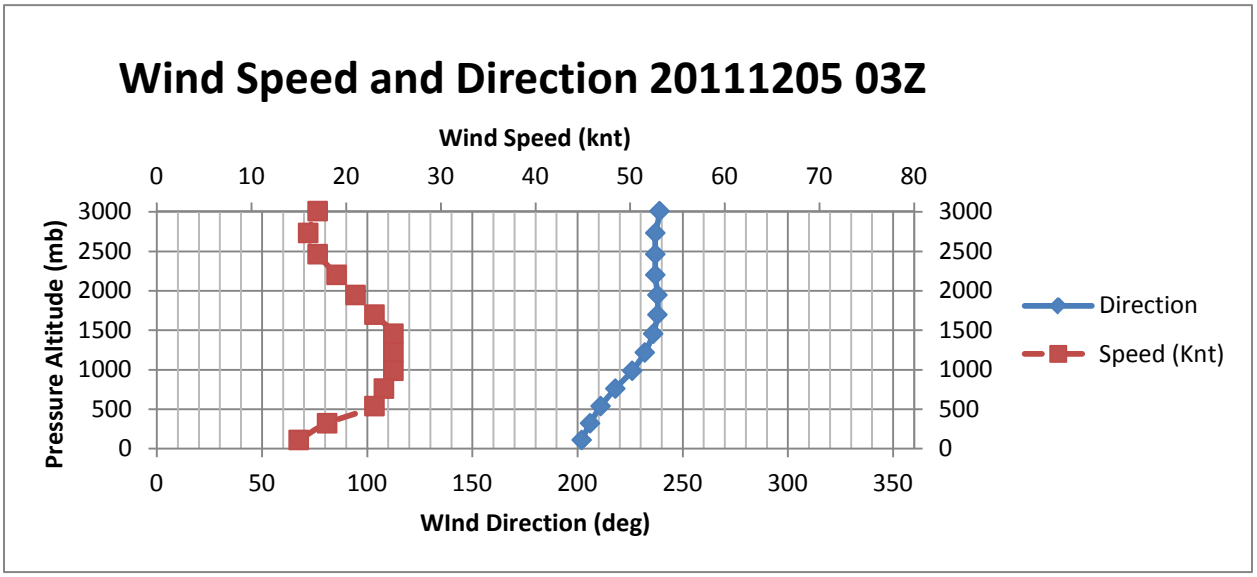


Figure 112, NAM 12km model wind speed and direction, 12/05/11 for 03Z.

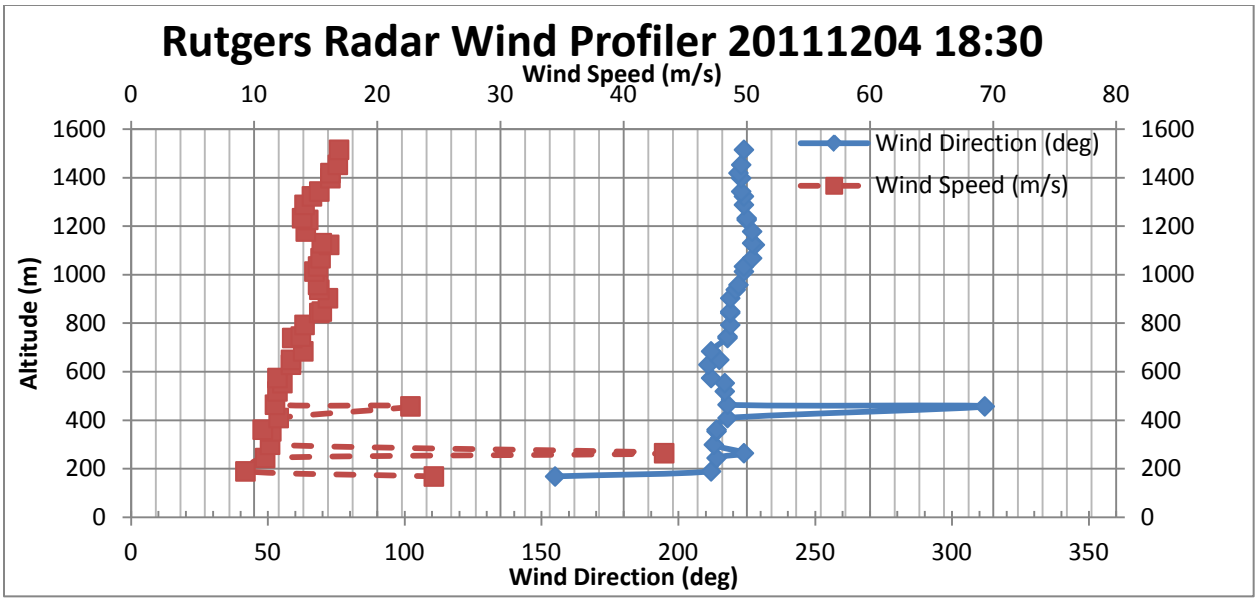


Figure 113, Radar wind profiler wind speed and direction for 12/04/11 18:30 local time, Rutgers University.

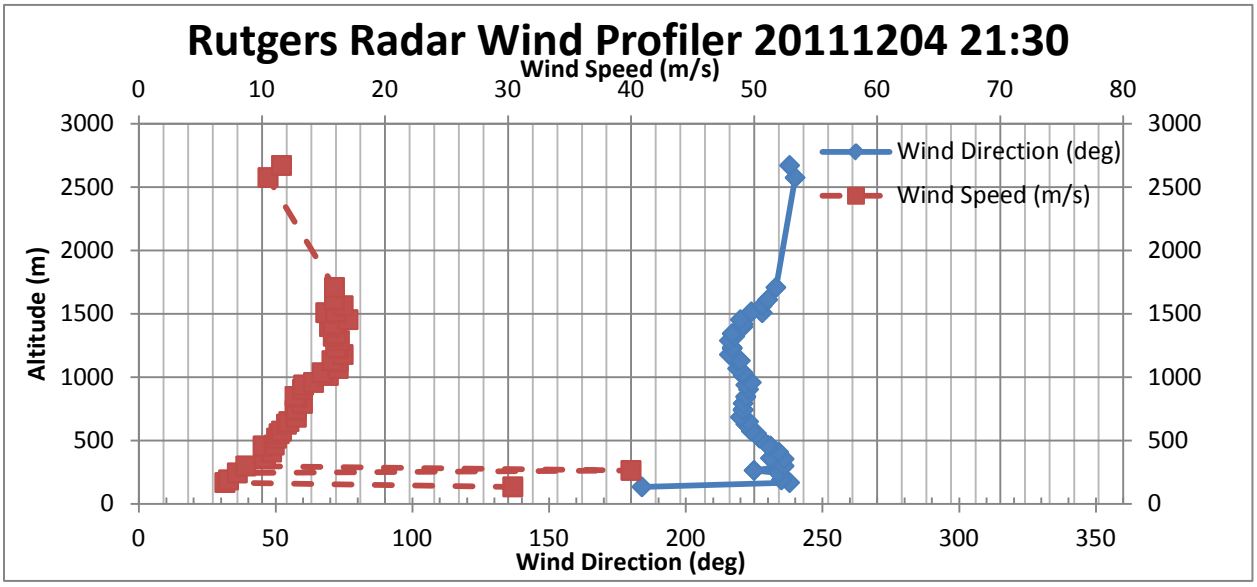


Figure 114, Radar wind profiler wind speed and direction for 12/04/11 21:30 local time, Rutgers University.

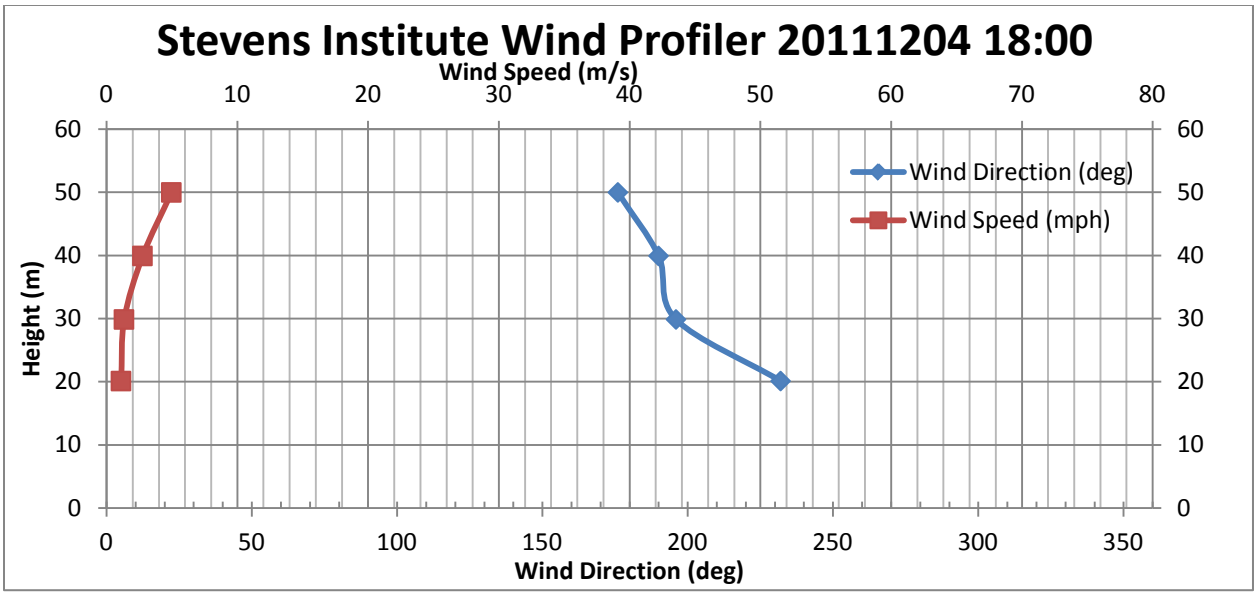


Figure 115, Stevens Institute of Technology sodar wind profile for 12/04/11 18:00 local time.

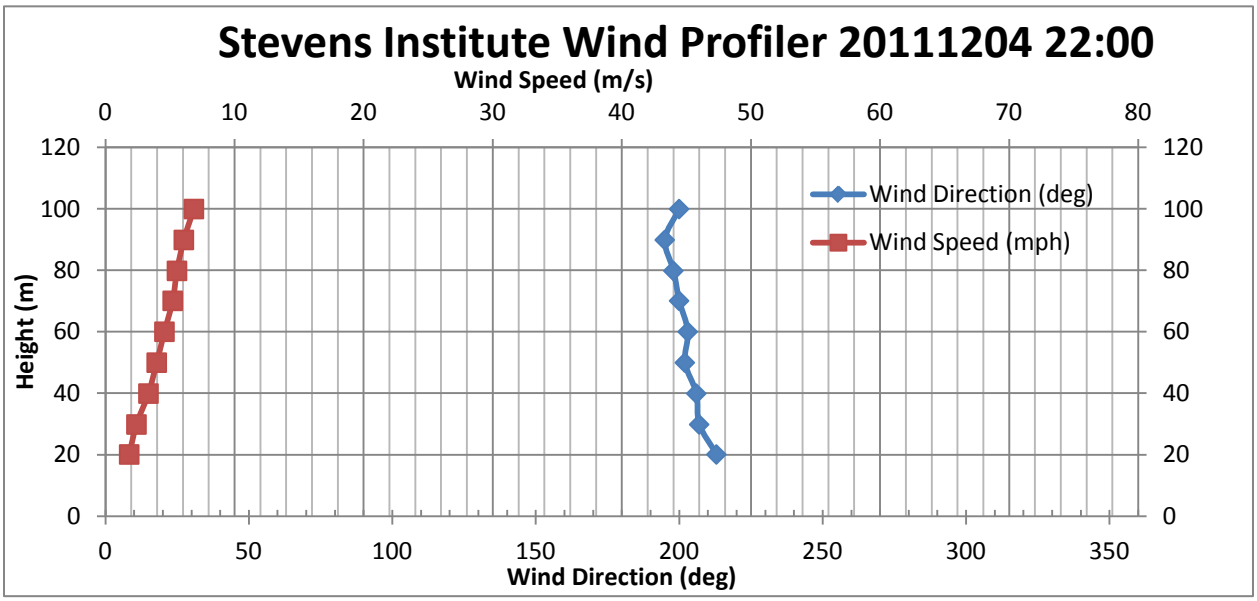


Figure 116, Stevens Institute of Technology sodar wind profile for 12/04/11 22:00 local time.

12/05/2011 Data

On December 5th, 2011 the Doppler Lidar system was operated for approximately four hours and 15 minutes. During two hour of this period the CCNY Direct detection Lidar was also in

operation. The profiles of the range corrected signal intensity of the Doppler Lidar (Figure 117) and the CCNY direct detection Lidar (Figure 119) show similar characteristic in the altitude profiles. After a range of 2200 m to 2500 m there is little or no signal return as there is a cloud layer that blocks most of the signal. The Doppler Lidar system in its current configuration only makes measurements to approximately 3000 m. The Direct detection Lidar can measure to a much greater distance as seen near 14:00 hours at an altitude of 6000 m. This target, most likely a cloud, is beyond the current range of the Doppler Lidar. The ceilometer data shown in Figure 120 shows the matching cloud layer and a region bellow the cloud base to approximately 1000 m where the return signal is weak.

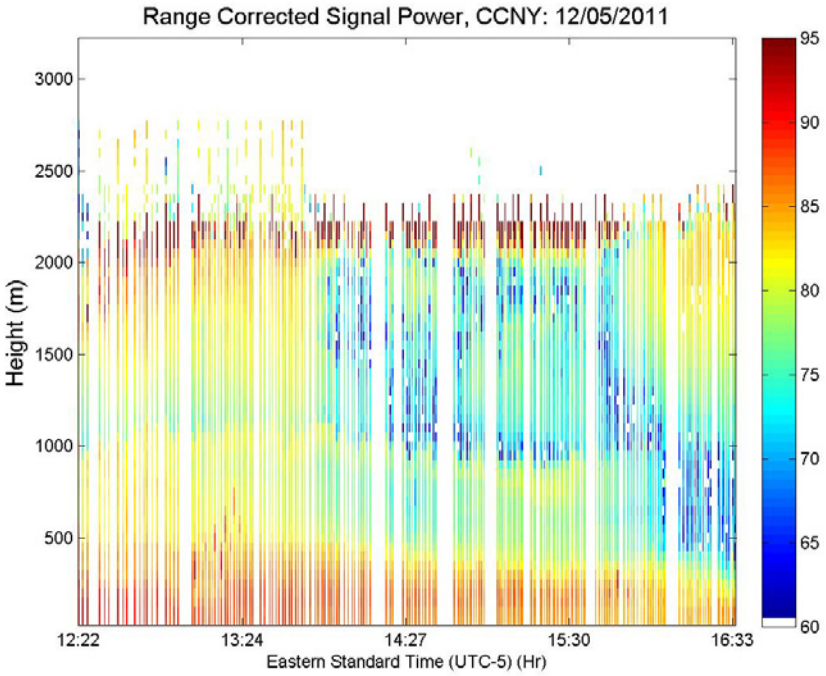


Figure 117, Range corrected signal intensity, 12/05/11.

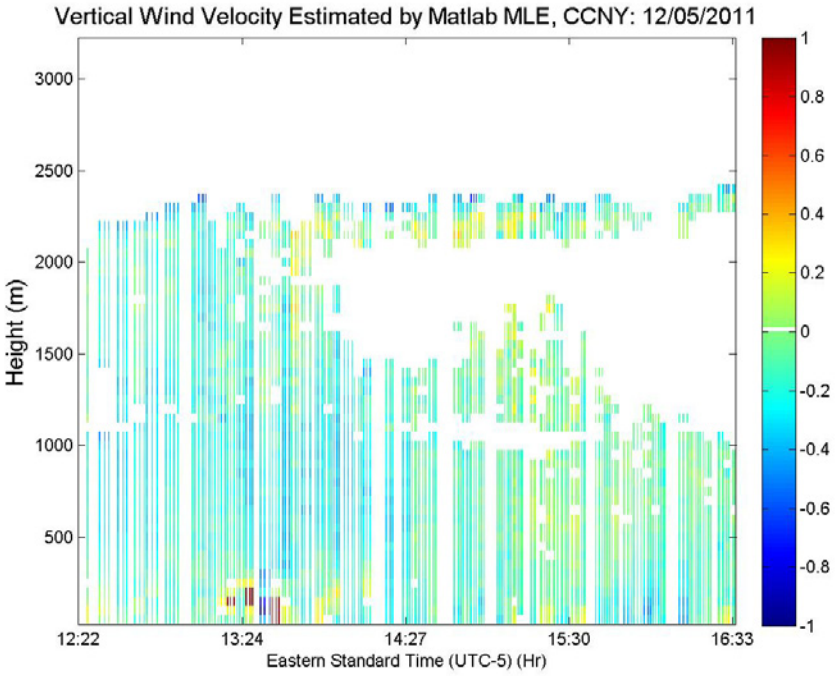


Figure 118, Vertical wind velocity 12/05/11. (colorbar m/s)

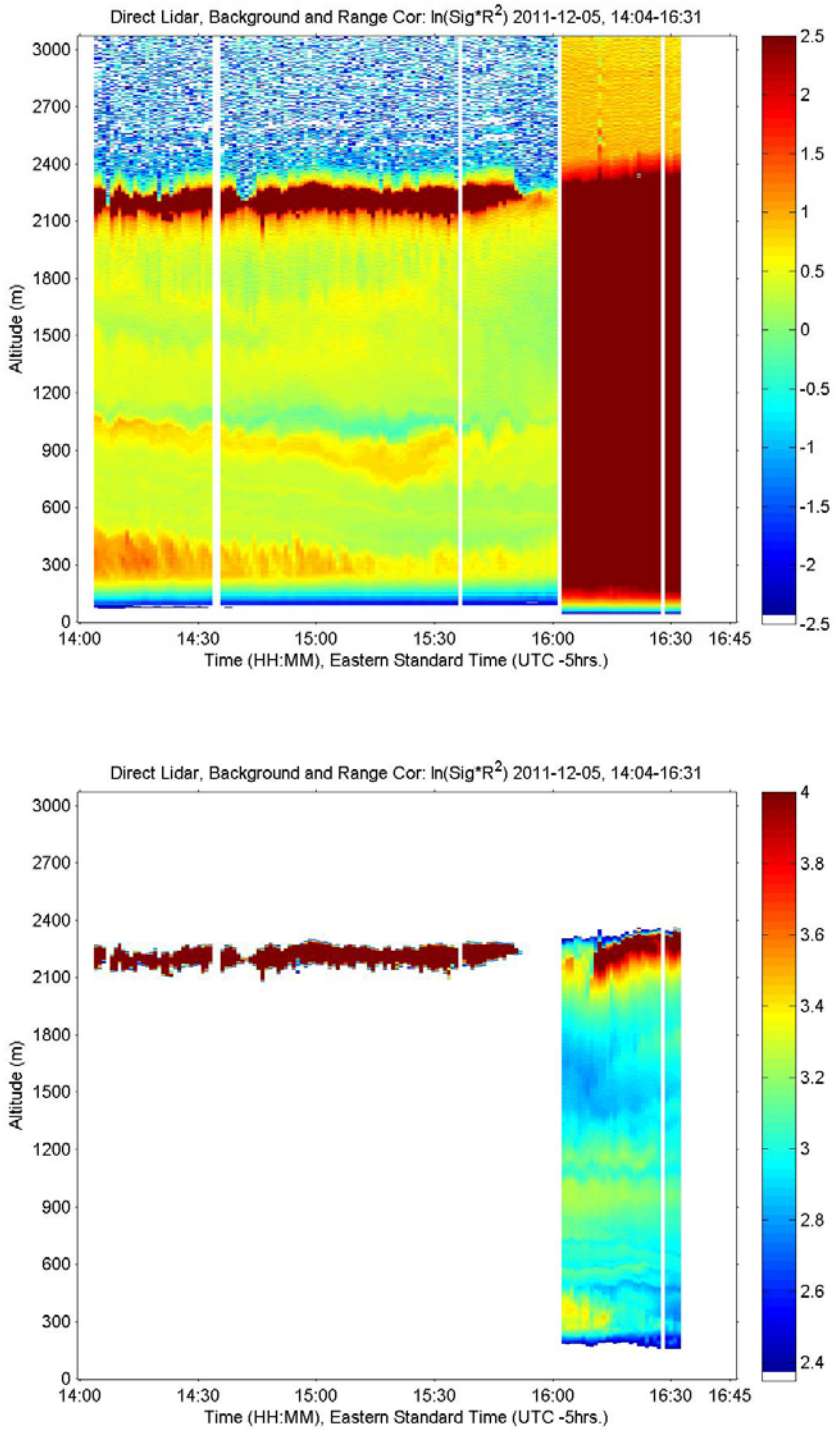


Figure 119, CCNY Direct detection Lidar return signal for 1064nm on 12/05/11. A neutral density filter was removed at 16:00. A single color scale cannot show the details for both cases so the top plot is scaled for the neutral density filter being in place while the lower plot is scaled for the case where there was no filter in use.

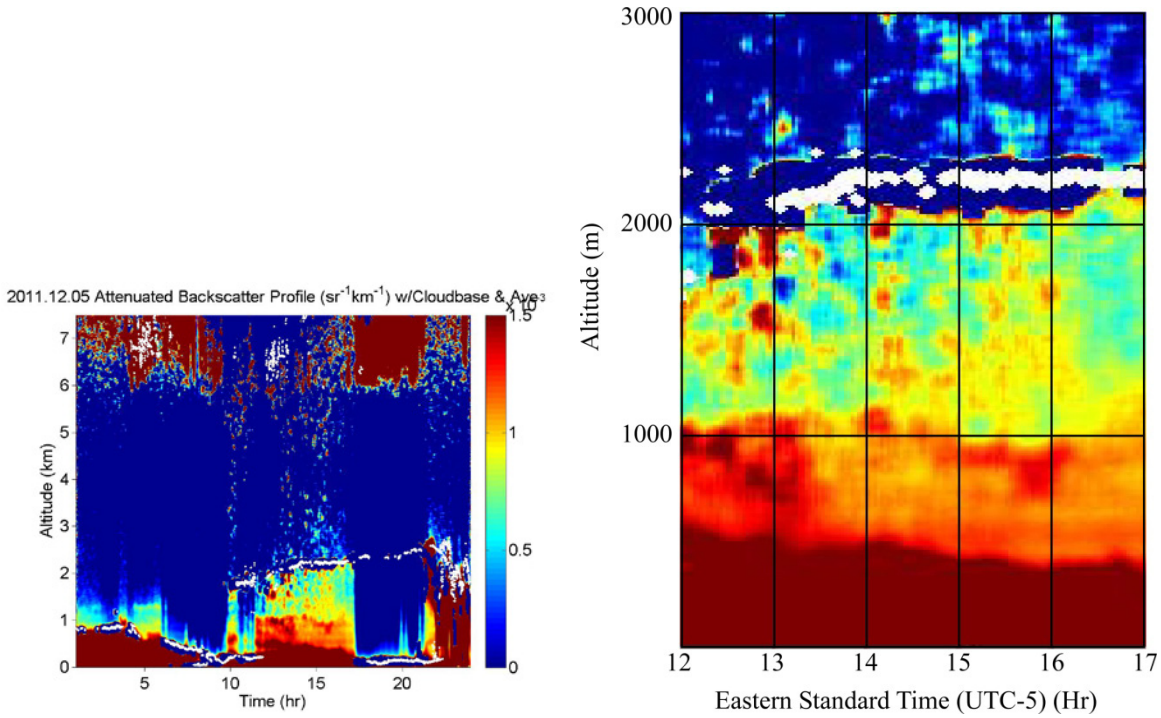


Figure 120, Ceilometer averaged data with clouds for 12/05/11. (Left) 24 hour period, (right) close up view of 12:00 to 17:00.

Wind barb plots from the Doppler Lidar for the same day are shown in Figure 121 while Figure 122 shows a Windgram based on the NAM 12 km model for 18Z and 21Z. The model's plot is of three hour averages while the Doppler Lidar plot is on the order of 15 minute averages. Even with this difference there is a correlation between the two plots. At 13:00 local time and 18Z the directions are similar and the speed increase and then decreases with altitude. At 16:00 local time and 21Z the direction of the wind is North or slightly North West at low altitudes but at higher altitudes the direction becomes more North Eastern.

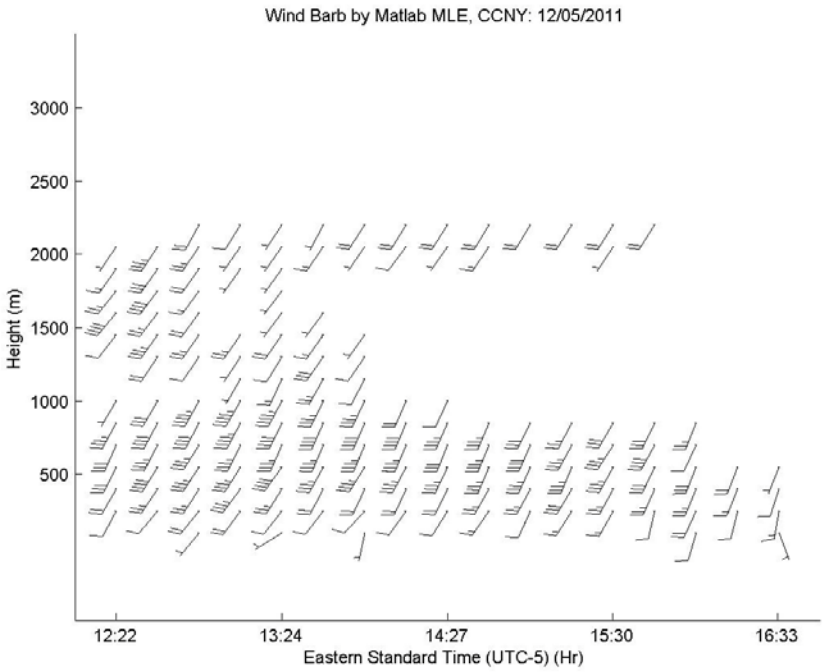


Figure 121, Doppler Lidar wind barb plot for 12/05/11.

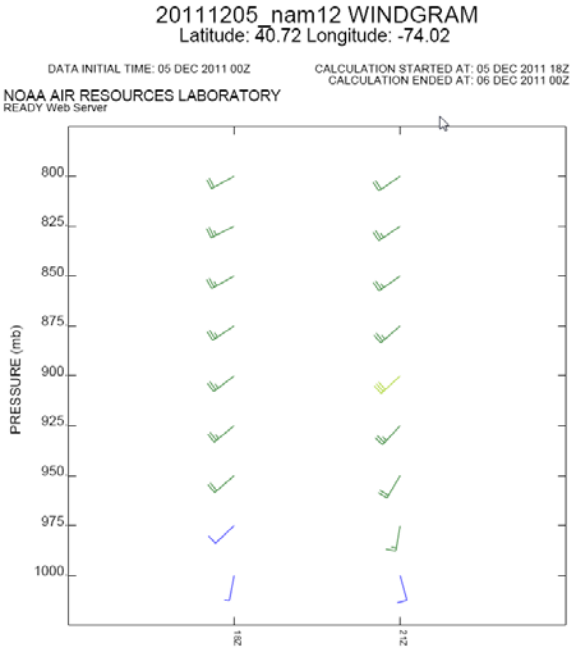


Figure 122, Windgram from NAM 12 km model for 12/05/11, 16Z and 21Z (11 and 16 EST).

The regional NAM maps shown in Figure 123 to Figure 130 illustrate how the wind at the same altitude and time of day can have different directions across the region. Since air is a compressible fluid some of the air mass moving into a region from a different direction may result in a rise in pressure but there will also be vertical components to the air mass's movement. At 18Z, at low altitude, a uniform but low speed northerly moving wind direction is seen across New Jersey in Figure 123 with a north westerly moving wind direction seen in the lower New York Harbor. At the same time, 18Z, but at greater altitude, 900 mb (~1000 m), the wind direction is uniformly north easterly moving at a greater speed (Figure 127). This means that the wind near the surface is moving in a direction that is perpendicular to the wind direction at 900 mb over the lower New York Harbor while over New Jersey the directional difference is half that.

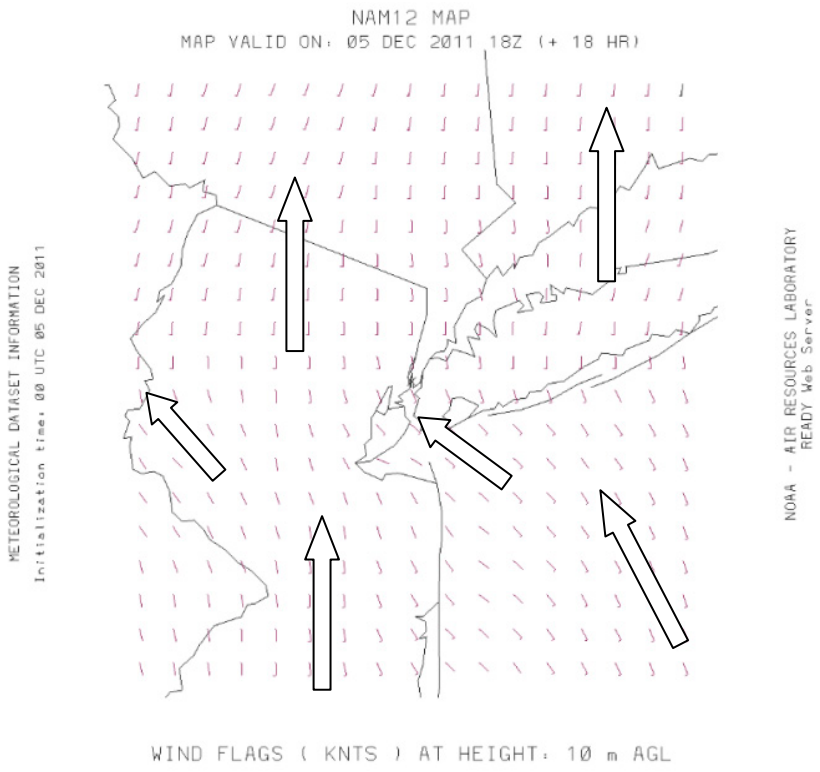


Figure 123, NAM map model for 12/05/11 at 10 m above ground level, 18Z (13 EST).

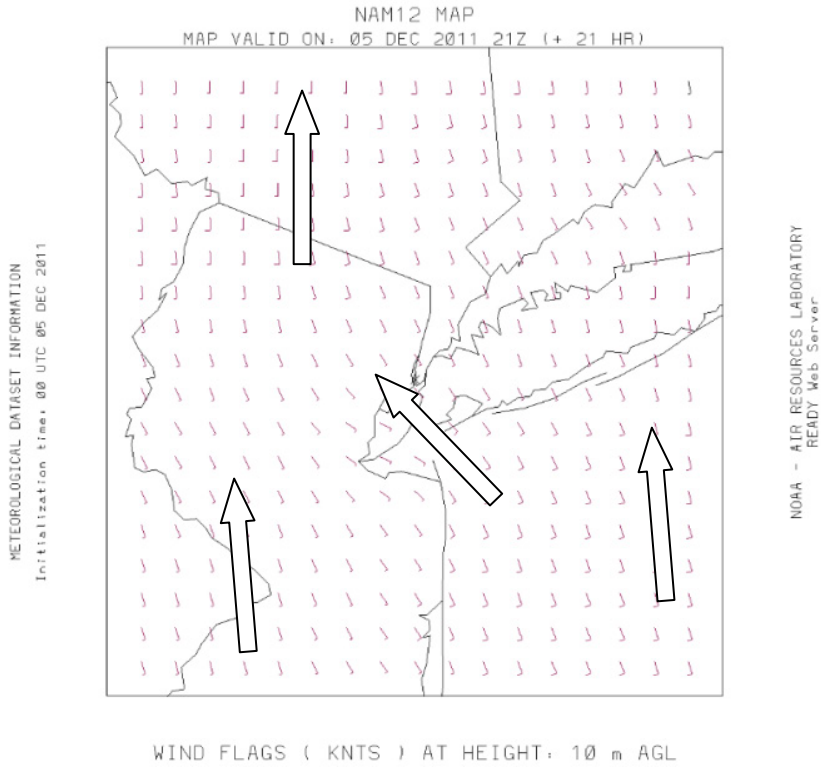


Figure 124, NAM map model for 12/05/11 at 10 m above ground level, 21Z (16 EST).

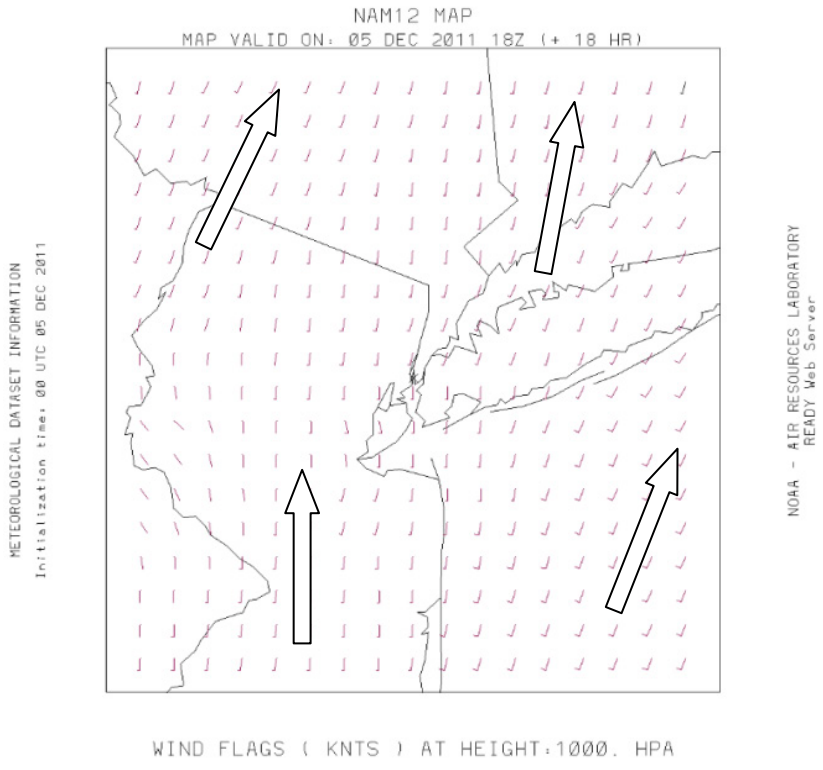


Figure 125, NAM map model for 12/05/11 at 1000 mb above ground level, 18Z (13 EST).

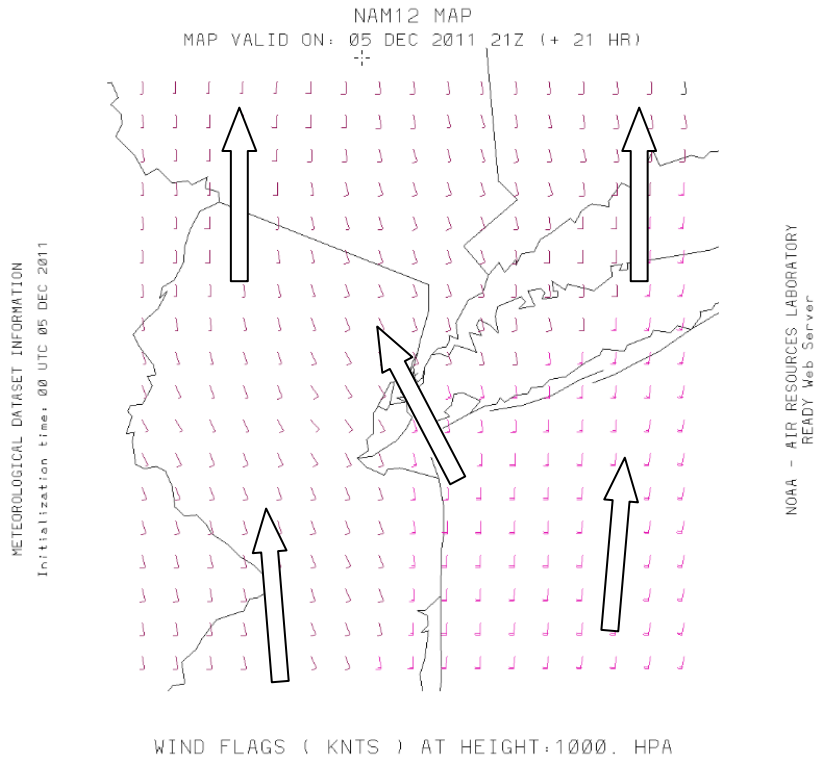


Figure 126, NAM map model for 12/05/11 at 1000 mb above ground level, 21Z (16 EST)..

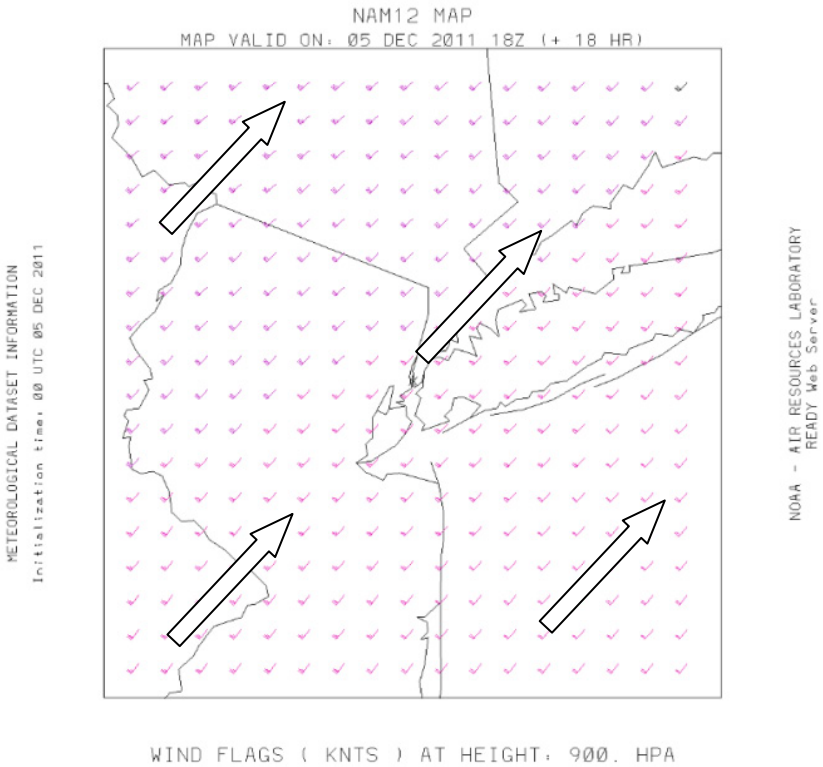


Figure 127, NAM map model for 12/05/11 at 900 mb above ground level, 18Z (13 EST).

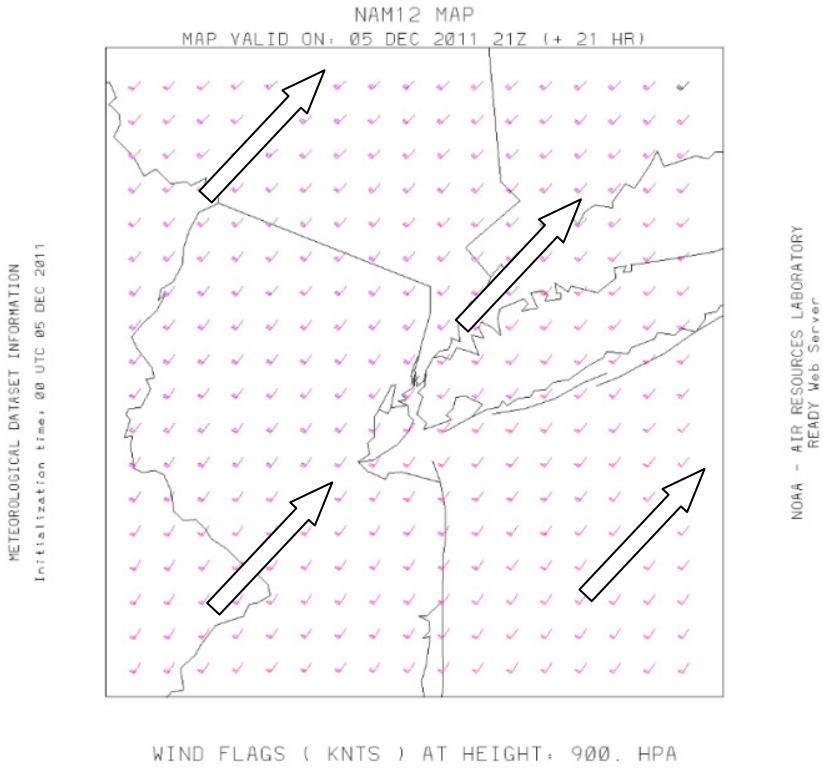


Figure 128, NAM map model for 12/05/11 at 900 mb above ground level, 21Z (16 EST)..

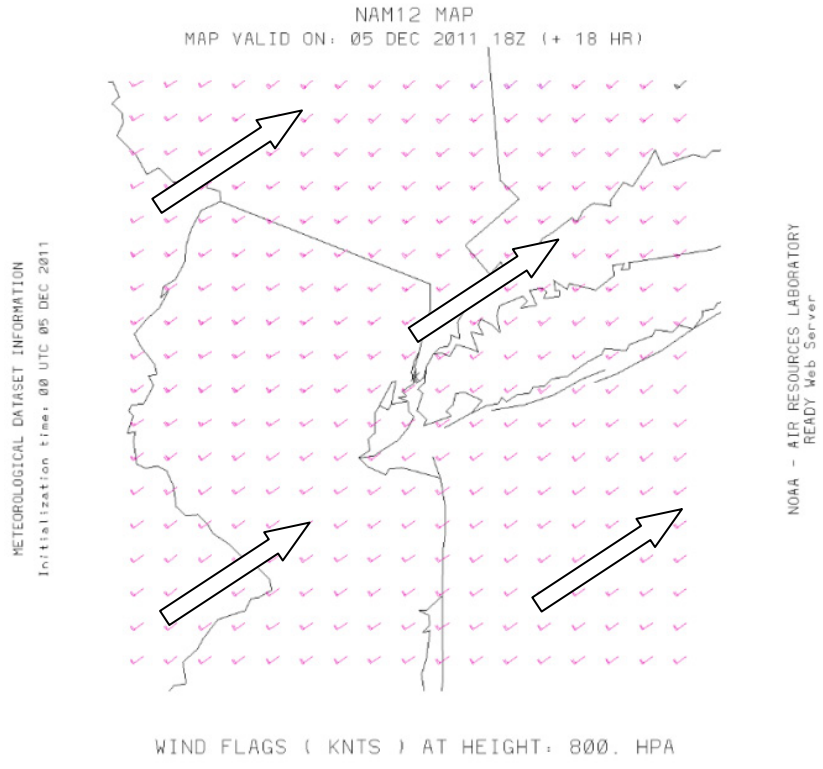


Figure 129, NAM map model for 12/05/11 at 800 mb above ground level, 18Z (13 EST).

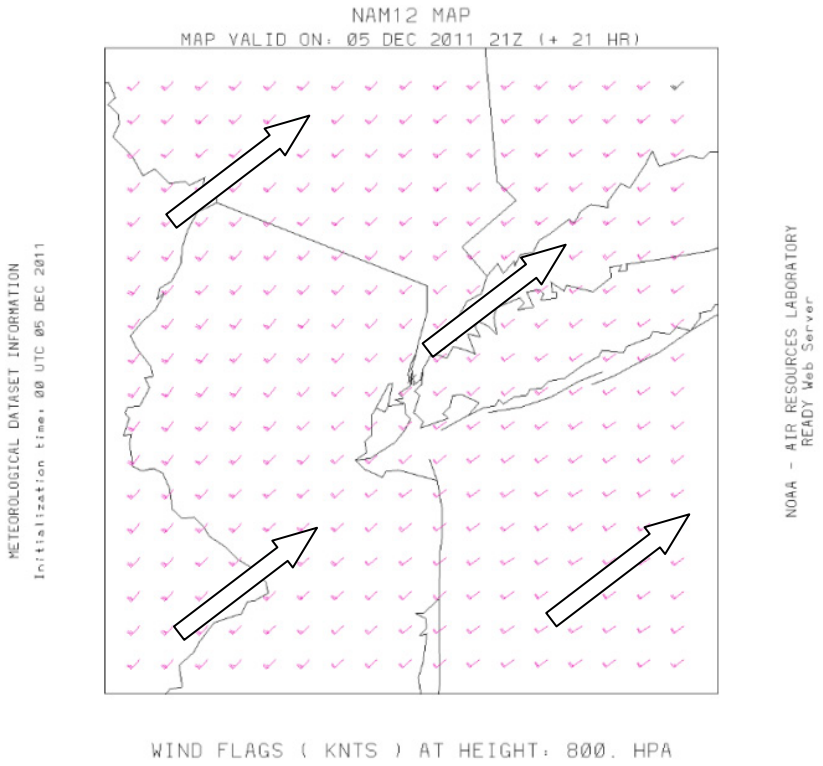


Figure 130, NAM map model for 12/05/11 at 800 mb above ground level, 21Z (16 EST)..

12/09/2011 Data

In the figures bellow the wind speed and direction indicated by the wind barbs (Figure 131) agree with the NAM windgram (Figure 132) in that they exhibit similar behavior in the direction and speeds. The windgram is model based for 12 km cells with 3 hour averages so the comparison is heuristic. The NAM stability plot in the figure bellow is in UTC time, Eastern Standard Time + 5 hours, covering 7:00 to 16:00 EST.

When the stability plot (Figure 133), showing the height of the planetary boundary layer indicates the layer increasing in altitude as time progresses, the Doppler wind Lidar (Figure 134) shows a similar behavior, as the return signals are too weak above the planetary boundary layer

to calculate a wind measurement. This is the result of the lack of aerosols to backscatter the outgoing pulse above the boundary layer.

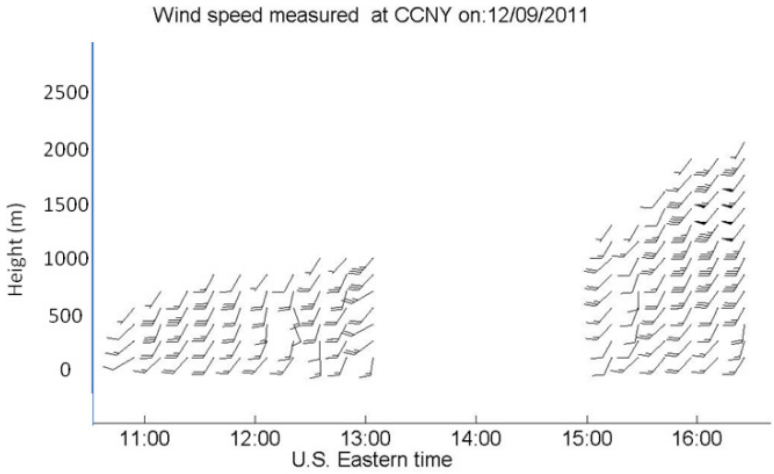


Figure 131, Doppler Lidar horizontal wind speed and direction versus altitude as wind barbs for 12/09/12.

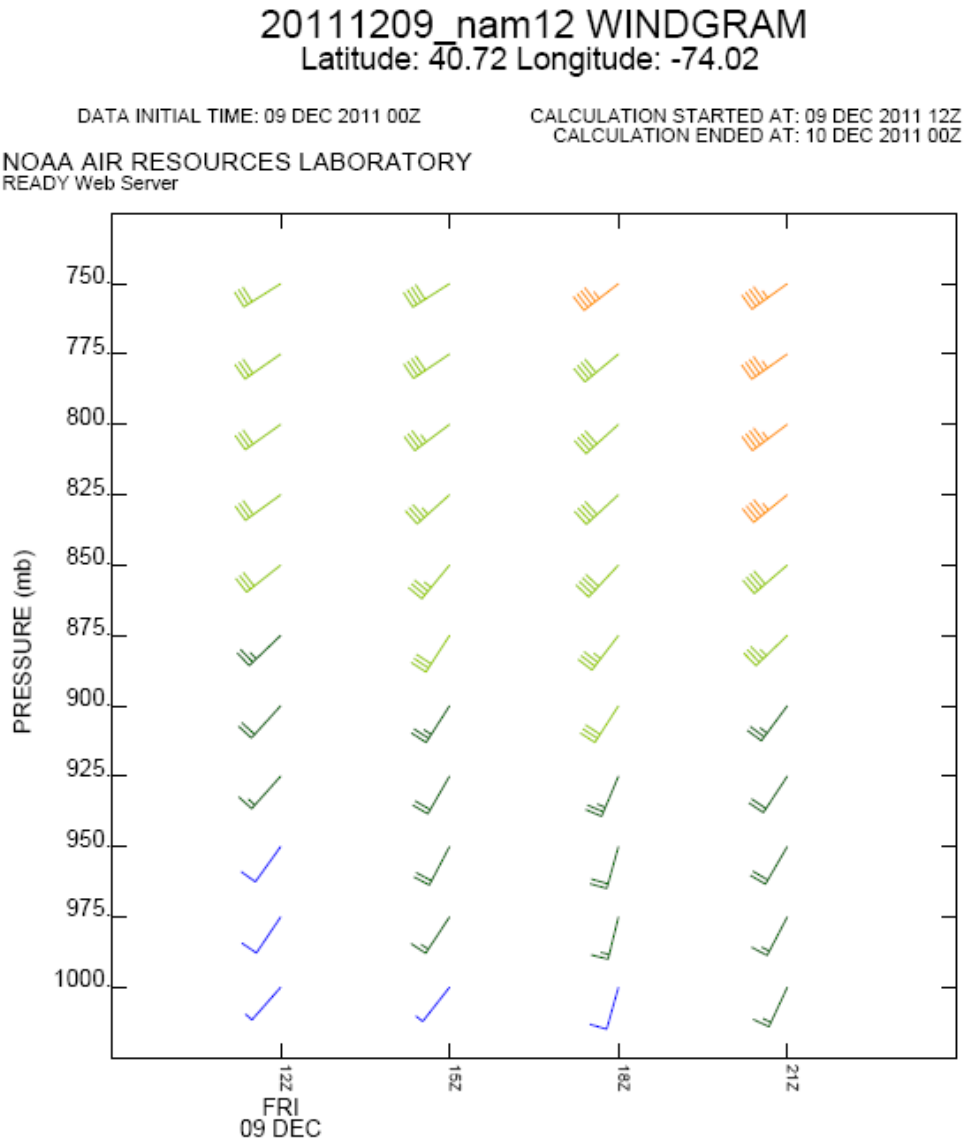
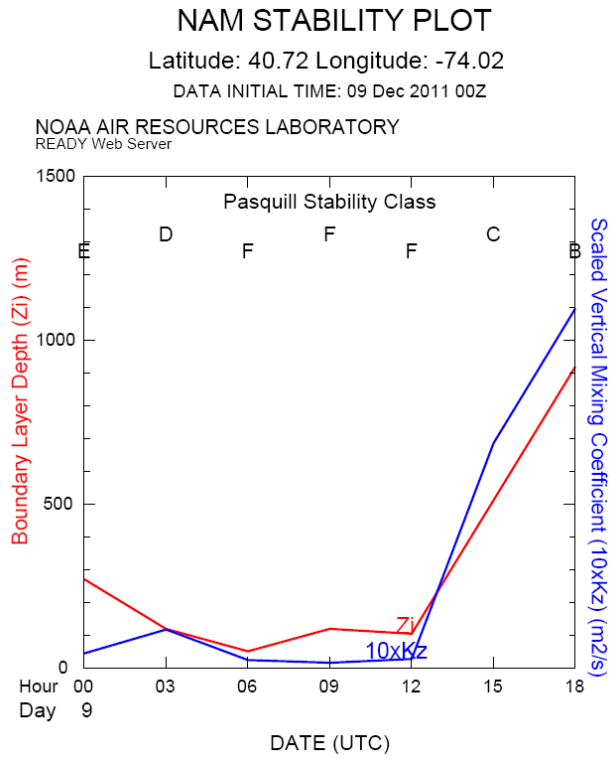


Figure 132, NAM model windgram for 12/09/12 , 12Z, 16Z, 18Z, 21Z in UTC time (7, 11, 13, 16 in EST).



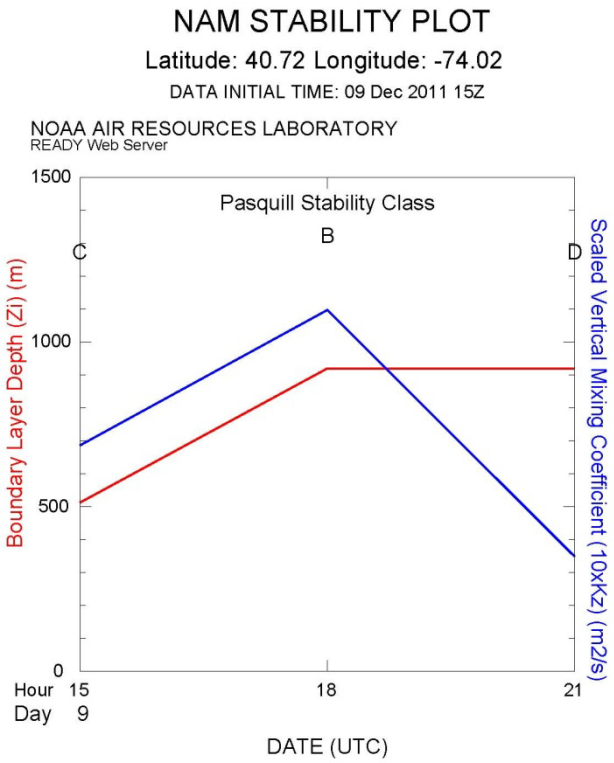


Figure 133, The NAM stability model for 12/09/12, (times: 0Z, 3Z, 6Z, 12Z, 15Z 18Z 21Z UTC time, previous day 19. Previous day 22, 1, 7, 10, 13, 16 EST.)

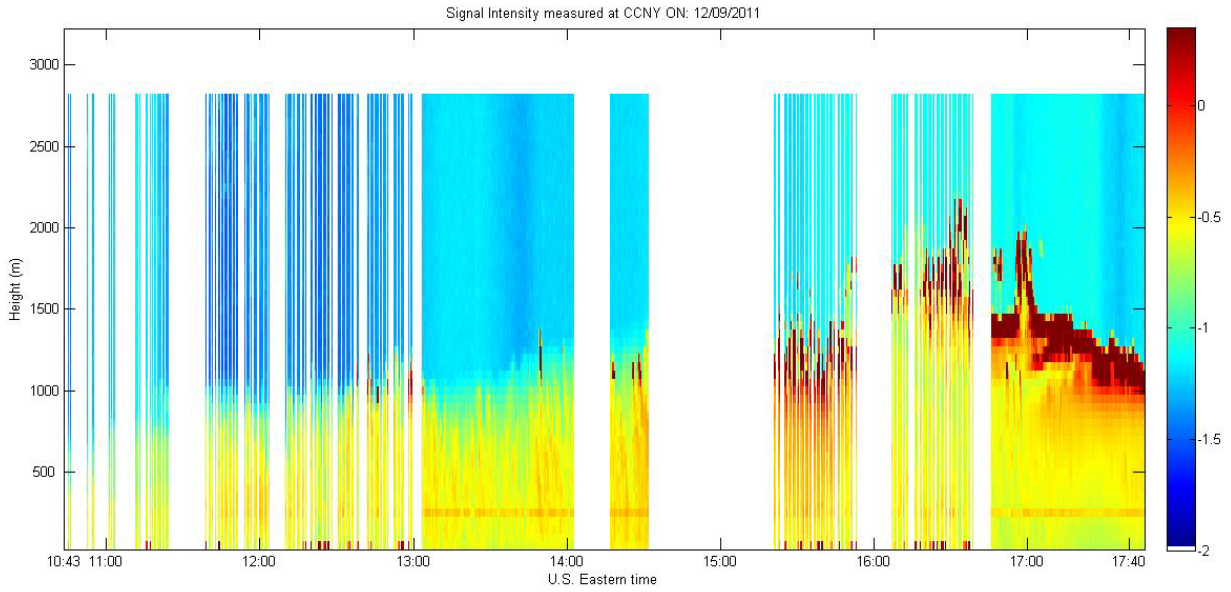


Figure 134, Range corrected coherent Lidar signal strength vs. time (EST) 12/09/11.

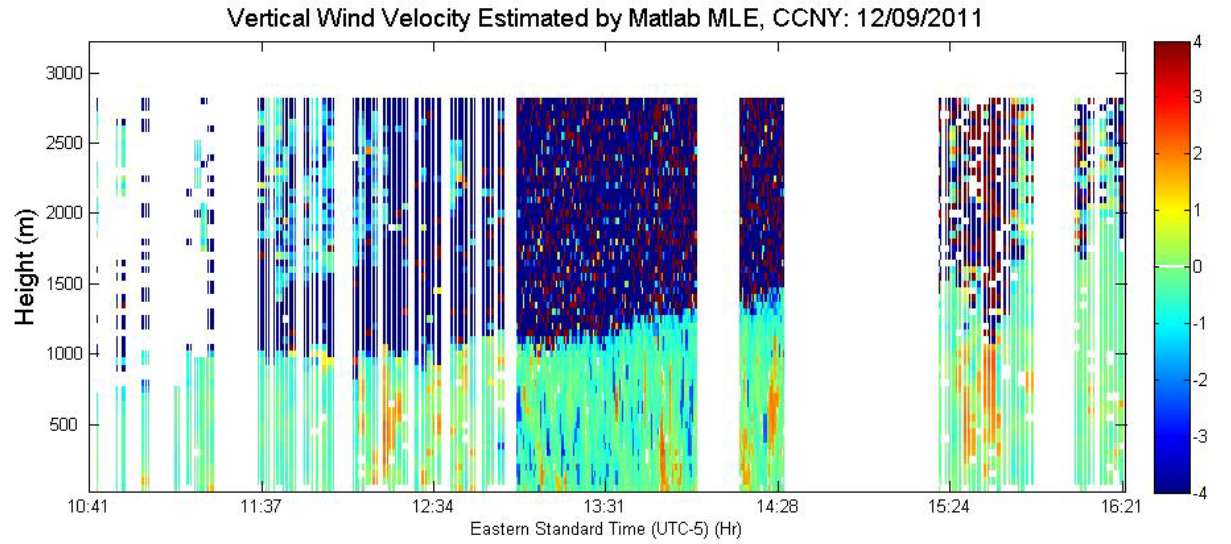


Figure 135, Vertical wind speed component as a function of time (EST) up to 3 km 12/09/11 (colorbar m/s, positive values are updrafts).

The range corrected direct detection Lidar (Figure 136) and the range corrected Doppler Lidar (Figure 134) exhibit similar patterns with the return signal intensity at altitude following a general pattern where the altitude of the highest valid return signal increase over time. There are periods in which the altitude decreases which also coincide between the two figures.

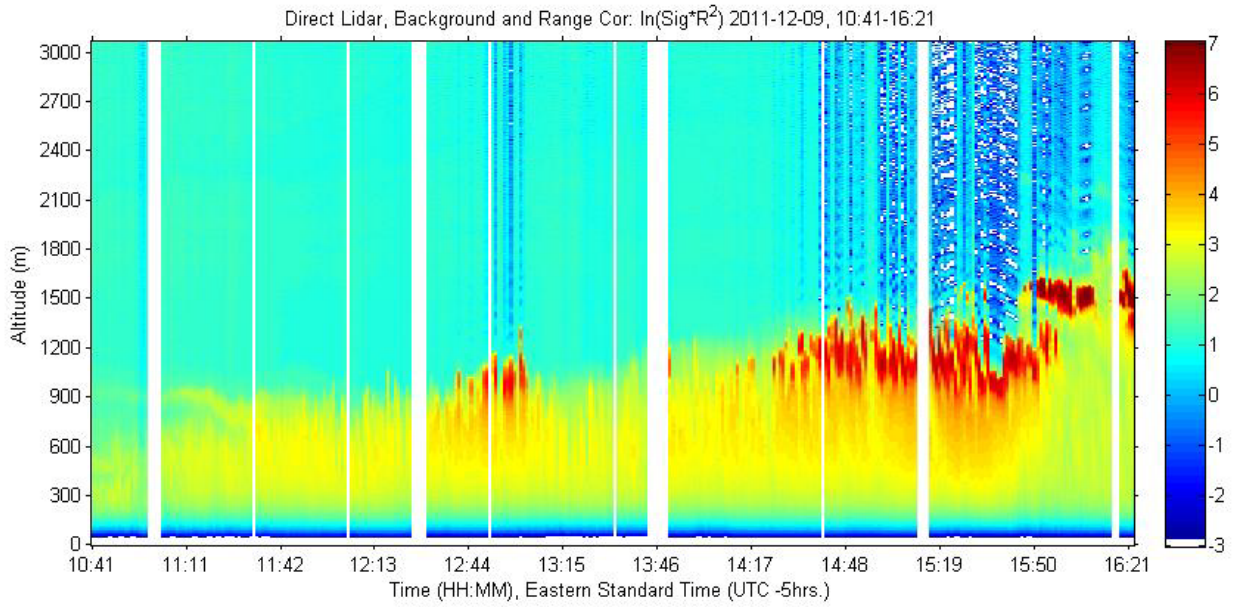


Figure 136, Range corrected direct detection Lidar signal (EST) (au)

.

The ceilometer (Figure 137) return signal correlates with the Doppler Lidar system.

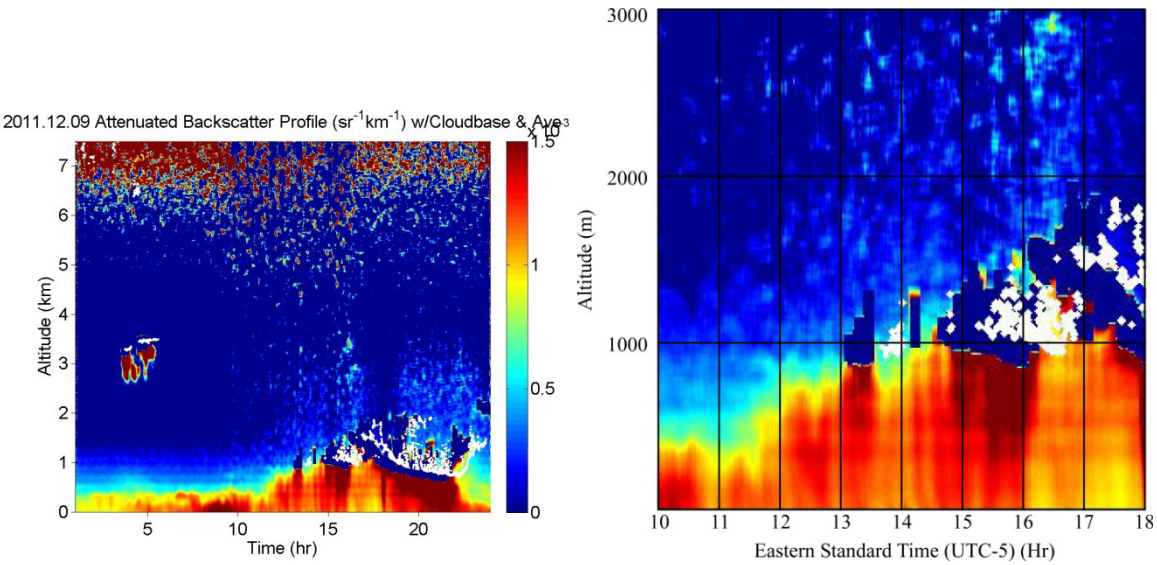


Figure 137, Ceilometer averaged data with clouds for 12/09/11. (Left) 24 hour period, (right) close up view of 10:00 to 18:00.

The microwave radiometer (Figure 138) shows relatively constant temperature profile while the water vapor profile (Figure 139) changes over time. The water vapor profile evolution during the day tracks the Doppler Lidar range corrected signal strength.

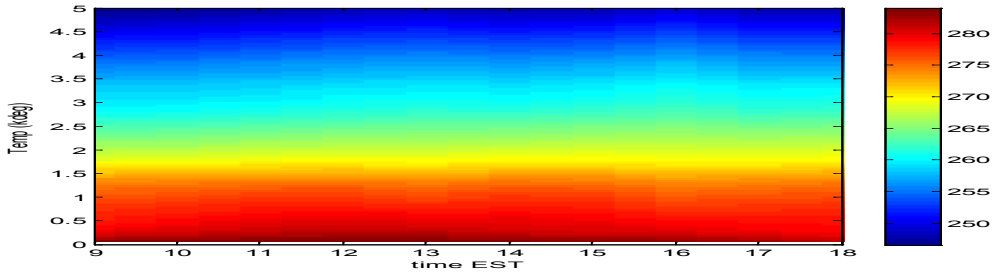


Figure 138, Temperature vertical profile vs. time measured using the hyperspectral microwave radiometer 12/09/11..

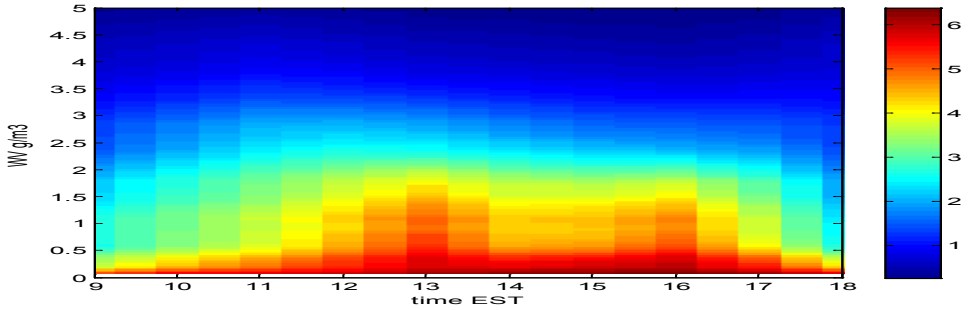


Figure 139, Water Vapor profile vs. time as measured by the hyperspectral Microwave radiometer.

11/30/2011 Data

On Noverber 30, 2011 the Doppler Lidar system was operated for approximetley one hour. The data is shown in and Figure 140 and Figure 141. The Direct detection was not in operation at this time. Ceilometer data is shown in Figure 143. The cloud base durring the Doppler Lidar observation period is approximetaly 2000 m which matches well with the ceilometer data. The

microwave radiometer water vapor profile also shows the vapor concentrations decreasing significantly after 2250 m (Figure 144).

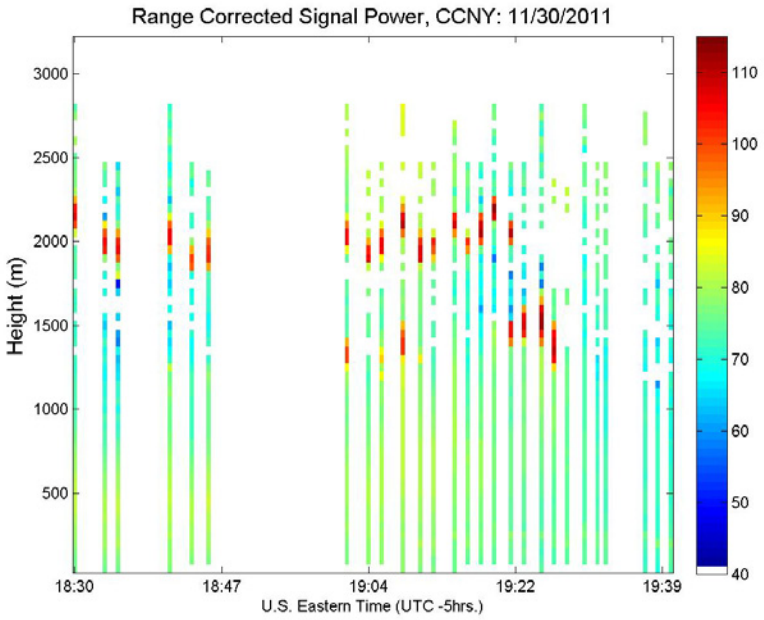


Figure 140, Doppler Lidar range corrected signal intensity 11/30/12.

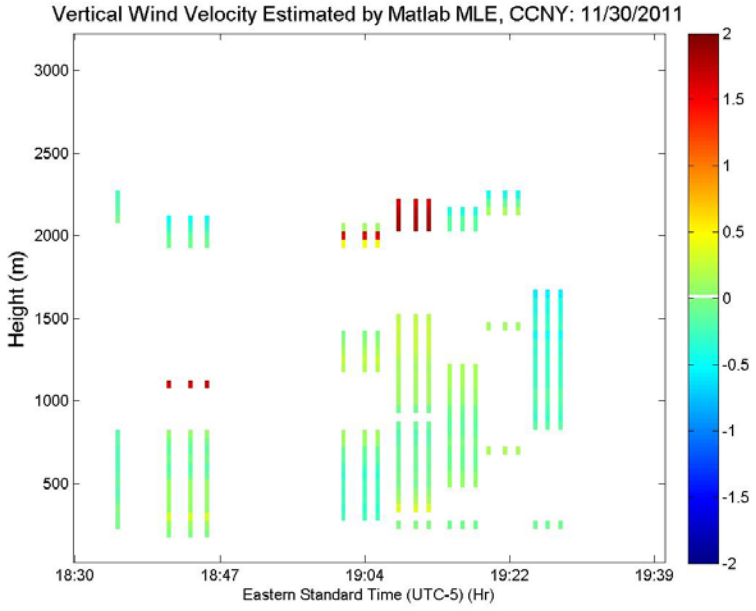


Figure 141, Doppler Lidar vertical velocity 11/30/12 (colorbar m/s, positive values are updrafts).

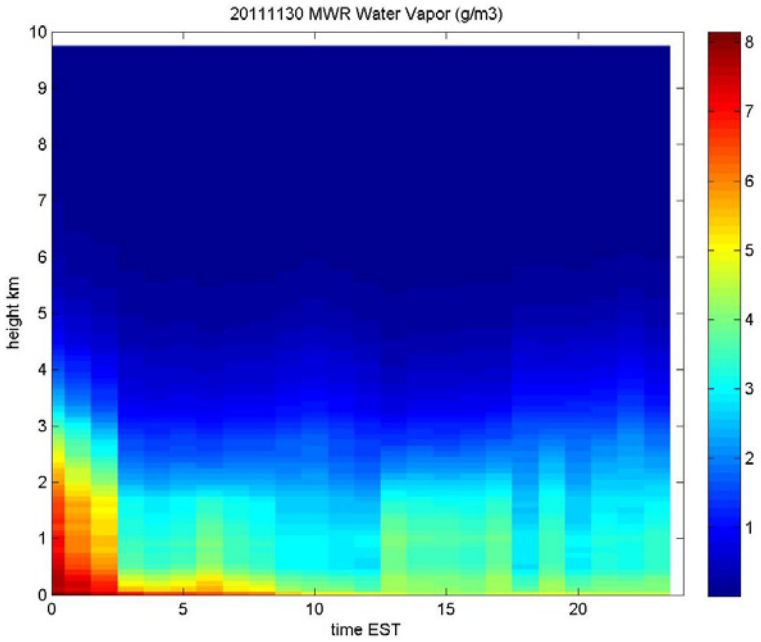


Figure 142, Microwave radiometer water vapor profile for 11/30/12.

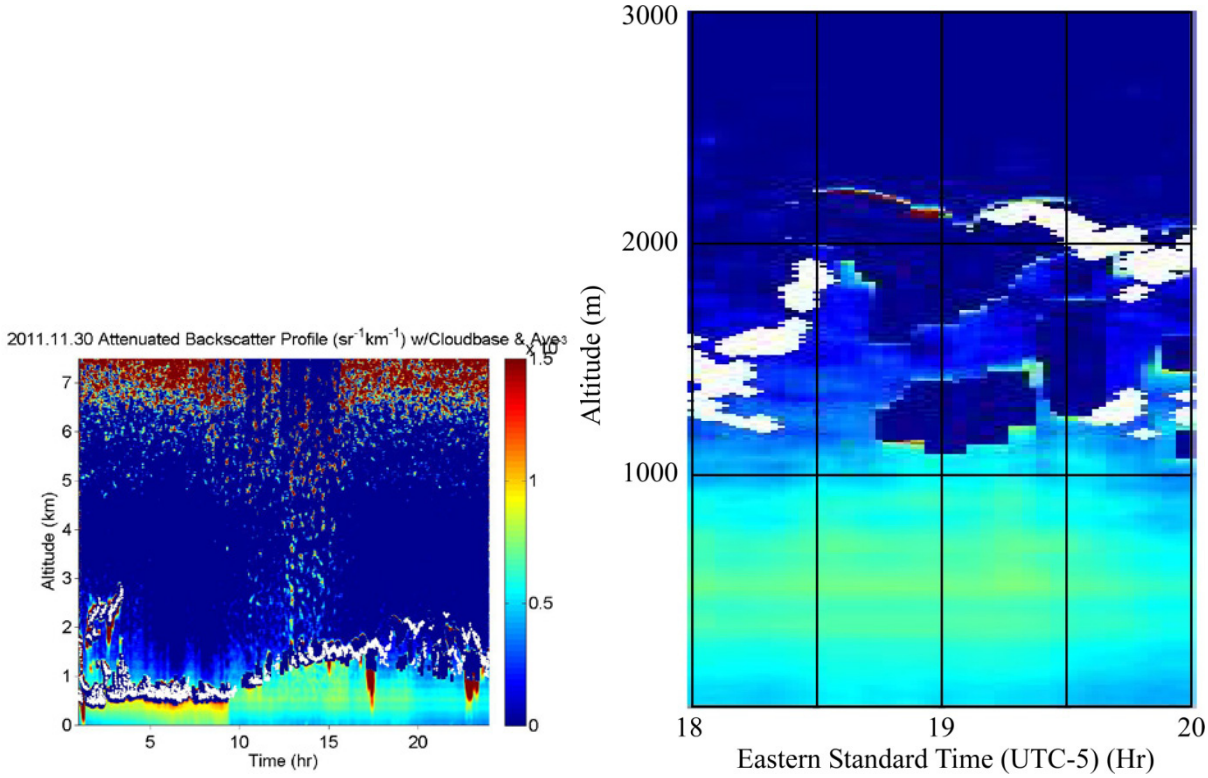


Figure 143, Ceilometer profile for 11/30/11. (Left) 24 hour period, (right) close up view of 18:00 to 20:00.

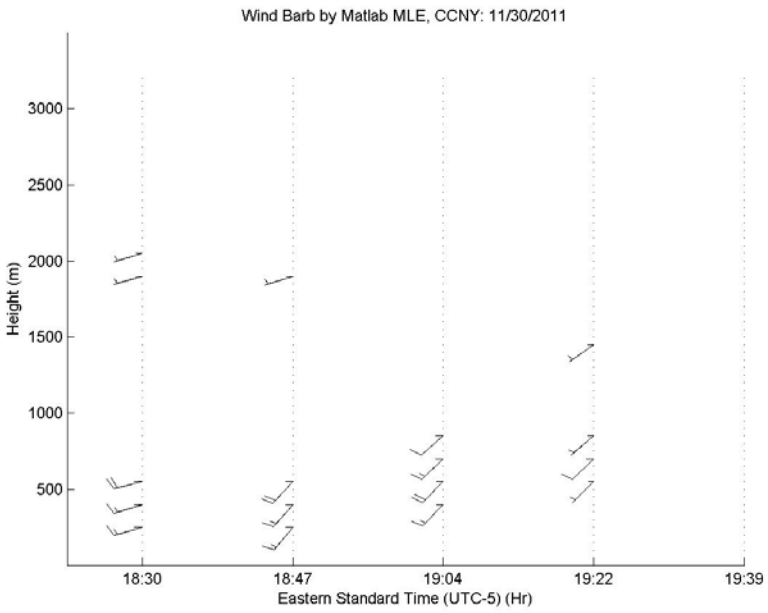


Figure 144, Doppler Lidar wind barbs for 11/30/11.

The wind barb plot in Figure 144 show very few barbs. With sparse data the horizontal wind speed and directions cannot be calculated. This results in many missing wind barbs.

12/12/2011 Data

The CCNY direct detection Lidar seen in Figure 145 shows very little signal return from approximately 500 m to 7 km. The next figure shows a close up view bellow 3 km. The North American Model (NAM) stability plots for the same date are show in Figure 147 and Figure 148. Note that the time on the NAM plots are UTC or Zulu time. The boundary layer depth from these graphs coincides with the intense return signals seen close to the ground by the direct detection Lidar. This is confirmed by the ceilometer (see Figure 149). The coherent Doppler Lidar was operated in a scanning mode on this particular date. The vertical components' signal intensities, corrected for range are shown in Figure 150 and Figure 151. The lack or return signal from altitudes greater than 500 m is due to lack of aerosols for the coherent Doppler Lidar to target.

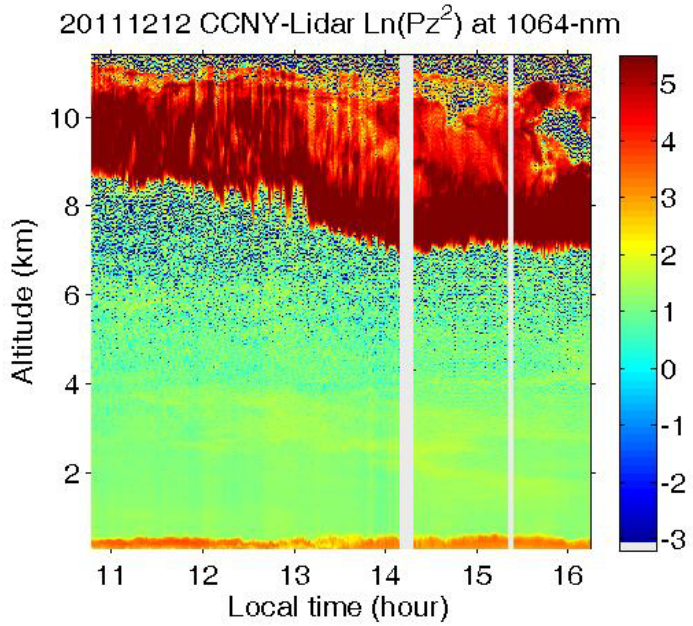


Figure 145, CCNY Direct detection Lidar data of 12/12/2011.

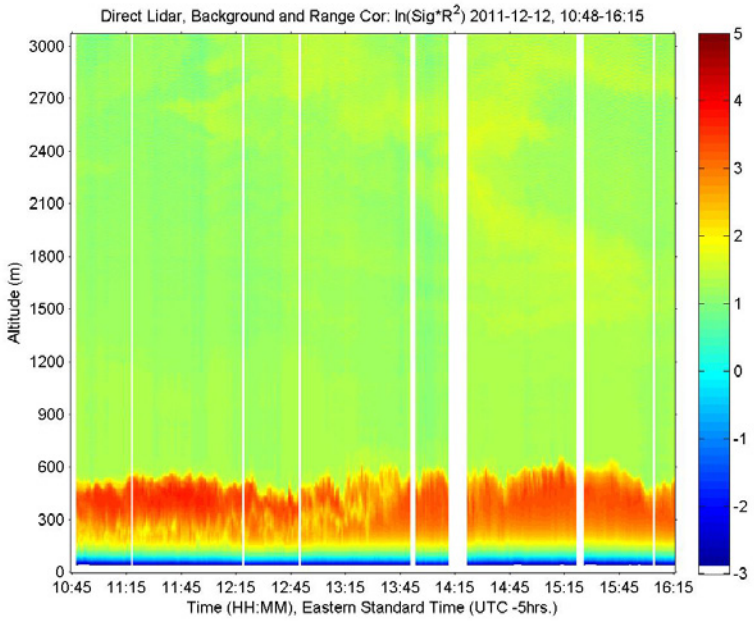


Figure 146, Close up view of CCNY Direct detection Lidar data of 12/12/2011.

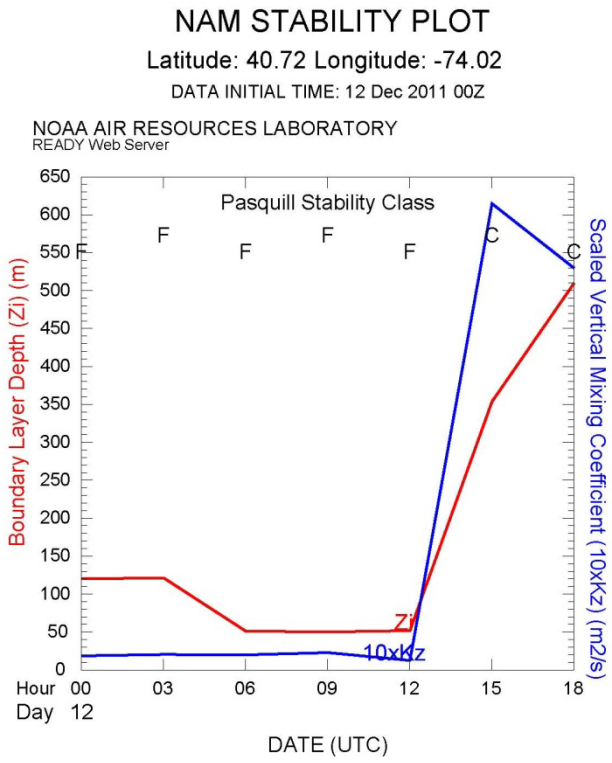


Figure 147, NAM Stability model plot for 12/12/2011, 00Z to 18Z UTC (previous day 19 to current day 13 EST).

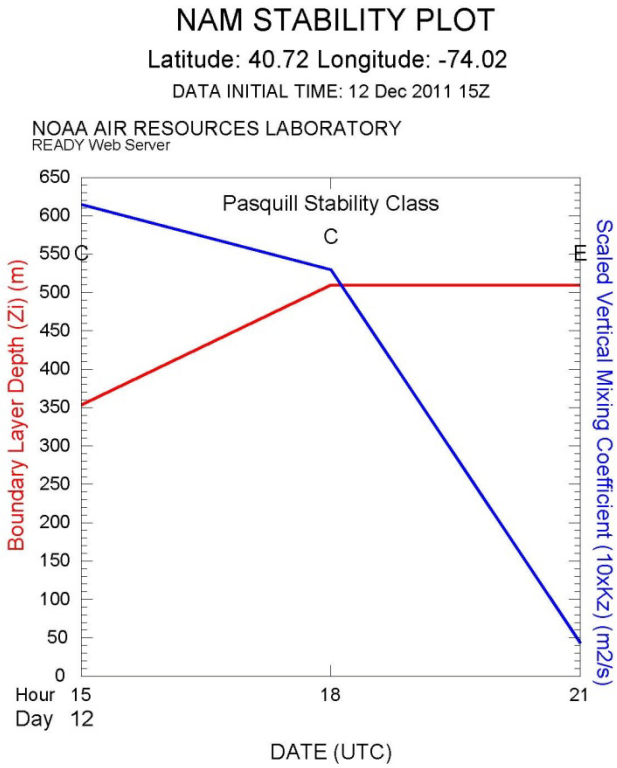


Figure 148, NAM Stability model plot for 12/12/2011, 15Z, 18Z, 21Z UTC (10, 13, 16 EST).

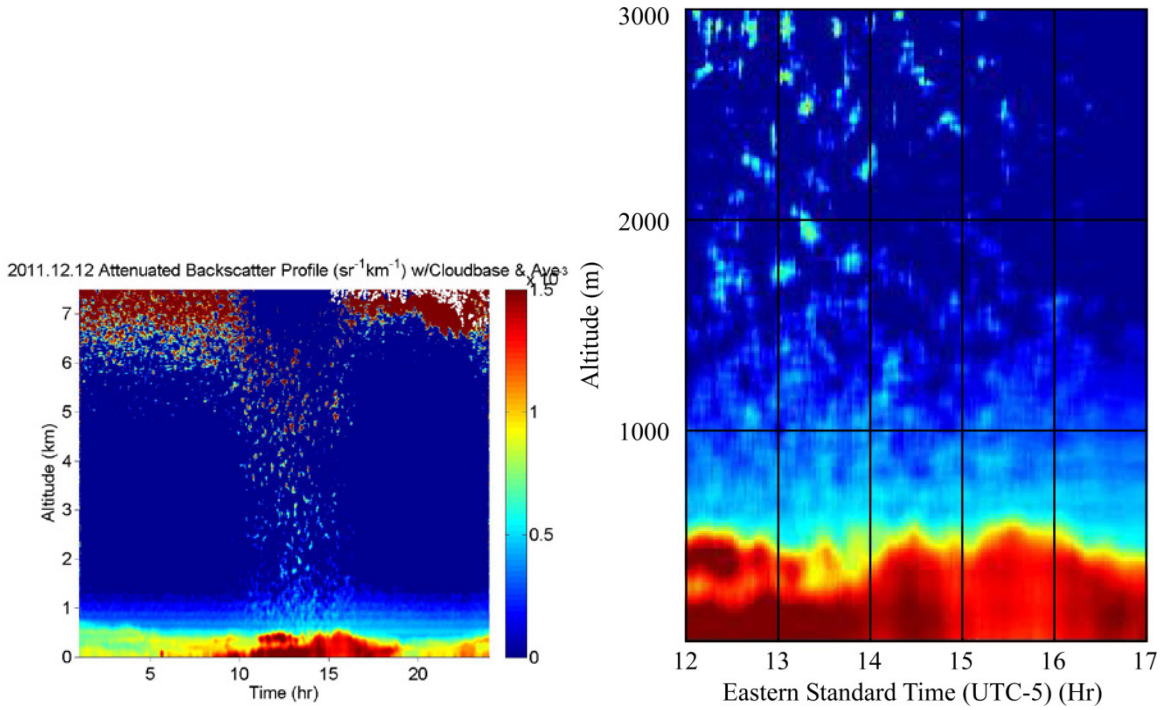


Figure 149, Ceilometer return signal for 12/12/2011. (Left) 24 hour period, (right) close up view of 12:00 to 17:00.

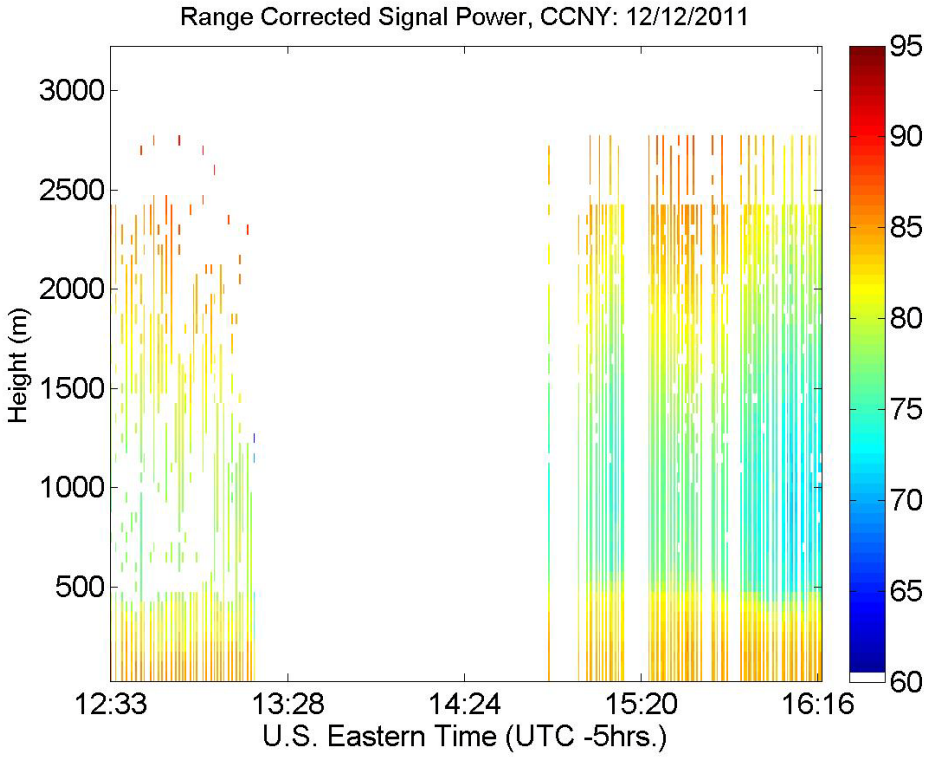


Figure 150, Coherent Doppler Lidar range corrected intensity signal for 12/12/2011, 12:33 to 13:19, and 14:51 to 16:17.

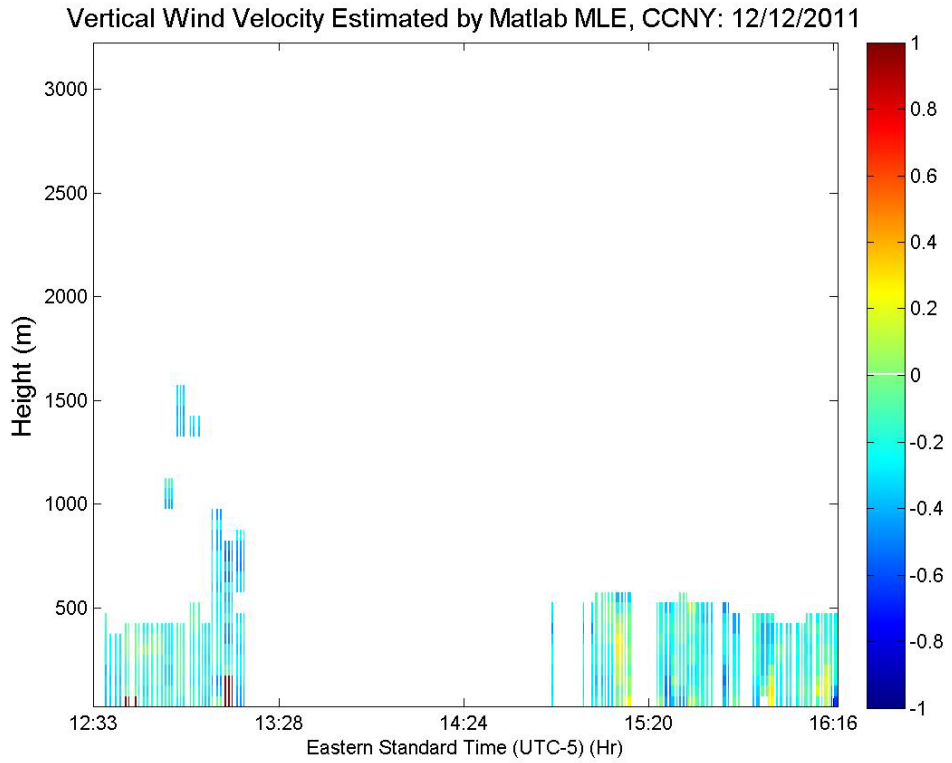


Figure 151, Coherent Doppler vertical velocity for 12/12/2011, 14:51 to 16:17 (colorbar in m/s).

12/13/2011 Data

On December 13, 2011 the Doppler Lidar system was operated in a vertical staring mode from 9:04-12:04, 12:21-15:21, and 15:28-17:28. The plot of signal intensity and vertical wind velocity are shown in Figure 152 and Figure 153. The ceilometer return signal for 12/13/2011 is shown in Figure 154 for a 24 hour period and the hours of interest. The CCNY direct Lidar was operated from 10:56-16:08 and a plot of the data can be seen in Figure 155. There is weak or little return signal from beyond 1 km. This can be seen in the data from all three instruments. The atmosphere is very "clean" past 1 km. The top of the layer of aerosols is low near 09:00 and rises through the day until 14:46. The top of the layer then lowers slightly until 16:00 after which it rises once again. The behavior measured by the Doppler Lidar system correlates well with the

other instruments. The NAM Stability model plot (Figure 156) for 12/13/2011 covering the period or 10:00 to 16:00 shows a boundary layer at approximately 250 m at 10:00. At 13:00 the model predicts the boundary layer to be at 700 m which drops to 500 m by 16:00. This pattern is seen to correlate with that seen in the Doppler Lidar plot bellow.

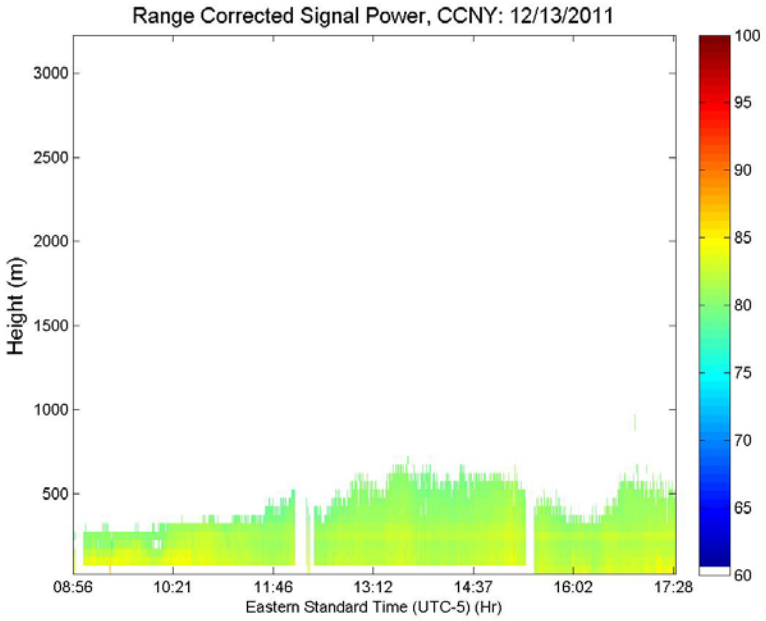


Figure 152, Coherent Doppler Lidar range corrected intensity signal for 12/13/2011, 08:56 to 17:28.

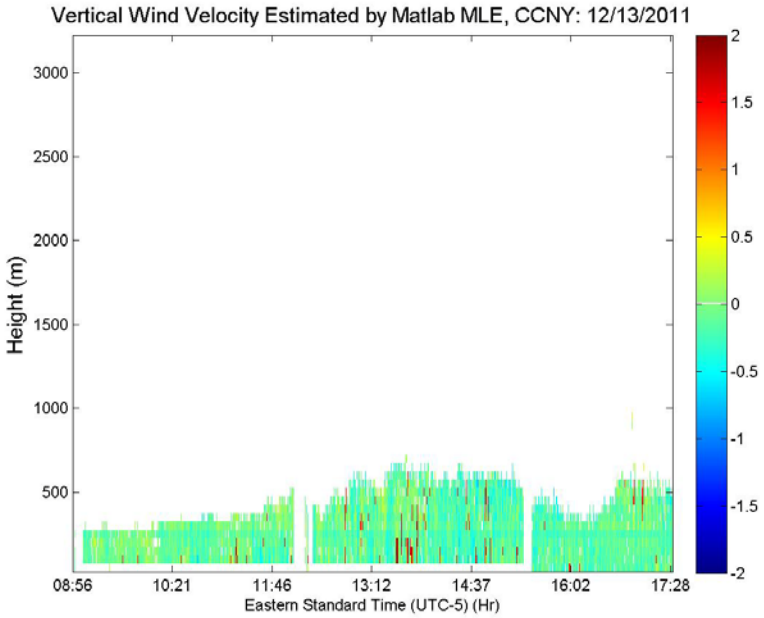


Figure 153, Coherent Doppler vertical velocity for 12/13/2011, 08:56 to 17:28. (colorbar in m/s)

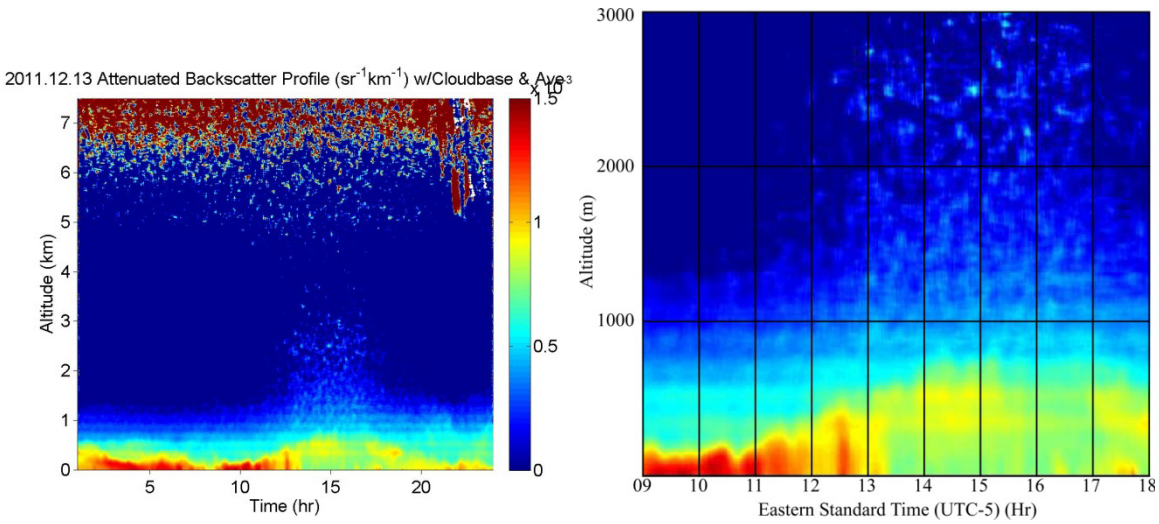


Figure 154, Ceilometer return signal for 12/13/2011. (Left) 24 hour period, (right) close up view of 09:00 to 18:00.

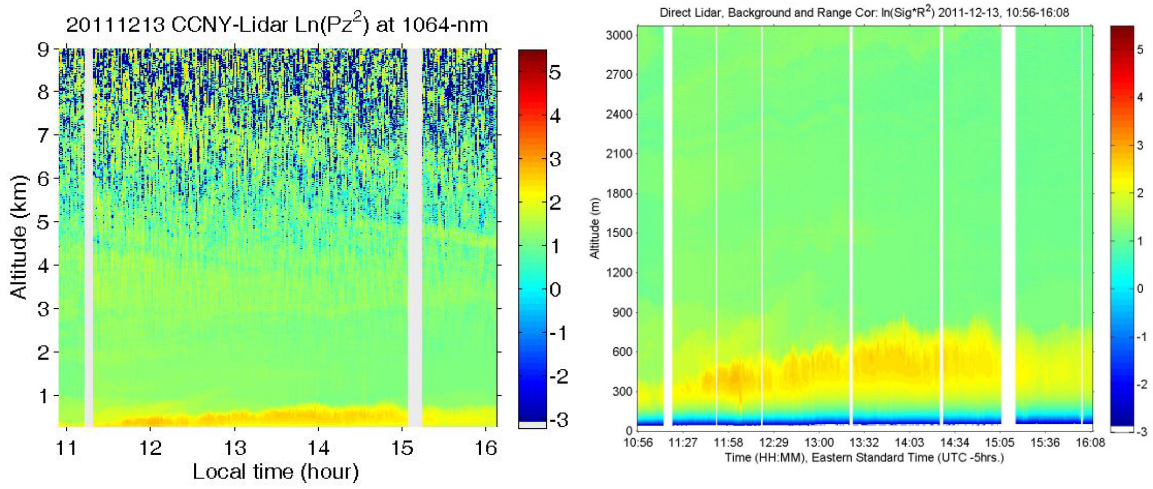


Figure 155, Full range and close up plot of CCNY Direct detection Lidar data for 12/13/2011.

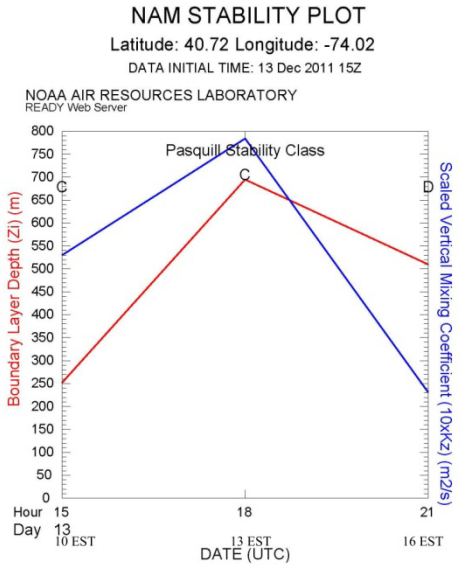


Figure 156, NAM Stability model plot for 12/13/2011, 13Z, 18Z, 21Z (10, 13, 16 EST).

Doppler Lidar and Planetary Boundary Layer

The structure of the Planetary Boundary Layer (PBL) has traditionally been measured with weather balloons, meteorological towers, and aircraft. Recently remote sensing techniques such as sodar, radar, and Lidar have been used to determine where the top of the PBL was located in a continuous manner. The temporally continuous determination of the PBL top is desirable to

support weather and air-quality predictions. Subjective visual estimates of the PBL top are operator dependent. Techniques and algorithms have been developed for the continuous processing of Lidar backscatter signals. These techniques are not without their weakness. Signal noise is one of these weaknesses. The definition of the top of the PBL is not itself clearly defined. One technique in use is based upon wavelets. [143-146]

Despite the issues with PBL top determination techniques a wavelet technique [143] was applied to the Doppler Lidar data as well as the CCNY direct detection Lidar data. This allows the comparison of the systems when their data sets are analyzed with the same techniques. The analysis is less of a determination of the PBL top level than it is a numerical validation of the ability of the Doppler wind Lidar system to make measurements that are comparable to an existing system at the same location. Comparative data presented earlier is less numerical than the top of PBL analysis.

The CCNY direct detection Lidar has a range resolution of 3.75 m while the Doppler wind Lidar has a range resolution of 48 m due to the spectral processing that is done on the data acquisition card. If the Doppler system were to directly record data the range resolution would be 0.75 m. Because the system is designed to be a wind sensing instrument this was not done. Despite the nearly 13 times difference in range resolution the data shows good correlations between the two systems. The Doppler wind Lidar system has less altitude resolution but greater time resolution than the CCNY direct detection Lidar. The Doppler system produced data with half minute temporal spacing while the direct detection system produced data with one minute spacing. In the plots of the PBL top the difference in altitude between the Doppler and direct detection Lidar systems has been corrected for. The Doppler system is located at local ground level while the direct detection Lidar is approximately 22 m above the Doppler Lidar.

12/01/2011 PBL Data

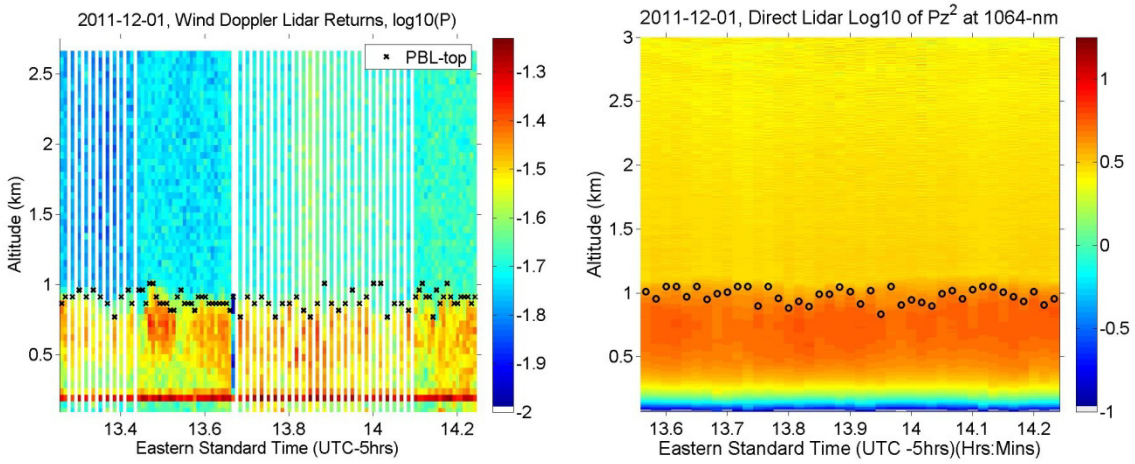


Figure 157, Return signal and PBL top for Doppler Lidar (left) and direct detection Lidar (right) for 12/01/11.

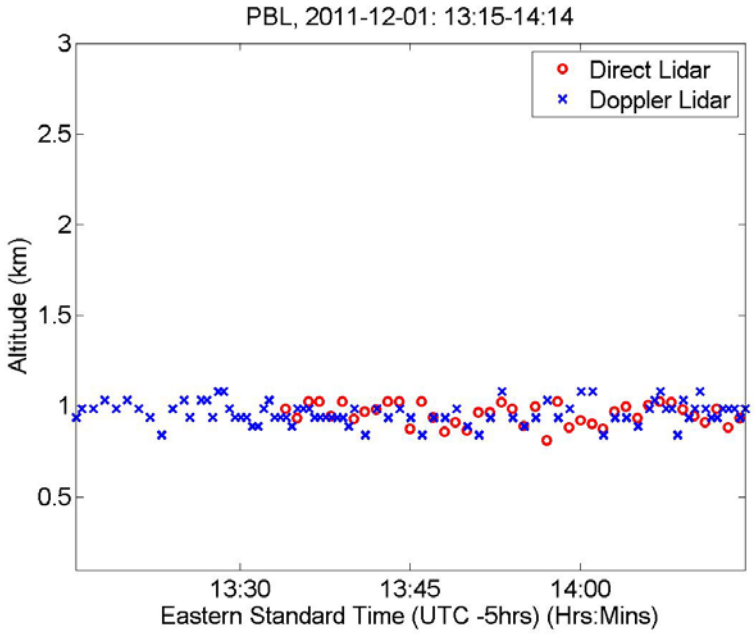


Figure 158, Planetary boundary layer top determined by a wavelet technique for data sets from Doppler Wind Lidar and CCNY direct detection Lidar on 12/01/11.

In Figure 157 the return signal and the PBL top are shown for the Doppler and direct detection Lidar systems. In Figure 158 only the PBL tops for the Doppler and direct detection Lidar systems are shown to allow for easier comparison. This figure shows a close tracking between the PBL tops for the two systems.

12/05/2011 PBL Data

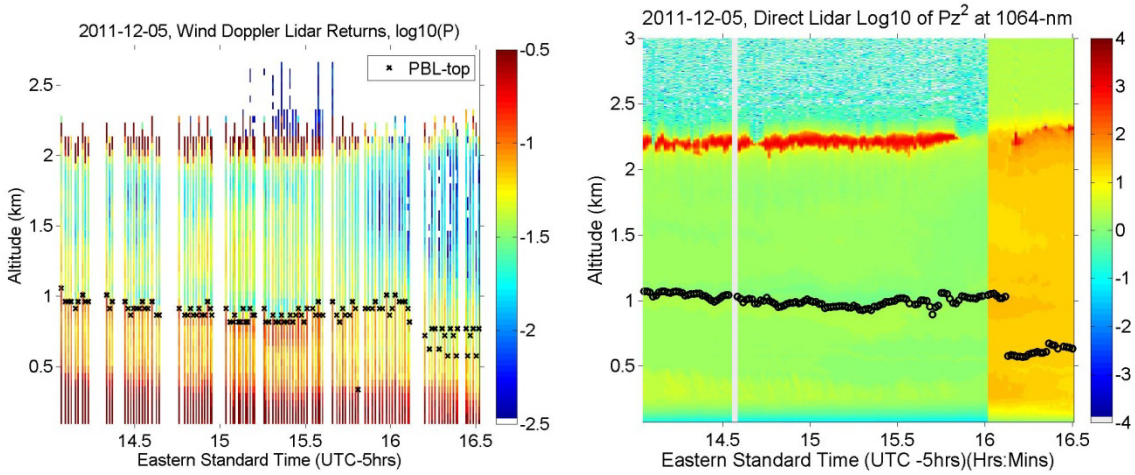


Figure 159, Return signal and PBL top for Doppler Lidar (left) and direct detection Lidar (right) for 12/05/11.

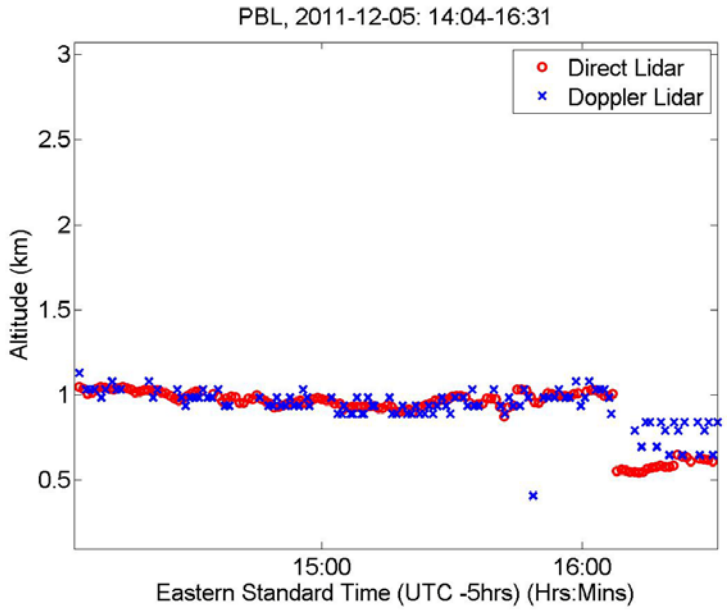


Figure 160, Planetary boundary layer top determined by wavelet technique for data sets from Doppler Wind Lidar and CCNY direct detection Lidar on 12/05/11.

In Figure 159 the return signal and the PBL top are shown for the Doppler and direct detection Lidar systems. In Figure 160 only the PBL tops for the Doppler and direct detection Lidar systems are shown to allow for easier comparison. This figure shows a close tracking between

the PBL tops for the two systems for most of the period. In the portion of the direct detection Lidar data prior to 16:00 a neutral density filter was in use over the detector due to the strong return signal from clouds near 2 km. Once the clouds dissipated and the signal power decreased the filter was removed. After 16:00 the plot shows a change in the color scheme as a result. Since the PBL top determination techniques operates on each column of data this should have little effect on the results. The Doppler and direct detection Lidar systems have very similar PBL tops until the 16:00.

12/09/2011 PBL Data

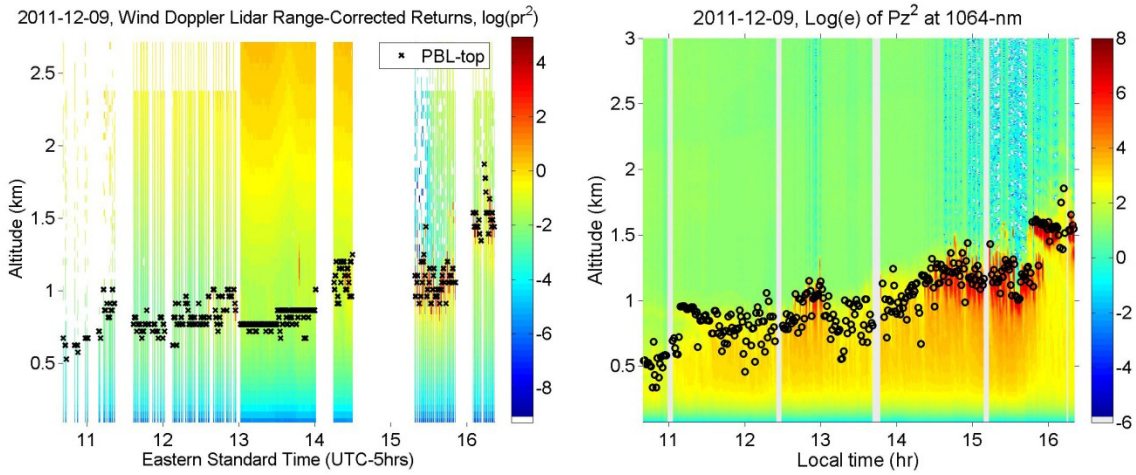


Figure 161, Return signal and PBL top for Doppler Lidar (left) and direct detection Lidar (right) for 12/09/11.

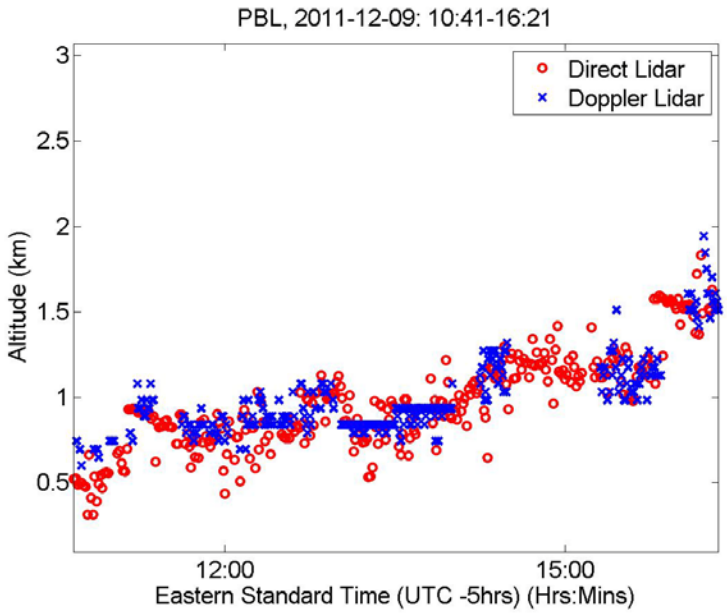


Figure 162, Planetary boundary layer top determined by wavelet technique for data sets from Doppler Wind Lidar and CCNY direct detection Lidar on 12/09/11.

In Figure 161 the return signal and the PBL top are shown for the Doppler and direct detection Lidar systems. In Figure 162 only the PBL tops for the Doppler and direct detection Lidar systems are shown to allow for easier comparison. The figure shows close tracking between the PBL top for both systems although the PBL seems to fluctuate more than in the previous data. This may be due to a dynamic atmosphere during data collection. The Pasquill atmospheric stability classes are the oldest and the most widely used techniques for the categorizing of atmospheric turbulence present. [147, 148] According to the NAM stability model, during this data collection the classes were as follows: 10:00 C (slightly unstable), 13:00 B (unstable), and 16:00 (neutral). This might contribute to the high frequency variability in the PBL top determination while a slow over change is also observed.

12/12/2011 PBL Data

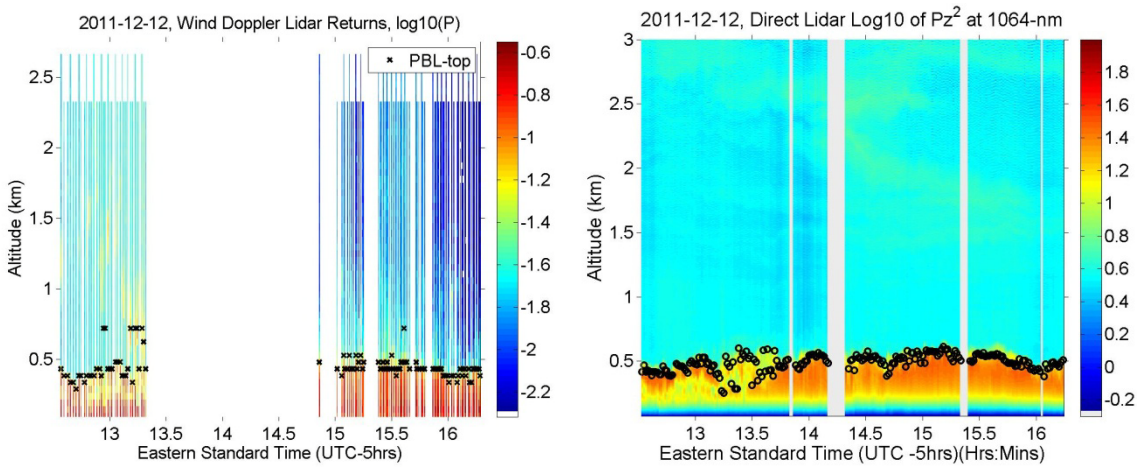


Figure 163, Return signal and PBL top for Doppler Lidar (left) and direct detection Lidar (right) for 12/12/11.

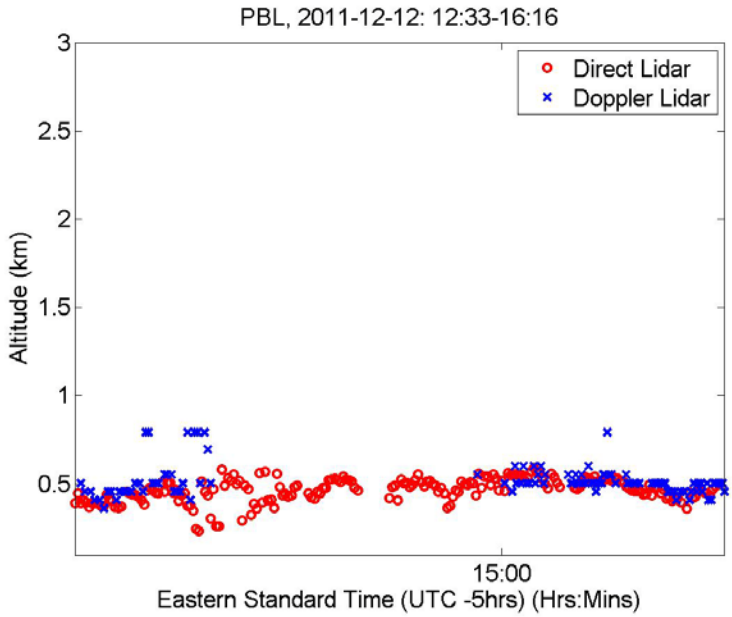


Figure 164, Planetary boundary layer top determined by wavelet technique for data sets from Doppler Wind Lidar and CCNY direct detection Lidar on 12/12/11.

In Figure 163 the return signal and the PBL top are shown for the Doppler and direct detection Lidar systems. In Figure 164 only the PBL tops for the Doppler and direct detection Lidar

systems are shown to allow for easier comparison. The two systems have close agreement in determining the top of the PBL with an occasional outlier.

12/13/2011 PBL Data

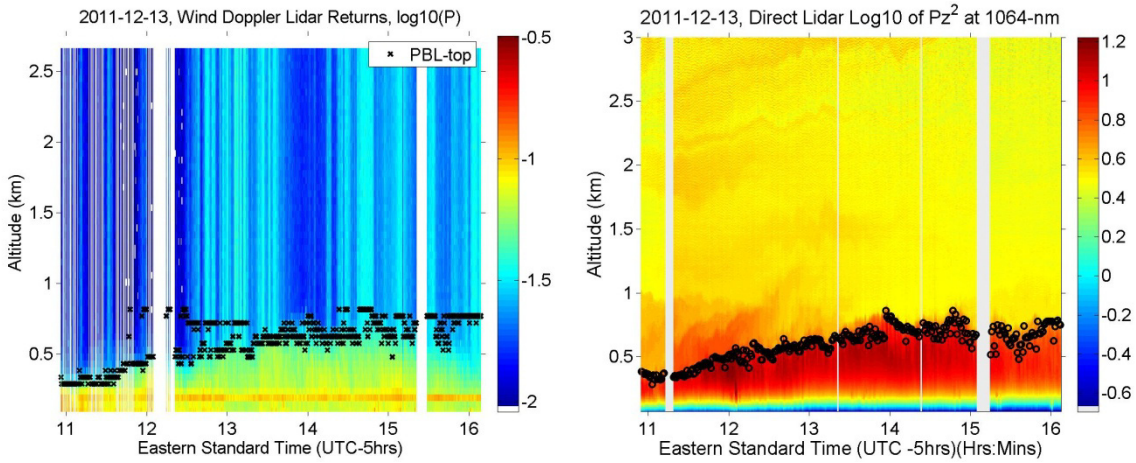


Figure 165, Return signal and PBL top for Doppler Lidar (left) and direct detection Lidar (right) for 12/13/11.

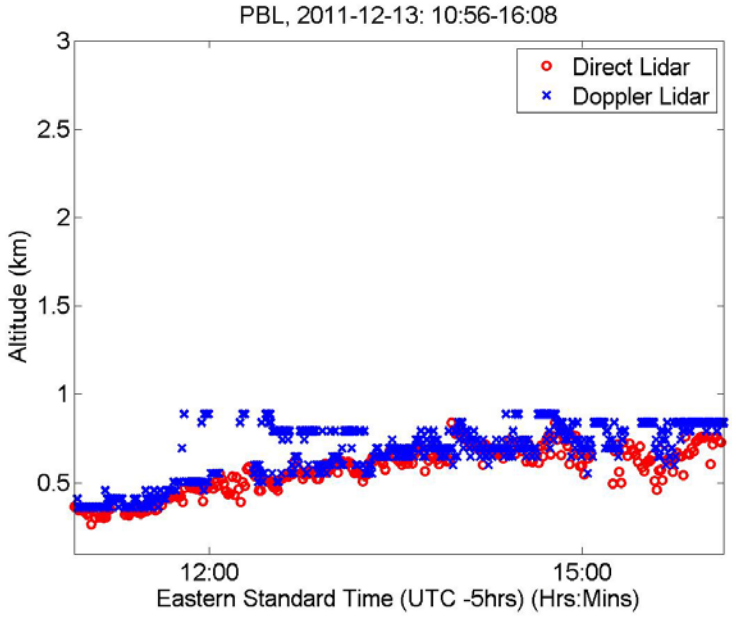


Figure 166, Planetary boundary layer top determined by wavelet technique for data sets from Doppler Wind Lidar and CCNY direct detection Lidar on 12/12/11.

In Figure 165 the return signal and the PBL top are shown for the Doppler and direct detection Lidar systems. In Figure 166 only the PBL tops for the Doppler and direct detection Lidar systems are shown to allow for easier comparison. The top of the PBL determined by both system's data seem to agree for the majority of the data although there does seem to be periods where the Doppler Lidar system is determining a higher PBL top than the direct detection Lidar system even though the two systems track the same long term changes.

Data Summary

The comparison of the Doppler Lidar to other instruments in the area shows a correlation between measurements. The Doppler Lidar system has weak return signals beyond the boundary layer or when the aerosol concentrations are low. The Doppler Lidar system is able to measure the border of the boundary layer when it is below 3000 m.

The Doppler Lidar system is able to determine the wind direction and speed with three beam measurements. Variations in the data between instruments in different locations are most likely due to variations in the wind fields at those locations. The fact that the system is located near the local maximum ground level and on the edge of a steep hill that transverses nearly 30 m in the span of two blocks may contribute to the variations seen in the measurements made at various locations.

When the data from the system is processed to determine the planetary boundary layer top altitudes and compared to the altitudes determined from data obtained from the CCNY direct detection Lidar which was processed in the same manner there is good agreement among the results. There are some variations between the results from the two systems but in general they correlate well.

CHAPTER 6
Assessment

The eye safe coherent Doppler wind Lidar system has been constructed from commodity components and the overall design has maintained simplicity when possible. The system has been able to perform wind velocity measurements and create wind components from these velocities. This has been accomplished by the determination of relatively small Doppler frequency shift as compared to the frequency of the carrier which is on the order of 200 THz.

Since the system is an aerosol based Lidar there is difficulty obtaining a return signal beyond the planetary boundary layer with a substantial vertical component when operating at eye safe emission limits. In a horizontal configuration a low concentration of aerosols should be less of a limiting factor on range as surface level aerosols are usually at higher concentrations.

Vertical atmospheric backscattered signals have been obtained from the current limit of the system which may be extended to nearly 7.5 km with software modifications. The current range can be affected by cloud layers that act to attenuate the outgoing and return signals.

The atmospheric turbulence also has an effect on the systems performance. High turbulence can cause many optical anomalies that can affect the system. Optical aberrations of the system's components are another source of signal degradation that has been addressed during the systems development.

The eye safe coherent Doppler wind Lidar system is a complex instrument where system component characteristics and constantly changing atmospheric conditions affect the performance. Optimization of one aspect of the system may have a deleterious effect on the overall performance of the system under certain atmospheric condition but not others. The

relationship may not be direct and may have multiple subsequent affects on secondary or tertiary components or performance. The rudimentary system specification of pulse repetition rate comes into play in the selection and configuration of many parts of the system. The pulse repetition rate will have an impact upon the system's eye safety, optical amplifier selection and performance, the system's maximum range, the minimal computer processing power and others. The system's configuration in order to maintain eye safety will affect the size of the lens, and the pulse power. The size of the lens will affect the focal length necessary for efficient coupling into and out of the fiber optic cable as well as the spatial coherence of the atmospheric signal. The focal length will affect the overall size, portability, and complexity of the system which in turn affects the overall robustness and immunity to vibrations. The permutations of the system parameters are large in number and in some aspects the changing operating environment makes optimization more of a compromise between desired performance and operating conditions.

The eye safe coherent Doppler wind Lidar system has been created from commodity components while taking into account the operating conditions and restrictions. It performance has been compared to other instruments and models and has shown good correlation.

CHAPTER 7
Future Work

The following are a number of possible improvements and suggestions on areas of further research that may help to advance the project based upon the experiences and knowledge accumulated during this work.

Automated Scanning and Positioning

With the current system, a motorized platform that could rotate the mirror to predefined positions under computer control could be installed without very much difficulty to allow for autonomous continuous data collection. Mounting the mirror at 45° on a stepper motor rotation stage would be one way of achieving this. This would not allow for adjustment of the pitch of the mirror but that too could be implemented with a little more effort and another rotation stage. Automated scanning would allow the operator to make other observations and reduce the variability in the periodicity of the measurements while also increasing the accuracy of the beam pointing.

Another possible solution for continuous autonomous rotation of the Lidar beam may be to mount an optical wedge in the center of a stepper motor rotation stage that has a clear aperture through its center. The mirror would be fixed so the beam would hit the bottom of the wedge. The beam would be tilted at an angle by the wedge and the wedge would then be rotated by the rotation stage. This would allow for beam pointing positions that are not possible with the current configuration, as the mirror can only be rotated so far before its profile becomes narrower than the outgoing beam. The wedge may introduce losses and distortions that may affect operations, so caution must be taken. Another possibility similar to a wedge is the use of a

holographic optical element (HOE). These can be flat holographic plates that act like optical elements. A HOE could be used to tilt the beam at an angle as the hologram is rotated.

The distance between the fiber tip and the lens could be adjusted with piezoelectric positioners. This could help to keep the beam at the proper position thus avoiding variations in focus during temperature variations which cause the optical rail to expand and contract.

Signal Processing

More sophisticated signal processing techniques could be explored. Such techniques as autocorrelation, short time fast Fourier transforms, wavelets and others designed to estimate the frequency of a sinusoid in noise is an area of research in communication theory that may have applications to Lidar. Offloading signal processing duties from the computer to the FPGA chip on the data acquisition card would help reduce the time necessary to make a measurement. Use of various data collection "windows" on the data may help to make the estimate of frequency more accurate.

Pulse Shaping and RF Driving Circuitry

The characteristics of the optical pulse driving the optical EDFA amplifier could be manipulated through variation of the signal driving the AOMs. Since the optical amplifier tends to amplify the leading edge of its input more than the tail end of the input, non-symmetric shaping of the optical input pulse, accomplished through manipulation of the RF signal driving the AOMs, could create a desired output profile. A circuit could be inserted before the RF amplifier to alter the profile of the signal driving the AOMs. A multiplier circuit incorporating the Analog Devices ANL5391 may be able to meet the needs for this task.

The RF driving circuitry could be integrated. The oscillator, directional coupler, RF switch, pulse shaping circuit, bandpass filters, RF amplifier, and RF splitter could be integrated into a single unit. The bandpass filters currently on the input to the RF amplifier could be moved to the output side if new filters with higher power capacity were located. This could help reduce noise introduced by the RF amplifier. Currently the system filters the input to the RF amplifier.

Semiconductor Laser

In the past semiconductor lasers have been too noisy for many applications in Doppler Lidar. Recently a company known as Redfern Integrated Optics Inc. (RIO Inc.) has manufactured a semiconductor laser with low noise and low cost, approximately $4,000. This company has also received a grant from the government to develop a semiconductor laser for a Lidar system that will be deployed on a satellite. [149] Chinese researchers are also working on 100 mW all fiber single frequency polarized seed lasers for coherent Doppler Lidar that have very narrow line width (4.1 kHz) and high extinction ratios (40.5 dB). [150]

Additional Optical Amplifier and Optical Switch

A second optical amplifier could be installed in between these optical circulator and variable optical coupler to amplify the return signal from the atmosphere. This could increase the range of the system. Calculations of the increase in noise due to this second optical amplifier would need to be performed in order to make certain it helped with the performance.

The back reflection from the fiber tip of the outgoing pulse would also be amplified by a second optical amplifier. This would create a large pulse to the detector every time a pulse was sent to

the atmosphere. This amplified back reflection from the tip may put the detector into a state where its output could be compromised until it recovered.

To reduce the effect of this back reflection from the fiber tip, the detector some means of attenuating or switching the signal during the outgoing pulse period would need to be implemented. It has been difficult to locate an optical device that is sufficiently fast, has low loss, and low cost but development of these photonic devices should be kept under observation.

Detector

The output of the detector could be amplified with a low noise amplifier. An amplifier with sufficient bandwidth, a low noise figure, and limits on its voltage output so as to not damage the analog to digital converter would be needed. An analysis of the effect on the signal to noise ratio would also be necessary. A customized balanced detector design might also improve the system performance. A balanced detector with a custom designed trans-impedance amplifier that amplified the frequency spectrum of interest and attenuated other frequencies might have better signal to noise performance then an off-the-shelf balanced detector. It is often advantageous to place gain circuitry close as possible to the sensor. A variable gain circuit might also be included to vary the gain. The return signals from just after the pulse leaves the fiber tip do not need as much amplification as a signal from a greater distance. This may also help with the back reflection from the outgoing pulse from the fiber optic tip of the circulator.

Packaging

The current state of development of the Lidar system is movable but not very portable as it is still a development system. The advancement of the design towards a unit that is small enough to be

carried by an individual is possible without major alterations. The suggestions concerning RF component integration mentioned above and the improvements to the optical system would be significant steps towards port ability. By offloading duties from the computer to the data acquisition card, a lower cost and lower power computer could be used. Perhaps an industrial style embedded computer system might be able to run the Lidar system and log the data.

CHAPTER 8
Summary

The main findings of this dissertation are:

Wind measurement instruments such as anemometers, weather balloons, sodar wind profilers, and radar wind profilers have physical, costs and operational limitations that can be overcome by an eye safe Doppler wind Lidar system.

A coherent Doppler wind Lidar system was designed, components purchased, tested, and implemented in a system for the detection of wind while remaining eye safe. This should allow for unsupervised operation.

The Doppler wind Lidar system was operated, and data was collected and processed to produce wind velocities as well as wind speeds and directions. These were compared to other measurements for verification of the system's operation and exhibited good agreement.

CHAPTER 9
Conclusions

An eye-safe coherent Doppler wind Lidar has been constructed utilizing many components readily available from telecommunication industry manufacturers. The use of commodity devices has avoided research and development of custom designed components, which has allowed for quicker development of the system while maintaining lower overall cost.

The eye-safe system can be operated in an unmanned mode requiring only minimal operator system preparations and attention during fixed line of sight measurement collection.

Three different sequenced line of sight measurements have been successfully combined to calculate the horizontal and vertical wind vectors. From these, wind direction and wind speed have been obtained.

Wind direction and wind speed calculated from the data collected by the system has been compared to other wind instruments and validates the system's performance to a good degree.

CHAPTER 10
Appendix

Computer Programs and Functions

Part 1: Matlab program: takelidar15.m

Part 2: Matlab function: wind_calculate_function2

Part 3: Matlab program: ProcLidarBeamVel_06.m

Part 4: Matlab function: function windbarb_DS01

Part 5: Matlab program: ProcLidarBeamVel_04_Plot_02.m

Part 6: Matlab program: test30s06.m

Part 1: Matlab program: takelidar15.m

```
% takelidar15.m
%
% clear all;
% David Santoro
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% PURPOSE:
% This porgram will help a user take Doppler Lidar data using a modified
% SNAP program with a FPGA code that takes 8192 data points, uses
% 128 concecutive points to compute 64 FFT spectrums, add the newest
% spectrums to a running sum of spectrums until 10,000 have been colleted.
% After 10000 summations these are appended to a file until 1 minutes has
% been collected.  Then the user will be instructed to readjsut the laser
% beams direction and to rerpeat the data collection.
%
% The wind vel is calculated from the measured laser pulse summations
% at a position.  The last 3 positions are used to calculate the u, v, and
% w vectors resulting from a 'north', 'east' and 'verticle' measurement.
%
% The horizontal (north, east) and verticale wind speeds are printed.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The text file of accumulated spectrums will have the naming format
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FILE NAME NOMENCLATURE
% The default text folder name is encoded with the START date and time
% yyyymmddTHHMMSS.txt
% 'yyyymmddTHHMMSS' (ISO 8601) standard- NOTE THE 'T' between date and time
% yyy= the year (i.e. 2011)
% mm  = the month (two digits, i.e January = 01)
% dd  = then day of the month (two digit day of the month, i.e. 07)
% HH  = the hour (24 hour clock, i.e. 3:45PM is 1545, NO daylight savings)
% MM  = the minutes of the hour (00 to 59)
% SS  = the seconds of the minute (00 to 59)
% Matlab has a Numeric Identifiers for Predefined Format of 30
% for the ISO 8601 format standard
% If the time right now was October 5th 2011, 10:51:15 PM
% ds = datestr(now,30) --> ds = 20111005T225115

time_format      = 30;% Time Format 30 = 'yyyymmddTHHMMSS' (ISO 8601)
datetimeStart    = datestr(now,time_format);% Get program start date & time
file_finish_time = 0; % Variable to hold time when SNAP finishes with file
file_wait_time   = 3; % Seconds to wait before testing to see
                      % if SNAP has finished writing to a file

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Set Up Debugging Variables
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
dBUG            = 1;% DEBUGGING SWITCH
                    % =1 (true) Do debugging,
                    % =0 (false) No debuging
                    %
dbugPC          = 0;% LOCATION OF COMPUTER
                    % Use to precnfigure filenames, directories, etc
                    % =0 for WinXP 64bit Doppler PC (CCNY),
                    % =1 for home (santoro),
                    % =2 for work (santoro),
                    % =3 for work T647 (santoro),
                    %
dbugPLOTS       = 0;% CREATE PLOTS FOR DEBUGGING
                    % =0 No debuging plots,
                    % =1 Plot Debug Plots
                    %
dBUGvelMethode = 3;% METHOD OF DETERMINING WIND VELOCITY
                    % Freq Peak Wind Vel Method
                    % =1 MLE, Maximum Likily Methode
                    % =2,
                    % =3, Frequency Estimators- dsp Tips
                    %     Fast, Accurate Frequency Estimators- dsp Tips &
                    %     Tricks Jacobsen, and Kootsookos IEEE Signal
                    %     PRocessing Magazine May 2007, Pg 123, Equation 2
                    %
dBUGfakeSNAP = 3;   % SOURCE OF DATA
                    % =0 Do NOT create Fake SNAP data, REAL DATA on HD with
                    %    Date/Time stamped file name
                    % -------------------------------------------------
                    % =1 Create fake SANP data for debugging (ramp func)
                    % -------------------------------------------------
                    % =2 Create fake background data from a data file of ONE
                    %    spectra. And generate sample data by modifying it.
                    %    Use
                    %    092311DelayT0_AB_2.24uS_my_output_file_3_60s.txt
                    %    as fake data Has delay of 2.24us, 9 pretrigger
                    %    spectra and the 1st two spectra seem to be
                    %    different from the next seven spectra.  There
                    %    seems to be 3 spectra effected by the outgoing
                    %    pulse tip reflection
                    % -------------------------------------------------
                    % =3 Do NOT create Fake SNAP data, REAL DATA on HD with
                    %    "my_30_Seconds_file.txt" file name
                    %
dbugNoiseSub = 0;   % METHOD OF REMOVEING BACKGROUND (NOISE)
                    % =0 10Log(S+N)-10Log(N), CHECK?????
                    % =1 ((S+N)-N)/N,
                    % =2 10Log((S+N)-N) - 10Log(N)

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Set default SNAP OUTPUT File Type
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
file_nameXXs = 'my_30_Seconds_file.txt';
SNAPfiletype = 1;  % FILE NOMENCLATURE EXPECTED FROM SNAP PROGRAM
                    % =0, Text File, yyyymmddTHHMMSS.txt
                    % =1, Text File, my_30_Seconds_file.txt
switch SNAPfiletype
  case 0
```

```
      file_name = [datetimeStart,'.txt'];% File Name with Date/Time Stamp
  case 1  % If SNAP outputs non date/time stamped file name
      file_name = file_nameXXs;
  case 3  % Other file format
end

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Set default directories and file names based on location of PC
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
switch dbugPC   % =0 for van ,
                % =1 for home,
                % =2 for work,
                % =3 Work T647
                % =4 for _______
    case 0 % =0 for van
      file_path = ...
       'D:\TEMP_SANTORO\20111213_Data_SNAPmod\20111213_Data_SNAPmod_Run2\'
       %'D:\TEMP_SANTORO\20111201_Data_SNAPmod\20111201_Data_SNAPmod_Test\'
       %'D:\TEMP_SANTORO\20111128_Data_SNAPmod\';
       %'D:\TEMP_SANTORO\20111125_Data focused at 2797m\20111125_Data SNAPmod
focused at 2797m\20111125_Data Vert Keopsys 170ma 1500ma DDG 1.1227us';
       %'D:\TEMP_SANTORO\PhD Matlab\Lidar_Text_Data_Matlab\20111117_Data\';

      file_name_fakeBK = ...
       'BackGround_Noise_092311DelayT0_AB_2.24uS_my_output_file_3_60s.txt';
      %file_name = '20111004T131022.txt';
    case 1 % =1 for home
      file_path = ...
       'D:\David\PhD\PhD Matlab\Lidar_Text_Data_Matlab\20111125_Data\';
      file_name_fakeBK = ...
       'BackGround_Noise_092311DelayT0_AB_2.24uS_my_output_file_3_60s.txt';
      %file_name = '20111004T131022.txt';
    case 2 % =2 for work
      file_path = 'C:\David\CCNY PhD Matlab\Lidar_Text_Data_Matlab\';
      file_name_fakeBK = ...
       'BackGround_Noise_092311DelayT0_AB_2.24uS_my_output_file_3_60s.txt';
      %file_name = '20111004T131022.txt';
    case 3 % =3 for work T647
      file_path = 'C:\Temp\santoro\PhD Matlab\Lidar_Text_Data_Matlab\';
      file_name_fakeBK = ...
       'BackGround_Noise_092311DelayT0_AB_2.24uS_my_output_file_3_60s.txt';
      %file_name = '20111004T131022.txt';
    case 4 % =4 for _____
      file_path = 'E:\PhD\PhD Matlab\Lidar_Text_Data_Matlab\';
      file_name_fakeBK = ...
       'BackGround_Noise_092311DelayT0_AB_2.24uS_my_output_file_3_60s.txt';
      %file_name = '20111004T131022.txt';
end

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Default Length of Time to Take Measurememts
Time_Length   = 30;     % Number of seconds to make a measurment
Time_LengthBK = 30;     % Number of  seconds to take an independent, before
                        % measuremnts start, background (noise) measurement

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Default File Name Suffixes to be Used
```

```
file_name_suffix_Background    = '_bk';  % Background (Noise) Spectrums
file_name_suffix_BackgroundAVG = '_avg'; % Average Background (Noise) Spec
file_name_suffix_MetaData      = '_meta';% Meta Data
file_name_suffix_Temp          = '_temp';% temporary Data
file_name_suffix_WindVlVcDr    = '_wind';% Wind Velocity,Vectors,Directions
file_name_suffix_BeamX         = '_Beam';% BeamX

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Constants and Default Values %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Physical Constants
c    = 299792458;       % Speed of Light (m/s) in a vacume
nAir = 1.000293;        % Refractive Index of air
                        % Understanding the properties of matter
                        % By Michael De Podesta page 131
h   = 6.62606957e-34;   % Planck's constant (Js)

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Digital Delay Generator (Standford DG-535)
DDGjitM    = 25e-9;     % Output jitter speced by manufacturer
DDG_TplusA = 1.1228e-6; % T+delay for A time (s)
DDG_AplusB = 300e-9;    % A+delay for B time (s)
DDG_PulseT = DDG_AplusB - DDG_TplusA;   % DDG pulse period (s)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ADC (Innovative Intergrations X5-400M verD) Constants and Defaults
FsamplingM= 400e6;  % ADC sampling Rate (spec manufacturer)
ADCbitsM  = 14;     % ADC number of bits (spec manufacturer)
ADCenobM  = 10.1;   % Effective Number of Bits for our ADC
                    % (spec manufacturer) with full scale sinosoidal input
                    % 70MHz, 850 mVRMS filtered sine sampled at 400 MSPS,
                    % DC-200 MHz, DC-coupled specified by manufacturer
ADCsnrM   = 73.6;   % SNR (dB) (spec manufacturer) for
                    % 70MHz, 850 mVRMS filtered sine sampled at 400 MSPS,
                    % DC-200 MHz, DC-coupled specified by manufacturer
ADCnoiseM = 350e-6; % Typical noise for grounded input (V), 1 standard
                    % deviation by manufacturer (spec manufacturer)
ADCoffseterrorM = 700e-6;   % Factory calibration Offset Error(V), averaged
                            % 64K samples after warmup (spec manufacturer)
ADCgainerrorM   = 0.0002;   % 0.02%, Factory calibration Gain Error
                            % after warmup (spec manufacturer)
ADVinMaxM       = +1;       % Input Voltage Max (V) (spec manufacturer)
ADVinMinM       = -1;       % Input Voltage Min (V) (spec manufacturer)
ADCjitAdditiveM = 80e-12;   % Additive jitter of ADC using external clock
                            % (s) (spec manufacturer)
ADCjitInterM    = 340e-15;  % Total jitter of ADC using internal clock
                            % (s) (spec manufacturer)

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Query User for Length of Time to Measure Signals and Background (Noise)
prompt1 = {['How Many Seconds Will The One Time Background (Noise)'...
            ' Measurement Be?'],...
            'How Many Seconds Will Each Beam Measurent Be?'};
dlg_title1         = 'Measurements Durration';
num_lines1         = 1;
def1               = { num2str( Time_LengthBK),...
                                num2str(Time_Length ) } ;
options.Resize     = 'on'; options.WindowStyle= 'normal';
```

```
options.Interpreter= 'tex';
answer1            = inputdlg(prompt1,dlg_title1,num_lines1,def1,options);
ATime_LengthBKZ    = str2num(answer1[15]);
Time_Length        = str2num(answer1{2});

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  System Parameters
SampPerPulse = 8192;     % Number of samples collected per laser pulse
UsepreTR     = 4;        % Use the nth pre trigger spectrum to find the
                         % average backgound noise spectrum of a
                         % particular pulse
Navg_spec = Time_Length*2;  % Default Number of spectrums from background
                            % measurements to use in order to find
                            % BACKGROUND (noise) is ALL
Naccsize = 4096;         % Number of points in FPGA output for one
                         % FFT accumulation accumulated measurement
NpreTRspec = 6;          % Number of pretrigger spectrums
                         % (FFT spectrums befor laser pulse)
Ntip_spec = 2;           % Number of spectrums effected by
                         % pulse reflection from fiber tip
NpreTRtip_spec = NpreTRspec + Ntip_spec;    % Number of spectrums effected
                                            % by reflections or pre trigger
PRF       = 20000;                 % Pulses Repatition Frequency (Hz)
                                   % (laser pulses per second)
Tpulse    = 1/PRF;                 % Pulse to pulse period (sec)
Npulseacc = 10000;                 % Number of laser pulses per
                                   % accumulation output from ADC/FPGA
Tpulseacc = Npulseacc*Tpulse;      % Period (S) for Npulseacc pulses
                                   % to be accumuled (Aquisition period)
Naccs     = Time_Length/Tpulseacc; % Number of FFT acumulation outputs
                                   % by FPGA for the measurement period
Nptsfile  = Naccsize*Naccs;        % Number of points SNAP writes to
                                   % a data file for the measurement period
NptsfileBK= Naccsize*Time_LengthBK/Tpulseacc; % Number of points SNAP
                                              % write to Background data
                                              % file
No_Averages = Naccs;     % Number of pulse accumulations to average
                         % to make a wind estimate

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Query User Number of Measurements To Average for BeamX
prompt1b = {[num2str(Naccs),' Accumulation Measurements, (', ...
             num2str(Npulseacc),' pulses each), Will '...
             'Be Made Durring The ',num2str(Time_Length),...
             ' Seconds Each Beam Position Is Measured. '...
             'How many will be averaged to determine the '...
             'line of site wind velocity?']...
             };
dlg_title1b = 'Measurement Averaging';
num_lines1b = 1;
def1b = {num2str(No_Averages)};
options.Resize='on';
options.WindowStyle='normal';
options.Interpreter='tex';
answer1b = inputdlg(prompt1b,dlg_title1b,num_lines1b,def1b,options);
No_Averages = str2num(answer1b[15]);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FFT related Constants and Defaults
Nfft = 128;              % Number of points input to the FPGA's FFT routine
N_pt_spec = Nfft/2;      % # of Points in a single spectrum for a single
                         % range gate output by the FPGA (i.e. Nfft/2)
N_spec_acc = Naccsize/N_pt_spec;  % Number of spectrum per accumulation
FFTbw = FsamplingM/Nfft;% Freq bandwidth of each FFT output/bin from FPGA
                         % CHECK THIS-does xilinx FFT library put the DC
                         % component in the 1st Bin???  Matlab may be
                         % different  GET Web PAGE or PDF REFERENCE

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Range Gate Size of FFT
% (Time to Take Samples for one FFT of data)*SpeedOfLight/2
RGS    = (Nfft/FsamplingM) * c/2;     % Range Gate Size in meters
Rrange = (RGS/2:RGS:RGS*N_spec_acc)'; % Radial ranges (vector)
                                      % from origin (telescope)

            % CHECK- SHOULD THIS BE POST PULES ONLY?????

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Laser Charateristics, AOM Constants and Defaults
%
% SEED LASER LOW POWER AND HIGH POWER OUTPUTS
% NP Photonics Part Number = RFLSA-500-1-1545.2-PM-S_NSL
% Oct 2008   Serial number = J10-000789-379
LambdaLO = 1545.2e-9;       % Wavelength of Seed Laser
fseed    = c/LambdaLO;      % Frequency of Laser
KW       = 2*pi/LambdaLO;   % Wave number, (angular wavenumber,
                            % circular wavenumber)
SeedLW   = 5200;            % Lorentzian Full Width Half Max (Hertz)
                            % (spec measured by manufacturer).  Full width
                            % of 104.0 kHz when 20 dB down from peak of
                            % signal.  (Se Manual- Final Inspection Report
                            % for 000780 RFLSA-500-1-1545.2-PM-S-NSI)
SeedStability = 50e6;       % Seed Laser stability over 1 hour
SeedRINpk     = -113.1;     % (dB/Hz),Relative Intensity Noise (RIN)
                            % Dominant peak at relaxation oscillation freg
                            % of laser at 2.3 MHz.  Shot noise limited
                            % greater then 50 MHz.
SeedPER       = 17;         % Seed laser Polarization Extintion Ration
                            % Minimum (dB)
SeedS_ASEi    = 35;         % Seed laser Signal to ASE Noise Ratio
                            % Intergrated Minimum (dB)
SeedS_ASE     = 55;         % Seed laser Signal to ASE Noise Ratio
                            % 50 pm, bandwidth, Minimum (dB)
SeedSMSR      = 50;         % Seed laser Side Mode Supression Ratio
                            % Minimum (dB)
SeedRINsnLP84 = -160;       % (dB/Hz), RIN of seed low power output
                            % at ~ 84Mhz, shot noise limited
SeedRINsnLP54 = -157;       % (dB/Hz), RIN of seed low power output, 54MHz
                            % shot noise limited
SeedRINsnLP114= -163;       % (dB/Hz), RIN of seed low power output, 114MHz
                            % shot noise limited
SeedRINsnHP84 = -155;       % (dB/Hz), RIN of seed high power output
                            % at ~ 84Mhz, shot noise limited
SeedRINsnHP54 = -153;       % (dB/Hz), RIN of seed high power output
```

```
                              % at ~ 84Mhz, shot noise limited
SeedRINsnHP114= -157;         % (dB/Hz), RIN of seed high power output
                              % at ~ 84Mhz, shot noise limited

thao     = 200e-9;            % Laser Pulse Width from AOMS FWHM (s)
freqAOM1 = 42e6;              % Acousto Optical Modulator (AOM) 1 freq shift
freqAOM2 = 42e6;              % Acousto Optical Modulator (AOM) 2 freq shift
fsAOM    =freqAOM1+freqAOM2;% Total freq shift to pulse by both AOMs

E        = 13e-6;             % Energy per Laser Pulse(J) Keopsys Pulse Amp

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Doppler Frequencies of Interest for Wind Measurments
FreqInterestStartUser = 46.875e6;   % Start Freq of interest
                                    % for Doppler Measurements
FreqInterestStopUser  = 125.0e6;    % Stop Freq of interest
                                    % for Doppler Measurements

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FFT spectral output Constants and Defaults
NpreTR = N_pt_spec*NpreTRspec;  % Number of points in all spectrums before
                                % laser pulse (i.e. pre trigger spectrums)

IpreTRstart = (UsepreTR-1)*N_pt_spec+1;    % Index of start of the pre
                                            % Trigger spectrum chosen
                                            % to use as background
                                            % (noise)
IpreTRend   = (UsepreTR*N_pt_spec);         % Index of end of the pre
                                            % Trigger spectrum chosen
                                            % to use as background
                                            % (noise)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Spectral Bin Indexies and Counts
% Index of START freq of spectrum range of interest
   % IpreTR_Spec_Start: Index of the frequency bin with the lowest
   % frequency which we are interested in, within a spectrum of NFFT/2
   % points.
IpreTR_Spec_Start = floor(FreqInterestStartUser/FFTbw);

% Index of Stop freq of spectrum range of interest
   % IpreTR_Spec_Stop: Index of the frequency bin within a spectrum of
   % NFFT/2 points with the highest frequency which we are interested in.
IpreTR_Spec_Stop = ceil(FreqInterestStopUser/FFTbw);

% Number of spectral bins in the freq range of interest
Nspec_bins = (IpreTR_Spec_Stop-IpreTR_Spec_Start)+1;

% Number of spectrums after pre trigger and fiber tip reflection spectrums
N_spec_acc_post_pulse = N_spec_acc - NpreTRspec - Ntip_spec;

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Constants for SNR simulation of Lidar signal based on equation 3
% of Mitsubishi paper titled compact all-fiber pulsed coherent doppler
%
% Compact all-fiber pulsed coherent Doppler lidar system for wind sensing
% S. Kameyama, T. Ando, K. Asaka, Y. Hirano, and S. Wadaka
% 10 April 2007,  Vol. 46, No. 11, APPLIED OPTICS 1953
```

```
R     = 1 ;      % reflectivity of a hard target
B     = 100e6;   % for wide band SNR use 100MHz,
                 % for narrow band SNR use 2MHz Bandwidth .
Cn_sq = 2e-14;   % Refractive Index Structure Constant;
D     = 0.1;     % Effective aperture Diameter (m)
beta  = 8.3e-7; %R/(fseed*thao);%atmospheric backscatter coefficiency(/m/sr)

% The following parameter needs to be checked
K     =  0.95;   % One way atmospheric transmittance (/km)
F     = 1500;    % Focal Range of Optical Antena (m)
Eta_F = 0.9;     %0.355 ; % Far field system efficiency (-4.5 dB)
Ac    = 0.76;    % Correction Factor.
Eta_I = 0.9;     %0.676; % Factor for insertion loss of optical components
                 % except for telescop (-1.7 dB)
Eta_RE = 0.977; % Telescope absorption and reflection loss (-0.1 dB)
Eta_A  = 0.91;  % Telescope reflection loss (-0.4 dB)
Eta_PP = 0.912;
Eta_Q  = 0.9;    % 0.794;

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  User/Equipment Constants, Defaults
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Users, Predefined Choices
DL_user1 = 'moshary';
DL_user2 = 'arend';
DL_user3 = 'abdelazim';
DL_user4 = 'santoro';
DL_user5 = 'jalloh';
DL_user6 = 'guest1';
DL_user7 = 'guest2';
DL_users = {DL_user1; DL_user2; DL_user3; DL_user4; ...
                      DL_user5; DL_user6; DL_user7};

% Locations, Predefined Choices
DL_location1 = 'CCNY';
DL_location2 = 'Liberty Science Center';
DL_location3 = 'Brookhaven National Lab';
DL_location4 = 'other';
DL_locationS = {DL_location1; DL_location2; DL_location3; DL_location4};

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Truck Cardinal Directions (North, South, East, West) and Tilt from Level
Truck_tiltZE = 0;% Zenithe angle = 0 if laser beam is verticle
                 % (and Truck optical table is level)
                 % If the edge of the optical table closest to the
                 % fron of the truck is lower than the egde of
                 % the optical table nearest the rear of the truck
                 % ZE will have a POSITIVE value, and a NEGATIVE value
                 % if the rear edge is lower than the front edge.
                 % If truck has right side of optical table lower than the
                 % left side of the optical table then __________
                 % Have to come up with a framework - don;t worry now

Truck_cardianlAZ = 299.33;   % Azmith angle = 0 if the direction of
                             % Truck's optical table taken along the
                             % long edge (length) is North.
                       % North is 0 degrees, East is 90 degrees,
```

```
                        % South is 180 degress and West is 270 degrees

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Software Versions Being Used - for Meta Data
%
% Matlab Information about curret .m file running
MlabFname      = mfilename;         % Matlab Filename running without '.m'
MlabFpath_name = mfilename('fullpath'); % Path and Filename of running .m

% FPGA Firmware version being used
X5fpgaLogicPath_Name = ['D:\TEMP_SAMEH\Simulink_Works\Accumulated_FFTs\'...
                        'FFT_10K_Accum_Reset_7_hwcosim_module.mdl'];
% SNAP Version being Used
SNAPverPath_Name     = ['C:\Innovative\X5-400M\Examples\'...
                            'Snap_07282011\FPGA_SNAP\Debug_x64'];

% X5-400M SNAP Default Settings
X5frameSize     = 8192;% Frame Size in decimal
X5frameSizeBK   = 8192;% Frame Size in decimal (BACKGROUND MEASUREMENT)
X5packetSize    = 8192;% Packet Size in decimal
X5packetSizeBK  = 8192;% Packet Size in decimal (BACKGROUND MEASUREMENT)
X5ch            = 0;   % ADC channel to use for aquizition
X5BytesPerSamp  = 4;   % Because FPGA FFT uses 32nits per sample
X5samples00m01s = (X5BytesPerSamp*SampPerPulse*PRF)/Npulseacc;
                       % Number of samples to set for 0 minutez, 1 second
                       % of FFT spectrums, output every 500ms
X5samples       = Time_Length*X5samples00m01s;   % # SNAP Samples to take
                                                 % for BeamX measurments
X5samplesBK     = Time_LengthBK*X5samples00m01s; % Number of SNAP Samples
                                                 % to take for background

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% WRITE SNAP INIZILIZATION DEFAULT FILE HERE
% variables that are default to the SNAP initilization
% TO BE DONE BEFORE STARTING THE SNAP PROGRAM - will help make sure the
% setting s will be correct- SHOULD CHECK TO SEE THAT THE HsData.bdd
% does not already exist- Will not need to write that file every time
% because the exisitance of the file means the user already set up SNAP
% the way they wanted it to run.
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Lasers' Operational Settings
KeopsysStage1 = 0.170;  % Keopsys 1st Stage Driving Current (A)
KeopsysStage2 = 1.700;  % Keopsys 2nd Stage Driving Current (A)
NP_piezeo     = 0;      % NP Photonics Piezeio Voltage Setting (V)(0=auto)
NP_hiP        = 0.500;  % NP Photonics High Power Output Settinf (W)

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Dummy Data Prototypes to Be Used for Backgrounds (noise) and Data

% OPTION 1: Waveform Based Fake Data Prototype
WfStart1 = 1-(N_pt_spec/2);
WfEnd1   = (N_pt_spec/2)-1;
WfWidth1 = ceil(N_pt_spec*1.09);
WfSkew1  = -0.2;

WfStart2 = 2-(N_pt_spec/2);
```

```
WfEnd2   = (N_pt_spec/2);
WfWidth2 = ceil(N_pt_spec*0.78);
WfSkew2  = 0.1;

dummy_data_Proto = [1,(tripuls(WfStart1:WfEnd1,WfWidth1,WfSkew1) - ...
                       0.63*tripuls(WfStart2:WfEnd2,WfWidth2,WfSkew2))]';
% OPTION 2: Add some if needed

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Maximum Likelyhood Estimation Default Parameters
SNR_Thresh       = 0.0000125; % SNR decision threshold for a measured value
No_Fitted_Points = 3;     % No of spectrum points to be fitted in MLE

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% QUERY User if they have setup FPGA Firmware, and Copied Modified
% SNAP to the Destination Directory
qf = warndlg({'(1) Have You Uploaded the FPGA Firmware? ';...
        X5fpgaLogicPath_Name;'';...
        '(2) Started the Modified SNAP copied from ';...
        SNAPverPath_Name;...
        'to the Data Destination Directory You Created?';'';...
        ['If No to Either Question, Quit MatLab By Pressing '...
        'ctrl + C, And Do So NOW !!']...
        },'','normal');
waitfor(qf); % Force execution to halt until user presses OK Button

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Tell User to start modified SNAP for BACKGROUND measurement (noise
% spectrum)
gf = warndlg({'SNAP Settings To Measure Background (Noise) Spectrum';'';...
            'Start the Modified SNAP - X5 Capture Application';...
            'CONFIGURE TAB';...
            'Set Busmaster Size to 24 MB, Open the Card';'';...
            'SETUP TAB';...
            ['Clock Sourse = Internal, 400.0 Freq (MHz)'];...
            ['Packet Size to ' num2str(X5packetSize) ,', 0x',...
            dec2hex(X5packetSize), ' in Hex'];...
            ['Uncheck all Communications Properties'];'';...
            ['Channels = Ch0, Trigger Source = External, De-activate Auto
Trigger'];...
            ['Frame Size to ' num2str(X5frameSize) ,', 0x',...
            dec2hex(X5frameSize), ' in Hex'];...
            ['Set Data Logging Ceiling to ', num2str(X5samplesBK), ' Uncheck
Auto Stop'];...
            },'','normal');
waitfor(gf); % Force execution to halt until user presses OK Button

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Truck Cardinal Directions (North, South, East, West) and Tilt from Level
%
% Truck_tiltZE:
% Zenithe angle = 0 if laser beam is verticle (Truck optical table is
% level) If the edge of the optical table closest to the front of the truck
% is lower than the egde of the optical table nearest the rear of the truck
% ZE will have a POSITIVE value, and a NEGATIVE value if the rear edge is
% lower than the front edge. ( Or if table is tilted back have Zieneth
% be positive and add 180 degrees to Azmith ??)
```

```
% If truck has the right side of optical table lower than the left side of
% the optical table then __________  Have to come up with a framework to
% decide what to do- don;t worry now
% Any time Truck_tiltZE NOT = Zero the Zenith and Azmith angles used to
% position the beam for a measuremnets relative to the truck's optical
% table will have to be adjusted to make sure beam is going towards the
% correct location in space.

% Truck_cardianlAZ:
% Azmith angle = 0 if the direction of Truck's optical table taken along
% the long edge (length) is North.  North is 0 degrees, East is 90 degrees,
% South is 180 degress and West is 270 degrees

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Query User for: Truck's Cardinal Orientation (Front faces 0 to 359 deg)
% and Truck's Optical Table Tilt from Level
%
% NOTE: Magnetic NORTH is approzimatley 13 deg 10 min off towards EAST
%       in the NYC area.  True North is NOT Magnetic North
prompt3 = {['Research Van Heading (Front Points Toward): North= 0 deg'...
            ' [Heading is Magnetic Heading +13.8 degrees]'],...
           'Optical Table Forward Tilt from Level (degrees):'};
dlg_title3 = 'Research Van Orientation (Cardinal Orientation/Tilt)';
num_lines3 = 1;
def3       = {num2str(Truck_cardianlAZ),num2str(Truck_tiltZE)};
options.Resize      = 'on';
options.WindowStyle = 'normal';
options.Interpreter = 'tex';
answer3 = inputdlg(prompt3,dlg_title3,num_lines3,def3,options);
AZ                  = str2num(answer3[15]);  % Cardinal Heading of Truck
ZE                  = str2num(answer3{2});  % Tilt of Truck

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Users Enters the Tilt (Zieneth Angle) of the Three Laser Beams
Truck_ZE0 = 0;     % Beam #0 Tilt (zenith) relative to optical table
Truck_ZE1 = 30;    % Beam #1 Tilt (zenith) relative to optical table
Truck_ZE2 = 30;    % Beam #2 Tilt (zenith) relative to optical table

prompt4 = {...
            ['Beam0: Zenith Angle (Tilt Relative to ',...
                    'Optical Table) of Verical Beam, degrees:'],...
            ['Beam1: Zenith Angle (Tilt Relative to ',...
                    'Optical Table) of North Beam, degrees:'],...
            ['Beam2: Zenith Angle (Tilt Relative to ',...
                    'Optical Table) of East Beam, degrees:']};
dlg_title4 = 'Zenith Angle of Beams Referenced to Optical Table';
num_lines4 = 1;
def4 = {num2str(Truck_ZE0),num2str(Truck_ZE1),num2str(Truck_ZE2)};
answer4 = inputdlg(prompt4,dlg_title4,num_lines4,def4);
ZE0 = str2num(answer4[15]);  % Zenith angle of verticle beam (Beam0)
ZE1 = str2num(answer4{2});  % Zenith angle of North-ish beam (Beam1)
ZE2 = str2num(answer4{3});  % Zenith angle of East-ish beam (Beam2)

Truck_relative_ZE = [ZE0 ZE1 ZE2];
ZE0offset = ZE0 + ZE;% Laser Beam #0, verticle
                     % Difference of Vector V0 zenith
                     % angle from earth normal
```

```
ZE1offset = ZE1 + ZE;% These would be used if multiple laser beams are used
ZE2offset = ZE2 + ZE;% These are the tilts of the laser beams relative to
                     % the optical table
ZEoffset  = [ZE0offset ZE1offset ZE2offset];

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Query User for the Azmith Angle of the 3 Laser Beams
%
% NOTE: Magnetic NORTH is NOT True North.
%       The Magnetic Decllination is 13 degrees 10 sec West
%       Declination is negative, Inclination is 67 degrees 13 minutes
%       Magnetic Field Strength is 52497.8 nT
%       City College van location
%       Latitude:  40 deg 49min 16.6 sec North
%       Longitude: 73 deg 56min 54.6 sec West
%       Convent Ave is heading 28.9 deg from True North
%       Convent Ave is heading 43.0 deg from Magnetic North
%       Van Heading is 299.33 degrees at City College
Truck_AZ0 = 0;  % Beam #0 deg from North (azmith) relative to optical table
                % (verticle has no compass directional
                % component if verticle)
Truck_AZ1 =120; % Beam #1 degrees from North (azmith)
                % relative to optical table
Truck_AZ2 =240;% Beam #2 degrees from East (azmith)
                % relative to optical table
prompt5 = {...
    ['Beam0: Azmith Angle (Heading Relative to Front of Optical Table)'...
        ' of Verical Beam, degrees:'],...
    ['Beam1: Azmith Angle (Heading Relative to Front of Optical Table)',...
        ' of North Beam, degrees:'],...
    ['Beam2: Azmith Angle (Heading Relative to Front of Optical Table)',...
        ' of East Beam, degrees:']};
dlg_title5 = 'Heading Angle of Beams Referenced to Front of Optical Table';
num_lines5 = 1;
def5 = {num2str(Truck_AZ0),num2str(Truck_AZ1),num2str(Truck_AZ2)};
answer5 = inputdlg(prompt5,dlg_title5,num_lines5,def5);
AZ0 = str2num(answer5[15]);  % Azmith angle of verticle beam (Beam0)
AZ1 = str2num(answer5{2});  % Azmith angle of North-ish beam (Beam1)
AZ2 = str2num(answer5{3});  % Azmith angle of East-ish beam (Beam2)

Truck_relative_AZ = [AZ0 AZ1 AZ2];
AZ0offset = AZ0 + AZ;   % Beam #0 should have no bearing as it should be
                        % verticle if the optical table was level
AZ1offset = mod(AZ1+AZ,360);    % Beam 1 is usually North
                                % and Beam 2 is usually East
AZ2offset = mod(AZ2+AZ,360);    % but YOU MUST look at the Azimith
                                % ZE0 angles to be sure
AZoffset     = [AZ0offset AZ1offset AZ2offset];
V0V1V2_files = {'none','none','none'};  % Cells of last three data files
                                        % used to find wind vectors
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% For the Spectrum of Interest (SOI) calculate the
% index within the pre Trigger spectrums for the start and end of SOI
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
IpreTRstart_SOI = IpreTRstart     + IpreTR_Spec_Start - 1;
            % IpreTRstart_SOI: Index of the pre trigger spectrum (spectrum
            % precceding the fiber tip back reflected pulse) to be used as
```

```
            % the Background (NOISE) of the current pulse having the lowest
            % frequency.
IpreTRend_SOI   = IpreTRstart_SOI + Nspec_bins - 1;
            % IpreTRend_SOI: Index of the pre trigger spectrum (spectrum
            % precceding the fiber tip back reflected pulse) to be used as
            % the Background (NOISE) of the current pulse having the
            % highest frequency.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Directory and File Name Initialize Information From User
%
% Have User Confirm Path to Directory and Approximate File Name
% Since names will be based on the date/time at the start of the run
% an approximate name will be a good starting point to find the
% first data file writen to the directory or the newest file after
% a long break in taking data (given not past midnight)
file_path = uigetdir(file_path,...
                     'Find/Create Directory to Hold FFT Spectrum Files');

% Have User Confirm Approximate Name of the First Data (SPECTRUM) File
prompt = {'If Path to FFT Spectrum Data files is correct:',...
        ['Enter Approximate (model) Name of the First FFT Spectrum File'...
        ' yyyymmddTHHMMSS (no extention):']};
name = 'Input Approximate Name (yyyymmddTHHMMSS) of 1st FFT Spectrum File';
promptWidth = ceil(max(length(file_path),length(file_name))*1.25);
numlines = [1, promptWidth; 1, promptWidth];  % Input Lines, Input Width
defaultanswer       = {file_path,file_name};
options.Resize      = 'on';
options.WindowStyle = 'normal';
answer = inputdlg(prompt,name,numlines,defaultanswer,options);

file_path      = (cell2mat(answer(1)));
file_name      = (cell2mat(answer(2)));
file_path_name = fullfile(file_path, file_name);
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Query Users to Select Names of Operators Taking Data
% Isel= the index to the selected item(s)
% SelEr is 0 if no selection is made, or 1 if a selection is made.

Isel = length(DL_users);    % Last user in list of users is default user
[Isel,SelEr] = listdlg('PromptString','Select Operators (ctrl for
multiple):',...
                'SelectionMode','multiple',...
                'Name','Current Operators',...
                'InitialValue',length(DL_users),...
                'CancelString','No User Name',...
                'ListSize',[200 120],...
                'ListString',DL_users);
DLoperators = DL_users(Isel);

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Query User to Select Location
% Isel = the index to the selected item
% SelEr is 0 if no selection is made, or 1 if a selection is made.
```

```
Isel = 1;       % 1st location in list is default
[Isel,SelEr] = listdlg('PromptString','Select Location:',...
                'SelectionMode','single',...
                'Name', 'Location',...
                'PromptString', 'Select Current Location',...
                'InitialValue',1,...
                'CancelString','No Selection',...
                'ListSize',[200 80],...
                'ListString',DL_locationS);
DL_location = DL_locationS(Isel);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Have User Check Camera Image of Sky
% Record Image of Sky YES/NO
% YES--> Display Image to See if alignment is OK
% Superimpose
% front, rear, right, left
% Red dot at the NORTH position
% north, east south, west

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Query User to Enter Number of Pre Trigger Spectrums In The Data
% User Enters Which Pre Trigger Spectrums to Use as Background Noise
% User Enters Number of Spectrums Effected by Out Going Pulse
prompt6 = {...
    ['Number of Pre-Trigger Spectrums Seen in FFT Data'],...
    ['Use the Nth Pre-Trigger Spectrum to Calculate '...
    'Background (Noise) Spectrum For Each Measurement Set']};
dlg_title6 = 'PRE-TRIGGER SPECTRUMS';
num_lines6 = 1;
def6       = {num2str(NpreTRspec),num2str(UsepreTR),num2str(Truck_AZ2)};
answer6    = inputdlg(prompt6,dlg_title6,num_lines6,def6);

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Query User to Enter Comments
user_comm  = 'none'; % Default User Coments
prompt2    = {'Enter General User Comments Here:'};
dlg_title2 = 'User Input Comments';
num_lines2 = 1;
def2       = {user_comm};
answer2    = inputdlg(prompt2,dlg_title2,num_lines2,def2);

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Prepare Data to be Saved to Meta File
% Can NOT PUT TEXT AND NUMBERS IN SAME ARRAY--USE CELLS?
%    VALUE NAME         ,VALUE
Meta_header_pre = ...
    {'Date_Time_Start' ,datetimeStart,...
    'SNAP_ver'         ,SNAPverPath_Name,...
    'FPGA_firm_ver'    ,X5fpgaLogicPath_Name,...
    'Matlab_file'      ,MlabFname,...
    'Matlab_path'      ,MlabFpath_name,...
    'X5frame'          ,X5frameSize,...
    'X5packet'         ,X5packetSize,...
    'X5_ch'            ,X5ch,...
    'SpeedofLight'     ,c,...
    'Keopsys_1st_Stage',KeopsysStage1,...
```

```
    'Keopsys_2nd_Stage',KeopsysStage2,...
    'NP_piezeoVolt'    ,NP_piezeo,...
    'NP_Hi_power'      ,NP_hiP,...
    'Operator(s)'      ,DLoperators{:},...
    'Van_Loc'          ,DL_location{:},...
    'Van_Zenith'       ,ZE,...
    'Van_Front_orien'  ,AZ,...
    'Vr0ZE'            ,ZE0,...
    'Vr1ZE1'           ,ZE1,...
    'Vr2ZE2'           ,ZE2,...
    'ZE0offset'        ,ZE0offset,...
    'ZE1offset'        ,ZE1offset,...
    'ZE2offset'        ,ZE2offset,...
    'Vr0AZ'            ,AZ0,...
    'Vr1AZ1'           ,AZ1,...
    'Vr2AZ2'           ,AZ2,...
    'AZ0offset'        ,AZ0offset,...
    'AZ1offset'        ,AZ1offset,...
    'AZ2offset'        ,AZ2offset,...
    'Number_PreTriggerSpectrums',char(answer6(1)),...
    'PreTrigger_Spectrums_Calculate_BkGrd',char(answer6(2))...
    'User_Comment'     , char(answer2(1))};

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Prepare to Write Meta Data
%
% Can NOT use 'csvwrite' with mixed data (strings and numberic)
% csvwrite(fullfile(file_path_M,file_name_M),Meta_header)
% does not accept cell arrays for the input matrix M.
% To export a cell array that contains only numeric data, use
% cell2mat to convert the cell array to a numeric matrix before
% calling csvwrite. (each cell has to be exacly same size???)
%
% To export cell arrays with mixed alphabetic and numeric data,
% where each cell contains a single element, you can create
% an Excel spreadsheet (if your system has Excel installed)
% using xlswrite. From Excel you can 'save as' .csv formate.
% For all other cases, you must use low-level
% export functions to write your data. For more information,
% see Exporting a Cell Array to a Text File in the MATLAB Data
% Import and Export documentation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SAVES an EXCEL FILE '.xls' with the filename AND extention
% found in file_name_M -can be changed to not be '.txt' but '.xls'
%[status,msg] = xlswrite(fullfile(file_path_M,file_name_M),Meta_header)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Can Not write to txt file easily because there are string and numberic
% 1- Turn each element of the Cell array into a string with the function
% 'ex_func' which accepts a Cell and turns numberics to strings
%Meta_header = cellfun(@ex_func,Meta_header_pre(:),'UniformOutput',0);

% If above function 'ex_func' is not available use a loop
Meta_header = Meta_header_pre;
for k = 1:length(Meta_header)
```

```
    in_datatype = class(Meta_header{k});
    switch in_datatype
    case 'char'
        % Cell is a string, do nothing
    case 'double'
        Meta_header{k} = num2str(Meta_header{k});
    end
end

[path_string,file_name_wo_ext,file_ext]= fileparts(file_path_name);
file_path_name = fullfile(file_path,[file_name_wo_ext, ...
                                  file_name_suffix_MetaData,file_ext]);
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Create Name/path for metafile
% Following lines only returns the name of new or existing
% file that you designate
[file_name_M, file_path_M, filterindex] = uiputfile( ...
            {'*.txt','Text (*.txt)';...
            '*.csv','Comma Seprerated Values (.csv)';...
            '*.*',  'All Files (*.*)'},...
            'Create .txt File to Hold All of the Meta Data',...
            file_path_name);
if isequal(file_name_M,0) || isequal(file_path_M,0)
   disp('User selected Cancel')
else
   disp(['User selected ',fullfile(file_path_M,file_name_M)])
end

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Write metafile data to file in directory
% The 1st Line of the Text File Will Have Info That Will Not Be Repeated
fid_M = fopen(fullfile(file_path_M,file_name_M), 'w');
fprintf(fid_M,'%s, ', Meta_header{:}); % Write Comma Seperated
fprintf(fid_M,'\n'); % Write Cariage Return to File
fclose(fid_M);

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Tell User: Make Background (Noise) Spectrum Measurements
qf = warndlg({'SNAP: Make Background Measurement (Noise Spectrums)';'';...
            'BEFORE Turning On KEYOPSYS Final Amplifier';'';...
            'STREAM TAB';...
            ['Press Running Man Button'];'';...
            'Hit OK Button Bellow After Pressing SNAP Running Man'...
            },'','normal');
waitfor(qf); % Force execution to halt until user presses OK Button
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Get Real or Create Fake SNAP Data File for BACKGROUND %%%%%%%%%%
switch dBUGfakeSNAP % =0, Get Real Data From Hard Drive Writen by SNAP
                    %     with Date/Time Stamped File Name
                    % =1, Waveform Based Data and Store to a File
                    % =2, Use 'file_name_fakeBK' to Make Data
                    % =3, Get Real Data From Hard Drive Writen by SNAP
                    %     with "my_30_Seconds_file.txt" Name
    case 0  % Get Real Data From Hard Drive Writen by SNAP
            % Do not make fake data, SNAP will write real data to HD
            % in Date/Time Stamped Filename
    case 1  % Make Some Waveform Based FAKE Data and Store to a File
```

```
      dummy_data_tmp = dummy_data_Proto;
      dummy_dataBK   = repmat(dummy_data_tmp,...
                           1,(N_spec_acc*Time_LengthBK/Tpulseacc));
      dummy_data_tmp = []; % empty temp variable

    case 2  % Use 'file_name_fakeBK' to Make a Background Noise File
            % 'file_name_fakeBK' has 1 'N_pt_spec' Long Point Spectrum
            % Make File Same Length That Would Have Been Created by Running
            % SNAP to take 'Time_LengthBK' Seconds of Accumulations
            % by Sequencially Reproducing the 'file_name_fakeBK' the
            % Proper Number of Times
            % Spectrums In A Single FPGA Output Accumulation = N_spec_acc
            % Total number of spectrums= Time_LengthBK*N_spec_acc/Tpulseacc

            % Read Data From Fake Data File  'file_name_fakeBK'
      fidFakeSNAP_BK = fopen(fullfile(file_path,file_name_fakeBK), 'r');
      dummy_data_tmp= fscanf(fidFakeSNAP_BK,'%f ',N_pt_spec);
      fclose(fidFakeSNAP_BK);
            % Reproduce the Fake Background Spectrum Into a Matrix
            % Resemblingg Real Data That Would Come From SNAP

            % pulstran creates a repetative pulse that can do many things
            % plot(pulstran(0:1/8192:0.5-(1/8192),...
            %                            0:1/128:1,dummy_dataBK,8192))
            %dummy_data_tmp=pulstran(0:1/8192:1-(1/8192),...
            %                              0:1/128:1,dummy_dataBK,8192);
      dummy_dataBK = repmat(dummy_data_tmp,N_spec_acc,...
                                           (Time_LengthBK/Tpulseacc));
      dummy_data_tmp = []; % empty temp variable

    case 3  % Get Real Data From Hard Drive Writen by SNAP
            % Do not make fake data, SNAP will write real data to HD
            % as an XX second file text file with know name
end
if (dBUGfakeSNAP==1 || dBUGfakeSNAP==2) % Save fake BACKGROUND data
                                        % to directory that SNAP would
                                        % have saved to with dateTtime
                                        % file format
    fidFakeSNAP = fopen(fullfile(file_path_M,...
                           [datestr(now,time_format),'.txt']), 'w');
    fprintf(fidFakeSNAP,'%u ',dummy_dataBK);% Write Comma Seperated Data
    fprintf(fidFakeSNAP,'\n');              % Write Cariage Return to File
    fclose(fidFakeSNAP);                    % Close File
    %save(fullfile(file_path_M,[datestr(now,time_format),'.txt']),...
    %                                    num2str(dummy_dataBK),'-ascii');
    pause(0.50);      % Slight delay
end
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% User Enters Comments about Background (Noise) Spectrum for Meta Data
user_comm  = ''; % Default User Coments
prompt7  = {'Enter User Comments About Background (Noise) Spectrum Here:'};
dlg_title7 = 'User Input Comments';
num_lines7 = 1;
def7       = {user_comm};
answer7    = inputdlg(prompt7,dlg_title7,num_lines7,def7);

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Check Directory, Has SNAP Finished Writing BACKGROUND (noise) Data,
% Rename File to indicate this is the Background measurement.
% Matlab can not move a file when it is open by other programs

switch dBUGfakeSNAP
    case {0,1,2}    % (0)Real, (1)Fake Ramp, and (2)Fake PreRecorded
                    % Background files saved to HD with dateTtime format.
                    % Get List of Files in Directory without
                    % Sub-Folders or Parent Folders
        dir_data  = dir(file_path);  % Get complete contents of current dir
        dir_index = [dir_data.isdir];% Check to see if entry is a directory
        file_list = {dir_data(~dir_index).name}'; % Make list of file names
                                                  % excluding directories
        % Find Most Recently Modified File in Directory of Saved Data
        %%%%%%%% May need to selectively find .txt file only ???? %%%%%%%%%
        file_time       = {dir_data(~dir_index).datenum}';% Make list of file
times
        last_file_time  = max(cell2mat(file_time));       % Newest time of a
file
        last_file_timeTmp = last_file_time; % Initial time of new file temp
        last_time_index = find(cell2mat(file_time)==last_file_time);
        % Find the index of the newest file time
        last_time_name  = cell2mat(file_list(last_time_index));
        % Get the newest file time

        [path_string,file_name_wo_ext,file_ext]= fileparts(last_time_name);
        % Get the parts of the file name
        % path_string      = String represneting path to file
        % file_name_wo_ext = file name without the extension
        % file_ext         = file extention

        % Change Name of File, Append Suffix (i.e. '_BK') for Background to
name
        last_path_file     = fullfile(file_path,last_time_name);
        last_path_file_new = fullfile(file_path,[file_name_wo_ext,....
                                file_name_suffix_Background,file_ext]);

        % Try to move the SNAP Data file Until sucessful,to New Name with
'_bk.txt'
        pb  = waitbar(0,'Collecting Data');     % Use wait bar to show
progress
        while ~movefile(last_path_file, last_path_file_new)
            % Wait bar shows progress
            waitbar(etime(datevec(last_file_time),...
                                datevec(last_file_timeTmp))...
                                /(Time_LengthBK/Tpulseacc))
            pause(file_wait_time);  % While unable to make name
                                    % change wait and try again after pause
            if dBUG ==1, datestr(now,31), end     % Print time if debugging

            file_data_tmp = dir(fullfile(file_path,last_time_name));
            last_file_time = (file_data_tmp.datenum);% Lastest modification
                                                     % time of file
            display(last_file_timeTmp,last_file_time)
        end
        delete (pb);                    % Close Wait Bar
```

```
case 3 % If file_name is "my_30_Seconds_file.txt" (file_nameXXs)
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Check To See When Text File is Writen Then Rename
    %
    % Each file written to the directory after XX seconds as
    % file_nameXXs will be renamed to yyyymmddThhmmss.txt (Year Month
    % Day 'T' Hours Minutes Seconds .txt) format and the orginal file
    % erased.
    % The normal state of the folder will not have a file_nameXXs in it

    % Generate full path name to XX second data file
    last_time_name = fullfile(file_path,file_name);

    %=====================================================================
    % IF SNAP WROTE 30 Second File AT END OF PREVIOUS RUN AND WAS NOT
    % PROCESSED (ORPHAN FILE) Delete It If It Is More than 10% Older
    % Than The Length of Time The Data Should Be Covering.
    %
    % dir_data = dir(file_nameXXs) % Get file attributes of XX sec file
    % filedate = dir_data.datenum  % Get files modification date
    %
    % If a previous text data file exist from old run delete it
    % if etime(clock,datevec(filedate))>=(Time_Length*1.1)
    %     delete(last_path_file_new)
    % end
    %=====================================================================
    [path_string,file_name_wo_ext,file_ext]= fileparts(last_time_name);
    % Get the parts of the file name
    % path_string      = String represneting path to file
    % file_name_wo_ext = file name without the extension
    % file_ext         = file extention
    while isempty(dir(last_time_name))% Check to see if file is created
        pause(Time_Length*0.15)       % Wait for 15% of total data time
        display(['Waiting ', num2str(Time_Length*0.15),...
                                       ' seconds for data file...'])
    end
    fileinfo = dir(last_time_name); % Get dir info on specific file
    % Rename File, Append Suffix (i.e. '_bk') for Background
    % Use Modification time of file to name file as dateTtime
    % format (yyymmmddThhmmss_bk.txt)
    % Modification time is when the file was opened, not when closed
    last_file_time     = fileinfo(1).datenum;
    file_name_wo_ext   = datestr(last_file_time,30);
                         % Use dateTtime as file name without extention
    last_path_file_new = fullfile(path_string,[file_name_wo_ext,....
                            file_name_suffix_Background,file_ext]);

    % When SNAP Has Finished Writing Data, Rename (Move) File
    % to New Name and Extention
    % NOTE: Even though the file name may be present in the directory
    % this does not mean the file is closed and ready to be read.  You
    % MUST check to make sure.
    while ~movefile(last_time_name, last_path_file_new)
        if dBUG ==1  % Print time if debugging
            display(['File still open,  Waiting ',...
                num2str(Time_Length*0.025),' seconds: ' ...
                                           (datestr(now,31))])
```

```
            end
            pause(Time_Length*0.025); % While unable to make
                                      % name change wait and try again
        end
        display ([file_name,' -->Renamed To--> ',[file_name_wo_ext,....
                                    file_name_suffix_Background,file_ext]])
%       data_file = last_path_file_new; % Record name of last data file
end
noise_file = last_path_file_new; % Name of background (noise) file


%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Append Background (Noise), Time, File Name, Comments Info to Meta File
% Prepare Data to be Saved to Meta File
% Can NOT PUT TEXT AND NUMBERS IN SAME ARRAY--USE CELLS?
Meta_Background_pre = ...
    {datestr(last_file_time,time_format),...
    [file_name_wo_ext,file_name_suffix_Background,file_ext],...
    'BackgroundNoise',...
    cell2mat(answer7)};

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Can Not write to txt file easily because there are string and numberic
% 1- Turn each element of the Cell array into a string with the function
% 'ex_func' which accepts a Cell and turns numberics to strings
% Meta_header = cellfun(@ex_func,Meta_header_pre(:),'UniformOutput',0);

% If above function 'ex_func' is not available use a loop
Meta_background = Meta_Background_pre;
for k = 1:length(Meta_background)
    in_datatype = class(Meta_background{k});
    switch in_datatype
    case 'char'
        % Cell is a string, do nothing
    case 'double'
        Meta_background{k} = num2str(Meta_background{k});
    end
end

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Write Metafile Data to File in Directory
% The 1st Line of the Text File Will Have Info That Will Not Be Repeated
fid_M = fopen(fullfile(file_path_M,file_name_M), 'a');
fprintf(fid_M,'%s, ', Meta_background{:}); % Write Comma Seperated
fprintf(fid_M,'\n'); % Write Cariage Return to File
fclose(fid_M);

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Average Spectrums Recorded to File as Background (Noise) Measurement
%
% Output a single spectrum in s file which is the average of all the
% spectrums collected as background (noise).
% Spectrums Recorded With Keyopsys (Optical Amplifier) Turned Off Are
% Read as Background Data File Saved by SNAP and
% Renamed 'filename'_bk_avg.txt by This Program

% Open Background Data File
```

```
fidSNAP_BK = fopen(noise_file, 'r');
data_BKrun = fscanf(fidSNAP_BK,'%f ',NptsfileBK);
fclose(fidSNAP_BK);

% Average Background Data File (One Spectrum Representing Avg Background)
data_BKavgTmp1 = reshape(data_BKrun,N_pt_spec,[]);% Reshape Matrix
data_BKavgSize = size(data_BKavgTmp1,2);          % Number of ALL Spectrums
                                                  % in Background (noise)
                                                  % measurement
data_BKavg1 = sum(data_BKavgTmp1,2)/data_BKavgSize;

% Average Background Data File (Multiple Spectrums Representing
% Avg Background vs Range)
data_BKavgTmp2 = reshape(data_BKrun,Naccsize,[]);% Reshape Matrix
data_BKavgSize = size(data_BKavgTmp2,2);          % Number of ALL Accums
                                                  % in Background (noise)
                                                  % spectrums
data_BKavg2 = sum(data_BKavgTmp2,2)/data_BKavgSize;

% Save the Average Background (Noise) Spectrum
[path_string, file_name_wo_ext, file_ext] = fileparts(noise_file);
file_name_BK_AVG1 = fullfile(path_string,...
              [file_name_wo_ext,file_name_suffix_BackgroundAVG,file_ext]);
fid_BK_AVG1 = fopen(file_name_BK_AVG1, 'w');
fprintf(fid_BK_AVG1,'%u ', data_BKavg1);  % Write Comma Seperated
fprintf(fid_BK_AVG1,'\n');                % Write Cariage Return to File
fclose(fid_BK_AVG1);

% Save the Averaged Background (Noise) Spectrums  for each Range
[path_string, file_name_wo_ext, file_ext] = fileparts(noise_file);
file_name_BK_AVG2 = fullfile(path_string,[file_name_wo_ext, ...
                    file_name_suffix_BackgroundAVG,num2str(N_spec_acc),...
                    file_ext]);
fid_BK_AVG2 = fopen(file_name_BK_AVG2, 'w');
fprintf(fid_BK_AVG2,'%u ', data_BKavg2);  % Write Comma Seperated
fprintf(fid_BK_AVG2,'\n');                % Write Cariage Return to File
fclose(fid_BK_AVG2);

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Show The User The Averaged Background and Ask if User Wants to Repeat
% the Measurement  <------background slope sometimes seen


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Have User Setup Keopsys Final Amplifier
gf = warndlg({'KEOPSYS Final Amplifier Activation';'';...
 'Follow Setup Procedures to Configure KEYOPSYS';'';...
 ['FIRST Stage Current      = ',num2str(KeopsysStage1,'%2.3f'),' Amps'];...
 ['SECOND Stage Current = ',num2str(KeopsysStage2,'%2.3f'),' Amps'];...
 },'','normal');
waitfor(gf); % Force execution to halt until user presses OK Button
%%% PUT IN OPTION TO HAVE USER CHANGE THE VALUES
%%%% See Test CODE at end of this program

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Instruct User to Setup SNAP to Take Measurement Spectrums
```

```
qf = warndlg({'SNAP SETTINGS TO MAKE MEASUREMENTS (SPECTRUMS)';'';...
            'Modified SNAP Should Already Be Running';'';...
            'Change the Following Settings';'';...
            'SETUP Tab';...
            ['De-Activate Auto Stop'];...
            ['Set Data Logging Ceiling to ', num2str(X5samples)];...
            ['Packet Size to ' num2str(X5packetSize) ,', 0x',...
            dec2hex(X5packetSize), ' in Hex'];...
            ['Frame Count to ' num2str(X5frameSize) ,', 0x',...
            dec2hex(X5frameSize), ' in Hex'];...
            },'','normal');
waitfor(qf); % Force execution to halt until user presses OK Button

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Instruct User Set Mirror for Beam0 (Verticle) (VERY FIRST MEASUREMENT)
qf = warndlg({'Prepare Mirror Position For FIRST Measurment';'';...
            'Move the Mirror To Make Beam0 Measurement (VERTICLE)';'';...
            ['Zenith (Tilt) Angle, Relative to the '...
        'Optical Table: ', num2str(Truck_relative_ZE(1)),' Degrees'];'';...
            ['Azmith Angle, Relative to the Optical Table: ',...
            num2str(Truck_relative_AZ(1)),' Degrees']},'','normal');
waitfor(qf); % Force execution to halt until user presses OK Button

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Prepare to Perform Multiple Measurments
iBeam      = 1;    % The Number of the Current Beam Measuremnt
UserQuit = false;% Flag to tell when user wants to stop

while UserQuit == false  % While the User has not quit
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Instruct User Set Mirror for Next Beam Position
if iBeam ~= 1 % If the Measuremnt Is NOT the First Move to New Position
    qf = warndlg({'REPOSITION MIRROR FOR NEXT MEASUREMENT';'';...
           ['Move the Mirror To Make Beam ',num2str(mod(iBeam-1,3)),...
           ' Measurement'];'';...
           ['BEAM Zenith (Tilt) Angle, Relative to the Optical Table: ',...
           num2str(Truck_relative_ZE(mod(iBeam-1,3)+1)),' Degrees'];'';...
           ['BEAM Azmith Angle, Relative to the Optical Table: ',...
           num2str(Truck_relative_AZ(mod(iBeam-1,3)+1)),' Degrees'];'';...
           'Press OK When Mirror is Re-positioned'},'','normal');
    waitfor(qf); % Force execution to halt until user presses OK Button
end

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Instruct User to Take Data With SNAP
qf = warndlg({'SNAP SETTINGS TO MAKE MEASUREMENTS (SPECTRUMS)';'';...
            'Modified SNAP Should Already Be Running';'';...
            'Settings';...
            ['SETUP Tab: De-Activate Auto Stop, '...
            'Set Data Logging Ceiling to ', num2str(X5samples)];...
            ['Packet Size to ', num2str(X5packetSize),', 0x',...
            dec2hex(X5packetSize), ' in Hex'];...
            ['Frame Count to ' num2str(X5frameSize) ,', 0x',...
            dec2hex(X5frameSize), ' in Hex'];'';...
            ['Press Running Man Button'];'';...
            ['Hit OK Button Bellow After Pressing SNAP Running Man'];...
            },'','normal');
```

```
waitfor(qf); % Force execution to halt until user presses OK Button

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% User Enters Comments about Making BeamX Measurment
user_comm  = ''; % Default User Coments
prompt2    = {['Enter User Comments for Beam ',num2str(mod(iBeam-1,3)),...
              ' Measurement Here:']};
dlg_title2 = 'User Input Comments';
num_lines2 = 1;
def2       = {user_comm};
answer2    = inputdlg(prompt2,dlg_title2,num_lines2,def2);


%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Create fake SNAP data file for Measuremnet BeamX %%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if iBeam==1  % If this is the first run make fake data
 switch dBUGfakeSNAP % =0, Get Real Data From Hard Drive Writen by SNAP
                     %     with Date/Time Stamped File Name
                     % =1, Waveform Based Data and Store to a File
                     % =2, Use 'file_name_fakeBK' to Make Data
                     % =3, Get Real Data From Hard Drive Writen by SNAP
                     %     with "my_30_Seconds_file.txt" Name
    case 0  % Do not make fake data
        % SNAP will generat the data

    case 1  % Make Some  Waveform Based Data and Store to a File
        % fake data (spectrum) template (prototype)
        dummy_data_tmp = dummy_data_Proto;

        % form pre trigger set of spectrums
        dummy_data_PRE = repmat(dummy_data_tmp,NpreTRspec,1);

        % form trigger set of spectrums (tip reflections)
        dummy_data_tmp_trig = dummy_data_tmp*10e6;
        dummy_data_TRIG = repmat(dummy_data_tmp_trig,Ntip_spec,1);

        % form post trigger set of spectrums
        dummy_data_tmp_post = dummy_data_tmp;
        dummy_data_POST = repmat(dummy_data_tmp_post,...
                                 N_spec_acc_post_pulse,1);

        % add a little to the extra to one point in the spectrum
        % and have it progress from AOM freq
        Pt_f0 = ceil(fsAOM/FFTbw);  % # of point (bin) of AOM freq
        Pts_f0_to_end = N_pt_spec - Pt_f0;  % # pts center to
                                            % end of spectrum
        % size of freg step to take
        Pts_to_jump =Pts_f0_to_end/(N_spec_acc_post_pulse);

        for pp = 1:N_pt_spec:length(dummy_data_POST)
         dummy_data_POST(pp+Pt_f0-1+floor((pp/N_pt_spec)*Pts_to_jump))=...
            ... %(1/pp)*... % A scalling factor PUT IN
            1.15*dummy_data_POST(pp+Pt_f0-
1+floor((pp/N_pt_spec)*Pts_to_jump));
```

```
    end

   % Create a Accumulation Output with Pre-trig, Trig, Post Spectrums
   dummy_data_tmp = [dummy_data_PRE; dummy_data_TRIG; dummy_data_POST];

   % Reproduce the Fake Accumulation Data for the Apropriot Length
   dummy_data = repmat(dummy_data_tmp,(Time_Length/Tpulseacc),1);
   dummy_data_tmp = []; % empty temp variable

case 2  % Create a more realistic data (with variation) for BeamX
        % Use 'file_name_fakeBK' to Make a BeamX Data File
        % The 'file_name_fakeBK' has 1 'N_pt_spec' Long Point Spectrum
        % Make File Same Length That Would Have Been Created by Running
        % SNAP to take 'Time_LengthBK' Seconds of Accumulations
        % by Reproducing the 'file_name_fakeBK' the Proper
        % Number of Times
        % Spectrums In A Single FPGA Output Accumulation = N_spec_acc
        % Total number of spectrums= Time_LengthBK*N_spec_acc/Tpulseacc

    % Read Data From Fake Data File  'file_name_fakeBK'
    fidFakeSNAP_BK = fopen(fullfile(file_path,...
                                    file_name_fakeBK), 'r');
    dummy_dataBK = fscanf(fidFakeSNAP_BK,'%f ',N_pt_spec);
    fclose(fidFakeSNAP_BK);

    % Reproduce the Fake Background Spectrum Into a Vector
    % Resembling Real Data That Would Come From SNAP

    % fake data (spectrum) template (prototype)
    dummy_data_tmp = dummy_dataBK;

    % form pre trigger set of spectrums
    dummy_data_PRE = repmat(dummy_data_tmp,NpreTRspec,1);

    % form trigger set of spectrums (tip reflections)
    % MAKE LARGE WITH SCALLING FACTOR TO BETTER MATCH REAL DATA
    dummy_data_tmp_trig = dummy_data_tmp*3.83864e+5; %
    dummy_data_TRIG = repmat(dummy_data_tmp_trig,Ntip_spec,1);

    % form post trigger set of spectrums
    dummy_data_tmp_post = dummy_data_tmp;
    dummy_data_POST = repmat(dummy_data_tmp_post,...
                             N_spec_acc_post_pulse,1);

    % Add a little to the extra to one point in the spectrum
    % and have it progress from AOM freq
    Pt_f0 = ceil(fsAOM/FFTbw);  % The Index of point (bin) of AOM freq
    Pts_f0_to_end = N_pt_spec - Pt_f0;  % Number of pts center to
                                        % end of spectrum
    % Size of freq step to take
    Pts_to_jump =Pts_f0_to_end/(N_spec_acc_post_pulse);

    for pp = 1:N_pt_spec:length(dummy_data_POST)
            dummy_data_POST(pp + Pt_f0 -1 ...
                    + floor((pp/N_pt_spec) * Pts_to_jump) ) = ...
            1.15 * dummy_data_POST(pp + Pt_f0 - 1 ...
                    + floor((pp/N_pt_spec) * Pts_to_jump));
```

```
        end

        % Create a Accumulation Output with Pre-trig, Trig, Post Spectrums
        dummy_data_tmp = [dummy_data_PRE; dummy_data_TRIG; dummy_data_POST];

        % Reproduce the Fake Accumulation Data for the Apropriot Length
        dummy_data = repmat(dummy_data_tmp,(Time_Length/Tpulseacc),1);
        dummy_data_tmp = []; % empty temp variable

    case 3  % Get Real Data From Hard Drive Writen by SNAP
            % Do not make fake data, SNAP will write real data to HD
            % as an XX second file text file with know name
 end
end
if (dBUGfakeSNAP==1 || dBUGfakeSNAP==2)
    % Save Fake Accumulation Data to directory that SNAP would have saved to
    fidFakeSNAP = fopen(fullfile(file_path,...
                            [datestr(now,time_format),'.txt']), 'w');
    fprintf(fidFakeSNAP,'%u ', dummy_data); % Write Comma Seperated
    fprintf(fidFakeSNAP,'\n'); % Write Cariage Return to File
    fclose(fidFakeSNAP);
    %save(fullfile(file_path_M,[datestr(now,time_format),'.txt'])...
    %                                    ,num2str(dummy_data),'-ascii');
    pause(0.50);      % Slight delay
end

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Determine If SNAP Has Completed the Writing of Data to the txt file
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Check Directory, Has SNAP Finished Writing Data,
% Rename File to indicate dtaeTtime file was opened.
% Matlab can not move a file when it is open by other programs

switch dBUGfakeSNAP
    case {0,1,2}    % (0)Real, (1)Fake Ramp, and (2)Fake PreRecorded
                    % Background files saved to HD with dateTtime format.
                    % Get List of Files in Directory without
                    % Sub-Folders or Parent Folders
        dir_data  = dir(file_path);  % Get complete contents of current dir
        dir_index = [dir_data.isdir];% Check to see if entry is a directory
        file_list = {dir_data(~dir_index).name}'; % Make list of file names
                                                  % excluding directories

        % Find Most Recently Modified File in Directory of Saved Data
        %%%%%%%% May need to selectively find .txt file only ???? %%%%%%%%%
        file_time       = {dir_data(~dir_index).datenum}';% Make list of file
times
        last_file_time  = max(cell2mat(file_time));       % Newest time of a
file
        last_file_timeTmp = last_file_time; % Initial time of new file temp
        last_time_index = find(cell2mat(file_time)==last_file_time);
        % Find the index of the newest file time
        last_time_name  = cell2mat(file_list(last_time_index));
        % Get the newest file time

        [path_string,file_name_wo_ext,file_ext]= fileparts(last_time_name);
        % Get the parts of the file name
```

```
        % path_string      = String represneting path to file
        % file_name_wo_ext = file name without the extension
        % file_ext         = file extention

        % Change Name of File, Append Suffix (i.e. '_temp') for name
        last_path_file     = fullfile(file_path,last_time_name);
        last_path_file_new = fullfile(file_path,[file_name_wo_ext,....
                                    file_name_suffix_Temp,file_ext]);

        % Try to move the SNAP Data file Until sucessful,to New Name with
'_temp.txt'
        while ~movefile(last_path_file, last_path_file_new)
            pause(file_wait_time);  % while unable to make
                                    % name change wait and try again
            if dBUG ==1, datestr(now,31), end       % Print time if debugging
            file_data_tmp = dir(fullfile(file_path,last_time_name));
            last_file_time = (file_data_tmp.datenum); % Lastest modification
                                                      % time of file
        end
        movefile(last_path_file_new,last_path_file); % Rename file to
                                                     % SNAP's name

    case 3 % If Data file_name is "my_30_Seconds_file.txt" (file_nameXXs)
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % Check To See When Text File is Writen Then Rename
        %
        % Each file written to the directory after XX seconds as
        % file_nameXXs will be renamed to yyyymmddThhmmss.txt (Year Month
        % Day 'T' Hours Minutes Seconds .txt) format and the orginal file
        % erased.
        % The normal state of the folder will not have a file_nameXXs in it

        % Generate full path name to XX second data file
        last_time_name = fullfile(file_path,file_name);

        %====================================================================
        % IF SNAP WROTE 30 Second File AT END OF PREVIOUS RUN AND WAS NOT
        % PROCESSED (ORPHAN FILE) Delete It If It Is More than 10% Older
        % Than The Length of Time The Data Should Be Covering.
        %
        % dir_data = dir(file_nameXXs) % Get file attributes of XX sec file
        % filedate = dir_data.datenum  % Get files modification date
        %
        % If a previous text data file exist from old run delete it
        % if etime(clock,datevec(filedate))>=(Time_Length*1.1)
        %     delete(last_path_file_new)
        % end
        %====================================================================
        [path_string,file_name_wo_ext,file_ext]= fileparts(last_time_name);
        % Get the parts of the file name
        % path_string      = String represneting path to file
        % file_name_wo_ext = file name without the extension
        % file_ext         = file extention
        while isempty(dir(last_time_name))% Check to see if file is created
            pause(Time_Length*0.15)       % Wait for 15% of total data time
            display(['Waiting ', num2str(Time_Length*0.15),...
                                        ' seconds for data file...'])
```

```
        end
        fileinfo = dir(last_time_name); % Get dir info on expected file
        % Rename File, Append Suffix (i.e. '_temp') for Temporary File
        % Use Modification time of file to name file as dateTtime
        % format (yyymmmddThhmmss_temp.txt)
        % Modification time is when the file was opened, not when closed
        last_file_time     = fileinfo(1).datenum;
        file_name_wo_ext   = datestr(last_file_time,30);
                             % Use dateTtime as file name without extention
        last_path_file_new = fullfile(path_string,[file_name_wo_ext,....
                                                              file_ext]);
% If we want to have atemp file to work with use the name bellow
%         last_path_file_new = fullfile(path_string,[file_name_wo_ext,....
%                                          file_name_suffix_Temp,file_ext]);

        % When SNAP Has Finished Writing Data, Rename (Move) File
        % to New Name and Extention
        % NOTE: Even though the file name may be present in the directory
        % this does not mean the file is closed and ready to be read.  You
        % MUST check to make sure.
        while ~movefile(last_time_name, last_path_file_new)
            if dBUG ==1  % If debugging and file NOT ready Inform User
                display(['File still open,  Waiting ',...
                    num2str(Time_Length*0.025),' seconds: ' ...
                                                 (datestr(now,31))])
            end
            pause(Time_Length*0.025); % While unable to make
                                      % name change wait and try again
        end
        display ([file_name,' -->Renamed To--> ',[file_name_wo_ext,....
                                                              file_ext]])
%       data_file = last_path_file_new; % Record name of last data file
end

data_file = last_path_file_new; % Name of last data file

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Append BeamX, Time, File Name, Comments Info to Meta File
% Prepare Data to be Saved to Meta File
% Can NOT PUT TEXT AND NUMBERS IN SAME ARRAY--USE CELLS?
Meta_BeamX_pre = ...
    {datestr(last_file_time,time_format),...
    [file_name_wo_ext,file_ext],...
    ['Beam'     ,num2str(mod(iBeam-1,3))],...
    ['TruckZE=' ,num2str(Truck_relative_ZE(mod(iBeam-1,3)+1))],...
    ['TruckAZ=' ,num2str(Truck_relative_AZ(mod(iBeam-1,3)+1))],...
                cell2mat(answer2)};

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Can Not write to txt file easily because there are string and numberic
% 1: Turn each element of the Cell array into a string with the function
% 'ex_func' which accepts a Cell and turns numberics to strings
% Meta_header = cellfun(@ex_func,Meta_header_pre(:),'UniformOutput',0);

% If above function 'ex_func' is not available use a loop to convert cells
Meta_BeamX = Meta_BeamX_pre;
for k = 1:length(Meta_BeamX)
```

```
    in_datatype = class(Meta_BeamX{k});
    switch in_datatype
    case 'char'
        % Cell is a string, do nothing
    case 'double'
        Meta_BeamX{k} = num2str(Meta_BeamX{k}); % make into string
    end
end

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Write/Append Metafile Data to File in Directory
% The 1st Line of the Text File Will Have Info That Will Not Be Repeated
fid_M = fopen(fullfile(file_path_M,file_name_M), 'a');
fprintf(fid_M,'%s, ', Meta_BeamX{:}); % Write Comma Seperated
fprintf(fid_M,'\n'); % Write Cariage Return to File
fclose(fid_M);

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Open Last Data File Saved by SNAP to Prepare for Doppler Shift Estimation

% data_BKavg1:  % Average Background Data (One Spectrum Representing
%                 Avg Background) from Spectrum From Data Collected
%                 by SNAP when Keyopsys (Optical Amplifier) was turned off
%                 Saved as dateTtime_bk_avg.txt
% data_BKavg2:  % Average Background Data (One Spectrum for Each Average
%                 Background at each range) from Data Collected
%                 by SNAP when Keyopsys (Optical Amplifier) was turned off
%                 Saved as dateTtime_bk_avgXX.txt where XX is the number
%                 of Range Gates (Spectrums) in the file

% Read New Data From File
fidSNAP_data = fopen(data_file,'r');
SNAP_data    = fscanf(fidSNAP_data,'%f ',Nptsfile);
fclose(fidSNAP_data);

% Record The Names of Last Three Data File for Beam0, Beam 1, and Beam2
[LastData_Path, LastData_Name, LastData_Ext] = fileparts(data_file);
V0V1V2_files{mod(iBeam-1,3)+1} = [LastData_Name, LastData_Ext];

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Average No_Averages Accumulations to form an Average

% Turn a single column vector of many accumulations (each made of many
% spectrums one after another), into a matrix where each column is a single
% accumulation consisting of multiple spectrums.  Each column represents
% an average of No_Averages accumulation output by the FPGA
SNAP_data_avg = zeros(Naccsize,Naccs/No_Averages); % Prepare result matrix
SNAP_data_tmp = reshape(SNAP_data,Naccsize,[]);
for n = 1:No_Averages:Naccs
 SNAP_data_avg(:,n)= sum(SNAP_data_tmp(:,n:n+No_Averages-1),2)/No_Averages;
end

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculate Doppler Shift from Collected Data
    % TODO- Make a function from MLE code where Inputs: filename.txt,
    % number of pre-trigger spectra, spectra to use as background
    % Outputs: Peak in each range gate
```

```
switch dBUGvelMethode   % Velocity Method to Find Wind
                        % =1, MLE, Maximum Likelihood Method
                        % =2, Use Peak Detection Method
                        % =3, Use Eq.3 from "Fast, Accurate Frequency
                        %     Estimators- dsp Tips & Tricks" for Detection
                        %     Method of Finding Wind Velocity

 case 1 % =1, Wind Velocity by Maximum Likelihood Method (MLE)
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Calculate Doppler Shift
    % in_file = fopen(data_file,'r');
    back_ground = zeros(1, NpreTR);
    One_Gate_Spectrum = zeros(1,N_pt_spec);
    Signal = zeros(Naccsize,1);
    Signal_Sum = zeros(Naccsize,1);
    Signal_Intensity = zeros(N_spec_acc,Naccs/No_Averages);
    Freq_Shift       = zeros(N_spec_acc,Naccs/No_Averages);
    MLE_Freq_Shift   = zeros(N_spec_acc,Naccs/No_Averages);
    Vel_MLE_Freq_Shift= zeros(N_spec_acc,Naccs/No_Averages);
    Freq_Shift_Est   = zeros(N_spec_acc,Naccs/No_Averages);
    Vel_Freq_Shift_Est=zeros(N_spec_acc,Naccs/No_Averages);
    Range_Corrected  =zeros(N_spec_acc_post_pulse,Naccs/No_Averages);
    Range_Corrected2 =zeros(N_spec_acc_post_pulse,Naccs/No_Averages);
    %Range_Corrected  = zeros(60,Naccs/No_Averages);
    %Range_Corrected2 = zeros(60,Naccs/No_Averages);
    %Precision        = zeros(fsAOM,Naccs/No_Averages);
    Precision        = zeros(N_spec_acc,Naccs/No_Averages);
    Power_Vector     = zeros(1,2*No_Fitted_Points +1);
    Frequency_Vector = zeros(1,2*No_Fitted_Points +1);
    MLE_Precision    = zeros(N_spec_acc,Naccs/No_Averages);
    Precision_Est    = zeros(N_spec_acc,Naccs/No_Averages);
    One_Gate_Frequency_Vector = zeros(1,2000);

    Back_Ground_noise = data_BKavg1;
%       output_file_str = strcat(data_file(1:length...
%                           (data_file)-13),'reorganized.txt');
%       output_file = fopen( output_file_str ,'w') ;

    for n =1:1:Naccs;
%      [Signal count] = fscanf(in_file,'%f',Naccsize); % count = #
%                                                     % of sucessesful reads
     Signal = SNAP_data((n-1)*Naccsize+1:n*Naccsize); % count = #
     count = length(Signal);                                    % of
sucessesful reads
     % if count ~= Naccsize THROW an ERROR
     Signal_Sum = Signal_Sum + Signal/No_Averages ;
     if(mod(n,No_Averages)==0)    % If Signal_Sum is average of No_Averages

      % NEEED TO NOT PROCESS THE PRE TRIGGER SPECTRUMS THE spectrum
      % processed should be after the NTH spectrum SEEN IN BEFORE THE
      % OUTGOING PULSE MAKE jj GO FROM  #OF PRE Trigger SPECTRUMS+1 jj is the
      % range GAte??????? NpreTRspec+1 ??????????????????????????????????
      % N_spec_acc NUMBER OF SPECRUMS FOR EACH ACCUMULATION

      % ???? for jj =NpreTRspec+1:1:N_spec_acc;
```

```
      for jj =1:1:N_spec_acc;% jj is the index of the ranged spectrum
        switch dbugNoiseSub % Select Method to Remove Background (Noise)
                            % S=Signal, N=Noise
                            % =0 10Log(S+N)-10Log(N),
                            % =1 ((S+N)-N)/N,
                            % =2 10Log((S+N)-N) - 10Log(N)
         % IpreTR_Spec_Start: Index of the frequency bin within a
            % spectrum of NFFT/2 points with the lowest frequency
            % which we are interested in.
         % IpreTR_Spec_Stop: Index of the frequency bin within a
            % spectrum of NFFT/2 points with the highest frequency
            % which we are interested in.
         % IpreTRstart_SOI: Index of the pre trigger spectrum (spectrum
            % precceding the fiber tip back reflected pulse) to be used as
            % the Background (NOISE) of the current pulse having the lowest
            % frequency.
         % IpreTRend_SOI: Index of the pre trigger spectrum (spectrum
            % precceding the fiber tip back reflected pulse) to be used as
            % the Background (NOISE) of the current pulse having the
            % highest frequency.
         case 0
           One_Gate_Spectrum = 10*log10(...
                    Signal_Sum(IpreTR_Spec_Start+(jj-1)*N_pt_spec:...
                                  IpreTR_Spec_Stop+(jj-1)*N_pt_spec))...
                    -10*log10(Back_Ground_noise(IpreTR_Spec_Start:...
                                                  IpreTR_Spec_Stop));
         case 1 % Subtract Background Noise Counts and then
                % normalise spectrum by dividing by Background Noise
           One_Gate_Spectrum = (Signal_Sum(...
                           IpreTR_Spec_Start+(jj-1)*N_pt_spec:...
                                  IpreTR_Spec_Stop+(jj-1)*N_pt_spec)...
               -(Back_Ground_noise(IpreTR_Spec_Start:IpreTR_Spec_Stop)))...
                   ./Back_Ground_noise(IpreTR_Spec_Start:IpreTR_Spec_Stop);
         case 2 % Subtract Background Noise Counts and then
                % take log of remaining spectrum and
                % subtract log Background Noise
           One_Gate_Spectrum_minusNoise = Signal_Sum(...
                                IpreTR_Spec_Start+(jj-1)*N_pt_spec:...
                                IpreTR_Spec_Stop+(jj-1)*N_pt_spec)...
                  -(Back_Ground_noise(IpreTR_Spec_Start:IpreTR_Spec_Stop));
           One_Gate_Spectrum = 10*log10(...
                    One_Gate_Spectrum_minusNoise...
                        (IpreTR_Spec_Start+(jj-1)*N_pt_spec:...
                                      IpreTR_Spec_Stop+(jj-1)*N_pt_spec))...
                       -10*log10(Back_Ground_noise(IpreTR_Spec_Start:...
                                        IpreTR_Spec_Stop));
        end

        % Dividing a Spectrum by the background noise spectrum
        % is the same as subtracting the 10*log10(spectrum)
        % valuse of the signal and background noise spectrum
% PROBLEM: DIVIDING THE SIGNAL SPECTRUM BY THE NOISE
% SPECTRUM WILL MAKE THE ERROR VERY LARGE WHEN THE
% SPECTRUM STRENGTH IS WEAK (AT THE HIGHER FREQUENCIES
% THERE IS ONLY A LITTLE POWER SO ANY ERROR WILL LARGE
% COMPARED TO THE MIDDLE FREQ WHERE THE POWER IS HIGHER)
```

```
%        for jj =1:1:fsAOM;
%           One_Gate_Spectrum = 10*log10(...
%               Signal_Sum(10+(jj-1)*128:50+(jj-1)*128))...
%                 -10*log10(Back_Ground_noise(10:50)) ;
          [Max_Signal, Max_Freq] = max(One_Gate_Spectrum);
% PROBLEM - if you do not use 10*log10 of Background Noise the Shape of
%           the spectrum will influnece the Max_Signal,Max_Freq

          Signal_Intensity(jj,n/No_Averages) = Max_Signal;
          if jj > 5
              Range_Corrected2(jj,n/No_Averages) = 10*log10...
                                        (10^(Max_Signal/10) * (jj*48)^2);
          else
          end

          Precision(jj,n/No_Averages)= 0.91*(1+2.1*Max_Signal)...
                                      /(Max_Signal*sqrt(No_Averages * 10e3));
        if (Max_Signal > SNR_Thresh)
              Freq_Shift(jj,n/No_Averages) = (fsAOM-(Max_Freq+8)*FFTbw )...
                                                    * ( (LambdaLO)/2) ;

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %%%% The following section will calculate MLE
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
             if (((Max_Freq+No_Fitted_Points)<length(One_Gate_Spectrum))...
                                      && ((Max_Freq-No_Fitted_Points)>0))
                  index1 = Max_Freq - No_Fitted_Points;
                  for k = 1:1: 2*No_Fitted_Points +1 ;
                      % peak_freq - No_Fitted_Points : 1 :
                      % peak_freq + No_Fitted_Points ;
                      Power_Vector(k) = One_Gate_Spectrum(index1 +k -1)';
                      % Power Spectrum Value
                      Frequency_Vector(k) = 8+index1 +k -1;
                      % (2 * k/No_Samples_w_Zero_Pad) * max(fseed);
                  end

                  counter_1 =0;
                  counter_2 =0;
                  for m = 1:1:2*No_Fitted_Points +1;
                      for i = 1:1:floor(abs((Power_Vector(m))*100))+1;
                         One_Gate_Frequency_Vector( i+counter_1 ) = ...
                                                    Frequency_Vector(m);
                         counter_2 = counter_2 +1 ;
                      end
                      counter_1 = counter_1 +i ;
                  end
                      Stat_Array = mle(One_Gate_Frequency_Vector...
                                                        (1:counter_2));
                      My_mu = Stat_Array(1);
                      My_sigma = Stat_Array(2);

                      MLE_Freq_Shift(jj,n/No_Averages) = (fsAOM - My_mu);

                      Vel_MLE_Freq_Shift(jj,n/No_Averages) = ...
                                (fsAOM - My_mu) * ( (LambdaLO)/2) ;
%                       MLE_Freq_Shift(jj,n/No_Averages) = ...
%                                 (fsAOM- My_mu * FFTbw ) * ( (LambdaLO)/2) ;
```

```
                    MLE_Precision(jj,n/No_Averages) = ...
                                                My_sigma-My_mu * 1.5/2;
            else
                   % If Threshold is not meet set freq shift to 999
                   MLE_Freq_Shift(mod(jj,N_pt_spec)+1,n/No_Averages)=999;
                   % If Threshold is not meet set precision to 10
                   MLE_Precision(jj,n/No_Averages) = 10 ;

            end
       else
            Freq_Shift(mod(jj,N_pt_spec)+1,n/No_Averages) = 999;
            Signal_Intensity(jj,n/No_Averages) = 0;
            Precision(jj,n/No_Averages) = 10 ;
            MLE_Freq_Shift(mod(jj,N_pt_spec)+1,n/No_Averages) = 999;
            MLE_Precision(jj,n/No_Averages) = 10 ;
       end
     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
     %%%% End of MLE
     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      end
             Signal_Sum  = zeros(Naccsize,1);
      else
  end

     %back_ground = back_ground + Signal(4993:5120)';
     Vel_Freq_Shift_Est = Vel_MLE_Freq_Shift;
     Precision_Est  = MLE_Precision;
end

case 2 % =2,  Wind Velocity by Peak Detection Method
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   % Calculate Doppler Shift




case 3 % =3, Wind Velocity by using Eq.3 from IEEE
       % Fast, Accurate Frequency Estimators- dsp Tips & Tricks
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   % Calculate Doppler Shift
   % Ref: Fast, Accurate Frequency Estimators- dsp Tips & Tricks
   % Jacobsen, and Kootsookos
   % IEEE Signal Processing Magazine May 2007, Pg 123, Equation 2
   %
   %                    Kpeak = k + gama
   %                    ftone = Kpeak*(fs/N)                    (EQ 1)
   %
   %                    (abs(X(k+1)) - abs(X(k-1)))
   %    gama  = ---------------------------------------------   (EQ 2)
   %              4*abs(X(k)) - 2*abs(X(k-1)) - 2*abs(X(k+1))
   %
   Power_Vector      = zeros(N_spec_acc,1);% Vector To hold Max return
                                           % signal power in each range
                                           % gate
   Signal_Sum        = SNAP_data_avg;
   Back_Ground_noise = data_BKavg1;
```

```
for n = 1:size(SNAP_data_avg,2); % Process each column of the
                                 % averaged accumulation matrix
One_Gate_Spectrum_History=[];
 for jj = 1:1:N_spec_acc;% jj is the range gate index of the spectrums
                         % The larger jj the farther away the gate
   IdxRanSpecStart = IpreTR_Spec_Start+(jj-1)*N_pt_spec;
   IdxRanSpecFinish= IpreTR_Spec_Stop +(jj-1)*N_pt_spec;
   switch dbugNoiseSub   % Method to Remove Background (Noise)
                         % S=Signal, N=Noise
                         % =0, 10Log(S+N)-10Log(N) = 10Log(S)=S in dB
                         % Diagnotic modes
                         % =1, ((S+N)-N)/N = S/N
                         % =2, 10Log((S+N)-N) - 10Log(N) = (S/N) in dB
      % IpreTR_Spec_Start: Index of the frequency bin within a
         % spectrum of NFFT/2 points with the lowest frequency
         % which we are interested in.
      % IpreTR_Spec_Stop: Index of the frequency bin within a
         % spectrum of NFFT/2 points with the highest frequency
         % which we are interested in.
      % IpreTRstart_SOI: Index of the pre trigger spectrum (spectrum
         % precceding the fiber tip back reflected pulse) to be used as
         % the Background (NOISE) of the current pulse having the lowest
         % frequency.
      % IpreTRend_SOI: Index of the pre trigger spectrum (spectrum
         % precceding the fiber tip back reflected pulse) to be used as
         % the Background (NOISE) of the current pulse having the
         % highest frequency.
   case 0 % =0, 10Log(S+N)-10Log(N)
        One_Gate_Spectrum = 10*log10(...
              Signal_Sum(IdxRanSpecStart:IdxRanSpecFinish))...
                          -10*log10(...
                 Back_Ground_noise(IpreTR_Spec_Start:IpreTR_Spec_Stop));

   case 1     % =1, ((S+N)-N)/N
              % Subtract Background (Noise) Counts from measurement
              % (Signal with Noise) and then normalise spectrum by
              % dividing by Background (Noise), i.e. SNR in counts
        One_Gate_Spectrum = (Signal_Sum(...
                          IdxRanSpecStart:IdxRanSpecFinish)...
            -(Back_Ground_noise(IpreTR_Spec_Start:IpreTR_Spec_Stop)))...
                ./Back_Ground_noise(IpreTR_Spec_Start:IpreTR_Spec_Stop);

   case 2     % =2, 10Log((S+N)-N) - 10Log(N)
              % Subtract Background (Noise) Counts from measurement
              % (Signal with Noise) and then take log of remaining
              % spectrum and subtract log Background Noise Equivelent to
              % 10Log[((S+N)-N)/N] = 10Log(S/N) Like 10Log of case=1
              % above, i.e. SNR in dB

        One_Gate_Spectrum_minusNoise = Signal_Sum(...
                             IdxRanSpecStart:IdxRanSpecFinish)...
               -(Back_Ground_noise(IpreTR_Spec_Start:IpreTR_Spec_Stop));
        One_Gate_Spectrum = 10*log10(One_Gate_Spectrum_minusNoise)...
                     -10*log10(Back_Ground_noise(IpreTR_Spec_Start:...
                                       IpreTR_Spec_Stop));
       % Something to try
       % One_Gate_Spectrum = abs(One_Gate_Spectrum).^2;
```

```
      end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Debugging plot- The Atmospheric range gate spectrums occur after
% NpreTRspec pretrigger spectrums and Ntip_spec spectrums where the tip
% back reflections has obviously affected the spectrum for the range gate
%
% Use One_Gate_Spectrum_History for debugging
One_Gate_Spectrum_History = [One_Gate_Spectrum_History One_Gate_Spectrum];
if dBUG == true
    figure(2)
    hold on
    FreqHorAxis =(IpreTR_Spec_Start*FFTbw: FFTbw: IpreTR_Spec_Stop*FFTbw);
    if jj <= NpreTRspec
        subplot(2,2,1),plot(FreqHorAxis/1e6,One_Gate_Spectrum)
        title('Pre-Trigger Spectrums')
        xlabel('Spec. of Interest (SOI), MHz')
        ylabel('Inten. (BKGD Corr), Counts')
    elseif jj <= NpreTRtip_spec
        subplot(2,2,2),plot(FreqHorAxis/1e6,One_Gate_Spectrum)
        title('Fiber Tip Reflection Spectrums')
        xlabel('Spec. of Interest (SOI), MHz')
        ylabel('Inten. (BKGD Corr), Counts')
    elseif jj > NpreTRtip_spec
        subplot(2,2,3:4),plot(FreqHorAxis/1e6,One_Gate_Spectrum)
        title('One Gate Spectrum Of Interest (SOI): Post Corrected
Spectrums')
        xlabel('Spectrum of Interest (SOI), MHz')
        ylabel('Intensity (Background Corrected), Counts')
        display(['Plot, Post Corected Spectrum Number --> ', num2str(jj)])
    end
    grid on

    pause
    if jj >= N_spec_acc
        clf     % clear figure after plotting all range gate spectrums
        %hold off % let the graph stay untile the next batch of data arrives
    end

end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
     if (jj >= (NpreTRspec+Ntip_spec))  % Only Process Range Gates
                                        % beyond the back reflection from
                                        % the tip, with the >= the spectrum
                                        % directly after the spectrum of
                                        % the tip reflection will be
                                        % processed BUT it is very Very
                                        % VERY large compared to the notmal
                                        % spectrums seen just one range
                                        % gate further along in time.
      [Max_Signal, Max_Freq] = max(One_Gate_Spectrum);

      if (Max_Signal > SNR_Thresh) % Calculate Equation 2 (estimate
                                   % of frequency between FFT bins) if
                                   % signal detected is sufficently strong
       % Avoid Max Signals close to the max(last) and min(first) freq bins
       if ( ((Max_Freq + 1) < length(One_Gate_Spectrum))...
                                              && ((Max_Freq - 1) > 0))
```

```
            gama_num  = abs(One_Gate_Spectrum(Max_Freq+1))...
                                    - abs(One_Gate_Spectrum(Max_Freq-1));
            gama_den  = 4*abs(One_Gate_Spectrum(Max_Freq))...
                                  - 2*abs(One_Gate_Spectrum(Max_Freq-1))...
                                  - 2*abs(One_Gate_Spectrum(Max_Freq+1));
            gama      = gama_num / gama_den;
            kMax_Freq = Max_Freq + gama;
            ftone     = (kMax_Freq + IpreTR_Spec_Start) * FFTbw;
            % Power Spectrum Value
            Power_Vector(jj) = One_Gate_Spectrum(Max_Freq);
            % NOTE- Using value of Max_Freq NOT kMax_Freq

            Frequency_Vector(jj) = ftone;
            Freq_Shift_Est(jj) = (fsAOM - ftone);   % <-----------------------
-Pos Dop Shift would be Neg
            Vel_Freq_Shift_Est(jj) = Freq_Shift_Est(jj) * (LambdaLO/2);
            Precision_Est(jj)  = 999; % FIX THIS FIND STD DEV or VARiance
        else
            % If Threshold is not meet, set freq shift, vel est to 999
            Freq_Shift_Est(jj) = 999;
            Vel_Freq_Shift_Est(jj) = 999;
            % If Threshold is not meet, set precision to 999
            Precision_Est(jj)  = 999;
        end % Peak is min bins from min or max of spectrum

      end  % SNR Threshold min requirement
     end   % Range Gate beyound tip reflection
    end    % Step through range gates
   end

end        % Velocity Method to Find Wind



%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SNR simulation of Lidar signal based on equation 3
% of Mitsubishi paper titled compact all-fiber pulsed coherent doppler
%
% Compact all-fiber pulsed coherent Doppler lidar system for wind sensing
% S. Kameyama, T. Ando, K. Asaka, Y. Hirano, and S. Wadaka
% 10 April 2007,  Vol. 46, No. 11, APPLIED OPTICS 1953
%
%            Eta_D(L) * Lambda * E * Beta * K^(2L/1000) * pi * D^2
% SNR(L)  = ------------------------------------------------------
%                              8 * h * B * L^2
%
%
%                                 Eta_S * Eta_F
% Eta_D(L)= ---------------------------------------------------------
%                                  pi*(Ac*D)^2           Ac*D
%           [ 1 + (1-(L/F))^2 * (--------------_)^2 + (---------)^2 ]
%                                  4*Lambda*L           2*So(L)
%
%
% Eta_S = Eta_I * Eta_RE * Eta_A * Eta_PP * Eta_Q
%
%
```

```
%                2*k*T*Fn*h*v          Eta_Q * PL * Rin * RB
% Eta_PP = ( 1 + ------------------ + --------------------- ) ^-1
%              Eta_q * e^2 *PL * RL      2 * h * V
%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Distance = zeros(60);
So       = zeros(60);
term_a   = zeros(60);
term_b   = zeros(60);
term_c   = zeros(60);
Eta_D    = zeros(60);
Term_1   = zeros(60);
SNR      = zeros(60);
Eta_S    = Eta_I * Eta_RE * Eta_A * Eta_PP * Eta_Q ; %
for L = 1:1:NpreTRtip_spec; % Distance (m)
    Distance(L) = L * RGS;
    So(L) = (1.1 * KW^2 * Distance(L) * Cn_sq)^(-3/5);
    term_a(L) = (1- Distance(L)/F)^2 ;
    term_b(L) = ([pi*(Ac*D)^2/(4*LambdaLO*Distance(L))]^2) ;
    term_c(L) = (Ac*D/(2*So(L)))^2;
    Eta_D(L) = ( Eta_S * Eta_F)/(1+ term_a(L)* term_b(L) + term_c(L));
    Term_1(L) = LambdaLO*E*beta*[(K)^(2*Distance(L)/1000)]*pi*...
                                        (D)^2/(8 * h* B* Distance(L)^2);
    SNR(L) = Eta_D(L)* Term_1(L) ;
end

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Dr. Arend Plot request from 11/21/11
%  Dr. Arend wants a plot that is updated every time SNAP saves a
%  measurement to the hard drive and data is processed.
%  He wants to have the plot of Intensity vs Range updated as the program
%  runs so the operator can make adjustments to the fiber position.
%  CHECK THIS IDEA TO:
%  plot last 3 results using axis limits of current data
%  Dr. Arend request, 11/28/11, horizontal axis be in meters instead of
%  range gates. Range Gate Size (RGS) currently 48m.
%  ---- Perhaps two horizontal axis can be created (range gates and meters)
figure(1)
plot(1:length(Power_Vector(NpreTRtip_spec+1:end)),...
                                    Power_Vector(NpreTRtip_spec+1:end))
grid on
title(['Max Signal Intensity (Corected for Background) vs Range Gate'])
xlabel(['Range Gate(',num2str(RGS,'%4.1f'),'m Each), After ',...
        num2str(NpreTRspec),' Pre-trigger,',...
        ' and ',num2str(Ntip_spec),' Tip Back Reflection Gates'])
ylabel('Intensity')

% Put 2nd X-axis (meters) on plot
hl1 = line(1:length(Power_Vector(NpreTRtip_spec+1:end)),...
                        Power_Vector(NpreTRtip_spec+1:end),'Color','b');
ax1 = gca;
set(ax1,'XColor','b','YColor','b')
ax2 = axes('Position',get(ax1,'Position'),...
           'XAxisLocation','top',...
           'YAxisLocation','right',...
           'Color','none',...
```

```
           'XColor','k','YColor','k');
% To draw a and line on same graph
hl2 = line((1:length(Power_Vector(NpreTRtip_spec+1:end)))*RGS,...
            Power_Vector(NpreTRtip_spec+1:end),'Color','k','Parent',ax2);




%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Save Wind Velocity Array for BeamX as yyyymmddThhmmss.BeamX.txt Format
%
% Suffix String for BeamX velocity file
BeamX = [file_name_suffix_BeamX,num2str(mod(iBeam-1,3)),'Vel'];

% Vel_Freq_Shift_Est is the BeamX Line of Sight Wind Velocity
Vel_Est_BeamX_file = fullfile(LastData_Path,...
                                [LastData_Name,BeamX,LastData_Ext]);
fid_Vel_Est = fopen(Vel_Est_BeamX_file, 'w');
%fid_Vel_Est = fopen(Vel_Est_BeamX_file, 'w','US-ASCII'); % character
encoding scheme
fprintf(fid_Vel_Est,'%6.4f ',Vel_Freq_Shift_Est);  % Write
fprintf(fid_Vel_Est,'\n');                  % Write Cariage Return to File
fclose(fid_Vel_Est);

% SAVE NAME OF DATA FILES USED TO MAKE THE CURRENT VEL VECT CAL
% REPLACE MLE_Freq_Shift with Vel_Freq_Shift_Est
% and MLE_Precision with Precision_Est

AZTruck_relativeX = num2str(Truck_relative_AZ(mod(iBeam-1,3)+1));
ZETruck_relativeX = num2str(Truck_relative_ZE(mod(iBeam-1,3)+1));
AZTruck_cardinalX = num2str(Truck_cardianlAZ);
ZETruck_cardinalX = num2str(Truck_tiltZE);
AZoffsetX = num2str(AZoffset(mod(iBeam-1,3)+1));
ZEoffsetX = num2str(ZEoffset(mod(iBeam-1,3)+1));

% Append BeamX Velocity, Time, File Name, Comments Info to Meta File
% Prepare Data to be Saved to Meta File
% Can NOT PUT TEXT AND NUMBERS IN SAME ARRAY--USE CELLS?
Meta_BeamX_Vel_pre = ...
    {datestr(last_file_time,time_format),...
    [file_name_wo_ext,file_ext],...
    ['Beam',num2str(mod(iBeam-1,3))],...
    ['TruckAZ=',num2str(Truck_relative_AZ(mod(iBeam-1,3)+1))],...
    ['TruckZE=',num2str(Truck_relative_ZE(mod(iBeam-1,3)+1))],...
    cell2mat(answer2),...
    ['TruckCardinalAZ=',AZTruck_cardinalX],...
    ['TruckCardinalZE=',ZETruck_cardinalX],...
    ['TruckOffsetAZ=',AZoffsetX],...
    ['TruckOffsetZE=',ZEoffsetX]...
        };

%If function 'ex_func' is not available use a loop
Meta_header = Meta_BeamX_Vel_pre;
for k = 1:length(Meta_header)
    in_datatype = class(Meta_header{k});                %<--------------fix
    switch in_datatype
```

```
    case 'char'  %Cell is a string, do nothing
    case 'double'
        Meta_header{k} = num2str(Meta_header{k});
    end
end

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Write Metafile Data to File in Directory for BeamX  <----------------------
---------------
% The 1st Line of the Text File Will Have Info That Will Not Be Repeated
fid_M = fopen(fullfile(file_path_M,file_name_M), 'a');
fprintf(fid_M,'%s, ', Meta_background{:}); % Write Comma Seperated
fprintf(fid_M,'\n'); % Write Cariage Return to File
fclose(fid_M);

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculate Wind VECTOR for Last Three BeamX Vector Measurements
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This program;
% INPUTS: Three raadial vectors of wind velocity from each laser beam
%                                    (VrBeam0, VrBeam1,VrBeam2)
% Azimuth Angles (AZ0, AZ1, AZ2)  degrees: Referenced from NORTH (0 deg)
% Zenith Angles  (ZE0, ZE1, ZE2)  degrees: Referenced from
%                                          Normal to the Earth Center
% Range Gate Size (RGS)
% OUTPUTS: Calculates the resultant wind vectors

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Wind VECTOR Azimuth, direction TOWARDS which the wind is blowing
% Phi_VEC
% METEOROLOGICAL WIND DIRECTION is direction FROM which the wind blows
% Phi_MET
% Phi_MET = Phi_VEC + 180
%
% Unit vectors
% iii = unit vector in the NORTH
% jjj = unit vector in the EAST
% kkk = unit vector in the VERTICLE
% rrr = unit vector in the RADIAL DIRECTION
%
% Wind Vectors use right-hand-rule with thumb pointing EAST
%
% thetaZE, laser beam directed at zenith angle theta
% phiAZ, laser beam directed at azimuth angle phi
% Vrbeam, the radial component of the wind vector along laser beam
% u, zonal      (towards East)
% v, meriodonal (towards North)
% w, vertical
%
% Radial Vind Vector at Zenith Angle and Azmith Angle (Direction);
% Vrbeam(thetaZE,phiAZ) = u * sin(thetaZE) * sin(phiAZ)
%                       + v * sin(thetaZE) * cos(phiAZ)
%                       + w * cos(thetaZE)
%
```

```
% Since v and u will NOT be aligned with the exact North (0 deg) or
% East (90 deg) the horizontal wind vector in the verticle plane
% defined by phiAZ is
% vH(phiAZ) = u * sin(phiAZ) + v * cos(phiAZ)
%
% Radial component of velocity
% Vrbeam(thetaZE,phiAZ) = vH(phiAZ) * sin(thetaZE) + w * cos(thetaZE)
%
% Using THREE laser beam directions (i.e. vertical, north-ish, East-ish)
% The vertical component vV(thetaZE) is determined by the w component

% VERTICAL Components
%   vV(thetaZE) = w * cos(thetaZE)
% HORIZONTAL Components
%   vH(phiAZ)   = (Vrbeam(thetaZE,phiAZ) - w*cos(thetaZE)) / sin(thetaZE)
%
% If beam is normal to earth: thetaZE=0, cos(thetaZE)=1
% then vV(thetaZE=0) = w*cos(0) = w*1 = w
% and  vH(phiAZ=NA)  =(Vrbeam(thetaZE=0,phiAZ=NA) - w*1)/0 = 0/0
%
% If beam is horizontal to earth: thetaZE=90, cos(thetaZE)=0
% then vV(thetaZE=90) = w*cos(90) = w*0 = 0
% and  vH(thetaZE=0,phiAZ=y)=(Vrbeam(thetaZE=0,phiAZ=y) - w*cos(90)/sin(90)
%                           =((Vrbeam(thetaZE=0,phiAZ=y) - w*0)/1
%                           =
switch mod(iBeam-1,3)    % Most recent BeamX will be used to Wind Vector
    case 0
        VrBeam0 = Vel_Freq_Shift_Est;
    case 1
        VrBeam1 = Vel_Freq_Shift_Est;
    case 2
        VrBeam2 = Vel_Freq_Shift_Est;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if iBeam >= 3 % Calculate Wind Vectors After 3 Vectors are Measured
  % wind_calculate_function from Dr. Gross (12/04/11)
  % wm1,wm2,wm3 are the 3 wind measurments
  % Angles in radians
  % gmrot= z axis rotation of mirror to get beam tilting (set to approx 29)
  %      "-" gama (counter Clockwise) equals Beam1 (Easterly direction)
  %      "+" gama (Clockwise) equals Beam2 (Northernly direction)
  % phic = azimuth rotation of the optical table in van (lab) coordinates
  % to geo-coordinates (E,N,V). The angle between the laser (defines as
direction 1)
  % and true East. The laser comes from the East of the u direction.
  % Strange- "+" rotation seems to be counter clockwise and Phic=0 seems to
  % have van at the -45 degree heading (clockwize 45 degrees)

  wm1=VrBeam1;  % wm1 is X direction
  wm2=VrBeam2;  % wm2 is Y direction
  wm3=VrBeam0;  % wm3 is Z direction

  gmrot=29.4*pi/180;
  Truck_cardianlAZ_gross = 360 - Truck_cardianlAZ;
  %phic = (Truck_cardianlAZ -90)*pi/180; % <-------------------- FIX
%   if ((Truck_cardianlAZ <= 90) && (Truck_cardianlAZ >= 0))
```

```
%      phic = (90 - Truck_cardianlAZ)*pi/180;
%   elseif ((Truck_cardianlAZ > 90) && (Truck_cardianlAZ <= 360))
%      phic = (90 - Truck_cardianlAZ)*pi/180;
%   end
phic = (90 - Truck_cardianlAZ)*pi/180; % For Counter Clockwise Rotation

  % <----------------------------------------FILTER THE 999 from functions

  for ww = 1:length(wm1)
    [we(ww),wn(ww),wv(ww)] = wind_calculate_function2...
                                (wm1(ww),wm2(ww),wm3(ww),gmrot,phic);
  end
%   [we,wn,wv]=wind_calculate_function(wm1',wm2',wm3',gmrot,phic);
  V0 = wv; % <---- REMOVE THE NEGATIVES ???/check
  V1 = we;
  V2 = wn;
  % Take East(V1) and North(V2) Vectors to find direction and speed of wind
  HorAZ =  mod((360 + atan2(V1,V1)*180/pi),360);
  HorSpeed = sqrt(V1.^2 + V2.^2);

if 10==8
Vertical_Elemnts = length(V1);
Horizontal_Elemnts = 4;

U=repmat(V1',1,Horizontal_Elemnts);
V=repmat(V2',1,Horizontal_Elemnts);


[yearW monthW dayW hourW minW secW] = datevec(now);

timeW   = [1:0.25:1-0.25+0.25*Horizontal_Elemnts];
heightW = [1: 2: 2*size(U,1)];

% [latgrat_T,longrat_T] = meshgrid(heightW,timeW);
% latgrat=latgrat_T';
% longrat=longrat_T';
[latgrat,longrat] = meshgrat(heightW,timeW);
% X = U(:);
% Y = V(:);
H= latgrat;
T= longrat;

% H= latgrat(:);
% T= longrat(:);
eee = 20;
for dd = 1:4
for ee = 1:size(U,1);
    if (U(ee,dd) > eee || U(ee,dd) < -eee)
        U(ee,dd)= 0;
    end
    if (V(ee,dd) > eee || V(ee,dd) < -eee)
        V(ee,dd)= 0;
    end
end
end
```

```
barb_fig = figure(3)  ;% Wind Barb PLOT
plot_windbarb(U*2, V*2, H,T,0.5,'ms')
%axis([xscale(1) xscale(16) -100 2200]);
ylabel('heightW (m)');
xlabel('timeW');
title(['Wind Barb measured on:  ', date]);

quiver_plot = figure(4) ;
quiver( -1*U,-1*V), grid on
% axis([0 17 0 23]);
title(['Wind Speed and Direction measured on:  ', date]);
xlabel('timeW ');
ylabel('heightW (m)')

end

%     V0 =  VrBeam0;
%     V1 = (VrBeam1 - VrBeam0.*cos(ZE1offset)) ./sin(ZE1offset);
%     V2 = (VrBeam2 - VrBeam0.*cos(ZE2offset)) ./sin(ZE2offset);
%
%     V0V1V2 = [V0;V1;V2];  % Output matrix: Vert, North, East Componets  <--
--------------VO is only one number (999)

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Calculate Wind Velocity and Direction from  Horizontal Components
    % (East Component, North Component), Direction:North=0 deg, East=90 deg
    % South=180 deg, West=270 deg
%     HorAZ = zeros(length(V0V1V2),1);
%     % Wind Direction & Wind Velocity Loop
%     for nwd = 1:length(V0V1V2)
%         if ((V1(nwd)==0) && (V2(nwd)==0)),HorAZ(nwd) = 999;, end
%         if ((V1(nwd)>0) && (V2(nwd)~=0)),
%             if (V2(nwd)>0), HorAZ(nwd) = atan(V2(nwd)/V1(nwd));
%              else, HorAZ(nwd) = 360 - atan(V2(nwd)/V1(nwd));, end
%         end
%         if ((V1(nwd)<0) && (V2(nwd)~=0))
%             if (V2(nwd)>0),HorAZ(nwd) = 180 - atan(V2(nwd)/V1(nwd));
%              else, HorAZ(nwd) = 180 + atan(V2(nwd)/V1(nwd));, end
%         end
%         if ((V1(nwd)~=0)||(V2(nwd)~=0)),HorSpeed(nwd) = sqrt(V1(nwd)^2 +
V2(nwd)^2);
%         else, HorSpeed(nwd) = 999;,end
%     end

    %% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Print to Screen: Wind Velcity Vectors for u, v, w for 1000 to 1500m

    % Use 'disp' to supress Matlab ptinting 'Var = ', new line the values
    % Range (Meters),
    % Horizontal Components (Vector EAST, Vector NORTH, Vector UP)
    % Horizontal SPEED and DIRECTION (wind is goinng TOWARDS)
    disp('    Range     Vu=Ve     Vv=Vn     Vw=Vup     Speedh    Dirh')
    %disp(gallery('normaldata',[8 6],0));% 'gallery' makes fake test data
    if iBeam == 1
        V0 =  VrBeam0;
        % Only Print first vector
```

```
        % rtemp = sprintf('%5.0f \n',(20:32)'*RGS); % try to format
        disp([(20:32)'*RGS, V0(20:32)'])
    elseif iBeam==2
        V1 = (VrBeam1 - VrBeam0.*cos(ZE1offset)) ./sin(ZE1offset);
        % Only Print 1st Two Vectors
        disp([(20:32)'*RGS,V1(20:32)', V0(20:32)'])
    elseif iBeam>=3
        % Print All Vectors and Calculations
        disp([(8:32)'*RGS,V2(8:32)' V1(8:32)' V0(8:32)' ...
            HorSpeed(8:32)' HorAZ(8:32)']) % WHYS is HorAZ col vec? <--------
-------???
    end

end
%% Save Wind Vector Calculation to File

% Save the 3 files used to find the wind vectors V0V1V2_files
% Append BeamX Velocity, Time, File Name, Comments Info to Meta File
% Prepare Data to be Saved to Meta File

% FIX THIS HERE
% Can NOT PUT TEXT AND NUMBERS IN SAME ARRAY--USE CELLS?
% Prepare Velocity, Vector, Wind Direction, Wind Velocity Meta Information
Meta_Vec_pre = ...
    {datestr(last_file_time,time_format),...    % Date & time
    [file_name_wo_ext,file_ext],...
    ['Data ',V0V1V2_files],...
    ['TruckZE=',num2str(Truck_relative_ZE(mod(iBeam-1,3)+1))],...
    ['TruckAZ=',num2str(Truck_relative_AZ(mod(iBeam-1,3)+1))],...
    cell2mat(answer2)};

% If function 'ex_func' is not available use a loop
Meta_Vec_header = Meta_Vec_pre;
for k = 1:length(Meta_Vec_header)
    in_datatype = class(Meta_Vec_header{k});                %<--------------
------fix
    switch in_datatype
    case 'char'  %Cell is a string, do nothing
    case 'double'
        Meta_Vec_header{k} = num2str(Meta_Vec_header{k});
    end
end
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Write Metafile Data to File in Directory for Horizontal Wind Vector
% The 1st Line of the Text File Will Have Info That Will Not Be Repeated
[path_string_M,file_name_wo_ext_M,file_ext_M]= fileparts(file_name_M);
file_path_MwVec = [file_name_wo_ext_M,...
                                file_name_suffix_WindVlVcDr,'.txt'];
% fid_M = fopen(fullfile(file_path_M,file_path_MwVec), 'a');
% fprintf(fid_M,'%s, ', Meta_Vec_header{:}); % Write Comma Seperated
% fprintf(fid_M,'\n'); % Write Cariage Return to File
% fclose(fid_M);


 %% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 % Increase index iBeam, the total number of the BeamX measurments
 iBeam = iBeam + 1;    %
```

```
 display(['Beam Measurement: ',num2str(iBeam)])

 %% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 % Query User If They Would Like to Continue Taking Data
 options.Interpreter = 'tex';
 % Include the desired Default answer
 options.Default = 'YES- Continue Taking Data';
 % Create a string for the question
 qstring = 'Would You Like To Continue To Take Data';
 choice = questdlg(qstring,'TAKE MORE DATA ?',...
                   'YES- Continue Taking Data','No',options);
 switch choice
    case 'YES- Continue Taking Data'
        % Do Nothing and continue in the while loop
    case 'No'
        UserQuit = true;    % User wants to quite
 end

end


% END OF NORMALLY EXECUTED CODE

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%                                                          %%%%%%%%%%
%%%%%%% BELLOW ARE PROTIONS OF TEST CODES I TRIED OUT %%%%%%%%%%
%%%%%%%                                                          %%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


if 5==9
A = randn(10,1);

% tequnique to have user enter a variable name and value
checkLabels = {'Save sum of A to variable named:' ...
               'Save mean of A to variable named:'};
varNames = {'sumA','meanA'};
items = {sum(A),mean(A)};
export2wsdlg(checkLabels,varNames,items,...
             'Save Sums to Workspace');


A = randn(10,1);
checkLabels = {'Save sum of A to variable named:' ...
               'Save mean of A to variable named:'};
varNames = {'sumA','meanA'};
items = {sum(A),mean(A)};
export2wsdlg(checkLabels,varNames,items,...
             'Save Sums to Workspace');


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
d = dir;
str = {d.name};
[s,v] = listdlg('PromptString','Select a file:',...
                'SelectionMode','single',...
                'ListString',str);



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% VIDEO CAMERA INPUT


% Select or enter a file name for saving a figure
% as an image in one of four formats, described in a cell array.
uiputfile({'*.jpg;*.tif;*.png;*.gif','All Image Files';...
          '*.*','All Files' },'Save Image',...
          'C:\Work\newfile.jpg')

%% TO MAKE A BUTTON ON A FIGURE WINDOW
qf = figure;
h = uicontrol('Position',[20 20 200 40],'String','Continue',...
              'Callback','uiresume(gcbf)');
disp('This will print immediately');
uiwait(gcf);
disp('This will print after you click Continue');
close(qf);
end
```

Part 2: Matlab function: wind_calculate_function2

```
function [we,wn,wv]=wind_calculate_function2(wm1,wm2,wm3,gmrot,phic)

% 01/04/12 Reworked version of function
% wm1,wm2,wm3 are the 3 wind measurments along
%                                Beam1 (u), Beam2 (v) and Beam0 (w)
% Angles are in radians
% gmrot= z axis rotation of mirror to get beam tilting (set to approx 30))
% phic = azimuth rotation of the optical table (lab) coordinates to geo
% coordinates (E,N,V).
% Clockwise rotation is "-" angle rotation,
% Counter-clockwise is "+" angle rotation.
% The angle is between the laser and the face of the mirror.

% Orientation of Van/Optical Table to Coordinate System
% The laser beam moves from the front of the van towards the rear of the
% van along the optical table.
% The mirror is rotated by Gama Rotation (gmrot) in the "+" (Counter clockwise) and
% "-" (clockwsie) angles as viewed from above the optical table.
% gmtot = 0     ----> wm3, Verticle
% gmtot = "+" X ----> wm2, Beam1
% gmtot = "-" X ----> wm1, Beam2

% A "+" wind velocity measurement is wind moving AWAY from the lidar
% A "-" wind velocity measurement is wind moving TOWARDS the lidar
%

gm = gmrot;
% gmtot = 0     ----> wm3, Verticle
% gmtot = "+" X ----> wm2, Beam1 ?? Beam2
% gmtot = "-" X ----> wm1, Beam2 ?? Beam1

M1 = wm3;   % Verticle Measurement
M2 = wm1;   % Northernly Beam before rotation of truck (Beam from East)
M3 = wm2;   % Southernly Beam before rotation of truck (Beam from East)



% Wind Vector R = Ax + By +Cz
An = M2 + M3 - (M1*(1+cos(gm)));
Ad = ( (cos(gm))^2 - cos(gm) - sqrt(2)*(sin(gm))^2 );
A = An / Ad;

Bn = M2 - M3;
Bd = (1-sqrt(2)) * sin(gm) * cos(gm) - sin(gm);
B = Bn/Bd;

C = M1;

wlab =  [A; B; C];
% Rotation Matrix (Counter Clockwise rotation ("+" angle rotation)
    R = [cos(phic)   -sin(phic)    0; ...
         sin(phic)    cos(phic)    0; ...
           0            0          1];

    wout = R*wlab;
    we   = wout(1);
    wn   = wout(2);
    wv   = wout(3);
```

```
% we = A;
% wn = B;
% wv = C;
```

```
% % set coordinate axis
% ax =[1,0,0];
% ay =[0,1,0];
% az =[0,0,1];
%
% % set gamma rotation angle (+/- ~30)
% gm = gmrot;
%
%
%    %  unit vectors describing laser beam direction in laboratory frame
%    %  based on mathematical calculation.
%
%     e1=[cos(gm)^2-1, -sin(gm)*cos(gm), cos(gm)];  % + gm solution
%     e2=[cos(gm)^2-1,  sin(gm)*cos(gm), cos(gm)];  % - gm solution
%     e3=[    0      ,         0       ,    1   ];  % nominal (gm=0 solution)
%
%
%     % storing the 3 wind measurments as a row vector
%     wm=[wm1;wm2;wm3];
%
%     % generating the matrix representing the projection from x-y-z to
%     % 1-2-3.
%
% %     B(1,:)=[dot(e1,ax) dot(e1,ay) dot(e1,az)];
% %     B(2,:)=[dot(e2,ax) dot(e2,ay) dot(e2,az)];
% %     B(3,:)=[dot(e3,ax) dot(e3,ay) dot(e3,az)];
% %
% %     A1=inv(B);
% %
% %   Matrix  that takes (c1,c2,c3) into (w1,w2,w3)
%
```

```
%     B(1,:)= [dot(e1,e1) dot(e1,e2) dot(e1,e3)];
%     B(2,:)= [dot(e2,e1) dot(e2,e2) dot(e2,e3)];
%     B(3,:)= [dot(e3,e1) dot(e3,e2) dot(e3,e3)];
%
%     cv=inv(B)*wm;
%
%
% % Matrix going from (1,2,3) to (x,y,z)   % note that this matrix operates
% % on (c1,c2,c3)  not (w1,w2,w3);
%
%     A(1,:)=[dot(e1,ax) dot(e2,ax) dot(e3,ax)];
%     A(2,:)=[dot(e1,ay) dot(e2,ay) dot(e3,ay)];
%     A(3,:)=[dot(e1,az) dot(e2,az) dot(e3,az)];
%
%     % inverse matrix makes the transformation from 1-2-3 ro x-y-z
%
% %     B=inv(A);
%
%     % matrix multiplication gives wind vector in lab (x-y-z) frame
%
% %    wlab=B*wm;
% %    wlab=A*wm;
%      wlab=A*cv;
%
%     % additional rotation needed to move the lab frame to the (E,N,V)
%     % frame assuming for simplicity no significant need for an azimuth.
%     % This angle is the angle made by the laser projected onto the true
%     % east axis
%
%     % R is the rotation matrix for a positive angle rotation (counter
%     % clockwise rotation) about the Z axis
%     R = [cos(phic)   -sin(phic)    0; ...
%          sin(phic)    cos(phic)    0; ...
%            0            0          1];
%     % For a negative angle rotation (counter clockwise rotation) about the
%     % Z axis Matrix RzNeg
%     % RzNeg = [ cos(phic)    sin(phic)    0; ...
%     %          -sin(phic)    cos(phic)    0; ...
%     %            0            0          1];
%
%
%     wout = R*wlab;
%     we   = wout(1);
%     wn   = wout(2);
%     wv   = wout(3);
```

Part 3: Matlab program: ProcLidarBeamVel_06.m

```
% ProcLidarBeamVel_06.m
%
% clear all;
% David Santoro
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% PURPOSE:
% This program will read, from the hard drive,  text file of a Beam's wind
% velocity created by Matlab program takelidarXX.m and create a 3D cell
% array of the data for further processing.
% OUTPUT:
% Matlab .mat file with the nameing convention.
% yyyymmddTHHMMSS_WindVec_CCCCC_ssssssS.mat
% _CCCCC   = Number of Beam Measurments Processed
% _ssssssS = Number of seconds the data collection period spans
%
% For Example: 20111205T122223_WindVec_00177_015038S.mat
% Year = 2011, Month = December, Day = 5th
% Time = 12:22:23 (22 minutes and 23 seconds past noon)
% Number of Measurments = 177
% Collection period spanns 150386 seconds (approz 4 hrs, 10 min, 38 sec)
%
% The modified verion of SNAP (Sameh) took 8192 data points and outputs 64,
% 64 point spectrums that resulted from accumulating the FFT of 10,000
% laser pulses (1/2 sec) and wrote a text file to the hard drive after
% taking a total of 30 sec worth of data (i.e. A text file concisting of
% series of 60 sets of spectrums where each set of spectrums had 64
% spectrums each of 64 points.)
%
% These text files were processed and turned into a Matlab vector made of
% 64 wind velocities (one for each range gate along the beam (NOTE: There
% are a few range gates that are from before the laser beam exited the tip
% of the fiber)).  The velocites along the beam were saved in a text file
% with the name yyymmddThhmmss_BeamXVel.txt, where X is 0|1|2.
% Beam0 is the verticle beam (w)
% Beam1 is the (u) : u beam is counter clockwise of the v beam (Top View)
% Beam2 is the (v) : v beam is clockwise of the u beam (Top View)
%
% In takelidarXX.m the SNAP program with a FPGA code that takes 8192 data
% points, uses 128 concecutive points to compute 64 FFT spectrums, add the
% newest spectrums to a running sum of spectrums until 10,000 have been
% colleted. After 10,000 summations these are appended to a file until 30
% seconds has been collected.  Then the user will be instructed to readjst
% the laser beams direction and to repeat the data collection.
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The text file of accumulated spectrums will have the naming format
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FILE NAME NOMENCLATURE
% The default text folder name is encoded with the START date and time
% yyyymmddTHHMMSS.txt
% yyyymmddTHHMMSS is ISO 8601 standard- NOTE:The 'T' between date and time
% yyyy = the year (i.e. 2011)
```

```
% mm   = the month (two digits, i.e January = 01)
% dd   = the day of the month (two digit day of the month, i.e. 07)
% HH   = the hour (24 hour clock, i.e. 3:45PM is 1545, NO daylight savings)
% MM   = the minutes of the hour (00 to 59)
% SS   = the seconds of the minute (00 to 59)
% Matlab has a Numeric Identifier Predefined Format of 30
% for the ISO 8601 format standard
% If the time right now was October 5th 2011, 10:51:15 PM
% ds = datestr(now,30) --> ds = 20111005T225115

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Useful Unit Converstions
% 1 mph            = 0.44704 meters / second
% 1 knot           = 0.514444444 meters / second
% 1 foot / second  = 0.3048 meters / second
% 1 meter / second = 1.94384449 knots
% 1 meter / second = 2.23693629 mph
% 1 meter / second = 3.2808399 feet / second


time_format      = 30;% Time Format 30 = 'yyyymmddTHHMMSS' (ISO 8601)
datetimeStart    = datestr(now,time_format);% Get program start date & time
file_finish_time = 0; % Variable to hold time when SNAP finishes with file
file_wait_time   = 3; % Seconds to wait before testing to see
                      % if SNAP has finished writing to a file

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Set Up Debugging Variables
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
dBUG            = 1;% DEBUGGING SWITCH
                    % =1 (true) Do debugging,
                    % =0 (false) No debuging
                    %
dbugPC          = 1;% LOCATION OF COMPUTER
                    % Use to precnfigure filenames, directories, etc
                    % =0 for WinXP 64bit Doppler PC (CCNY),
                    % =1 for home (santoro),
                    % =2 for work (santoro),
                    % =3 for work T647 (santoro),
                    % =4 for temporary site
dbugPLOTS       = 0;% CREATE PLOTS FOR DEBUGGING
                    % =0 No debuging plots,
                    % =1 Plot Debug Plots
                    %
dBUGvelMethode = 3;% METHOD OF DETERMINING WIND VELOCITY
                    % Freq Peak Wind Vel Method
                    % =1 MLE, Maximum Likily Methode
                    % =2,
                    % =3, Frequency Estimators- dsp Tips
                    %     Fast, Accurate Frequency Estimators- dsp Tips &
                    %     Tricks Jacobsen, and Kootsookos IEEE Signal
                    %     PRocessing Magazine May 2007, Pg 123, Equation 2
                    %
dBUGfakeSNAP = 3;  % SOURCE OF DATA
                    % =0 Do NOT create Fake SNAP data, REAL DATA on HD with
                    %    Date/Time stamped file name
                    % ------------------------------------------------
                    % =1 Create fake SANP data for debugging (ramp func)
                    % ------------------------------------------------
                    % =2 Create fake background data from a data file of ONE
                    %    spectra. And generate sample data by modifying it.
                    %    Use
```

```
                    %    092311DelayT0_AB_2.24uS_my_output_file_3_60s.txt
                    %    as fake data Has delay of 2.24us, 9 pretrigger
                    %    spectra and the 1st two spectra seem to be
                    %    different from the next seven spectra.  There
                    %    seems to be 3 spectra effected by the outgoing
                    %    pulse tip reflection
                    % ------------------------------------------------
                    % =3 Do NOT create Fake SNAP data, REAL DATA on HD with
                    %    "my_30_Seconds_file.txt" file name
                    %
dbugNoiseSub = 0;   % METHOD OF REMOVEING BACKGROUND (NOISE)
                    % =0 10Log(S+N)-10Log(N), CHECK?????
                    % =1 ((S+N)-N)/N,
                    % =2 10Log((S+N)-N) - 10Log(N)

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Set default SNAP OUTPUT File Type
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
file_nameXXs = 'my_30_Seconds_file.txt';
SNAPfiletype = 1;   % FILE NOMENCLATURE EXPECTED FROM SNAP PROGRAM
                    % =0, Text File, yyyymmddTHHMMSS.txt
                    % =1, Text File, my_30_Seconds_file.txt
switch SNAPfiletype
  case 0
      file_name = [datetimeStart,'.txt'];% File Name with Date/Time Stamp
  case 1  % If SNAP outputs non date/time stamped file name
      file_name = file_nameXXs;
  case 3  % Other file format
end

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Set default directories and file names based on location of PC
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
switch dbugPC    % =0 for van ,
                 % =1 for home,
                 % =2 for work,
                 % =3 Work T647
                 % =4 for _______
    case 0 % =0 for van
      file_path = ...
       'D:\TEMP_SANTORO\20111213_Data_SNAPmod\20111213_Data_SNAPmod_Run2';
       %'D:\TEMP_SANTORO\20111201_Data_SNAPmod\20111201_Data_SNAPmod_Test'
       %'D:\TEMP_SANTORO\20111128_Data_SNAPmod\';
       %'D:\TEMP_SANTORO\20111125_Data focused at 2797m\20111125_Data SNAPmod focused
at 2797m\20111125_Data Vert Keopsys 170ma 1500ma DDG 1.1227us';
       %'D:\TEMP_SANTORO\PhD Matlab\Lidar_Text_Data_Matlab\20111117_Data\';

      file_name_fakeBK = ...
       'BackGround_Noise_092311DelayT0_AB_2.24uS_my_output_file_3_60s.txt';
      %file_name = '20111004T131022.txt';
    case 1 % =1 for home
%       file_path = ...
%        'D:\David\PhD\PhD
Data\Data\20111205_Data_SNAPmod\20111205_Data_SNAPmod_Run2_temp';
      file_path = ...
        'D:\David\PhD\PhD
Data\Data\20111208_Data_SNAPandSNAPmod\20111208_Data_SNAPmod_Run1_scanFor2h50m_temp';
      file_name_fakeBK = ...
       'BackGround_Noise_092311DelayT0_AB_2.24uS_my_output_file_3_60s.txt';
      %file_name = '20111004T131022.txt';
    case 2 % =2 for work
      %file_path = 'C:\David\CCNY PhD Matlab\Lidar_Text_Data_Matlab';
```

```
      file_path = ...
       'C:\David\CCNY PhD Data\20111205_Data_SNAPmod\20111205_Data_SNAPmod_Run2_temp';
      file_name_fakeBK = ...
       'BackGround_Noise_092311DelayT0_AB_2.24uS_my_output_file_3_60s.txt';
      %file_name = '20111004T131022.txt';
    case 3 % =3 for work T647
      file_path ='C:\Temp\santoro\CCNY PhD\PhD
Data\20111205_Data_SNAPmod\20111205_Data_SNAPmod_Run2_temp';
      %file_path = 'C:\Temp\santoro\PhD Matlab\Lidar_Text_Data_Matlab';
      file_name_fakeBK = ...
       'BackGround_Noise_092311DelayT0_AB_2.24uS_my_output_file_3_60s.txt';
      %file_name = '20111004T131022.txt';
    case 4 % =4 for _____
      file_path = 'E:\PhD\PhD
Data\Data\20111205_Data_SNAPmod\20111205_Data_SNAPmod_Run2_temp';
      file_name_fakeBK = ...
       'BackGround_Noise_092311DelayT0_AB_2.24uS_my_output_file_3_60s.txt';
      %file_name = '20111004T131022.txt';
end

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Default Length of Time to Take Measurememts
Time_Length   = 30;    % Number of seconds to make a measurment
Time_LengthBK = 30;    % Number of  seconds to take an independent, before
                       % measuremnts start, background (noise) measurement

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Default File Name Suffixes to be Used
file_name_suffix_Background    = '_bk';  % Background (Noise) Spectrums
file_name_suffix_BackgroundAVG = '_avg'; % Average Background (Noise) Spec
file_name_suffix_MetaData      = '_meta';% Meta Data
file_name_suffix_Temp          = '_temp';% temporary Data
file_name_suffix_WindVlVcDr    = '_WindVec';% Wind Velocity,Vectors,Directions
file_name_suffix_BeamX         = '_Beam';% BeamX
file_name_suffix2_BeamX         = 'Vel';% BeamX Velocity

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Constants and Default Values %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Physical Constants
c    = 299792458;       % Speed of Light (m/s) in a vacume
nAir = 1.000293;        % Refractive Index of air
                        % Understanding the properties of matter
                        % By Michael De Podesta page 131
h    = 6.62606957e-34;  % Planck's constant (Js)

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Digital Delay Generator (Standford DG-535)
DDGjitM    = 25e-9;     % Output jitter speced by manufacturer
DDG_TplusA = 1.1228e-6; % T+delay for A time (s)
DDG_AplusB = 300e-9;    % A+delay for B time (s)
DDG_PulseT = DDG_AplusB - DDG_TplusA;   % DDG pulse period (s)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ADC (Innovative Intergrations X5-400M verD) Constants and Defaults
FsamplingM= 400e6;  % ADC sampling Rate (spec manufacturer)
ADCbitsM  = 14;     % ADC number of bits (spec manufacturer)
ADCenobM  = 10.1;   % Effective Number of Bits for our ADC
                    % (spec manufacturer) with full scale sinosoidal input
                    % 70MHz, 850 mVRMS filtered sine sampled at 400 MSPS,
                    % DC-200 MHz, DC-coupled specified by manufacturer
ADCsnrM   = 73.6;   % SNR (dB) (spec manufacturer) for
                    % 70MHz, 850 mVRMS filtered sine sampled at 400 MSPS,
```

```
                    % DC-200 MHz, DC-coupled specified by manufacturer
ADCnoiseM = 350e-6; % Typical noise for grounded input (V), 1 standard
                    % deviation by manufacturer (spec manufacturer)
ADCoffseterrorM = 700e-6;   % Factory calibration Offset Error(V), averaged
                            % 64K samples after warmup (spec manufacturer)
ADCgainerrorM   = 0.0002;   % 0.02%, Factory calibration Gain Error
                            % after warmup (spec manufacturer)
ADVinMaxM       = +1;       % Input Voltage Max (V) (spec manufacturer)
ADVinMinM       = -1;       % Input Voltage Min (V) (spec manufacturer)
ADCjitAdditiveM = 80e-12;   % Additive jitter of ADC using external clock
                            % (s) (spec manufacturer)
ADCjitInterM    = 340e-15;  % Total jitter of ADC using internal clock
                            % (s) (spec manufacturer)

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Query User for Length of Time to Measure Signals and Background (Noise)
% prompt1 = {['How Many Seconds Will The One Time Background (Noise)'...
%             ' Measurement Be?'],...
%             'How Many Seconds Will Each Beam Measurent Be?'};
% dlg_title1         = 'Measurements Durration';
% num_lines1         = 1;
% def1               = { num2str( Time_LengthBK), num2str(Time_Length ) } ;
% options.Resize     = 'on';
% options.WindowStyle= 'normal';
% options.Interpreter= 'tex';
% answer1            = inputdlg(prompt1,dlg_title1,num_lines1,def1,options);
% ATime_LengthBKZ    = str2num(answer1[15]);
% Time_Length        = str2num(answer1{2});

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  System Parameters
SampPerPulse = 8192;    % Number of samples collected per laser pulse
UsepreTR     = 4;       % Use the nth pre trigger spectrum to find the
                        % average backgound noise spectrum of a
                        % particular pulse
Navg_spec = Time_Length*2;  % Default Number of spectrums from background
                            % measurements to use in order to find
                            % BACKGROUND (noise) is ALL
Naccsize = 4096;        % Number of points in FPGA output for one
                        % FFT accumulation accumulated measurement
NpreTRspec = 6;         % Number of pretrigger spectrums
                        % (FFT spectrums befor laser pulse)
Ntip_spec = 2;          % Number of spectrums effected by
                        % pulse reflection from fiber tip
NpreTRtip_spec = NpreTRspec + Ntip_spec;    % Number of spectrums effected
                                            % by reflections or pre trigger
PRF       = 20000;                % Pulses Repatition Frequency (Hz)
                                  % (laser pulses per second)
Tpulse    = 1/PRF;                % Pulse to pulse period (sec)
Npulseacc = 10000;                % Number of laser pulses per
                                  % accumulation output from ADC/FPGA
Tpulseacc = Npulseacc*Tpulse;     % Period (S) for Npulseacc pulses
                                  % to be accumuled (Aquisition period)
Naccs     = Time_Length/Tpulseacc; % Number of FFT acumulation outputs
                                  % by FPGA for the measurement period
Nptsfile  = Naccsize*Naccs;       % Number of points SNAP writes to
                                  % a data file for the measurement period
NptsfileBK= Naccsize*Time_LengthBK/Tpulseacc; % Number of points SNAP
                                              % write to Background data
                                              % file
No_Averages = Naccs;    % Number of pulse accumulations to average
                        % to make a wind estimate
```

```
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Query User Number of Measurements To Average for BeamX
% prompt1b = {[num2str(Naccs),' Accumulation Measurements, (', ...
%             num2str(Npulseacc),' pulses each), Will '...
%             'Be Made Durring The ',num2str(Time_Length),...
%             ' Seconds Each Beam Position Is Measured. '...
%             'How many will be averaged to determine the '...
%             'line of site wind velocity?']...
%             };
% dlg_title1b = 'Measurement Averaging';
% num_lines1b = 1;
% def1b = {num2str(No_Averages)};
% options.Resize='on';
% options.WindowStyle='normal';
% options.Interpreter='tex';
% answer1b = inputdlg(prompt1b,dlg_title1b,num_lines1b,def1b,options);
% No_Averages = str2num(answer1b[15]);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FFT related Constants and Defaults
Nfft = 128;             % Number of points input to the FPGA's FFT routine
N_pt_spec = Nfft/2;     % # of Points in a single spectrum for a single
                        % range gate output by the FPGA (i.e. Nfft/2)
N_spec_acc = Naccsize/N_pt_spec;  % Number of spectrum per accumulation
FFTbw = FsamplingM/Nfft;% Freq bandwidth of each FFT output/bin from FPGA
                        % CHECK THIS-does xilinx FFT library put the DC
                        % component in the 1st Bin???  Matlab may be
                        % different  GET Web PAGE or PDF REFERENCE

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Range Gate Size of FFT
% (Time to Take Samples for one FFT of data)*SpeedOfLight/2
RGS    = (Nfft/FsamplingM) * c/2;     % Range Gate Size in meters
Rrange = (RGS/2:RGS:RGS*N_spec_acc)'; % Radial ranges (vector)
                                      % from origin (telescope)

            % CHECK- SHOULD THIS BE POST PULES ONLY?????

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Laser Charateristics, AOM Constants and Defaults
%
% SEED LASER LOW POWER AND HIGH POWER OUTPUTS
% NP Photonics Part Number = RFLSA-500-1-1545.2-PM-S_NSL
% Oct 2008   Serial number = J10-000789-379
LambdaLO = 1545.2e-9;       % Wavelength of Seed Laser
fseed    = c/LambdaLO;      % Frequency of Laser
KW       = 2*pi/LambdaLO;   % Wave number, (angular wavenumber,
                            % circular wavenumber)
SeedLW   = 5200;            % Lorentzian Full Width Half Max (Hertz)
                            % (spec measured by manufacturer).  Full width
                            % of 104.0 kHz when 20 dB down from peak of
                            % signal.  (Se Manual- Final Inspection Report
                            % for 000780 RFLSA-500-1-1545.2-PM-S-NSI)
SeedStability = 50e6;       % Seed Laser stability over 1 hour
SeedRINpk     = -113.1;     % (dB/Hz),Relative Intensity Noise (RIN)
                            % Dominant peak at relaxation oscillation freg
                            % of laser at 2.3 MHz.  Shot noise limited
                            % greater then 50 MHz.
SeedPER       = 17;         % Seed laser Polarization Extintion Ration
                            % Minimum (dB)
```

```
SeedS_ASEi    = 35;         % Seed laser Signal to ASE Noise Ratio
                            % Intergrated Minimum (dB)
SeedS_ASE     = 55;         % Seed laser Signal to ASE Noise Ratio
                            % 50 pm, bandwidth, Minimum (dB)
SeedSMSR      = 50;         % Seed laser Side Mode Supression Ratio
                            % Minimum (dB)
SeedRINsnLP84 = -160;       % (dB/Hz), RIN of seed low power output
                            % at ~ 84Mhz, shot noise limited
SeedRINsnLP54 = -157;       % (dB/Hz), RIN of seed low power output, 54MHz
                            % shot noise limited
SeedRINsnLP114= -163;       % (dB/Hz), RIN of seed low power output, 114MHz
                            % shot noise limited
SeedRINsnHP84 = -155;       % (dB/Hz), RIN of seed high power output
                            % at ~ 84Mhz, shot noise limited
SeedRINsnHP54 = -153;       % (dB/Hz), RIN of seed high power output
                            % at ~ 84Mhz, shot noise limited
SeedRINsnHP114= -157;       % (dB/Hz), RIN of seed high power output
                            % at ~ 84Mhz, shot noise limited


thao     = 200e-9;          % Laser Pulse Width from AOMS FWHM (s)
freqAOM1 = 42e6;            % Acousto Optical Modulator (AOM) 1 freq shift
freqAOM2 = 42e6;            % Acousto Optical Modulator (AOM) 2 freq shift
fsAOM    =freqAOM1+freqAOM2;% Total freq shift to pulse by both AOMs

E        = 13e-6;           % Energy per Laser Pulse(J) Keopsys Pulse Amp

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Doppler Frequencies of Interest for Wind Measurments
FreqInterestStartUser = 46.875e6;   % Start Freq of interest
                                    % for Doppler Measurements
FreqInterestStopUser  = 125.0e6;    % Stop Freq of interest
                                    % for Doppler Measurements

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FFT spectral output Constants and Defaults
NpreTR = N_pt_spec*NpreTRspec;  % Number of points in all spectrums before
                                % laser pulse (i.e. pre trigger spectrums)

IpreTRstart = (UsepreTR-1)*N_pt_spec+1;    % Index of start of the pre
                                            % Trigger spectrum chosen
                                            % to use as background
                                            % (noise)
IpreTRend   = (UsepreTR*N_pt_spec);         % Index of end of the pre
                                            % Trigger spectrum chosen
                                            % to use as background
                                            % (noise)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Spectral Bin Indexies and Counts
% Index of START freq of spectrum range of interest
   % IpreTR_Spec_Start: Index of the frequency bin with the lowest
   % frequency which we are interested in, within a spectrum of NFFT/2
   % points.
IpreTR_Spec_Start = floor(FreqInterestStartUser/FFTbw);

% Index of Stop freq of spectrum range of interest
   % IpreTR_Spec_Stop: Index of the frequency bin within a spectrum of
   % NFFT/2 points with the highest frequency which we are interested in.
IpreTR_Spec_Stop = ceil(FreqInterestStopUser/FFTbw);

% Number of spectral bins in the freq range of interest
Nspec_bins = (IpreTR_Spec_Stop - IpreTR_Spec_Start)+1;
```

```
% Number of spectrums after pre trigger and fiber tip reflection spectrums
N_spec_acc_post_pulse = N_spec_acc - NpreTRspec - Ntip_spec;

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Constants for SNR simulation of Lidar signal based on equation 3
% of Mitsubishi paper titled compact all-fiber pulsed coherent doppler
%
% Compact all-fiber pulsed coherent Doppler lidar system for wind sensing
% S. Kameyama, T. Ando, K. Asaka, Y. Hirano, and S. Wadaka
% 10 April 2007,  Vol. 46, No. 11, APPLIED OPTICS 1953
R     = 1 ;     % reflectivity of a hard target
B     = 100e6;  % for wide band SNR use 100MHz,
                % for narrow band SNR use 2MHz Bandwidth .
Cn_sq = 2e-14;  % Refractive Index Structure Constant;
D     = 0.1;    % Effective aperture Diameter (m)
beta  = 8.3e-7; %R/(fseed*thao);%atmospheric backscatter coefficiency(/m/sr)

% The following parameter needs to be checked
K     =  0.95;  % One way atmospheric transmittance (/km)
F     = 1500;   % Focal Range of Optical Antena (m)
Eta_F = 0.9;    %0.355 ; % Far field system efficiency (-4.5 dB)
Ac    = 0.76;   % Correction Factor.
Eta_I = 0.9;    %0.676; % Factor for insertion loss of optical components
                % except for telescop (-1.7 dB)
Eta_RE = 0.977; % Telescope absorption and reflection loss (-0.1 dB)
Eta_A  = 0.91;  % Telescope reflection loss (-0.4 dB)
Eta_PP = 0.912;
Eta_Q  = 0.9;   % 0.794;

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  User/Equipment Constants, Defaults
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Users, Predefined Choices
% DL_user1 = 'moshary';
% DL_user2 = 'arend';
% DL_user3 = 'abdelazim';
% DL_user4 = 'santoro';
% DL_user5 = 'jalloh';
% DL_user6 = 'guest1';
% DL_user7 = 'guest2';
% DL_users = {DL_user1; DL_user2; DL_user3; DL_user4; ...
%                       DL_user5; DL_user6; DL_user7};
%
% % Locations, Predefined Choices
% DL_location1 = 'CCNY';
% DL_location2 = 'Liberty Science Center';
% DL_location3 = 'Brookhaven National Lab';
% DL_location4 = 'other';
% DL_locationS = {DL_location1; DL_location2; DL_location3; DL_location4};

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Truck Cardinal Directions (North, South, East, West) and Tilt from Level
Truck_tiltZE = 0;% Zenithe angle = 0 if laser beam is verticle
                 % (and Truck optical table is level)
                 % If the edge of the optical table closest to the
                 % fron of the truck is lower than the egde of
                 % the optical table nearest the rear of the truck
                 % ZE will have a POSITIVE value, and a NEGATIVE value
                 % if the rear edge is lower than the front edge.
                 % If truck has right side of optical table lower than the
```

```
                % left side of the optical table then __________
                % Have to come up with a framework - don;t worry now

Truck_cardianlAZ = 299.33;    % Azmith angle = 0 if the direction of
                              % Truck's optical table taken along the
                              % long edge (length) is North.
                         % North is 0 degrees, East is 90 degrees,
                         % South is 180 degress and West is 270 degrees

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Software Versions Being Used - for Meta Data
%
% Matlab Information about curret .m file running
MlabFname      = mfilename;         % Matlab Filename running without '.m'
MlabFpath_name = mfilename('fullpath'); % Path and Filename of running .m

% FPGA Firmware version being used
X5fpgaLogicPath_Name = ['D:\TEMP_SAMEH\Simulink_Works\Accumulated_FFTs\'...
                        'FFT_10K_Accum_Reset_7_hwcosim_module.mdl'];
% SNAP Version being Used
SNAPverPath_Name     = ['C:\Innovative\X5-400M\Examples\'...
                            'Snap_07282011\FPGA_SNAP\Debug_x64'];

% X5-400M SNAP Default Settings
X5frameSize     = 8192;% Frame Size in decimal
X5frameSizeBK   = 8192;% Frame Size in decimal (BACKGROUND MEASUREMENT)
X5packetSize    = 8192;% Packet Size in decimal
X5packetSizeBK  = 8192;% Packet Size in decimal (BACKGROUND MEASUREMENT)
X5ch            = 0;   % ADC channel to use for aquizition
X5BytesPerSamp  = 4;   % Because FPGA FFT uses 32nits per sample
X5samples00m01s = (X5BytesPerSamp*SampPerPulse*PRF)/Npulseacc;
                       % Number of samples to set for 0 minutez, 1 second
                       % of FFT spectrums, output every 500ms
X5samples       = Time_Length*X5samples00m01s;   % # SNAP Samples to take
                                                 % for BeamX measurments
X5samplesBK     = Time_LengthBK*X5samples00m01s; % Number of SNAP Samples
                                                 % to take for background

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% WRITE SNAP INIZILIZATION DEFAULT FILE HERE
% variables that are default to the SNAP initilization
% TO BE DONE BEFORE STARTING THE SNAP PROGRAM - will help make sure the
% setting s will be correct- SHOULD CHECK TO SEE THAT THE HsData.bdd
% does not already exist- Will not need to write that file every time
% because the exisitance of the file means the user already set up SNAP
% the way they wanted it to run.
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Lasers' Operational Settings
KeopsysStage1 = 0.170;  % Keopsys 1st Stage Driving Current (A)
KeopsysStage2 = 1.700;  % Keopsys 2nd Stage Driving Current (A)
NP_piezeo     = 0;      % NP Photonics Piezeio Voltage Setting (V)(0=auto)
NP_hiP        = 0.500;  % NP Photonics High Power Output Settinf (W)

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Dummy Data Prototypes to Be Used for Backgrounds (noise) and Data

% OPTION 1: Waveform Based Fake Data Prototype
% WfStart1 = 1-(N_pt_spec/2);
% WfEnd1   = (N_pt_spec/2)-1;
% WfWidth1 = ceil(N_pt_spec*1.09);
```

```
% WfSkew1  = -0.2;
%
% WfStart2 = 2-(N_pt_spec/2);
% WfEnd2   = (N_pt_spec/2);
% WfWidth2 = ceil(N_pt_spec*0.78);
% WfSkew2  = 0.1;
%
% dummy_data_Proto = [1,(tripuls(WfStart1:WfEnd1,WfWidth1,WfSkew1) - ...
%                       0.63*tripuls(WfStart2:WfEnd2,WfWidth2,WfSkew2))]';
% OPTION 2: Add some if needed

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Maximum Likelyhood Estimation Default Parameters
SNR_Thresh       = 0.0000125; % SNR decision threshold for a measured value
No_Fitted_Points = 3;     % No of spectrum points to be fitted in MLE

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Cell Array Result Storage Indeies
MeasNo_i         = 1;
Range_i          = 2;
TimeOfCal_i      = 3;
TimeDelta_1      = 4;
Beam0Vel_i       = 5;
Beam0Int_i       = 6;
Beam0Time_i      = 7;
Beam1Vel_i       = 8;
Beam1Int_i       = 9;
Beam1Time_i      = 10;
Beam2Vel_i       = 11;
Beam2Int_i       = 12;
Beam2Time_i      = 13;
WindVelEAST_i    = 14;
WindVelNORTH_i   = 15;
WindVelVert_i    = 16;
WindSpeedHoriz_i = 17;
WindHeading_i    = 18;
WindCalcConf_1   = 19;




%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% QUERY User if they have setup FPGA Firmware, and Copied Modified
% SNAP to the Destination Directory
% qf = warndlg({'(1) Have You Uploaded the FPGA Firmware? ';...
%         X5fpgaLogicPath_Name;'';...
%         '(2) Started the Modified SNAP copied from ';...
%         SNAPverPath_Name;...
%         'to the Data Destination Directory You Created?';'';...
%         ['If No to Either Question, Quit MatLab By Pressing '...
%         'ctrl + C, And Do So NOW !!!']...
%         },'','normal');
% waitfor(qf); % Force execution to halt until user presses OK Button

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Tell User to start modified SNAP for BACKGROUND measurement (noise
% spectrum)
% gf = warndlg({'SNAP Settings To Measure Background (Noise) Spectrum';'';...
%             'Start the Modified SNAP - X5 Capture Application';...
```

```
%             'CONFIGURE TAB';...
%             'Set Busmaster Size to 24 MB, Open the Card';'';...
%             'SETUP TAB';...
%             ['Clock Sourse = Internal, 400.0 Freq (MHz)'];...
%             ['Packet Size to ' num2str(X5packetSize) ,', 0x',...
%             dec2hex(X5packetSize), ' in Hex'];...
%             ['Uncheck all Communications Properties'];'';...
%             ['Channels = Ch0, Trigger Source = External, De-activate Auto
Trigger'];...
%             ['Frame Size to ' num2str(X5frameSize) ,', 0x',...
%             dec2hex(X5frameSize), ' in Hex'];...
%             ['Set Data Logging Ceiling to ', num2str(X5samplesBK), ' Uncheck Auto
Stop'];...
%             },'','normal');
% waitfor(gf); % Force execution to halt until user presses OK Button

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Truck Cardinal Directions (North, South, East, West) and Tilt from Level
%
% Truck_tiltZE:
% Zenithe angle = 0 if laser beam is verticle (Truck optical table is
% level) If the edge of the optical table closest to the front of the truck
% is lower than the egde of the optical table nearest the rear of the truck
% ZE will have a POSITIVE value, and a NEGATIVE value if the rear edge is
% lower than the front edge. ( Or if table is tilted back have Zieneth
% be positive and add 180 degrees to Azmith ??)
% If truck has the right side of optical table lower than the left side of
% the optical table then __________  Have to come up with a framework to
% decide what to do- don;t worry now
% Any time Truck_tiltZE NOT = Zero the Zenith and Azmith angles used to
% position the beam for a measuremnets relative to the truck's optical
% table will have to be adjusted to make sure beam is going towards the
% correct location in space.

% Truck_cardianlAZ:
% Azmith angle = 0 if the direction of Truck's optical table taken along
% the long edge (length) is North.  North is 0 degrees, East is 90 degrees,
% South is 180 degress and West is 270 degrees

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Query User for: Truck's Cardinal Orientation (Front faces 0 to 359 deg)
% and Truck's Optical Table Tilt from Level
%
% NOTE: Magnetic NORTH is approzimatley 13 deg 10 min off towards EAST
%       in the NYC area.  True North is NOT Magnetic North
prompt3 = {['Research Van Heading (Front Points Toward): North= 0 deg'...
            ' [Heading is Magnetic Heading +13.8 degrees]'],...
           'Optical Table Forward Tilt from Level (degrees):'};
dlg_title3 = 'Research Van Orientation (Cardinal Orientation/Tilt)';
num_lines3 = 1;
def3       = {num2str(Truck_cardianlAZ),num2str(Truck_tiltZE)};
options.Resize      = 'on';
options.WindowStyle = 'normal';
options.Interpreter = 'tex';
answer3 = inputdlg(prompt3,dlg_title3,num_lines3,def3,options);
AZ                  = str2num(answer3[15]);  % Cardinal Heading of Truck
ZE                  = str2num(answer3{2});  % Tilt of Truck

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Users Enters the Tilt (Zieneth Angle) of the Three Laser Beams
Truck_ZE0 = 0;     % Beam #0 Tilt (zenith) relative to optical table
Truck_ZE1 = 30;    % Beam #1 Tilt (zenith) relative to optical table
```

```
Truck_ZE2 = 30;    % Beam #2 Tilt (zenith) relative to optical table

prompt4 = {...
            ['Beam0: Zenith Angle (Tilt Relative to ',...
                    'Optical Table) of Verical Beam, degrees:'],...
            ['Beam1: Zenith Angle (Tilt Relative to ',...
                    'Optical Table) of North Beam, degrees:'],...
            ['Beam2: Zenith Angle (Tilt Relative to ',...
                    'Optical Table) of East Beam, degrees:']};
dlg_title4 = 'Zenith Angle of Beams Referenced to Optical Table';
num_lines4 = 1;
def4 = {num2str(Truck_ZE0),num2str(Truck_ZE1),num2str(Truck_ZE2)};
answer4 = inputdlg(prompt4,dlg_title4,num_lines4,def4);
ZE0 = str2num(answer4[15]);  % Zenith angle of verticle beam (Beam0)
ZE1 = str2num(answer4{2});  % Zenith angle of North-ish beam (Beam1)
ZE2 = str2num(answer4{3});  % Zenith angle of East-ish beam (Beam2)

Truck_relative_ZE = [ZE0 ZE1 ZE2];
ZE0offset = ZE0 + ZE;% Laser Beam #0, verticle
                     % Difference of Vector V0 zenith
                     % angle from earth normal
ZE1offset = ZE1 + ZE;% These would be used if multiple laser beams are used
ZE2offset = ZE2 + ZE;% These are the tilts of the laser beams relative to
                     % the optical table
ZEoffset  = [ZE0offset ZE1offset ZE2offset];

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Query User for the Azmith Angle of the 3 Laser Beams
%
% NOTE: Magnetic NORTH is NOT True North.
%       The Magnetic Decllination is 13 degrees 10 sec West
%       Declination is negative, Inclination is 67 degrees 13 minutes
%       Magnetic Field Strength is 52497.8 nT
%       City College van location
%       Latitude:  40 deg 49min 16.6 sec North
%       Longitude: 73 deg 56min 54.6 sec West
%       Convent Ave is heading 28.9 deg from True North
%       Convent Ave is heading 43.0 deg from Magnetic North
%       Van Heading is 299.33 degrees at City College
Truck_AZ0 = 0;  % Beam #0 deg from North (azmith) relative to optical table
                % (verticle has no compass directional
                % component if verticle)
Truck_AZ1 =120; % Beam #1 degrees from North (azmith)
                % relative to optical table
Truck_AZ2 =240;% Beam #2 degrees from East (azmith)
                % relative to optical table
prompt5 = {...
    ['Beam0: Azmith Angle (Heading Relative to Front of Optical Table)'...
        ' of Verical Beam, degrees:'],...
    ['Beam1: Azmith Angle (Heading Relative to Front of Optical Table)',...
        ' of North Beam, degrees:'],...
    ['Beam2: Azmith Angle (Heading Relative to Front of Optical Table)',...
        ' of East Beam, degrees:']};
dlg_title5 = 'Heading Angle of Beams Referenced to Front of Optical Table';
num_lines5 = 1;
def5 = {num2str(Truck_AZ0),num2str(Truck_AZ1),num2str(Truck_AZ2)};
answer5 = inputdlg(prompt5,dlg_title5,num_lines5,def5);
AZ0 = str2num(answer5[15]);  % Azmith angle of verticle beam (Beam0)
AZ1 = str2num(answer5{2});  % Azmith angle of North-ish beam (Beam1)
AZ2 = str2num(answer5{3});  % Azmith angle of East-ish beam (Beam2)

Truck_relative_AZ = [AZ0 AZ1 AZ2];
```

```
AZ0offset = AZ0 + AZ;   % Beam #0 should have no bearing as it should be
                        % verticle if the optical table was level
AZ1offset = mod(AZ1+AZ,360);    % Beam 1 is usually North
                                % and Beam 2 is usually East
AZ2offset = mod(AZ2+AZ,360);    % but YOU MUST look at the Azimith
                                % ZE0 angles to be sure
AZoffset     = [AZ0offset AZ1offset AZ2offset];
V0V1V2_files = {'none','none','none'};  % Cells of last three data files
                                        % used to find wind vectors
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% For the Spectrum of Interest (SOI) calculate the
% index within the pre Trigger spectrums for the start and end of SOI
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
IpreTRstart_SOI = IpreTRstart     + IpreTR_Spec_Start - 1;
            % IpreTRstart_SOI: Index of the pre trigger spectrum (spectrum
            % prec ceding the fiber tip back reflected pulse) to be used as
            % the Background (NOISE) of the current pulse having the lowest
            % frequency.
IpreTRend_SOI   = IpreTRstart_SOI + Nspec_bins - 1;
            % IpreTRend_SOI: Index of the pre trigger spectrum (spectrum
            % prec ceding the fiber tip back reflected pulse) to be used as
            % the Background (NOISE) of the current pulse having the
            % highest frequency.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Directory and File Name Initialize Information From User
%
% Have User Confirm Path to Directory and Approximate File Name
% Since names will be based on the date/time at the start of the run
% an approximate name will be a good starting point to find the
% first data file writen to the directory or the newest file after
% a long break in taking data (given not past midnight)
file_path = uigetdir(file_path,...
                     'Find/Create Directory Holding Beam Velocity Files');

% Have User Confirm Approximate Name of the First Data (SPECTRUM) File
prompt = {'If Path to Beam Velocity Files is correct:',...
        ['Enter Approximate (model) Name of First Beam Velocity File'...
        ' yyyymmddTHHMMSS_BeamXVel (no extention):']};
name = 'Input Approximate Name (yyyymmddTHHMMSS) of 1st Beam Velocity Files';
promptWidth = ceil(max(length(file_path),length(file_name))*1.25);
numlines = [1, promptWidth; 1, promptWidth];  % Input Lines, Input Width
defaultanswer       = {file_path,file_name};
options.Resize      = 'on';
options.WindowStyle = 'normal';
answer = inputdlg(prompt,name,numlines,defaultanswer,options);

file_path      = (cell2mat(answer(1)));
file_name      = (cell2mat(answer(2)));
file_path_name = fullfile(file_path, file_name);

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Query Users to Select Names of Operators Taking Data
% Isel= the index to the selected item(s)
% SelEr is 0 if no selection is made, or 1 if a selection is made.

% Isel = length(DL_users);    % Last user in list of users is default user
% [Isel,SelEr] = listdlg('PromptString','Select Operators (ctrl for multiple):',...
```

```
%                 'SelectionMode','multiple',...
%                 'Name','Current Operators',...
%                 'InitialValue',length(DL_users),...
%                 'CancelString','No User Name',...
%                 'ListSize',[200 120],...
%                 'ListString',DL_users);
% DLoperators = DL_users(Isel);
%
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Query User to Select Location
% Isel = the index to the selected item
% SelEr is 0 if no selection is made, or 1 if a selection is made.
% Isel = 1;       % 1st location in list is default
% [Isel,SelEr] = listdlg('PromptString','Select Location:',...
%                 'SelectionMode','single',...
%                 'Name', 'Location',...
%                 'PromptString', 'Select Current Location',...
%                 'InitialValue',1,...
%                 'CancelString','No Selection',...
%                 'ListSize',[200 80],...
%                 'ListString',DL_locationS);
% DL_location = DL_locationS(Isel);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Have User Check Camera Image of Sky
% Record Image of Sky YES/NO
% YES--> Display Image to See if alignment is OK
% Superimpose
% front, rear, right, left
% Red dot at the NORTH position
% north, east south, west

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Query User to Enter Number of Pre Trigger Spectrums In The Data
% User Enters Which Pre Trigger Spectrums to Use as Background Noise
% User Enters Number of Spectrums Effected by Out Going Pulse
prompt6 = {...
    ['Number of Pre-Trigger/Tip Reflection Spectrums Seen in FFT Data'],...
    ['Use the Nth Pre-Trigger Spectrum to Calculate '...
    'Background (Noise) Spectrum For Each Measurement Set']};
dlg_title6 = 'PRE-TRIGGER SPECTRUMS';
num_lines6 = 1;
def6       = {num2str(NpreTRspec),num2str(UsepreTR),num2str(Truck_AZ2)};
answer6    = inputdlg(prompt6,dlg_title6,num_lines6,def6);

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Get Contents of Current Directory Excluding none file items
%
dir_data  = dir([file_path,filesep]);% Get complete contents of current dir
dir_index = [dir_data.isdir];       % Check to see if entry is a directory
file_list = {dir_data(~dir_index).name}'; % Make list of file names
                                          % excluding directories
% Finding Earliest Modified File in Directory of Saved Data may NOT work as
% the spectrum txt files, SNAP and other files will be included.
% Need to selectively make list xxxxxxxxTxxxxxx_BeamXVel.txt files
file_list_Vel = dir([file_path,filesep,'*Beam*Vel.txt']);
file_list_Vel_all_BeamVel = [];
for dd = 1: numel(file_list_Vel)
    file_list_Vel_all_BeamVel = [file_list_Vel_all_BeamVel; file_list_Vel(dd).name];
end
N_BeamVelfiles = size(file_list_Vel_all_BeamVel,1); % Number of files of Beam
Velocities
```

```
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Data Format to be placed in Cell Array
%  1)Largest Beam Measuremnt Number used in Current Calculations
%  2)Range (meters)
%  3)Time Last Measurment used in calculations
%  4)Delta Time (Time Last Measurment - Time Earliest Measurment)
%-----
%  5)Velocity of Beam0 used in Calculations (m/s)
%  6)Intensity of Peak used to calculate Velocity of Beam0
%  7)Time of Beam0 used in Calculations (yyyymmddThhhmmmss)
%-----
%  8)Velocity of Beam1 used in Calculations (m/s)
%  9)Intensity of Peak used to calculate Velocity of Beam1
% 10)Time of Beam1 used in Calculations (yyyymmddThhhmmmss)
%-----
% 11)Velocity of Beam2 used in Calculations (m/s)
% 12)Intensity of Peak used to calculate Velocity of Beam1
% 13)Time of Beam2 used in Calculations (yyyymmddThhhmmmss)
%-----
% 14)Horizontal Wind Velocity EAST Component (m/s)
% 15)Horizontal Wind Velocity NORTH Component (m/s)
% 16)Verticle Wind Velocity Component (m/s)
%-----
% 17)Wind Speed (Horizontal) (m/s)
% 18)Wind Heading (Horizontal) (degrees)
WindVecArrayHead =['Meas.No. ', 'Range(m) ','TimeLast ','DeltaT ',...
                   'VelBeam0 ','PeakInstenity0 ','TimeBeam0 ',...
                   'VelBeam1 ','PeakInstenity1 ','TimeBeam1 ',...
                   'VelBeam2 ','PeakInstenity2 ','TimeBeam2 ',...
                   'HorVelEAST ','HorVelNORTH ','VerVel ',...
                   'WindSpeedH ','WindHeadingH '];
% Create Cell Array to hold data that will be calculated bellow
WindVecData01 = cell(1,18,N_BeamVelfiles);



file_time_0 = [];
file_time_1 = [];
file_time_2 = [];

VrBeam0 = [];
VrBeam1 = [];
VrBeam2 = [];

V0 = [];
V1 = [];
V2 = [];

HorSpeed = [];
HorAZ    = [];
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Read Data Files and Make Calculations

iBeam = 1;
for iBeam = 1:N_BeamVelfiles
    % Read Beam Velocities
    fidVel = fopen(fullfile(file_path,file_list_Vel_all_BeamVel(iBeam,:)), 'r');
    Vrbeam = fscanf(fidVel,'%f ',N_spec_acc);
    fclose(fidVel);

    file_time_new = file_list_Vel_all_BeamVel(iBeam,1:15); % Get Date/Time in file
name
```

```
switch mod(iBeam-1,3)% Time of last three Beams used to find Wind Vector (0|1|2)
    case 0
        file_time_0 = file_time_new;
    case 1
        file_time_1 = file_time_new;
    case 2
        file_time_2 = file_time_new;
end



% file_time          = {dir_data(~dir_index).datenum}';% List file times
% first_file_time    = min(cell2mat(file_time));       % Oldest time of files
% first_file_timeTmp = first_file_time;     % Initial time of new file temp
% first_time_index = find(cell2mat(file_time)==first_file_time);
% % Find the index of the newest file time
% first_time_name  = cell2mat(file_list(first_time_index));
% % Get the oldest file time
%
% [path_string,file_name_wo_ext,file_ext]= fileparts(first_time_name);
% % Get the parts of the file name
% % path_string      = String represneting path to file
% % file_name_wo_ext = file name without the extension
% % file_ext         = file extention
%
% % Change Name of File, Append Suffix (i.e. '_BK') for Background to name
% first_path_file     = fullfile(file_path,first_time_name);
% first_path_file_new = fullfile(file_path,[file_name_wo_ext,....
%     file_name_suffix_Background,file_ext]);



% % Record The Names of Last Three Data File for Beam0, Beam 1, and Beam2
% [LastData_Path, LastData_Name, LastData_Ext] = fileparts(data_file);
% V0V1V2_files{mod(iBeam-1,3)+1} = [LastData_Name, LastData_Ext];



%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Dr. Arend Plot request from 11/21/11
%  Dr. Arend wants a plot that is updated every time SNAP saves a
%  measurement to the hard drive and data is processed.
%  He wants to have the plot of Intensity vs Range updated as the program
%  runs so the operator can make adjustments to the fiber position.
%  CHECK THIS IDEA TO:
%  plot last 3 results using axis limits of current data
%  Dr. Arend request, 11/28/11, horizontal axis be in meters instead of
%  range gates. Range Gate Size (RGS) currently 48m.
%  ---- Perhaps two horizontal axis can be created (range gates and meters)
% figure(1)
% plot(1:length(Power_Vector(NpreTRtip_spec+1:end)),...
%                                    Power_Vector(NpreTRtip_spec+1:end))
% grid on
% title(['Max Signal Intensity (Corected for Background) vs Range Gate'])
% xlabel(['Range Gate(',num2str(RGS,'%4.1f'),'m Each), After ',...
%         num2str(NpreTRspec),' Pre-trigger,',...
%         ' and ',num2str(Ntip_spec),' Tip Back Reflection Gates'])
% ylabel('Intensity')
%
% % Put 2nd X-axis (meters) on plot
% hl1 = line(1:length(Power_Vector(NpreTRtip_spec+1:end)),...
```

```
    %                         Power_Vector(NpreTRtip_spec+1:end),'Color','b');
    % ax1 = gca;
    % set(ax1,'XColor','b','YColor','b')
    % ax2 = axes('Position',get(ax1,'Position'),...
    %            'XAxisLocation','top',...
    %            'YAxisLocation','right',...
    %            'Color','none',...
    %            'XColor','k','YColor','k');
    % % To draw a and line on same graph
    % hl2 = line((1:length(Power_Vector(NpreTRtip_spec+1:end)))*RGS,...
    %             Power_Vector(NpreTRtip_spec+1:end),'Color','k','Parent',ax2);
    %

    %% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Save Wind Velocity Array for BeamX as yyyymmddThhmmss.BeamX.txt Format
    %
    % Suffix String for BeamX velocity file
    % BeamX = [file_name_suffix_BeamX,num2str(mod(iBeam-1,3)),'Vel'];
    %
    % % Vel_Freq_Shift_Est is the BeamX Line of Sight Wind Velocity
    % Vel_Est_BeamX_file = fullfile(LastData_Path,...
    %                               [LastData_Name,BeamX,LastData_Ext]);
    % fid_Vel_Est = fopen(Vel_Est_BeamX_file, 'w');
    % %fid_Vel_Est = fopen(Vel_Est_BeamX_file, 'w','US-ASCII'); % character encoding
scheme
    % fprintf(fid_Vel_Est,'%6.4f ',Vel_Freq_Shift_Est);  % Write
    % fprintf(fid_Vel_Est,'\n');                % Write Cariage Return to File
    % fclose(fid_Vel_Est);

    % SAVE NAME OF DATA FILES USED TO MAKE THE CURRENT VEL VECT CAL
    % REPLACE MLE_Freq_Shift with Vel_Freq_Shift_Est
    % and MLE_Precision with Precision_Est

    % AZTruck_relativeX = num2str(Truck_relative_AZ(mod(iBeam-1,3)+1));
    % ZETruck_relativeX = num2str(Truck_relative_ZE(mod(iBeam-1,3)+1));
    % AZTruck_cardinalX = num2str(Truck_cardianlAZ);
    % ZETruck_cardinalX = num2str(Truck_tiltZE);
    % AZoffsetX = num2str(AZoffset(mod(iBeam-1,3)+1));
    % ZEoffsetX = num2str(ZEoffset(mod(iBeam-1,3)+1));

    % Append BeamX Velocity, Time, File Name, Comments Info to Meta File
    % Prepare Data to be Saved to Meta File
    % Can NOT PUT TEXT AND NUMBERS IN SAME ARRAY--USE CELLS?
    % Meta_BeamX_Vel_pre = ...
    %     {datestr(last_file_time,time_format),...
    %     [file_name_wo_ext,file_ext],...
    %     ['Beam',num2str(mod(iBeam-1,3))],...
    %     ['TruckAZ=',num2str(Truck_relative_AZ(mod(iBeam-1,3)+1))],...
    %     ['TruckZE=',num2str(Truck_relative_ZE(mod(iBeam-1,3)+1))],...
    %     cell2mat(answer2),...
    %     ['TruckCardinalAZ=',AZTruck_cardinalX],...
    %     ['TruckCardinalZE=',ZETruck_cardinalX],...
    %     ['TruckOffsetAZ=',AZoffsetX],...
    %     ['TruckOffsetZE=',ZEoffsetX]...
    %         };
    %
    % %If function 'ex_func' is not available use a loop
    % Meta_header = Meta_BeamX_Vel_pre;
    % for k = 1:length(Meta_header)
    %     in_datatype = class(Meta_header{k});                %<--------------------
fix
    %     switch in_datatype
```

```
    %     case 'char'  %Cell is a string, do nothing
    %     case 'double'
    %         Meta_header{k} = num2str(Meta_header{k});
    %     end
    % end

    %% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Write Metafile Data to File in Directory for BeamX  <-----------------------------
----------
    % The 1st Line of the Text File Will Have Info That Will Not Be Repeated
    % fid_M = fopen(fullfile(file_path_M,file_name_M), 'a');
    % fprintf(fid_M,'%s, ', Meta_background{:}); % Write Comma Seperated
    % fprintf(fid_M,'\n'); % Write Cariage Return to File
    % fclose(fid_M);

    %% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Calculate Wind VECTOR for Last Three BeamX Vector Measurements
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % This program;
    % INPUTS: Three raadial vectors of wind velocity from each laser beam
    %                                    (VrBeam0, VrBeam1,VrBeam2)
    % Azimuth Angles (AZ0, AZ1, AZ2)  degrees: Referenced from NORTH (0 deg)
    % Zenith Angles  (ZE0, ZE1, ZE2)  degrees: Referenced from
    %                                          Normal to the Earth Center
    % Range Gate Size (RGS)
    % OUTPUTS: Calculates the resultant wind vectors

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %
    % Wind VECTOR Azimuth, direction TOWARDS which the wind is blowing TO:
    % Phi_VEC.
    % METEOROLOGICAL WIND DIRECTION is direction FROM which the wind blows
    % Phi_MET
    %
    % Phi_MET = Phi_VEC + 180
    %
    % Unit vectors
    % iii = unit vector in the NORTH
    % jjj = unit vector in the EAST
    % kkk = unit vector in the VERTICLE
    % rrr = unit vector in the RADIAL DIRECTION
    %
    % Wind Vectors use right-hand-rule with thumb pointing EAST
    %
    % thetaZE, laser beam directed at zenith angle theta
    % phiAZ, laser beam directed at azimuth angle phi
    % Vrbeam, the radial component of the wind vector along laser beam
    % u, zonal      (towards East)
    % v, meriodonal (towards North)
    % w, vertical

    switch mod(iBeam-1,3)    % BeamX will be used for Wind Vector (0|1|2)
        case 0
            VrBeam0 = Vrbeam;
        case 1
            VrBeam1 = Vrbeam;
        case 2
            VrBeam2 = Vrbeam;
    end
```

```
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    if iBeam >= 3 % Calculate Wind Vectors After 3 Vectors are Measured
        % wind_calculate_function from Dr. Gross (12/21/11)
        % wm1,wm2,wm3 are the 3 wind measurments
        % Angles are in radians
        % gmrot = z axis rotation of mirror that is tilted at 45 degrees
        %         "+" gama (Counter Clockwise) = wind measurement 1 (wm1)
        %         "-" gama         (Clockwise) = wind measurement 2 (wm2)
        % phic = azimuth rotation of the optical table (van 'lab'
        %        coordinates) in the real world (cardinal coordinates).
        %        to geo-coordinates (E,N,V). The angle between the laser
        %        (defines as direction 1) and true East.
        % Before the heading of the van (the front of the van is pointing
        % towards a cardinal driection) is corrected for it is assumed that
        % the laser beam comes from the "East" and goes to the "West" along
        % the u axis and the East Wind Vector, North Wind Vector and
        % Verticle Wind Vector are calculated based on the laser beam
        % comming from the East.  If the front of the Research Van was
        % facing East (90 Degrees) then the laser beam would be coming
        % from the East in the Van's lab coordinate system and the real
        % world (cardinal) coordinate system. No adjustment would be
        % nessasary and phic would be 0.
        %
%         wm1=VrBeam1;  % wm1 is X direction (1st Horiz Measurement)
%         wm2=VrBeam2;  % wm2 is Y direction (2nd Horiz Measurement)
%         wm3=VrBeam0;  % wm3 is Z direction
%         wm1=VrBeam1;  % wm1 is X direction (CCW rotation = + Gama Rot)
%         wm2=VrBeam2;  % wm2 is Y direction (CW roatation = - Gama Rot)
%         wm3=VrBeam0;  % wm3 is Z direction

% function [we,wn,wv]=wind_calculate_function2(wm1,wm2,wm3,gmrot,phic)
% 01/04/12 Reworked version of function
% wm1,wm2,wm3 are the 3 wind measurments along
%                              Beam1 (u), Beam2 (v) and Beam0 (w)
% Angles are in radians
% gmrot= z axis rotation of mirror to get beam tilting (set to approx 30))
% phic = azimuth rotation of the optical table (lab) coordinates to geo
% coordinates (E,N,V).
% Clockwise rotation is "-" angle rotation,
% Counter-clockwise is "+" angle rotation.
% The angle is between the laser (defines as direction 1) and true East?.
% The laser beam approches the mirror from the EAST (90 deg in cardianl
% coordinates but 270 deg (or -90 deg) in this coordinate system).

% Orientation of Van/Optical Table to Coordinate System
% The laser beam moves from the front of the van towards the rear of the
% van along the optical table.
% The mirror is rotated by Gama Rotation (gmrot) in the "+" (clockwise) and
% "-" (counter clockwsie) angles as viewed from above the optical table.
% gmtot = 0     ----> wm3, Verticle
% gmtot = "+" X ----> wm2, Beam1
% gmtot = "-" X ----> wm1, Beam2

% A "+" wind velocity measurement is wind moving AWAY from the lidar
% A "-" wind velocity measurement is wind moving TOWARDS the lidar
%
% Rotation of van has "+" angles as counter clockwise.
% Rotation of van has "-" angles as clockwise.

        % THIS DEPENDS ON WHICH WAY THE MIRROR WAS TILTED FIRST CW or CCW
```

```
        wm1=VrBeam1;  % wm1 is X direction (+ Gama Rot, CCW)
        wm2=VrBeam2;  % wm2 is Y direction (- Gama Rot, CW)
        wm3=VrBeam0;  % wm3 is Z direction
        gmrot=29.4*pi/180;  %<------------------- Move up for user to enter

        % Cardianl Coordinates
        % North = 0  , East = 90
        % South = 180, West = 270
        % Cardinal Degrees INCREASE in the CLOCKWISE direction.
        % This is the opposite of how the angles are handled in the
        % wind_calculation function.
%         Truck_cardianlAZ_gross = 270 - Truck_cardianlAZ -180;
%         phic = (Truck_cardianlAZ_gross)*pi/180; %

% wind_calculate_function2 Moshary
% phic is + for CCW rotation
% Assumes the normal to the mirror with gmrot = 0
% to be pointing in the X direction (EAST)
% If the front of the Van were facing EAST then this would be mean no
% additional rotation was needed.
% If the Van were faciung SOUTH (180 deg cardianal directions) then a
% rotation of 180 -90 = +90 would be needed to align van's coordinates with
% the cardinal coordinates.
%       phic = (360 - (Truck_cardianlAZ+90))*pi/180; % For Clockwise Rotation
%        phic = (Truck_cardianlAZ - 90)*pi/180; % For Counter Clockwise Rotation
%         if ((Truck_cardianlAZ <= 90) && (Truck_cardianlAZ >= 0))
%             phic = (90 - Truck_cardianlAZ)*pi/180;
%         elseif ((Truck_cardianlAZ > 90) && (Truck_cardianlAZ <= 360))
%             phic = (90 - Truck_cardianlAZ)*pi/180;
%         end
        phic = (90 - Truck_cardianlAZ)*pi/180; % For Counter Clockwise Rotation


        % FILTER THE 999 error code placed in vectors with Not a Number
        % (NaN) to feed to the wind calculation function.
        for ww = 1:length(N_spec_acc)
            if wm1(ww)== 999, wm1(ww)=NaN; end % Make all missing data NaN
            if wm2(ww)== 999, wm2(ww)=NaN; end
            if wm3(ww)== 999, wm3(ww)=NaN; end
        end

        % Calculate Wind Vectors for East, North, Verticle
        % wm1 is counter clockwise (+ Gama Rot) rotation of mirror (Beam1)
        %
        % wm2 is clockwise (- Gama Rot) rotation of mirror (Beam2)
        % From a TOP VIEW wm1 is counter clockwise of wm2
        % RIGHT HAND RULE- Thumb: wm1 (Beam1), Pointer Finger: wm2 (Beam2)
        %                  - Middle Finger: wm3 (Beam0)
        for ww = 1:N_spec_acc
            [we(ww),wn(ww),wv(ww)] = wind_calculate_function2...
                                     (wm1(ww),wm2(ww),wm3(ww),gmrot,phic);
        end
        V0 = wv;
        V1 = we;
        V2 = wn;
        % Take vectors, East(V1) (x-axis) and North(V2) (y-axis) to find
        % the direction the wind is going towards
        % atan2(Y/X) : Four-quadrant inverse tangent (arctangent)
        HorAZ =  mod((360 + atan2(V2,V1)*180/pi),360);  % direction and
        HorSpeed = sqrt(V1.^2 + V2.^2);                 % speed of wind
```

```
%     V0V1V2 = [V0;V1;V2];  % Output matrix: Vert, North, East Componets  <-----------
-----VO is only one number (999)


        %% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % Print to Screen: Wind Velcity Vectors for u, v, w for 1000 to 1500m

        % Use 'disp' to supress Matlab ptinting 'Var = ', new line the values
        % Range (Meters),
        % Horizontal Components (Vector EAST, Vector NORTH, Vector UP)
        % Horizontal SPEED and DIRECTION (wind is goinng TOWARDS)
        disp('    Range     Vu=Ve     Vv=Vn     Vw=Vup    SpeedH    DirH')
        %disp(gallery('normaldata',[8 6],0));% 'gallery' makes fake test data
        if iBeam == 1
            V0 =  VrBeam0;
            % Only Print first vector
            % rtemp = sprintf('%5.0f \n',(20:32)'*RGS); % try to format
            disp([(20:32)'*RGS, V0(20:32)'])
        elseif iBeam==2
            % V1 = (VrBeam1 - VrBeam0.*cos(ZE1offset)) ./sin(ZE1offset);
            % Only Print 1st Two Vectors
            disp([(20:32)'*RGS,V1(20:32)', V0(20:32)'])
        elseif iBeam>=3
            % Print All Vectors and Calculations
            disp([(8:32)'*RGS,V2(8:32)' V1(8:32)' V0(8:32)' ...
                HorSpeed(8:32)' HorAZ(8:32)']) % WHYS is HorAZ col vec? <-------------
--???
        end

    end
    %% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Calculations to Fill Cell Array
    %-----
    % Calculate Range Vector
    Range_vec = RGS*([zeros(NpreTRtip_spec,1);...
                                          (1:N_spec_acc-NpreTRtip_spec)']);
    %-----
    % Calculate Delta Time
    t0        = datenum(datevec(file_time_0,'yyyymmddTHHMMSS'));
    t1        = datenum(datevec(file_time_1,'yyyymmddTHHMMSS'));
    t2        = datenum(datevec(file_time_2,'yyyymmddTHHMMSS'));
    t012      = [t0, t1, t2];
    TimeMax_i = find(t012==max(t012));
    TimeMin_i = find(t012==min(t012));
    if sum(isnan(t012))==0  % If there are no NaN (not a number)
       deltat = t012(TimeMax_i) - t012(TimeMin_i);    % Cal delta time
       [dtY dtMo dtD dtH dtM dtS] = datevec(deltat);
       deltatS = dtH*3600 + dtM*60 + dtS;   % Delta Time in Seconds
       deltaS_vec = deltatS*ones(N_spec_acc,1);
    end



    %% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     Place Data Calculated in Cell Array
%      1)Largest Beam Measuremnt Number used in Current Calculations
%      2)Range (meters)
%      3)Time Last Measurment used in calculations
%      4)Delta Time (sec) (Time Last Measurment - Time Earliest Measurment)
```

```
%     -----
%      5)Velocity of Beam0 used in Calculations (m/s)
%      6)Intensity of Peak used to calculate Velocity of Beam0
%      7)Time of Beam0 used in Calculations (yyyymmddThhhmmmss)
%     -----
%      8)Velocity of Beam1 used in Calculations (m/s)
%      9)Intensity of Peak used to calculate Velocity of Beam1
%     10)Time of Beam1 used in Calculations (yyyymmddThhhmmmss)
%     -----
%     11)Velocity of Beam2 used in Calculations (m/s)
%     12)Intensity of Peak used to calculate Velocity of Beam1
%     13)Time of Beam2 used in Calculations (yyyymmddThhhmmmss)
%     -----
%     14)Horizontal Wind Velocity EAST Component (m/s)
%     15)Horizontal Wind Velocity NORTH Component (m/s)
%     16)Verticle Wind Velocity Component (m/s)
%     -----
%     17)Wind Speed (Horizontal) (m/s)
%     18)Wind Heading (Horizontal) (degrees)
%     19)Wind Calculation Confidence Index

    %  1)Largest Beam Measuremnt Number
    WindVecData01{1, MeasNo_i,iBeam} = iBeam*ones(N_spec_acc,1);

    %  2)Range (meters)
    WindVecData01{1, Range_i,iBeam} = Range_vec;

    %  3)Time of calculations
    WindVecData01{1, TimeOfCal_i,iBeam} = repmat(file_time_new,N_spec_acc,1);

    %  4)Delta Time (sec) (Time Last Measurment - Time Earliest Measurment)
    WindVecData01{1, TimeDelta_1,iBeam} = deltaS_vec;

    % datevec('20111205T134520','yyyymmddTHHMMSS')
    % t1=datenum(datevec('20111205T134530','yyyymmddTHHMMSS'))
    % t2=datenum(datevec(file_time_new,'yyyymmddTHHMMSS'))
    % deltat=datevec(t2-t1)

    %  5)Velocity of Beam0 used in Calculations (m/s)
    WindVecData01{1, Beam0Vel_i,iBeam} = VrBeam0;

    %  6)Intensity of Peak used to calculate Velocity of Beam0
    WindVecData01{1, Beam0Int_i,iBeam} = [];

    %  7)Time of Beam0 used in Calculations (yyyymmddThhhmmmss)
    WindVecData01{1, Beam0Time_i,iBeam} = repmat(file_time_0,N_spec_acc,1);

    %  8)Velocity of Beam1 used in Calculations (m/s)
    WindVecData01{1, Beam1Vel_i,iBeam} = VrBeam1;

    %  9)Intensity of Peak used to calculate Velocity of Beam1
    WindVecData01{1, Beam1Int_i,iBeam} = [];

    % 10)Time of Beam1 used in Calculations (yyyymmddThhhmmmss)
    WindVecData01{1,Beam1Time_i,iBeam} = repmat(file_time_1,N_spec_acc,1);

    % 11)Velocity of Beam2 used in Calculations (m/s)
    WindVecData01{1,Beam2Vel_i,iBeam} = VrBeam2;
```

```
    % 12)Intensity of Peak used to calculate Velocity of Beam1
    WindVecData01{1,Beam2Int_i,iBeam} = [];

    % 13)Time of Beam2 used in Calculations (yyyymmddThhhmmmss)
    WindVecData01{1,Beam2Time_i,iBeam} = repmat(file_time_2,N_spec_acc,1);

    % 14)Horizontal Wind Velocity EAST Component (m/s)
    WindVecData01{1,WindVelEAST_i,iBeam} = V1';

    % 15)Horizontal Wind Velocity NORTH Component (m/s)
    WindVecData01{1,WindVelNORTH_i,iBeam} = V2';

    % 16)Verticle Wind Velocity Component (m/s)
    WindVecData01{1,WindVelVert_i,iBeam} = V0';

    % 17)Wind Speed (Horizontal) (m/s)
    WindVecData01{1,WindSpeedHoriz_i,iBeam} = HorSpeed';

    % 18)Wind Heading (Horizontal) (degrees)
    WindVecData01{1,WindHeading_i,iBeam} = HorAZ';

    % 19)Wind Calculation Confidence Index - to indicate closeness of
    % measurments (t2-t2) / (t1-t0) If 1 then the time between measurments
    % are balanced The measurment time diffreence between t0 and t1 is less
    % significant than the time difference between t1 and t2
    % or
    % ((t2-t2) / (t1-t0)) / ((t2-t1)+(t1-t0))
end

%  TEST SPLOT
% quiver(WindVecData01{1, 16,177}',WindVecData01{1, 14,177}',WindVecData01{1,
15,177}',(1:64))
%% Save Wind Vector Calculation to File

[fileSy fileSm fileSd fileSH fileSM fileSS]=...
    (datevec(file_list_Vel_all_BeamVel(1,1:15),'yyyymmddTHHMMSS'));
[fileFy fileFm fileFd fileFH fileFM fileFS]=...
    (datevec(file_list_Vel_all_BeamVel(end,1:15),'yyyymmddTHHMMSS'));

tstart  = datenum(datevec(file_list_Vel_all_BeamVel(1,1:15),'yyyymmddTHHMMSS'));
tfinish = datenum(datevec(file_time_1,'yyyymmddTHHMMSS'));
tdur    = tfinish - tstart;

% Create file name suffix to indicate length of data processed
% Suffix description, Use TWO digits to Represent Dyas, Hours, Minutes,
% Seconds spanning the Durration From the Start to the Finish of the
% data collection period
[durtY durtMo durtD durtH durtM durtS] = datevec(tdur);
file_Days    = [sprintf('%02d',durtD),'D'];
file_Hours   = [sprintf('%02d',durtH),'H'];
file_Minutes = [sprintf('%02d',durtM),'M'];
file_Seconds = [sprintf('%02d',durtS),'S'];
file_Suffix_durTime = [file_Days file_Hours file_Minutes file_Seconds];

file_Suffix_durSeconds = [sprintf('%06d',...
                          durtD*86400+durtH*3600+durtM*60+durtS*1),'S'];

% TEST OF MAT save
% Save Cell Array of Data and Calculations
```

```
% Query User if they want to save the Wind Calculations
%save (fullfile(file_path,'WindVecData01'),'WindVecData01')

save (fullfile(file_path,[file_list_Vel_all_BeamVel(1,1:15),...
            file_name_suffix_WindVlVcDr,'_',...
                sprintf('%05d',size(file_list_Vel_all_BeamVel,1)),...
                                        '_',file_Suffix_durSeconds]),...
                                                    'WindVecData01')



% Save the 3 files used to find the wind vectors V0V1V2_files
% Append BeamX Velocity, Time, File Name, Comments Info to Meta File
% Prepare Data to be Saved to Meta File

% FIX THIS HERE
% Can NOT PUT TEXT AND NUMBERS IN SAME ARRAY--USE CELLS?
% Prepare Velocity, Vector, Wind Direction, Wind Velocity Meta Information
% Meta_Vec_pre = ...
%     {datestr(last_file_time,time_format),...    % Date & time
%     [file_name_wo_ext,file_ext],...
%     ['Data ',V0V1V2_files],...
%     ['TruckZE=',num2str(Truck_relative_ZE(mod(iBeam-1,3)+1))],...
%     ['TruckAZ=',num2str(Truck_relative_AZ(mod(iBeam-1,3)+1))],...
%     cell2mat(answer2)};
%
% % If function 'ex_func' is not available use a loop
% Meta_Vec_header = Meta_Vec_pre;
% for k = 1:length(Meta_Vec_header)
%     in_datatype = class(Meta_Vec_header{k});                %<--------------------
fix
%     switch in_datatype
%     case 'char'  %Cell is a string, do nothing
%     case 'double'
%         Meta_Vec_header{k} = num2str(Meta_Vec_header{k});
%     end
% end
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Write Metafile Data to File in Directory for Horizontal Wind Vector
% The 1st Line of the Text File Will Have Info That Will Not Be Repeated
% [path_string_M,file_name_wo_ext_M,file_ext_M]= fileparts(file_name_M);
% file_path_MwVec = [file_name_wo_ext_M,...
%                                 file_name_suffix_WindVlVcDr,'.txt'];
% fid_M = fopen(fullfile(file_path_M,file_path_MwVec), 'a');
% fprintf(fid_M,'%s, ', Meta_Vec_header{:}); % Write Comma Seperated
% fprintf(fid_M,'\n'); % Write Cariage Return to File
% fclose(fid_M);


 %% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 % Increase index iBeam, the total number of the BeamX measurments
 iBeam = iBeam + 1;   %
 display(['Beam Measurement: ',num2str(iBeam)])

 %% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 % Query User If They Would Like to Continue Taking Data
%  options.Interpreter = 'tex';
%  % Include the desired Default answer
%  options.Default = 'YES- Continue Taking Data';
%  % Create a string for the question
```

```
%  qstring = 'Would You Like To Continue To Take Data';
%  choice = questdlg(qstring,'TAKE MORE DATA ?',...
%                    'YES- Continue Taking Data','No',options);
%  switch choice
%     case 'YES- Continue Taking Data'
%         % Do Nothing and continue in the while loop
%     case 'No'
%         UserQuit = true;    % User wants to quite
%  end
%
% end


% END OF NORMALLY EXECUTED CODE

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%                                                                 %%%%%%%%%%%%%
%%%%%%% BELLOW ARE PROTIONS OF TEST CODES I TRIED OUT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%                                                                 %%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


if 5==9
A = randn(10,1);

% tequnique to have user enter a variable name and value
checkLabels = {'Save sum of A to variable named:' ...
               'Save mean of A to variable named:'};
varNames = {'sumA','meanA'};
items = {sum(A),mean(A)};
export2wsdlg(checkLabels,varNames,items,...
             'Save Sums to Workspace');


A = randn(10,1);
checkLabels = {'Save sum of A to variable named:' ...
               'Save mean of A to variable named:'};
varNames = {'sumA','meanA'};
items = {sum(A),mean(A)};
export2wsdlg(checkLabels,varNames,items,...
             'Save Sums to Workspace');


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
d = dir;
str = {d.name};
[s,v] = listdlg('PromptString','Select a file:',...
                'SelectionMode','single',...
                'ListString',str);



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%% VIDEO CAMERA INPUT


% Select or enter a file name for saving a figure
% as an image in one of four formats, described in a cell array.
uiputfile({'*.jpg;*.tif;*.png;*.gif','All Image Files';...
          '*.*','All Files' },'Save Image',...
          'C:\Work\newfile.jpg')

%% TO MAKE A BUTTON ON A FIGURE WINDOW
qf = figure;
h = uicontrol('Position',[20 20 200 40],'String','Continue',...
              'Callback','uiresume(gcbf)');
disp('This will print immediately');
uiwait(gcf);
disp('This will print after you click Continue');
close(qf);
end
```

Part 4: Matlab function: function windbarb_DS01

```
% function windbarb_DS01(x,y,spd,dir,scale,width,color)
% From Google Code
% Draw wind barb given wind speed and direction on current plot.
% The size of the barb is scaled by the diagonal length of the plot.


% Normaly a windbarb has the tail (the feathers, the short lines that
% indicate the speed) towards the direction from which the wind is arriving
% and the tip (the end of the long line without the feathers) is pointing
% in the direction the wind is moving towards.
% The Windbard function has the opposit conventionn and plots in meters per sec.

function windbarb_DS01(x,y,spd,dir,scale,width,color)

xm = get(gca,'XLim');
ym = get(gca,'YLim');

as = pbaspect;

ll = sqrt((xm(2)-xm(1))^2+(ym(2)-ym(1))^2);
l1 = ll * scale;
l2 = 0.5 * l1;

x0 = x;
y0 = y;

% (x,y) 0/1/3/5/7/9
if dir <= 90
    x1 = x0 + l1 * abs(sin(pi*dir/180)) * (xm(2)-xm(1))/ll;
    y1 = y0 + l1 * abs(cos(pi*dir/180)) * (ym(2)-ym(1))/ll * as(1)/as(2);
elseif dir <= 180
    x1 = x0 + l1 * abs(sin(pi*dir/180)) * (xm(2)-xm(1))/ll;
    y1 = y0 - l1 * abs(cos(pi*dir/180)) * (ym(2)-ym(1))/ll * as(1)/as(2);
elseif dir <= 270
    x1 = x0 - l1 * abs(sin(pi*dir/180)) * (xm(2)-xm(1))/ll;
    y1 = y0 - l1 * abs(cos(pi*dir/180)) * (ym(2)-ym(1))/ll * as(1)/as(2);
else
    x1 = x0 - l1 * abs(sin(pi*dir/180)) * (xm(2)-xm(1))/ll;
    y1 = y0 + l1 * abs(cos(pi*dir/180)) * (ym(2)-ym(1))/ll * as(1)/as(2);
end

x3 = x0 + 0.875 * (x1-x0);
y3 = y0 + 0.875 * (y1-y0);
x5 = x0 + 0.750 * (x1-x0);
y5 = y0 + 0.750 * (y1-y0);
x7 = x0 + 0.625 * (x1-x0);
y7 = y0 + 0.625 * (y1-y0);
x9 = x0 + 0.500 * (x1-x0);
y9 = y0 + 0.500 * (y1-y0);

% (x,y) 2/4/6/8/x
l3 = sqrt(l1^2+l2^2-2*l1*l2*cos(pi*(90+10)/180));

%l3 = sqrt(l1^2+l2^2-2*l1*l2*cos(pi*(90+30)/180));
d1 = 180/pi*acos((l1^2+l3^2-l2^2)/2/l1/l3);
```

```
d0 = dir + d1;
if d0 > 360
    d0 = d0 - 360;
end

if d0 <= 90
    x2 = x0 + l3 * abs(sin(pi*d0/180)) * (xm(2)-xm(1))/ll;
    y2 = y0 + l3 * abs(cos(pi*d0/180)) * (ym(2)-ym(1))/ll * as(1)/as(2);
elseif d0 <= 180
    x2 = x0 + l3 * abs(sin(pi*d0/180)) * (xm(2)-xm(1))/ll;
    y2 = y0 - l3 * abs(cos(pi*d0/180)) * (ym(2)-ym(1))/ll * as(1)/as(2);
elseif d0 <= 270
    x2 = x0 - l3 * abs(sin(pi*d0/180)) * (xm(2)-xm(1))/ll;
    y2 = y0 - l3 * abs(cos(pi*d0/180)) * (ym(2)-ym(1))/ll * as(1)/as(2);
else
    x2 = x0 - l3 * abs(sin(pi*d0/180)) * (xm(2)-xm(1))/ll;
    y2 = y0 + l3 * abs(cos(pi*d0/180)) * (ym(2)-ym(1))/ll * as(1)/as(2);
end

x4 = x3 + (x2 - x1);
y4 = y3 + (y2 - y1);
x6 = x5 + (x2 - x1);
y6 = y5 + (y2 - y1);
x8 = x7 + (x2 - x1);
y8 = y7 + (y2 - y1);
xx = x9 + (x2 - x1);
yx = y9 + (y2 - y1);

% wind barb
line([x0 x1],[y0 y1],'linewidth',width,'color',color)

if spd >= 90
    line([x1 x2],[y1 y2],'linewidth',width,'color',color)
    line([x2 x3],[y2 y3],'linewidth',width,'color',color)
    line([x3 x4],[y3 y4],'linewidth',width,'color',color)
    line([x5 x6],[y5 y6],'linewidth',width,'color',color)
    line([x7 x8],[y7 y8],'linewidth',width,'color',color)
    line([x9 xx],[y9 yx],'linewidth',width,'color',color)
elseif spd >= 85
    line([x1 x2],[y1 y2],'linewidth',width,'color',color)
    line([x2 x3],[y2 y3],'linewidth',width,'color',color)
    line([x3 x4],[y3 y4],'linewidth',width,'color',color)
    line([x5 x6],[y5 y6],'linewidth',width,'color',color)
    line([x7 x8],[y7 y8],'linewidth',width,'color',color)
    line([x9 (xx+x9)/2],[y9 (yx+y9)/2],'linewidth',width,'color',color)
elseif spd >= 80
    line([x1 x2],[y1 y2],'linewidth',width,'color',color)
    line([x2 x3],[y2 y3],'linewidth',width,'color',color)
    line([x3 x4],[y3 y4],'linewidth',width,'color',color)
    line([x5 x6],[y5 y6],'linewidth',width,'color',color)
    line([x7 x8],[y7 y8],'linewidth',width,'color',color)
elseif spd >= 75
    line([x1 x2],[y1 y2],'linewidth',width,'color',color)
    line([x2 x3],[y2 y3],'linewidth',width,'color',color)
    line([x3 x4],[y3 y4],'linewidth',width,'color',color)
    line([x5 x6],[y5 y6],'linewidth',width,'color',color)
    line([x7 (x8+x7)/2],[y7 (y8+y7)/2],'linewidth',width,'color',color)
elseif spd >= 70
    line([x1 x2],[y1 y2],'linewidth',width,'color',color)
    line([x2 x3],[y2 y3],'linewidth',width,'color',color)
    line([x3 x4],[y3 y4],'linewidth',width,'color',color)
    line([x5 x6],[y5 y6],'linewidth',width,'color',color)
```

```
elseif spd >= 65
    line([x1 x2],[y1 y2],'linewidth',width,'color',color)
    line([x2 x3],[y2 y3],'linewidth',width,'color',color)
    line([x3 x4],[y3 y4],'linewidth',width,'color',color)
    line([x5 (x6+x5)/2],[y5 (y6+y5)/2],'linewidth',width,'color',color)
elseif spd >= 60
    line([x1 x2],[y1 y2],'linewidth',width,'color',color)
    line([x2 x3],[y2 y3],'linewidth',width,'color',color)
    line([x3 x4],[y3 y4],'linewidth',width,'color',color)
elseif spd >= 55
    line([x1 x2],[y1 y2],'linewidth',width,'color',color)
    line([x2 x3],[y2 y3],'linewidth',width,'color',color)
    line([x3 (x4+x3)/2],[y3 (y4+y3)/2],'linewidth',width,'color',color)
elseif spd >= 50
    line([x1 x2],[y1 y2],'linewidth',width,'color',color)
    line([x2 x3],[y2 y3],'linewidth',width,'color',color)
elseif spd >= 45
    line([x1 x2],[y1 y2],'linewidth',width,'color',color)
    line([x3 x4],[y3 y4],'linewidth',width,'color',color)
    line([x5 x6],[y5 y6],'linewidth',width,'color',color)
    line([x7 x8],[y7 y8],'linewidth',width,'color',color)
    line([x9 (xx+x9)/2],[y9 (yx+y9)/2],'linewidth',width,'color',color)
elseif spd >= 40
    line([x1 x2],[y1 y2],'linewidth',width,'color',color)
    line([x3 x4],[y3 y4],'linewidth',width,'color',color)
    line([x5 x6],[y5 y6],'linewidth',width,'color',color)
    line([x7 x8],[y7 y8],'linewidth',width,'color',color)
elseif spd >= 35
    line([x1 x2],[y1 y2],'linewidth',width,'color',color)
    line([x3 x4],[y3 y4],'linewidth',width,'color',color)
    line([x5 x6],[y5 y6],'linewidth',width,'color',color)
    line([x7 (x8+x7)/2],[y7 (y8+y7)/2],'linewidth',width,'color',color)
elseif spd >= 30
    line([x1 x2],[y1 y2],'linewidth',width,'color',color)
    line([x3 x4],[y3 y4],'linewidth',width,'color',color)
    line([x5 x6],[y5 y6],'linewidth',width,'color',color)
elseif spd >= 25
    line([x1 x2],[y1 y2],'linewidth',width,'color',color)
    line([x3 x4],[y3 y4],'linewidth',width,'color',color)
    line([x5 (x6+x5)/2],[y5 (y6+y5)/2],'linewidth',width,'color',color)
elseif spd >= 20
    line([x1 x2],[y1 y2],'linewidth',width,'color',color)
    line([x3 x4],[y3 y4],'linewidth',width,'color',color)
elseif spd >= 15
    line([x1 x2],[y1 y2],'linewidth',width,'color',color)
    line([x3 (x4+x3)/2],[y3 (y4+y3)/2],'linewidth',width,'color',color)
elseif spd >= 10
    line([x1 x2],[y1 y2],'linewidth',width,'color',color)
elseif spd >= 5
    line([x1 (x2+x1)/2],[y1 (y2+y1)/2],'linewidth',width,'color',color)
else
    line([x1 (x2+3*x1)/4],[y1 (y2+3*y1)/4],'linewidth',width,'color',color)
end
```

Part 5: Matlab program: ProcLidarBeamVel_04_Plot_02.m

```
% ProcLidarBeamVel_04_Plot_02.m
%
% clear all;
% David Santoro
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% PURPOSE:
% This program will read, from the hard drive,  Matlab MET file of cell
% array containing wind vectors and Wind speed/heading
%  Data Format to be placed in Cell Array
%  1)Largest Beam Measuremnt Number used in Current Calculations
%  2)Range (meters)
%  3)Time Last Measurment used in calculations
%  4)Delta Time (Time Last Measurment - Time Earliest Measurment)
% -----
%  5)Velocity of Beam0 used in Calculations (m/s)
%  6)Intensity of Peak used to calculate Velocity of Beam0
%  7)Time of Beam0 used in Calculations (yyyymmddThhhmmmss)
% -----
%  8)Velocity of Beam1 used in Calculations (m/s)
%  9)Intensity of Peak used to calculate Velocity of Beam1
% 10)Time of Beam1 used in Calculations (yyyymmddThhhmmmss)
% -----
% 11)Velocity of Beam2 used in Calculations (m/s)
% 12)Intensity of Peak used to calculate Velocity of Beam1
% 13)Time of Beam2 used in Calculations (yyyymmddThhhmmmss)
% -----
% 14)Horizontal Wind Velocity EAST Component (m/s)
% 15)Horizontal Wind Velocity NORTH Component (m/s)
% 16)Verticle Wind Velocity Component (m/s)
% -----
% 17)Wind Speed (Horizontal) (m/s)
% 18)Wind Heading (Horizontal) (degrees)
%
% The modified verion of SNAP (Sameh) took 8192 data points and outputs 64,
% 64 point spectrums that resulted from accumulating the FFT of 10,000
% laser pulses (1/2 sec) and wrote a text file to the hard drive after
% taking a total of 30 sec worth of data (i.e. A text file concisting of
% series of 60 sets of spectrums where each set of spectrums had 64
% spectrums each of 64 points.)
%
% These text files were processed and turned into a Matlab vector made of
% 64 wind velocities (one for each range gate along the beam (NOTE: There
% are a few range gates that are from before the laser beam exited the tip
% of the fiber)).  The velocites along the beam were saved in a text file
% with the name yyymmddThhmmss_BeamXVel.txt, where X is 0|1|2.
% Beam0 is the verticle beam (w)
% Beam1 is the (u) : u beam is counter clockwise of the v beam (Top View)
% Beam2 is the (v) : v beam is clockwise of the u beam (Top View)
%
% In takelidarXX.m the SNAP program with a FPGA code that takes 8192 data
% points, uses 128 concecutive points to compute 64 FFT spectrums, add the
% newest spectrums to a running sum of spectrums until 10,000 have been
```

```
% colleted. After 10,000 summations these are appended to a file until 30
% seconds has been collected.  Then the user will be instructed to readjst
% the laser beams direction and to repeat the data collection.
%
%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The text file of accumulated spectrums will have the naming format
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FILE NAME NOMENCLATURE
% The default text folder name is encoded with the START date and time
% yyyymmddTHHMMSS.txt
% yyyymmddTHHMMSS is ISO 8601 standard- NOTE:The 'T' between date and time
% yyyy = the year (i.e. 2011)
% mm   = the month (two digits, i.e January = 01)
% dd   = the day of the month (two digit day of the month, i.e. 07)
% HH   = the hour (24 hour clock, i.e. 3:45PM is 1545, NO daylight savings)
% MM   = the minutes of the hour (00 to 59)
% SS   = the seconds of the minute (00 to 59)
% Matlab has a Numeric Identifier Predefined Format of 30
% for the ISO 8601 format standard
% If the time right now was October 5th 2011, 10:51:15 PM
% ds = datestr(now,30) --> ds = 20111005T225115

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The MAT file of processed beam velocities saved as cell arrays
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FILE NAME NOMENCLATURE
% yyyymmddTHHMMSS_WindVec_bbbbb_ssssssS.mat
% Start Date and Time as per ISO 8601 standard followed by "_WindVec_",
% bbbbb   = the number of Beam Measurments
% ssssssS = the number of seconds covered, "S"
% For Example: 20111205T122223_WindVec_00177_015038S
% 177 Measurements were made on December 5th, 2011 starting at 12:22:23
% and span 15,038 seconds (4.17 Hours)

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Useful Unit Converstions
% 1 mph            = 0.44704 meters / second
% 1 knot           = 0.514444444 meters /
% 1 foot / second  = 0.3048 meters / second
% 1 meter / second = 1.94384449 knots
% 1 meter / second = 2.23693629 mph
% 1 meter / second = 3.2808399 feet / second

orient landscape      % Set print orientation of papaer to Landscape
time_format      = 30;% Time Format 30 = 'yyyymmddTHHMMSS' (ISO 8601)
datetimeStart    = datestr(now,time_format);% Get program start date & time
file_finish_time = 0; % Variable to hold time when SNAP finishes with file
file_wait_time   = 3; % Seconds to wait before testing to see
                      % if SNAP has finished writing to a file
datetimeWindVec  ='20111205T122223_WindVec_00177_015038S';

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Set Up Debugging Variables
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
dBUG           = 1;% DEBUGGING SWITCH
                   % =1 (true) Do debugging,
                   % =0 (false) No debuging
                   %
dbugPC         = 1;% LOCATION OF COMPUTER
                   % Use to precnfigure filenames, directories, etc
                   % =0 for WinXP 64bit Doppler PC (CCNY),
```

```
                    % =1 for home (santoro),
                    % =2 for work (santoro),
                    % =3 for work T647 (santoro),
                    % =4 for temporary site
dbugPLOTS       = 0;% CREATE PLOTS FOR DEBUGGING
                    % =0 No debuging plots,
                    % =1 Plot Debug Plots
                    %
dBUGvelMethode = 3;% METHOD OF DETERMINING WIND VELOCITY
                    % Freq Peak Wind Vel Method
                    % =1 MLE, Maximum Likily Methode
                    % =2,
                    % =3, Frequency Estimators- dsp Tips
                    %     Fast, Accurate Frequency Estimators- dsp Tips &
                    %     Tricks Jacobsen, and Kootsookos IEEE Signal
                    %     PRocessing Magazine May 2007, Pg 123, Equation 2
                    %
dBUGfakeSNAP = 3;   % SOURCE OF DATA
                    % =0 Do NOT create Fake SNAP data, REAL DATA on HD with
                    %    Date/Time stamped file name
                    % ------------------------------------------------
                    % =1 Create fake SANP data for debugging (ramp func)
                    % ------------------------------------------------
                    % =2 Create fake background data from a data file of ONE
                    %    spectra. And generate sample data by modifying it.
                    %    Use
                    %    092311DelayT0_AB_2.24uS_my_output_file_3_60s.txt
                    %    as fake data Has delay of 2.24us, 9 pretrigger
                    %    spectra and the 1st two spectra seem to be
                    %    different from the next seven spectra.  There
                    %    seems to be 3 spectra effected by the outgoing
                    %    pulse tip reflection
                    % ------------------------------------------------
                    % =3 Do NOT create Fake SNAP data, REAL DATA on HD with
                    %    "my_30_Seconds_file.txt" file name
                    %
dbugNoiseSub = 0;   % METHOD OF REMOVEING BACKGROUND (NOISE)
                    % =0 10Log(S+N)-10Log(N), CHECK?????
                    % =1 ((S+N)-N)/N,
                    % =2 10Log((S+N)-N) - 10Log(N)

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Set default SNAP OUTPUT File Type
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
file_nameXXs = 'my_30_Seconds_file.txt';
SNAPfiletype = 3;  % FILE NOMENCLATURE EXPECTED FROM SNAP PROGRAM
                    % =0, Text File, yyyymmddTHHMMSS.txt
                    % =1, Text File, my_30_Seconds_file.txt
switch SNAPfiletype
  case 0
      file_name = [datetimeStart,'.txt'];% File Name with Date/Time Stamp
  case 1  % If SNAP outputs non date/time stamped file name
      file_name = file_nameXXs;
  case 3  % Wind Vector Data Cell Array on Hard Drive
      file_name = [datetimeWindVec,'.mat'];% File Name with Date/Time Wind
end

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Set default directories and file names based on location of PC
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
switch dbugPC    % =0 for van ,
                 % =1 for home,
```

```
                % =2 for work,
                % =3 Work T647
                % =4 for _______
    case 0 % =0 for van
      file_path = ...
       'D:\TEMP_SANTORO\20111213_Data_SNAPmod\20111213_Data_SNAPmod_Run2';
       %'D:\TEMP_SANTORO\20111201_Data_SNAPmod\20111201_Data_SNAPmod_Test'
       %'D:\TEMP_SANTORO\20111128_Data_SNAPmod\';
       %'D:\TEMP_SANTORO\20111125_Data focused at 2797m\20111125_Data SNAPmod focused
at 2797m\20111125_Data Vert Keopsys 170ma 1500ma DDG 1.1227us';
       %'D:\TEMP_SANTORO\PhD Matlab\Lidar_Text_Data_Matlab\20111117_Data\';

      file_name_fakeBK = ...
       'BackGround_Noise_092311DelayT0_AB_2.24uS_my_output_file_3_60s.txt';
      %file_name = '20111004T131022.txt';
    case 1 % =1 for home
%       file_path = ...
%        'D:\David\PhD\PhD
Data\Data\20111205_Data_SNAPmod\20111205_Data_SNAPmod_Run2_temp';
      file_path = ...
        'D:\David\PhD\PhD
Data\Data\20111208_Data_SNAPandSNAPmod\20111208_Data_SNAPmod_Run1_scanFor2h50m_temp';
      file_name_fakeBK = ...
       'BackGround_Noise_092311DelayT0_AB_2.24uS_my_output_file_3_60s.txt';
      %file_name = '20111004T131022.txt';
    case 2 % =2 for work
      file_path = ...
       'C:\David\CCNY PhD Data\20111205_Data_SNAPmod\20111205_Data_SNAPmod_Run2_temp';
      file_name_fakeBK = ...
       'BackGround_Noise_092311DelayT0_AB_2.24uS_my_output_file_3_60s.txt';
      %file_name = '20111004T131022.txt';
    case 3 % =3 for work T647
      file_path ='C:\Temp\santoro\CCNY PhD\PhD
Data\20111205_Data_SNAPmod\20111205_Data_SNAPmod_Run2_temp';
      %file_path = 'C:\Temp\santoro\PhD Matlab\Lidar_Text_Data_Matlab';
      file_name_fakeBK = ...
       'BackGround_Noise_092311DelayT0_AB_2.24uS_my_output_file_3_60s.txt';
      %file_name = '20111004T131022.txt';
    case 4 % =4 for _____
      file_path = 'C:\temp\santoro\PhD
Data\Datat\20111205_Data_SNAPmod\20111205_Data_SNAPmod_Run2_temp';
      file_name_fakeBK = ...
       'BackGround_Noise_092311DelayT0_AB_2.24uS_my_output_file_3_60s.txt';
      %file_name = '20111004T131022.txt';
end

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Default Length of Time to Take Measurememts
Time_Length   = 30;    % Number of seconds to make a measurment
Time_LengthBK = 30;    % Number of  seconds to take an independent, before
                       % measuremnts start, background (noise) measurement

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Default File Name Suffixes to be Used
file_name_suffix_Background    = '_bk';  % Background (Noise) Spectrums
file_name_suffix_BackgroundAVG = '_avg'; % Average Background (Noise) Spec
file_name_suffix_MetaData      = '_meta';% Meta Data
file_name_suffix_Temp          = '_temp';% temporary Data
file_name_suffix_WindVlVcDr    = '_WindVec';% Wind Velocity,Vectors,Directions
file_name_suffix_BeamX         = '_Beam';% BeamX
file_name_suffix2_BeamX         = 'Vel';% BeamX Velocity
```

```
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Constants and Default Values %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Physical Constants
c    = 299792458;       % Speed of Light (m/s) in a vacume
nAir = 1.000293;        % Refractive Index of air
                        % Understanding the properties of matter
                        % By Michael De Podesta page 131
h   = 6.62606957e-34;   % Planck's constant (Js)

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Digital Delay Generator (Standford DG-535)
DDGjitM    = 25e-9;     % Output jitter speced by manufacturer
DDG_TplusA = 1.1228e-6; % T+delay for A time (s)
DDG_AplusB = 300e-9;    % A+delay for B time (s)
DDG_PulseT = DDG_AplusB - DDG_TplusA;   % DDG pulse period (s)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ADC (Innovative Intergrations X5-400M verD) Constants and Defaults
FsamplingM= 400e6;  % ADC sampling Rate (spec manufacturer)
ADCbitsM  = 14;     % ADC number of bits (spec manufacturer)
ADCenobM  = 10.1;   % Effective Number of Bits for our ADC
                    % (spec manufacturer) with full scale sinosoidal input
                    % 70MHz, 850 mVRMS filtered sine sampled at 400 MSPS,
                    % DC-200 MHz, DC-coupled specified by manufacturer
ADCsnrM   = 73.6;   % SNR (dB) (spec manufacturer) for
                    % 70MHz, 850 mVRMS filtered sine sampled at 400 MSPS,
                    % DC-200 MHz, DC-coupled specified by manufacturer
ADCnoiseM = 350e-6; % Typical noise for grounded input (V), 1 standard
                    % deviation by manufacturer (spec manufacturer)
ADCoffseterrorM = 700e-6;   % Factory calibration Offset Error(V), averaged
                            % 64K samples after warmup (spec manufacturer)
ADCgainerrorM   = 0.0002;   % 0.02%, Factory calibration Gain Error
                            % after warmup (spec manufacturer)
ADVinMaxM       = +1;       % Input Voltage Max (V) (spec manufacturer)
ADVinMinM       = -1;       % Input Voltage Min (V) (spec manufacturer)
ADCjitAdditiveM = 80e-12;   % Additive jitter of ADC using external clock
                            % (s) (spec manufacturer)
ADCjitInterM    = 340e-15;  % Total jitter of ADC using internal clock
                            % (s) (spec manufacturer)

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  System Parameters
SampPerPulse = 8192;    % Number of samples collected per laser pulse
UsepreTR     = 4;       % Use the nth pre trigger spectrum to find the
                        % average backgound noise spectrum of a
                        % particular pulse
Navg_spec = Time_Length*2;  % Default Number of spectrums from background
                            % measurements to use in order to find
                            % BACKGROUND (noise) is ALL
Naccsize = 4096;        % Number of points in FPGA output for one
                        % FFT accumulation accumulated measurement
NpreTRspec = 6;         % Number of pretrigger spectrums
                        % (FFT spectrums befor laser pulse)
Ntip_spec = 2;          % Number of spectrums effected by
                        % pulse reflection from fiber tip
NpreTRtip_spec = NpreTRspec + Ntip_spec;    % Number of spectrums effected
                                            % by reflections or pre trigger
PRF       = 20000;                % Pulses Repatition Frequency (Hz)
                                  % (laser pulses per second)
Tpulse    = 1/PRF;                % Pulse to pulse period (sec)
Npulseacc = 10000;                % Number of laser pulses per
                                  % accumulation output from ADC/FPGA
```

```
Tpulseacc = Npulseacc*Tpulse;      % Period (S) for Npulseacc pulses
                                   % to be accumuled (Aquisition period)
Naccs     = Time_Length/Tpulseacc; % Number of FFT acumulation outputs
                                   % by FPGA for the measurement period
Nptsfile  = Naccsize*Naccs;        % Number of points SNAP writes to
                                   % a data file for the measurement period
NptsfileBK= Naccsize*Time_LengthBK/Tpulseacc; % Number of points SNAP
                                              % write to Background data
                                              % file
No_Averages = Naccs;    % Number of pulse accumulations to average
                        % to make a wind estimate

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FFT related Constants and Defaults
Nfft = 128;             % Number of points input to the FPGA's FFT routine
N_pt_spec = Nfft/2;     % # of Points in a single spectrum for a single
                        % range gate output by the FPGA (i.e. Nfft/2)
N_spec_acc = Naccsize/N_pt_spec;  % Number of spectrum per accumulation
FFTbw = FsamplingM/Nfft;% Freq bandwidth of each FFT output/bin from FPGA
                        % CHECK THIS-does xilinx FFT library put the DC
                        % component in the 1st Bin???  Matlab may be
                        % different  GET Web PAGE or PDF REFERENCE

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Range Gate Size of FFT
% (Time to Take Samples for one FFT of data)*SpeedOfLight/2
RGS    = (Nfft/FsamplingM) * c/2;     % Range Gate Size in meters
Rrange = (RGS/2:RGS:RGS*N_spec_acc)'; % Radial ranges (vector)
                                      % from origin (telescope)

            % CHECK- SHOULD THIS BE POST PULES ONLY?????

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Laser Charateristics, AOM Constants and Defaults
%
% SEED LASER LOW POWER AND HIGH POWER OUTPUTS
% NP Photonics Part Number = RFLSA-500-1-1545.2-PM-S_NSL
% Oct 2008   Serial number = J10-000789-379
LambdaLO = 1545.2e-9;       % Wavelength of Seed Laser
fseed    = c/LambdaLO;      % Frequency of Laser
KW       = 2*pi/LambdaLO;   % Wave number, (angular wavenumber,
                            % circular wavenumber)
SeedLW   = 5200;            % Lorentzian Full Width Half Max (Hertz)
                            % (spec measured by manufacturer).  Full width
                            % of 104.0 kHz when 20 dB down from peak of
                            % signal.  (Se Manual- Final Inspection Report
                            % for 000780 RFLSA-500-1-1545.2-PM-S-NSI)
SeedStability = 50e6;       % Seed Laser stability over 1 hour
SeedRINpk     = -113.1;     % (dB/Hz),Relative Intensity Noise (RIN)
                            % Dominant peak at relaxation oscillation freg
                            % of laser at 2.3 MHz.  Shot noise limited
                            % greater then 50 MHz.
SeedPER       = 17;         % Seed laser Polarization Extintion Ration
                            % Minimum (dB)
SeedS_ASEi    = 35;         % Seed laser Signal to ASE Noise Ratio
                            % Intergrated Minimum (dB)
SeedS_ASE     = 55;         % Seed laser Signal to ASE Noise Ratio
                            % 50 pm, bandwidth, Minimum (dB)
SeedSMSR      = 50;         % Seed laser Side Mode Supression Ratio
                            % Minimum (dB)
SeedRINsnLP84 = -160;       % (dB/Hz), RIN of seed low power output
                            % at ~ 84Mhz, shot noise limited
```

```
SeedRINsnLP54 = -157;       % (dB/Hz), RIN of seed low power output, 54MHz
                            % shot noise limited
SeedRINsnLP114= -163;       % (dB/Hz), RIN of seed low power output, 114MHz
                            % shot noise limited
SeedRINsnHP84 = -155;       % (dB/Hz), RIN of seed high power output
                            % at ~ 84Mhz, shot noise limited
SeedRINsnHP54 = -153;       % (dB/Hz), RIN of seed high power output
                            % at ~ 84Mhz, shot noise limited
SeedRINsnHP114= -157;       % (dB/Hz), RIN of seed high power output
                            % at ~ 84Mhz, shot noise limited

thao     = 200e-9;          % Laser Pulse Width from AOMS FWHM (s)
freqAOM1 = 42e6;            % Acousto Optical Modulator (AOM) 1 freq shift
freqAOM2 = 42e6;            % Acousto Optical Modulator (AOM) 2 freq shift
fsAOM    =freqAOM1+freqAOM2;% Total freq shift to pulse by both AOMs

E        = 13e-6;           % Energy per Laser Pulse(J) Keopsys Pulse Amp

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Doppler Frequencies of Interest for Wind Measurments
FreqInterestStartUser = 46.875e6;   % Start Freq of interest
                                    % for Doppler Measurements
FreqInterestStopUser  = 125.0e6;    % Stop Freq of interest
                                    % for Doppler Measurements

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FFT spectral output Constants and Defaults
NpreTR = N_pt_spec*NpreTRspec;  % Number of points in all spectrums before
                                % laser pulse (i.e. pre trigger spectrums)

IpreTRstart = (UsepreTR-1)*N_pt_spec+1;     % Index of start of the pre
                                             % Trigger spectrum chosen
                                             % to use as background
                                             % (noise)
IpreTRend   = (UsepreTR*N_pt_spec);          % Index of end of the pre
                                             % Trigger spectrum chosen
                                             % to use as background
                                             % (noise)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Spectral Bin Indexies and Counts
% Index of START freq of spectrum range of interest
   % IpreTR_Spec_Start: Index of the frequency bin with the lowest
   % frequency which we are interested in, within a spectrum of NFFT/2
   % points.
IpreTR_Spec_Start = floor(FreqInterestStartUser/FFTbw);

% Index of Stop freq of spectrum range of interest
   % IpreTR_Spec_Stop: Index of the frequency bin within a spectrum of
   % NFFT/2 points with the highest frequency which we are interested in.
IpreTR_Spec_Stop = ceil(FreqInterestStopUser/FFTbw);

% Number of spectral bins in the freq range of interest
Nspec_bins = (IpreTR_Spec_Stop - IpreTR_Spec_Start)+1;

% Number of spectrums after pre trigger and fiber tip reflection spectrums
N_spec_acc_post_pulse = N_spec_acc - NpreTRspec - Ntip_spec;

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Constants for SNR simulation of Lidar signal based on equation 3
% of Mitsubishi paper titled compact all-fiber pulsed coherent doppler
```

```
%
% Compact all-fiber pulsed coherent Doppler lidar system for wind sensing
% S. Kameyama, T. Ando, K. Asaka, Y. Hirano, and S. Wadaka
% 10 April 2007,  Vol. 46, No. 11, APPLIED OPTICS 1953
R     = 1 ;     % reflectivity of a hard target
B     = 100e6;  % for wide band SNR use 100MHz,
                % for narrow band SNR use 2MHz Bandwidth .
Cn_sq = 2e-14;  % Refractive Index Structure Constant;
D     = 0.1;    % Effective aperture Diameter (m)
beta  = 8.3e-7; %R/(fseed*thao);%atmospheric backscatter coefficiency(/m/sr)

% The following parameter needs to be checked
K     =  0.95;  % One way atmospheric transmittance (/km)
F     = 1500;   % Focal Range of Optical Antena (m)
Eta_F = 0.9;    %0.355 ; % Far field system efficiency (-4.5 dB)
Ac    = 0.76;   % Correction Factor.
Eta_I = 0.9;    %0.676; % Factor for insertion loss of optical components
                % except for telescop (-1.7 dB)
Eta_RE = 0.977; % Telescope absorption and reflection loss (-0.1 dB)
Eta_A  = 0.91;  % Telescope reflection loss (-0.4 dB)
Eta_PP = 0.912;
Eta_Q  = 0.9;   % 0.794;


%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Truck Cardinal Directions (North, South, East, West) and Tilt from Level
Truck_tiltZE = 0;% Zenithe angle = 0 if laser beam is verticle
                 % (and Truck optical table is level)
                 % If the edge of the optical table closest to the
                 % fron of the truck is lower than the egde of
                 % the optical table nearest the rear of the truck
                 % ZE will have a POSITIVE value, and a NEGATIVE value
                 % if the rear edge is lower than the front edge.
                 % If truck has right side of optical table lower than the
                 % left side of the optical table then __________
                 % Have to come up with a framework - don;t worry now

Truck_cardianlAZ = 299.33;   % Azmith angle = 0 if the direction of
                             % Truck's optical table taken along the
                             % long edge (length) is North.
                         % North is 0 degrees, East is 90 degrees,
                         % South is 180 degress and West is 270 degrees

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Software Versions Being Used - for Meta Data
%
% Matlab Information about curret .m file running
MlabFname      = mfilename;         % Matlab Filename running without '.m'
MlabFpath_name = mfilename('fullpath'); % Path and Filename of running .m

% FPGA Firmware version being used
X5fpgaLogicPath_Name = ['D:\TEMP_SAMEH\Simulink_Works\Accumulated_FFTs\'...
                        'FFT_10K_Accum_Reset_7_hwcosim_module.mdl'];
% SNAP Version being Used
SNAPverPath_Name     = ['C:\Innovative\X5-400M\Examples\'...
                            'Snap_07282011\FPGA_SNAP\Debug_x64'];

% X5-400M SNAP Default Settings
X5frameSize     = 8192;% Frame Size in decimal
X5frameSizeBK   = 8192;% Frame Size in decimal (BACKGROUND MEASUREMENT)
X5packetSize    = 8192;% Packet Size in decimal
```

```
X5packetSizeBK  = 8192;% Packet Size in decimal (BACKGROUND MEASUREMENT)
X5ch            = 0;   % ADC channel to use for aquizition
X5BytesPerSamp  = 4;   % Because FPGA FFT uses 32nits per sample
X5samples00m01s = (X5BytesPerSamp*SampPerPulse*PRF)/Npulseacc;
                       % Number of samples to set for 0 minutez, 1 second
                       % of FFT spectrums, output every 500ms
X5samples       = Time_Length*X5samples00m01s;   % # SNAP Samples to take
                                                 % for BeamX measurments
X5samplesBK     = Time_LengthBK*X5samples00m01s; % Number of SNAP Samples
                                                 % to take for background

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Lasers' Operational Settings
KeopsysStage1 = 0.170;  % Keopsys 1st Stage Driving Current (A)
KeopsysStage2 = 1.700;  % Keopsys 2nd Stage Driving Current (A)
NP_piezeo     = 0;      % NP Photonics Piezeio Voltage Setting (V)(0=auto)
NP_hiP        = 0.500;  % NP Photonics High Power Output Settinf (W)

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Dummy Data Prototypes to Be Used for Backgrounds (noise) and Data

% OPTION 1: Waveform Based Fake Data Prototype
% WfStart1 = 1-(N_pt_spec/2);
% WfEnd1   = (N_pt_spec/2)-1;
% WfWidth1 = ceil(N_pt_spec*1.09);
% WfSkew1  = -0.2;
%
% WfStart2 = 2-(N_pt_spec/2);
% WfEnd2   = (N_pt_spec/2);
% WfWidth2 = ceil(N_pt_spec*0.78);
% WfSkew2  = 0.1;
%
% dummy_data_Proto = [1,(tripuls(WfStart1:WfEnd1,WfWidth1,WfSkew1) - ...
%                        0.63*tripuls(WfStart2:WfEnd2,WfWidth2,WfSkew2))]';
% OPTION 2: Add some if needed

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Maximum Likelyhood Estimation Default Parameters
SNR_Thresh       = 0.0000125; % SNR decision threshold for a measured value
No_Fitted_Points = 3;     % No of spectrum points to be fitted in MLE

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Cell Array Results Storage Indeies/Names
MeasNo_i         = 1;
Range_i          = 2;
TimeOfCal_i      = 3;
TimeDelta_1      = 4;
Beam0Vel_i       = 5;
Beam0Int_i       = 6;
Beam0Time_i      = 7;
Beam1Vel_i       = 8;
Beam1Int_i       = 9;
Beam1Time_i      = 10;
Beam2Vel_i       = 11;
Beam2Int_i       = 12;
Beam2Time_i      = 13;
WindVelEAST_i    = 14;
WindVelNORTH_i   = 15;
WindVelVert_i    = 16;
WindSpeedHoriz_i = 17;
WindHeading_i    = 18;
```

```
WindCalcConf_1   = 19;


%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Truck Cardinal Directions (North, South, East, West) and Tilt from Level
%
% Truck_tiltZE:
% Zenithe angle = 0 if laser beam is verticle (Truck optical table is
% level) If the edge of the optical table closest to the front of the truck
% is lower than the egde of the optical table nearest the rear of the truck
% ZE will have a POSITIVE value, and a NEGATIVE value if the rear edge is
% lower than the front edge. ( Or if table is tilted back have Zieneth
% be positive and add 180 degrees to Azmith ??)
% If truck has the right side of optical table lower than the left side of
% the optical table then __________  Have to come up with a framework to
% decide what to do- don;t worry now
% Any time Truck_tiltZE NOT = Zero the Zenith and Azmith angles used to
% position the beam for a measuremnets relative to the truck's optical
% table will have to be adjusted to make sure beam is going towards the
% correct location in space.

% Truck_cardianlAZ:
% Azmith angle = 0 if the direction of Truck's optical table taken along
% the long edge (length) is North.  North is 0 degrees, East is 90 degrees,
% South is 180 degress and West is 270 degrees

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% % Query User for: Truck's Cardinal Orientation (Front faces 0 to 359 deg)
% % and Truck's Optical Table Tilt from Level
% %
% % NOTE: Magnetic NORTH is approzimatley 13 deg 10 min off towards EAST
% %       in the NYC area.  True North is NOT Magnetic North
% prompt3 = {['Research Van Heading (Front Points Toward): North= 0 deg'...
%             ' [Heading is Magnetic Heading +13.8 degrees]'],...
%            'Optical Table Forward Tilt from Level (degrees):'};
% dlg_title3 = 'Research Van Orientation (Cardinal Orientation/Tilt)';
% num_lines3 = 1;
% def3       = {num2str(Truck_cardianlAZ),num2str(Truck_tiltZE)};
% options.Resize      = 'on';
% options.WindowStyle = 'normal';
% options.Interpreter = 'tex';
% answer3 = inputdlg(prompt3,dlg_title3,num_lines3,def3,options);
% AZ                  = str2num(answer3[15]);  % Cardinal Heading of Truck
% ZE                  = str2num(answer3{2});  % Tilt of Truck

% %% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% % Users Enters the Tilt (Zieneth Angle) of the Three Laser Beams
% Truck_ZE0 = 0;     % Beam #0 Tilt (zenith) relative to optical table
% Truck_ZE1 = 30;    % Beam #1 Tilt (zenith) relative to optical table
% Truck_ZE2 = 30;    % Beam #2 Tilt (zenith) relative to optical table
%
% prompt4 = {...
%             ['Beam0: Zenith Angle (Tilt Relative to ',...
%                     'Optical Table) of Verical Beam, degrees:'],...
%             ['Beam1: Zenith Angle (Tilt Relative to ',...
%                     'Optical Table) of North Beam, degrees:'],...
%             ['Beam2: Zenith Angle (Tilt Relative to ',...
%                     'Optical Table) of East Beam, degrees:']};
% dlg_title4 = 'Zenith Angle of Beams Referenced to Optical Table';
% num_lines4 = 1;
% def4 = {num2str(Truck_ZE0),num2str(Truck_ZE1),num2str(Truck_ZE2)};
```

```
% answer4 = inputdlg(prompt4,dlg_title4,num_lines4,def4);
% ZE0 = str2num(answer4[15]);  % Zenith angle of verticle beam (Beam0)
% ZE1 = str2num(answer4{2});  % Zenith angle of North-ish beam (Beam1)
% ZE2 = str2num(answer4{3});  % Zenith angle of East-ish beam (Beam2)
%
% Truck_relative_ZE = [ZE0 ZE1 ZE2];
% ZE0offset = ZE0 + ZE;% Laser Beam #0, verticle
%                      % Difference of Vector V0 zenith
%                      % angle from earth normal
% ZE1offset = ZE1 + ZE;% These would be used if multiple laser beams are used
% ZE2offset = ZE2 + ZE;% These are the tilts of the laser beams relative to
%                      % the optical table
% ZEoffset  = [ZE0offset ZE1offset ZE2offset];
%
% %% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% % Query User for the Azmith Angle of the 3 Laser Beams
% %
% % NOTE: Magnetic NORTH is NOT True North.
% %       The Magnetic Decllination is 13 degrees 10 sec West
% %       Declination is negative, Inclination is 67 degrees 13 minutes
% %       Magnetic Field Strength is 52497.8 nT
% %       City College van location
% %       Latitude:  40 deg 49min 16.6 sec North
% %       Longitude: 73 deg 56min 54.6 sec West
% %       Convent Ave is heading 28.9 deg from True North
% %       Convent Ave is heading 43.0 deg from Magnetic North
% %       Van Heading is 299.33 degrees at City College
% Truck_AZ0 = 0;  % Beam #0 deg from North (azmith) relative to optical table
%                 % (verticle has no compass directional
%                 % component if verticle)
% Truck_AZ1 =120; % Beam #1 degrees from North (azmith)
%                 % relative to optical table
% Truck_AZ2 =240;% Beam #2 degrees from East (azmith)
%                 % relative to optical table
% prompt5 = {...
%     ['Beam0: Azmith Angle (Heading Relative to Front of Optical Table)'...
%         ' of Verical Beam, degrees:'],...
%     ['Beam1: Azmith Angle (Heading Relative to Front of Optical Table)',...
%         ' of North Beam, degrees:'],...
%     ['Beam2: Azmith Angle (Heading Relative to Front of Optical Table)',...
%         ' of East Beam, degrees:']};
% dlg_title5 = 'Heading Angle of Beams Referenced to Front of Optical Table';
% num_lines5 = 1;
% def5 = {num2str(Truck_AZ0),num2str(Truck_AZ1),num2str(Truck_AZ2)};
% answer5 = inputdlg(prompt5,dlg_title5,num_lines5,def5);
% AZ0 = str2num(answer5[15]);  % Azmith angle of verticle beam (Beam0)
% AZ1 = str2num(answer5{2});  % Azmith angle of North-ish beam (Beam1)
% AZ2 = str2num(answer5{3});  % Azmith angle of East-ish beam (Beam2)
%
% Truck_relative_AZ = [AZ0 AZ1 AZ2];
% AZ0offset = AZ0 + AZ;   % Beam #0 should have no bearing as it should be
%                         % verticle if the optical table was level
% AZ1offset = mod(AZ1+AZ,360);    % Beam 1 is usually North
%                                 % and Beam 2 is usually East
% AZ2offset = mod(AZ2+AZ,360);    % but YOU MUST look at the Azimith
%                                 % ZE0 angles to be sure
% AZoffset     = [AZ0offset AZ1offset AZ2offset];
% V0V1V2_files = {'none','none','none'};  % Cells of last three data files
%                                         % used to find wind vectors
% %% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% % For the Spectrum of Interest (SOI) calculate the
% % index within the pre Trigger spectrums for the start and end of SOI
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% IpreTRstart_SOI = IpreTRstart     + IpreTR_Spec_Start - 1;
%             % IpreTRstart_SOI: Index of the pre trigger spectrum (spectrum
%             % precceding the fiber tip back reflected pulse) to be used as
%             % the Background (NOISE) of the current pulse having the lowest
%             % frequency.
% IpreTRend_SOI   = IpreTRstart_SOI + Nspec_bins - 1;
%             % IpreTRend_SOI: Index of the pre trigger spectrum (spectrum
%             % precceding the fiber tip back reflected pulse) to be used as
%             % the Background (NOISE) of the current pulse having the
%             % highest frequency.
%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Directory and File Name Initialize Information From User
%
% Have User Confirm Path to Directory and Approximate File Name
% Since names will be based on the date/time at the start of the run
% an approximate name will be a good starting point to find the
% first data file writen to the directory or the newest file after
% a long break in taking data (given not past midnight)
file_path = uigetdir(file_path,...
                     'Find Directory Holding Wind Vector Data to Plot');

% Have User Confirm Approximate Name of the First Data (SPECTRUM) File
prompt = {'If Path to Wind Vectors Files is correct:',...
        ['Enter Approximate (model) Name of Wind Vector File'...
        ' yyyymmddTHHMMSS_WindVec_CCCCC_ssssssS:']};
name = 'Input Approximate Name (yyyymmddTHHMMSS) of Wind Vector File';
promptWidth = ceil(max(length(file_path)+60,length(file_name))*1.25);
numlines = [1, promptWidth; 1, promptWidth];  % Input Lines, Input Width
defaultanswer       = {file_path,file_name};
options.Resize      = 'on';
options.WindowStyle = 'normal';
answer = inputdlg(prompt,name,numlines,defaultanswer,options);

file_path      = (cell2mat(answer(1)));
file_name      = (cell2mat(answer(2)));
file_path_name = fullfile(file_path, file_name);

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Get Contents of Current Directory Excluding none file items
%
dir_data  = dir([file_path,filesep]);% Get complete contents of current dir
dir_index = [dir_data.isdir];        % Check to see if entry is a directory
file_list = {dir_data(~dir_index).name}'; % Make list of file names
                                          % excluding directories
% Finding Earliest Modified File in Directory of Saved Data may NOT work as
% the spectrum txt files, SNAP and other files will be included.
% Need to selectively make list xxxxxxxxTxxxxxx_BeamXVel.txt files
file_list_WindVec = dir([file_path,filesep,'*',file_name_suffix_WindVlVcDr,'*.mat']);
file_list_all_WindVec = [];
for dd = 1: numel(file_list_WindVec)
    file_list_all_WindVec = [file_list_all_WindVec; file_list_WindVec(dd).name];
end
N_WindVecfiles = size(file_list_all_WindVec,1); % Number of files of Wind Vectors
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Data Format to be placed in Cell Array
%  1)Largest Beam Measuremnt Number used in Current Calculations
%  2)Range (meters)
%  3)Time Last Measurment used in calculations
```

```
%  4)Delta Time (Time Last Measurment - Time Earliest Measurment)
%-----
%  5)Velocity of Beam0 used in Calculations (m/s)
%  6)Intensity of Peak used to calculate Velocity of Beam0
%  7)Time of Beam0 used in Calculations (yyyymmddThhhmmmss)
%-----
%  8)Velocity of Beam1 used in Calculations (m/s)
%  9)Intensity of Peak used to calculate Velocity of Beam1
% 10)Time of Beam1 used in Calculations (yyyymmddThhhmmmss)
%-----
% 11)Velocity of Beam2 used in Calculations (m/s)
% 12)Intensity of Peak used to calculate Velocity of Beam1
% 13)Time of Beam2 used in Calculations (yyyymmddThhhmmmss)
%-----
% 14)Horizontal Wind Velocity EAST Component (m/s)
% 15)Horizontal Wind Velocity NORTH Component (m/s)
% 16)Verticle Wind Velocity Component (m/s)
%-----
% 17)Wind Speed (Horizontal) (m/s)
% 18)Wind Heading (Horizontal) (degrees)
% 19)Wind Calculation Confidence Index % to be added
WindVecArrayHead =['Meas.No. ', 'Range(m) ','TimeLast ','DeltaT ',...
                   'VelBeam0 ','PeakInstenity0 ','TimeBeam0 ',...
                   'VelBeam1 ','PeakInstenity1 ','TimeBeam1 ',...
                   'VelBeam2 ','PeakInstenity2 ','TimeBeam2 ',...
                   'HorVelEAST ','HorVelNORTH ','VerVel ',...
                   'WindSpeedH ','WindHeadingH '];
% Create Cell Array to hold data that will be calculated bellow
%WindVecData01 = cell(1,18,N_BeamVelfiles);

file_time_0 = [];
file_time_1 = [];
file_time_2 = [];

VrBeam0 = [];
VrBeam1 = [];
VrBeam2 = [];

V0 = [];
V1 = [];
V2 = [];

HorSpeed = [];
HorAZ    = [];
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Read Data Files and Make Calculations

iDataSet = 1;
for iDataSet = 1:N_WindVecfiles
    % Data in Cell Array
    %  1)Largest Beam Measuremnt Number used in Current Calculations
    %  2)Range (meters)
    %  3)Time Last Measurment used in calculations
    %  4)Delta Time (sec) (Time Last Measurment - Time Earliest Measurment)
    %-----
    %  5)Velocity of Beam0 used in Calculations (m/s)
    %  6)Intensity of Peak used to calculate Velocity of Beam0
    %  7)Time of Beam0 used in Calculations (yyyymmddThhhmmmss)
    %-----
    %  8)Velocity of Beam1 used in Calculations (m/s)
    %  9)Intensity of Peak used to calculate Velocity of Beam1
```

```
% 10)Time of Beam1 used in Calculations (yyyymmddThhhmmmss)
%-----
% 11)Velocity of Beam2 used in Calculations (m/s)
% 12)Intensity of Peak used to calculate Velocity of Beam1
% 13)Time of Beam2 used in Calculations (yyyymmddThhhmmmss)
%-----
% 14)Horizontal Wind Velocity EAST Component (m/s)
% 15)Horizontal Wind Velocity NORTH Component (m/s)
% 16)Verticle Wind Velocity Component (m/s)
%-----
% 17)Wind Speed (Horizontal) (m/s)
% 18)Wind Heading (Horizontal) (degrees)
% 19)Wind Calculation Confidence Index % to be added
% Read Wind Vector Cell Array
load(fullfile(file_path,file_list_all_WindVec(iDataSet,:)));

%     load(fullfile(file_path,file_list_Vel_all_WindVec(iDataSet,:)),...
%                                         ['wind_vecs', num2str(iDataSet)]);
%     genvarname(['wind_vecs', num2str(iDataSet)] )= ...
%          load(fullfile(file_path,file_list_Vel_all_WindVec(iDataSet,:)));



% Ask user how many meaurements to plot
N_meas_plot     = 20; % Number of Wind Measurments to plot on one graph
RangeGate_start = 1;  % Range Gate of interest to start plot
RangeGate_end   = 56; % Range Gate of interest to end plot (Max=64-NpreTRtip_spec)

% Index of Range Gate after pre Trigger and tip relection to start plot
iRange_start = NpreTRtip_spec + RangeGate_start;
% Index of Range Gate after pre Trigger and tip relection to end plot
iRange_end   = NpreTRtip_spec + RangeGate_end;

% Get the times of all calculations in the cell array .met file
WtimeAllnum = [];  % Variable to hold the times of the measurments
N_BeamVelMeas = size(WindVecData01,3);
for tvc = 1:N_BeamVelMeas
%       posT = regexp(WindVecData01{1,TimeOfCal_i,tvc}(1,:),'T');
%       % Just the hhmmss of 1st element
%       WtimeAll = [WtimeAll; WindVecData01{1,3,tvc}(1,posT+1:end)];
        WtimeAllnum = [WtimeAllnum, datenum(datevec(WindVecData01....
                        {1,TimeOfCal_i,tvc}(1,:),'yyyymmddTHHMMSS'))];
end
WtimeAllmin = WtimeAllnum(1);
WtimeAllmax = WtimeAllnum(end);

% Test Plot of Irregular spaced time on X axis
% plot(WtimeAllnum,[1:177])
% set(gca,'XTick',WtimeAllnum)
% datetick('x','HH:MM','keepticks')
%
% To allow zooming see "Datetick with zooming panning and subplots"
% www.mathworks.com/matlabcentral/fileexchange/20422-datetick-with-
% zooming-panning-and-subplots
% Date formatted tick labels, automatically updated when zoomed or
% panned. Arguments are identical to those of DATETICK. Allows 2 or
% more subplots, keeps all of them synched using LINKPROP.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Query User to Time to Plot
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
prompt100 = {['Start Time of Plot (Max= ',...
```

```
                    datestr(WtimeAllmax,'HH:MM'),')'],...
                    ['Number of Measurments to Plot After Start Time']};
    dlg_title100 = 'Times to Plot';
    num_lines100 = 1;
    def100       = {num2str(datestr(WtimeAllmin,'HH:MM')),num2str(N_meas_plot)};
    answer100    = inputdlg(prompt100,dlg_title100,num_lines100,def100);
    TstartUser   = [answer100[15],':00']; % Time User wants to start to plot
    N_meas_plot  = str2num(answer100{2});% Number of calculations to plot

% Find the closest Time to start the plots from and go to start+N_meas_plot
TstartPlot_i = 1; % Time to start the plot - Have this Adjust to users entry
TstopPlot_i  = TstartPlot_i + N_meas_plot - 1; % Have this Adjust to users entry



    % For best results with multiple subplots, call datetick2 from the axis
    % that has the widest range of dates. This functionality could be
    % included, but would probably needlessly bloat the code.

    % Cardianl Coordinates
    % North = 0   , East = 90
    % South = 180, West = 270
    % Cardinal Degrees INCREASE in the CLOCKWISE direction.
    % This is the opposite of how the angles are handled in the
    % wind_calculation function.
    %Truck_cardianlAZ_gross = 270 - Truck_cardianlAZ -180;
    %phic = (Truck_cardianlAZ_gross)*pi/180; %
    %Truck_cardianlAZ_moshary = 270 - Truck_cardianlAZ -180;
    %phic = (Truck_cardianlAZ_moshary)*pi/180; %
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Prepare Variables to hold Range, Wind Speed, Wind Direction, Time
    WrangeSet = zeros(size(WindVecData01{1,Range_i,3},1),N_meas_plot);
    WSpSet    = zeros(size(WindVecData01{1,WindSpeedHoriz_i,3},1),N_meas_plot);
    WHdSet    = zeros(size(WindVecData01{1,WindHeading_i,3},1),N_meas_plot);
    WtimeSet  = zeros(size(WindVecData01{1,TimeOfCal_i,3},1),N_meas_plot);

    % Create Figure for Wind Barb to be Display inside of *** If you do
    % not define the plot first the sucessive wind barbs are not correctly
    % sized and will be distored.
    figure(1)
    AxisStartX = WtimeAllnum(TstartPlot_i) - (WtimeAllnum(TstartPlot_i+1)-
(WtimeAllnum(TstartPlot_i)));
    AxisStopX  = WtimeAllnum(TstopPlot_i) + (WtimeAllnum(TstopPlot_i)-
(WtimeAllnum(TstopPlot_i-1)));

    %line([1 N_meas_plot],[0 RGS*(iRange_end - iRange_start)],'color','w')
    %line([1 20],[0 2600],'color','w')
    line([AxisStartX  AxisStopX],...
                        [0 RGS*(iRange_end - iRange_start)],'color','w')
    pbaspect([1 1 1])   % Manually setting plot box aspect ratio
                        % disables the MATLAB stretch-to-fill feature
                        % (stretching of the axes to fit the window)
    %axis([0 N_meas_plot+1 0 RGS*(iRange_end - iRange_start)])
    %axis([WtimeAllnum(TstartPlot_i) WtimeAllnum(TstopPlot_i) 0 RGS*(iRange_end -
iRange_start)])
    axis([AxisStartX AxisStopX 0 RGS*(iRange_end - iRange_start)])
    box on
```

```
      for iWindVec = 1:N_BeamVelMeas
        if iWindVec>= 3
            Wrange = WindVecData01{1,Range_i,iWindVec};
            WSp    = WindVecData01{1,WindSpeedHoriz_i,iWindVec}; % Wind Speed (Note
Curly Brackets)
            WHd    = WindVecData01{1,WindHeading_i,iWindVec}; % Wind Heading (Note
Curly Brackets)
            Wtime1 = WindVecData01{1,TimeOfCal_i,iWindVec};
            Wtime  = datenum(datevec(WindVecData01....
                        {1,TimeOfCal_i,iWindVec},'yyyymmddTHHMMSS'));
            % WrangeSet(:,mod(iWindVec-3,N_meas_plot)+1) =  Wrange;
            WrangeSet                = circshift(WrangeSet,[0 -1]);
            WrangeSet(:,N_meas_plot) = Wrange;
            WSpSet                   = circshift(WSpSet,[0 -1]);
            WSpSet(:,N_meas_plot)    = WSp;
            WHdSet                   = circshift(WHdSet,[0 -1]);
            WHdSet(:,N_meas_plot)    = WHd;
            WtimeSet                 = circshift(WtimeSet,[0 -1]);
            WtimeSet(:,N_meas_plot)  = Wtime;
            % Note: There will be some data before the iRange_start
            % index because in the TakelidarXX.m program the data was
            % processed for the gate of the pre trigger spectrums plus the
            % tip back reflection not pre trigger spectrums plus the
            % tip back reflection plus 1.
if 8==10
    % Remember to plot knots or keep as m/s??
    % Use Time on X axis
            for ttt = 1:N_meas_plot
                for rrr = iRange_start:iRange_end %  - iRange_start
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                    windbarb_DS01(WtimeSet(1,ttt),(rrr - iRange_start)*RGS,...
                            WSpSet(rrr,ttt),WHdSet(rrr,ttt),0.01,1,'b')
                end
            end
            title('Doppler Wind Lidar') % Move to end and put date in title and time
on axis
            xlabel('Measuremsnt Number')
            ylabel('Range (meter)')
            axis([AxisStartX AxisStopX 0 RGS*(iRange_end - iRange_start)])

end
if 8==10  % GOOD PLOTTING ROUTINE 1
    % Remember: ploted as m/s
    % Plot Time (Hours:Minutes) on X axis
    % NOTE : Important to set up the plot area befor plotting windbarbs by
    % using the line command. For example:
    % line([WtimeAllnum(157)  WtimeAllnum(177)],[0 RGS*(iRange_end -
iRange_start)],'color','w')

            for ttt = 1:N_meas_plot
                for rrr = iRange_start:iRange_end % Using WtimeSet(rrr,ttt)
                  windbarb_DS01(WtimeSet(rrr,ttt),(rrr - iRange_start)*RGS,...
                            WSpSet(rrr,ttt),...
                            mod(180 + WHdSet(rrr,ttt),360),0.01,1,'b')
                end
            end
            title('Doppler Wind Lidar') % Move to end and put date in title and time
on axis
            xlabel('Measuremsnt Time (HH:MM)')
            ylabel('Range (meter)')
           %set(gca,'XTick',WtimeAllnum(N_BeamVelMeas-N_meas_plot-1: N_BeamVelMeas))
           %set(gca,'XTick',WtimeSet(1,TstartPlot_i:TstopPlot_i))
```

```
            set(gca,'XTick',WtimeSet(1,[TstartPlot_i:4:TstopPlot_i,TstopPlot_i]))
            datetick('x','HH:MM','keepticks')
%            axis([AxisStartX AxisStopX 0 RGS*(iRange_end - iRange_start)])

            % Dynamically adjust space at start and finish of x-axis so the
            % wind barbs can be seen otherwise they may trail off the edge
            % of the visible plot
            axisBoarder = ((WtimeSet(1,TstopPlot_i) - ...
                                WtimeSet(1,TstartPlot_i))/N_meas_plot);
            axis([WtimeSet(1,TstartPlot_i) - axisBoarder...
                   WtimeSet(1,TstopPlot_i) + axisBoarder ...
                                0 RGS*(iRange_end - iRange_start)])

end
if 8==10   % GOOD PLOTTING ROUTINE 2
    % Remember to plot as m/s
    % Plot Time (Hours:Minutes) on X axis
    % NOTE : Important to set up the plot area befor plotting windbarbs by
    % using the line command. For example:
    % line([WtimeAllnum(157)  WtimeAllnum(177)],[0 RGS*(iRange_end -
iRange_start)],'color','w')

    % Note: the windbarb.m function from Google ploted the tail (feathers)
    % of the wind barb pointing in to the direction the wind is going
    % towrds.  This is opposit of the normal wind barb convention.  The tip
    % The tip of the wind barb, end of the long line (no feathers), usually
    % points in the direction the wind is going towards.
    % Use mod(180+Heading,360)


            for ttt = 1:N_meas_plot
                for rrr = iRange_start:iRange_end % USING WtimeAllnum
                  windbarb_DS01(WtimeAllnum(ttt),(rrr - iRange_start)*RGS,...
                            WSpSet(rrr,ttt),...
                            mod(180 + WHdSet(rrr,ttt),360),0.01,1,'b')
                end
            end
            title('Doppler Wind Lidar') % Move to end and put date in title and time
on axis
                                        % Put in Date and min max time in
                                        % title
            xlabel('Measuremsnt Time (HH:MM)')
            ylabel('Range (meter)')
           %set(gca,'XTick',WtimeAllnum(N_BeamVelMeas-N_meas_plot-1: N_BeamVelMeas))
%            set(gca,'XTick',WtimeAllnum(TstartPlot_i:TstopPlot_i))   %  <-----------
Check time for correctness
            % Do Not number every x axis data but do make sure to get the
            % 1st and the last data sets
            set(gca,'XTick',WtimeAllnum([TstartPlot_i:4:TstopPlot_i,...
                                                    TstopPlot_i]))  %  <----------
-Check time for correctness

%              set(gca,'XTick',WtimeSet(1,TstartPlot_i:TstopPlot_i))
            datetick('x','HH:MM','keepticks')
            axis([AxisStartX AxisStopX 0 RGS*(iRange_end - iRange_start)])

end
% if 8==9
%     % Remember to plot knots or keep as m/s??
%     % Plot Time (Hours:Minutes) on X axis
%     % NOTE : Important to set up the plot area befor plotting windbarbs by
```

```
%     % using the line command. For example:
%     % line([WtimeAllnum(157)  WtimeAllnum(177)],[0 RGS*(iRange_end -
iRange_start)],'color','w')
%
%             for ttt = 1:N_meas_plot
%                 for rrr = iRange_start:iRange_end %- iRange_start
%%%%%%%%%%%%%%%%%%%%%%%%
%                     windbarb(ttt,(rrr - iRange_start)*RGS,...
%                             WSpSet(rrr,ttt),WHdSet(rrr,ttt),0.01,1,'b')
%                 end
%             end
%             title('Doppler Wind Lidar') % Move to end and put date in title and time
on axis
%             xlabel('Measuremsnt Time (HH:MM)')
%             ylabel('Range (meter)')
% %             set(gca,'XTick',WtimeAllnum(N_BeamVelMeas-N_meas_plot-1:
N_BeamVelMeas))
%             set(gca,'XTick',WtimeAllnum(158:177))
%             datetick('x','HH:MM','keepticks')
%
% end
% if 8==10
%     % Remember to plot knots or keep as m/s??
%     % Use Time on X axis
%             for ttt = 1:N_meas_plot
%                 for rrr = iRange_start:iRange_end - iRange_start
%                     windbarb(ttt,(rrr - iRange_start)*RGS,...
%                             WSpSet(rrr,ttt),WHdSet(rrr,ttt),0.01,1,'b')
%                 end
%             end
%             title('Doppler Wind Lidar') % Move to end and put date in title and time
on axis
%             xlabel('Measuremsnt Number')
%             ylabel('Range (meter)')
% end


            if 10==8 % barb plot not working well-Code From Sameh
                Vertical_Elemnts = length(V1);
                Horizontal_Elemnts = 4;
                U=repmat(V1',1,Horizontal_Elemnts);
                V=repmat(V2',1,Horizontal_Elemnts);
                [yearW monthW dayW hourW minW secW] = datevec(now);
                timeW   = [1:0.25:1-0.25+0.25*Horizontal_Elemnts];
                heightW = [1: 2: 2*size(U,1)];
                % [latgrat_T,longrat_T] = meshgrid(heightW,timeW);
                % latgrat=latgrat_T';
                % longrat=longrat_T';
                [latgrat,longrat] = meshgrat(heightW,timeW);
                % X = U(:);
                % Y = V(:);
                H= latgrat;
                T= longrat;
                % H= latgrat(:);
                % T= longrat(:);
                eee = 20;
                for dd = 1:4
                    for ee = 1:size(U,1);
                        if (U(ee,dd) > eee || U(ee,dd) < -eee)
                            U(ee,dd)= 0;
                        end
                        if (V(ee,dd) > eee || V(ee,dd) < -eee)
```

```
                    V(ee,dd)= 0;
                end
            end
        end
        barb_fig = figure(3)  ;% Wind Barb PLOT
        plot_windbarb(U*2, V*2, H,T,0.5,'ms')
        %axis([xscale(1) xscale(16) -100 2200]);
        ylabel('heightW (m)');
        xlabel('timeW');
        title(['Wind Barb measured on:  ', date]);

        quiver_plot = figure(4) ;
        quiver( -1*U,-1*V), grid on
        % axis([0 17 0 23]);
        title(['Wind Speed and Direction measured on:  ', date]);
        xlabel('timeW ');
        ylabel('heightW (m)')
    end

    %% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Print to Screen: Wind Velcity Vectors for u, v, w for 1000 to 1500m

    % Use 'disp' to supress Matlab ptinting 'Var = ', new line the values
    % Range (Meters),
    % Horizontal Components (Vector EAST, Vector NORTH, Vector UP)
    % Horizontal SPEED and DIRECTION (wind is goinng TOWARDS)
    disp('    Range     Vu=Ve     Vv=Vn     Vw=Vup    SpeedH    DirH')
    %disp(gallery('normaldata',[8 6],0));% 'gallery' makes fake test data
    %         if iBeam == 1
    %             V0 =  VrBeam0;
    %             % Only Print first vector
    %             % rtemp = sprintf('%5.0f \n',(20:32)'*RGS); % try to format
    %             disp([(20:32)'*RGS, V0(20:32)'])
    %         elseif iBeam==2
    %             % V1 = (VrBeam1 - VrBeam0.*cos(ZE1offset)) ./sin(ZE1offset);
    %             % Only Print 1st Two Vectors
    %             disp([(20:32)'*RGS,V1(20:32)', V0(20:32)'])
    %         elseif iBeam>=3
    %             % Print All Vectors and Calculations
    %             disp([(8:32)'*RGS,V2(8:32)' V1(8:32)' V0(8:32)' ...
    %                 HorSpeed(8:32)' HorAZ(8:32)']) % WHYS is HorAZ col vec?
<----------------???
    %         end




    %% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %  Data in Cell Array
    %  1)Largest Beam Measuremnt Number used in Current Calculations
    %  2)Range (meters)
    %  3)Time Last Measurment used in calculations
    %  4)Delta Time (sec) (Time Last Measurment - Time Earliest Measurment)
    %-----
    %  5)Velocity of Beam0 used in Calculations (m/s)
    %  6)Intensity of Peak used to calculate Velocity of Beam0
    %  7)Time of Beam0 used in Calculations (yyyymmddThhhmmmss)
    %-----
    %  8)Velocity of Beam1 used in Calculations (m/s)
    %  9)Intensity of Peak used to calculate Velocity of Beam1
    % 10)Time of Beam1 used in Calculations (yyyymmddThhhmmmss)
```

```
%-----
% 11)Velocity of Beam2 used in Calculations (m/s)
% 12)Intensity of Peak used to calculate Velocity of Beam1
% 13)Time of Beam2 used in Calculations (yyyymmddThhhmmmss)
%-----
% 14)Horizontal Wind Velocity EAST Component (m/s)
% 15)Horizontal Wind Velocity NORTH Component (m/s)
% 16)Verticle Wind Velocity Component (m/s)
%-----
% 17)Wind Speed (Horizontal) (m/s)
% 18)Wind Heading (Horizontal) (degrees)
% 19)Wind Calculation Confidence Index

%     %  1)Largest Beam Measuremnt Number
%     WindVecData01{1, 1,iBeam} = iBeam*ones(N_spec_acc,1);
%
%     %  2)Range (meters)
%     WindVecData01{1, 2,iBeam} = Range_vec;
%
%     %  3)Time of calculations
%     WindVecData01{1, 3,iBeam} = repmat(file_time_new,N_spec_acc,1);
%
%     %  4)Delta Time (sec) (Time Last Measurment - Time Earliest
Measurment)
%     WindVecData01{1, 4,iBeam} = deltaS_vec;
%
%     % datevec('20111205T134520','yyyymmddTHHMMSS')
%     % t1=datenum(datevec('20111205T134530','yyyymmddTHHMMSS'))
%     % t2=datenum(datevec(file_time_new,'yyyymmddTHHMMSS'))
%     % deltat=datevec(t2-t1)
%
%     %  5)Velocity of Beam0 used in Calculations (m/s)
%     WindVecData01{1, 5,iBeam} = VrBeam0;
%
%     %  6)Intensity of Peak used to calculate Velocity of Beam0
%     WindVecData01{1, 6,iBeam} = [];
%
%     %  7)Time of Beam0 used in Calculations (yyyymmddThhhmmmss)
%     WindVecData01{1, 7,iBeam} = repmat(file_time_0,N_spec_acc,1);
%
%     %  8)Velocity of Beam1 used in Calculations (m/s)
%     WindVecData01{1, 8,iBeam} = VrBeam1;
%
%     %  9)Intensity of Peak used to calculate Velocity of Beam1
%     WindVecData01{1, 9,iBeam} = [];
%
%     % 10)Time of Beam1 used in Calculations (yyyymmddThhhmmmss)
%     WindVecData01{1,10,iBeam} = repmat(file_time_1,N_spec_acc,1);
%
%     % 11)Velocity of Beam2 used in Calculations (m/s)
%     WindVecData01{1,11,iBeam} = VrBeam2;
%
%     % 12)Intensity of Peak used to calculate Velocity of Beam1
%     WindVecData01{1,12,iBeam} = [];
%
%     % 13)Time of Beam2 used in Calculations (yyyymmddThhhmmmss)
%     WindVecData01{1,13,iBeam} = repmat(file_time_2,N_spec_acc,1);
%
%     % 14)Horizontal Wind Velocity EAST Component (m/s)
%     WindVecData01{1,14,iBeam} = V1';
%
%     % 15)Horizontal Wind Velocity NORTH Component (m/s)
%     WindVecData01{1,15,iBeam} = V2';
```

```
            %
            %     % 16)Verticle Wind Velocity Component (m/s)
            %     WindVecData01{1,16,iBeam} = V0';
            %
            %     % 17)Wind Speed (Horizontal) (m/s)
            %     WindVecData01{1,17,iBeam} = HorSpeed';
            %
            %     % 18)Wind Heading (Horizontal) (degrees)
            %     WindVecData01{1,18,iBeam} = HorAZ';
            %
            %     % 19)Wind Calculation Confidence Index - to indicate closeness of
            %     % measurments (t2-t2) / (t1-t0) If 1 then the time between
measurments
            %     % are balanced The measurment time diffreence between t0 and t1 is
less
            %     % significant than the time difference between t1 and t2
            %     % or
            %     % ((t2-t2) / (t1-t0)) / ((t2-t1)+(t1-t0))
        end
    end

end

%  TEST PLOT
% quiver(WindVecData01{1, 16,177}',WindVecData01{1, 14,177}',WindVecData01{1,
15,177}',(1:64))
%% Save Wind Vector Calculation to File


% TEST OF MAT save
% Save Cell Array of Data and Calculations
% Query User if tehy want to save the Wind Calculations
%save (fullfile(file_path,'WindVecData01'),'WindVecData01')




% Save the 3 files used to find the wind vectors V0V1V2_files
% Append BeamX Velocity, Time, File Name, Comments Info to Meta File
% Prepare Data to be Saved to Meta File

% FIX THIS HERE
% Can NOT PUT TEXT AND NUMBERS IN SAME ARRAY--USE CELLS?
% Prepare Velocity, Vector, Wind Direction, Wind Velocity Meta Information
% Meta_Vec_pre = ...
%     {datestr(last_file_time,time_format),...    % Date & time
%     [file_name_wo_ext,file_ext],...
%     ['Data ',V0V1V2_files],...
%     ['TruckZE=',num2str(Truck_relative_ZE(mod(iBeam-1,3)+1))],...
%     ['TruckAZ=',num2str(Truck_relative_AZ(mod(iBeam-1,3)+1))],...
%     cell2mat(answer2)};
%
% % If function 'ex_func' is not available use a loop
% Meta_Vec_header = Meta_Vec_pre;
% for k = 1:length(Meta_Vec_header)
%     in_datatype = class(Meta_Vec_header{k});                %<--------------------
fix
%     switch in_datatype
%     case 'char'  %Cell is a string, do nothing
%     case 'double'
```

```
%         Meta_Vec_header{k} = num2str(Meta_Vec_header{k});
%     end
% end
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Write Metafile Data to File in Directory for Horizontal Wind Vector
% The 1st Line of the Text File Will Have Info That Will Not Be Repeated
% [path_string_M,file_name_wo_ext_M,file_ext_M]= fileparts(file_name_M);
% file_path_MwVec = [file_name_wo_ext_M,...
%                                 file_name_suffix_WindVlVcDr,'.txt'];
% fid_M = fopen(fullfile(file_path_M,file_path_MwVec), 'a');
% fprintf(fid_M,'%s, ', Meta_Vec_header{:}); % Write Comma Seperated
% fprintf(fid_M,'\n'); % Write Cariage Return to File
% fclose(fid_M);


 %% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 % Increase index iDataSet, the total number of the Data sets
 display(['Wind Vector Data Set: ',num2str(iDataSet)])
 iDataSet = iDataSet + 1;   %

 %% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 % Query User If They Would Like to Continue Taking Data
%  options.Interpreter = 'tex';
%  % Include the desired Default answer
%  options.Default = 'YES- Continue Taking Data';
%  % Create a string for the question
%  qstring = 'Would You Like To Continue To Take Data';
%  choice = questdlg(qstring,'TAKE MORE DATA ?',...
%                    'YES- Continue Taking Data','No',options);
%  switch choice
%     case 'YES- Continue Taking Data'
%         % Do Nothing and continue in the while loop
%     case 'No'
%         UserQuit = true;    % User wants to quite
%  end
%
% end


% END OF NORMALLY EXECUTED CODE

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%                                                          %%%%%%%%%%%%
%%%%%%% BELLOW ARE PROTIONS OF TEST CODES I TRIED OUT %%%%%%%%%%%%
%%%%%%%                                                          %%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


if 5==9
A = randn(10,1);

% tequnique to have user enter a variable name and value
checkLabels = {'Save sum of A to variable named:' ...
               'Save mean of A to variable named:'};
varNames = {'sumA','meanA'};
items = {sum(A),mean(A)};
```

```
export2wsdlg(checkLabels,varNames,items,...
             'Save Sums to Workspace');


A = randn(10,1);
checkLabels = {'Save sum of A to variable named:' ...
               'Save mean of A to variable named:'};
varNames = {'sumA','meanA'};
items = {sum(A),mean(A)};
export2wsdlg(checkLabels,varNames,items,...
             'Save Sums to Workspace');


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
d = dir;
str = {d.name};
[s,v] = listdlg('PromptString','Select a file:',...
                'SelectionMode','single',...
                'ListString',str);



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% VIDEO CAMERA INPUT


% Select or enter a file name for saving a figure
% as an image in one of four formats, described in a cell array.
uiputfile({'*.jpg;*.tif;*.png;*.gif','All Image Files';...
          '*.*','All Files' },'Save Image',...
          'C:\Work\newfile.jpg')

%% TO MAKE A BUTTON ON A FIGURE WINDOW
qf = figure;
h = uicontrol('Position',[20 20 200 40],'String','Continue',...
              'Callback','uiresume(gcbf)');
disp('This will print immediately');
uiwait(gcf);
disp('This will print after you click Continue');
close(qf);
end
```

Part 6: Matlab program: test30s06.m

```
% test30s06.m
%
% clear all;
% David Santoro
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% PURPOSE:
% This porgram will read the last 'my_30_Seconds_file.txt' file saved y the
% modified SNAP program. Doppler lidar data taken by using a modified SNAP
% program with a FPGA code that takes 8192 data points, uses 128
% concecutive points to compute 64 FFT spectrums, add the newest spectrums
% to a running sum of spectrums until 10,000 have been colleted. After
% 10000 summations these are appended to a file until 1 minutes has
% been collected.  Then the user will be instructed to readjsut the laser
% beams direction and to rerpeat the data collection.


time_format      = 30;% Time Format 30 = 'yyyymmddTHHMMSS' (ISO 8601)
datetimeStart    = datestr(now,time_format);% Get program start date & time
file_finish_time = 0; % Variable to hold time when SNAP finishes with file
file_wait_time   = 3; % Seconds to wait before testing to see
                      % if SNAP has finished writing to a file

dBUG            = 1;% DEBUGGING SWITCH
                    % =1 (true) Do debugging,
                    % =0 (false) No debuging

dbugPC    = 0;      % LOCATION OF COMPUTER
                    % Use to precnfigure filenames, directories, etc
                    % =0 for WinXP 64bit Doppler PC (CCNY),
                    % =1 for home (santoro),
                    % =2 for work (santoro),
                    % =3 for work T647 (santoro),
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Set default SNAP OUTPUT File Type
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
SNAPfiletype = 1;   % File Nomenclature expected from SNAP program
                    % =0, Text File, yyyymmddTHHMMSS.txt
                    % =1, Text File, my_30_Seconds_file.txt
switch SNAPfiletype
case 0
    file_name = datetimeStart;
case 1
% Use if SNAP does not put out Time Stamped File Name format
    file_name = 'my_30_Seconds_file.txt';
end
```

```
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Set default directories and file names based on location of PC
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
switch dbugPC   % =0 for van ,
                % =1 for home,
                % =2 for work,
                % =3 Work T647
                % =4 for _______
    case 0 % =0 for van
      %file_path = 'D:\TEMP_SANTORO\PhD
Matlab\Lidar_Text_Data_Matlab\20111117_Data\' ;
      %file_path =
'D:\TEMP_SANTORO\20111128_Data_SNAPmod\20111128_Data_SNAPmod_60min\';
      %file_path
='D:\TEMP_SANTORO\20111201_Data_SNAPmod\20111201_Data_SNAPmod_Run_60min\';
      file_path
='D:\TEMP_SANTORO\20111213_Data_SNAPmod\20111213_Data_SNAPmod_Run2\';
      %file_path
='D:\TEMP_SANTORO\20111201_Data_SNAPmod\20111201_Data_SNAPmod_Run_60min\20111
201_Data_SNAPmod_Run_60min-30deg\';
      file_name = datetimeStart;
      file_nameXXs = 'my_30_Seconds_file.txt'; % Use if SNAP does not put out
a Time Stamed File Name format
      %file_name_fakeBK =
'BackGround_Noise_092311DelayT0_AB_2.24uS_my_output_file_3_60s.txt';
      file_name_fakeBK = '20111128T125827_bk.txt';
      %file_name = 'HsData_FPGA_ACCUM_7_08172011.txt';
      %file_name = '20111004T131022_my_output_file_3.txt';
      %file_name = '20111004T131022';
      %Time_Length = 14400;    % this should be the # of 500ms FFT
                               % accumulations that have been made *2
    case 1 % =1 for home
      file_path = 'D:\David\PhD\PhD
Matlab\Lidar_Text_Data_Matlab\20111019_Data\';
      %file_path = 'D:\David\PhD\PhD
Matlab\Lidar_Text_Data_Matlab\20111019_Data';
      file_name = datetimeStart;
      file_nameXXs = 'my_30_Seconds_file.txt'; % Use if SNAP does not put out
a Time Stamed File Name format
      file_name_fakeBK = ...
       'BackGround_Noise_092311DelayT0_AB_2.24uS_my_output_file_3_60s.txt';
      %file_name = '20111004T131022_my_output_file_3.txt';
      %file_name = '090911_Hatch_open_170ma_1700ma_my_30_Seconds_file.txt';
      %file_name = '8192x2_data_growing_sawtooth.txt';
     %file_name = '090911_Hatch_closed_170ma_1700ma_my_30_Seconds_file.txt'
      %file_name = '090911_long group of 30s runs my_output_file_3.txt'
      %file_name = '092311DelayT0_AB_2.24uS_my_output_file_3_60s.txt'
      %Time_Length = 30;
    case 2 % =2 for work
      file_path = 'C:\David\CCNY PhD Matlab\Lidar_Text_Data_Matlab\';
      file_nameXXs = 'my_30_Seconds_file.txt'; % Use if SNAP does not put out
a Time Stamed File Name format
      file_name = datetimeStart;
      file_name_fakeBK = ...
       'BackGround_Noise_092311DelayT0_AB_2.24uS_my_output_file_3_60s.txt';
      %file_name = '20111004T131022_my_output_file_3.txt';
      %file_name = '090911_Hatch_open_170ma_1700ma_my_30_Seconds_file.txt';
```

```
      %file_name = '8192x2_data_growing_sawtooth.txt';
     %file_name = '090911_Hatch_closed_170ma_1700ma_my_30_Seconds_file.txt'
     %file_name = '090911_long group of 30s runs my_output_file_3.txt'
      %file_name = '092311DelayT0_AB_2.24uS_my_output_file_3_60s.txt'
      %Time_Length = 30;
    case 3 % =3 for work T647
      file_path = 'C:\Temp\santoro\PhD Matlab\Lidar_Text_Data_Matlab\';
      file_name = datetimeStart;
      file_nameXXs = 'my_30_Seconds_file.txt'; % Use if SNAP does not put out
a Time Stamed File Name format
      file_name_fakeBK = ...
       'BackGround_Noise_092311DelayT0_AB_2.24uS_my_output_file_3_60s.txt';
      %file_name = '20111004T131022_my_output_file_3.txt';
      %file_name = '090911_Hatch_open_170ma_1700ma_my_30_Seconds_file.txt';
      %file_name = '8192x2_data_growing_sawtooth.txt';
     %file_name = '090911_Hatch_closed_170ma_1700ma_my_30_Seconds_file.txt'
     %file_name = '090911_long group of 30s runs my_output_file_3.txt'
      %file_name = '092311DelayT0_AB_2.24uS_my_output_file_3_60s.txt'
      %Time_Length = 30;
    case 4 % =4 for ______
end

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Default File Name Suffixes to be Used
%file_name_suffix_Background = '_bk';     % Background (Noise) Spectrums
file_name_suffix_Background = '';     % temporary
file_name_suffix_BackgroundAVG = '_avg'; % Average Background (Noise) Spec
file_name_suffix_MetaData = '_meta';     % Meta Data

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Query User for Length of Time to Measure Signals and Background (Noise)
Time_Length   = 30;     % Number of seconds to make a measurment
Time_LengthBK = 30;     % Number of  seconds to take an independent, before
                        % measuremnts start, background (noise) measurement

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Check To See When Text File is Writen Then Rename
%
% Each file writen to the directory after XX seconds as file_nameXXs will
% be renamed to yyyymmddThhmmss.txt (Year Month Day 'T' Hours Minutes
% Seconds .txt) format and the orginal file erased. The normal state of the
% folder will not have a file_nameXXs in it.

last_path_file_new = file_name;
%=========================================================================
% IF SNAP WROTE 30 Second File AT END OF PREVIOUS RUN AND WAS NOT
% PROCESSED (ORPHAN FILE) Delete It If It Is More than 10% Older Than
% The Length of Time The Data Should Be Covering.
%
% dir_data = dir(file_nameXXs) % Get the file attributes of XX sec file
% filedate = dir_data.datenum  % Get the files modification date
%
% If a previous text data file exist from old run delete it
% if etime(clock,datevec(filedate))>=(Time_Length*1.1)
%     delete(last_path_file_new)
% end
```

```
%=========================================================================
last_time_name = file_nameXXs;
[path_string, file_name_wo_ext, file_ext] = fileparts(last_time_name);
% Get the parts of the file name
% path_string      = String represneting path to file
% file_name_wo_ext = file name without the extension
% file_ext         = file extentionwhile 1==1
while 1==1
while isempty(dir([file_path,'\',file_nameXXs]))% Check to see if file is
created
    pause(Time_Length*0.15)     % Wait for 15% of total data record length
    display(['Waiting ', num2str(Time_Length*0.15),...
        ' seconds for data file...'])
end

% Now thatthe file exist in the directory get the modification time to use
% as the new file name yyyymmddThhmmss

fileinfo = dir([file_path,'\',file_nameXXs]);

% Change Name of File, Append Suffix (i.e. '_BK') for Background to name
last_path_file     = fullfile(file_path,last_time_name);
% file_name_wo_ext = datestr(now,30);
file_name_wo_ext = datestr(fileinfo(1).datenum,30);
last_path_file_new = fullfile(file_path,[file_name_wo_ext,....
    file_name_suffix_Background,file_ext]);


% If SNAP Has Finished Writing Data, Move File to New Name and extention
% NOTE: Even though the file name may be present in the directory that
% does not mean the file is closed and ready to be read.  You MUST
% check to make sure.
while ~movefile(last_path_file, last_path_file_new)
    if dBUG ==1
        display(['File still open,  Waiting ',...
            num2str(Time_Length*0.025),' seconds: ',datestr(now,31)])
        %display(datestr(now,31))
    end       % Print time if debugging
    pause(Time_Length*0.025); % while unable to make
                              % name change wait and try again
end
display ([file_nameXXs,' -->Renamed To--> ',  [file_name_wo_ext,....
    file_name_suffix_Background,file_ext]])
%last_path_file = datestr(now,30);
%movefile(last_path_file_new,[last_path_file,file_ext]) % Rename file to
SNAP's name
data_file = last_path_file; % Record name of last data file
end
```

CHAPTER 11
Bibliography

1. Angelou, N., et al., *Advancements in Wind Energy Metrology – UPWIND 1A2.3*, T.F. Pedersen and R. Wagner, Editors., 2011. p. 98.

2. Fernando, H.J.S., et al., *Urban Fluid Mechanics: Air Circulation and Contaminant Dispersion in Cities.* Environmental Fluid Mechanics, 2001. **1**(1): p. 107-164.

3. Frehlich, R., et al., *Measurements of Boundary Layer Profiles in an Urban Environment.* Journal of Applied Meteorology and Climatology, 2006. **45**(6): p. 821-837.

4. Harris, M., et al., *Lidar for Turbine Control*, 2006, National Renewable Energy Laboratory: Golden, CO. p. 55.

5. Allwine, K.J., et al., *Overview of URBAN 2000: A Multiscale Field Study of Dispersion through an Urban Environment.* Bulletin of the American Meteorological Society, 2002. **83**(4): p. 521-536.

6. Frehlich, R., Y. Meillier, and M.L. Jensen, *Measurements of Boundary Layer Profiles with In Situ Sensors and Doppler Lidar.* Journal of Atmospheric and Oceanic Technology, 2008. **25**(8): p. 1328-1340.

7. Gryning, S.-E. and E. Batchvarova, *Measuring Meteorology in Urban Areas – Some Progress and Many Problems, Meteorological and Air Quality Models for Urban Areas*, A. Baklanov, et al., Editors., 2009, Springer Berlin Heidelberg. p. 125-131.

8. Hasager, C.B., et al., *Advances in Offshore Wind Resource Estimation, Advances in Wind Energy Conversion Technology*, M. Sathyajith and G.S.S. Philip, Editors., 2011, Springer Berlin Heidelberg. p. 85-106.

9. Ogasawara, T., et al., *Measurement of Aircraft Wake Vortices Using Doppler LIDAR.* Journal of Fluid Science and Technology, 2008. **3**(4): p. 488-499.

10. Settles, G.S., *Fluid Mechanics abd Homeland Security.* Annual Review of Fluid Mechanics, 2006. **38**(1): p. 87-110.

11. McCormick, M.P., *Use of Lidar for Stratospheric Measurements*, L.R. Center, Editor, 1977, NASA: Hampton, VA. p. 26.

12. Murray, J., *Lidar Systems Move to EyeSafe Wavelengths*, in *Lasers & Optronics Magazine*, 1997, Reed Business Information: New York. p. 13.

13. Lidar-UK. *The uses of LiDAR, What applications are there for LiDAR systems?* 2012; Find out all you need to know about LiDAR, its history, its uses and the technologies behind it...]. Available from: http://www.lidar-uk.com/usage-of-lidar/.

14. APS Physics, *This Month in Physics History, December 1958: Invention of the Laser*, in *APS News*, 2003, American Physical Society: College Park, MD. p. 1.

15. Optics 1 Inc. *History of Optics*. [Web Page] 2012; Available from: http://www.optics1.com/optics_history.php.

16. Miller, F.A., *A Postage Stamp History of Optics.* Appl. Spectrosc., 1992. **46**(1): p. 1-17.

17. Natural Sciences Sector of the United Nations Educational, S.a.C.O., *Miracle of Light* in *Natural Sciences Quarterly News Letter*, 2005, Natural Sciences Sector of the United Nations Educational, Scientific and Cultural Organization (UNESCO). p. 2.

18. Alhazen, *Book of Optics* 1021.

19. Clegg, R.M. *Physics 598OM- Optical Microscopy Lectures Spring 2012- Short historical look*. 2012; Course material]. Available from: http://online.physics.uiuc.edu/courses/phys598om/spring12/Lectures/.

20. Taylor, L.S. *Optics Highlight:s An Anecdotal History of Optics from Aristophanes to Zernike- III The Telescope Optics*. 2012; Available from: http://www.ee.umd.edu/~taylor/optics3.htm.

21. Leblanc, D.T. *NDACC Lidar Work Group- Antarctica*. 2009; Available from: http://ndacc-lidar.org/index.php?id=66/Dumont+d%27Urville.htm.

22. Newton, *An experiment to put pressure on the eye*, N.E.B. 8small.jpg, Editor, 1665, Cambridge University Library: Cambridge. p. An experiment to put pressure on the eye, from 'Of Colours'.

23. Tinke, A.P. *History of Particle Characterization*. 2010; Available from: http://www.benelux-scientific.be/fileadmin/user_files/pdf/TD2010/J_J_Tinke_history_particle_characterization_part2.pdf.

24. Technologies, N. *Theory of Light*. 2005; Available from: http://www.nightlase.com.au/education/optics/light.htm.

25. Deshmukh, P.C. and S. Venkataraman, *100 Years of Einstein's Photoelectric Effect*, in *Joint Talk of Indian Physics Association – Chennai Chapter, The Humboldt Club, Chennai, and the Department of Physics, IIT-Madras*, 2006, Indian Physics Association – Chennai Chapter, The Humboldt Club, Chennai, and the Department of Physics, IIT-Madras on April 20, 2005. Published in the Bulletin of Indian Physics Teachers Association: Madras Chennai, India. p. 16.

26. *The Use of the Searchlight in Meteorology.* Monthly Weather Review, 1897. **25**(5): p. 206-207.

27. Novotny, L., *Chapter 5 The history of near-field optics*, in *Progress in Optics*, E. Wolf, Editor, 2007, Elsevier. p. 137-184.

28. Clemesha, B.R., G.S. Kent, and R.W.H. Wright, *Laser Probing the Lower Atmosphere.* Nature, 1966. **209**(5019): p. 184-185.

29. Flamant, P.H., *Atmospheric and Meteorological Lidar: From Pioneers to Space Applications.* Comptes Rendus Physique, 2005. **6**(8): p. 864-875.

30. Elterman, L., *Seasonal Trends of Temperature, Density, and Pressure to 67.6 km Obtained with the Searchlight Probing Technique.* J. Geophys. Res., 1954. **59**(3): p. 351-358.

31. Hecht, J., *Laser Pioneers*. 1st ed, 1991, San Diego: Academic Press. 298.

32. She, C.Y., et al., *Laser Applications in Remote Sensing (SPIE Milestone Series)*, ed. W.B. Grant, 1997.

33. Fiocco, G. and L.D. Smullin, *Detection of Scattering Layers in the Upper Atmosphere (60-140 km) by Optical Radar.* Nature, 1963. **199**(4900): p. 1275-1276.

34. Fiocco, G. and G. Grams, *Observations of the Aerosol Layer at 20 km by Optical Radar.* Journal of the Atmospheric Sciences, 1964. **21**(3): p. 323-324.

35. Amzajerdian, F., et al. *2-micron coherent Doppler lidar for space-based global wind field mapping*. in *Geoscience and Remote Sensing Symposium, 2003. IGARSS '03. Proceedings. 2003 IEEE International*. 2003. Toulouse, France.

36. New York Times, *Laser May be Used to Detect Rough Air*, in *New York Times (1923-Current file)*, 1964, New York Times Publishing Company: New York, N.Y., United States. p. 66-66.

37. University of Washington. *LIDAR Overview*. 2012; Available from: http://forsys.cfr.washington.edu/JFSP06/lidar_technology.htm.

38. Lautenbacher, V.C.C.J., *Urban Meteorology, Meeting Weather Needs in the Urban Community*, 2004, Office of the Federal Coordinator for Meteorological Services and Supporting Research. p. 25.

39. Hicks, B.B., et al., *Urban Turbulence in Space and in Time.* Journal of Applied Meteorology and Climatology, 2011. **51**(2): p. 205-218.

40. Inokuchi, H., H. Tanaka, and T. Ando, *Development of an Onboard Doppler Lidar for Flight Safety - ICAS 2008*, in *26th Congress of International Council of the Aeronautical*

Sciences including the 8th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, 2008: Anchorage, Alaska, USA. p. 8.

41. Chan, P.W., *LIDAR-based turbulence intensity calculation using glide-path scans of the Doppler LIght Detection And Ranging (LIDAR) systems at the Hong Kong International Airport and comparison with flight data and a turbulence alerting system.* Meteorologische Zeitschrift, 2010. **19**(6): p. 549-563.

42. Hannon, S.M., et al., *Application of pulsed Doppler lidar in the airport terminal area*, in *Lidar Remote Sensing for Industry and Environmental Monitoring V, Upendra N. Singh; Kohei Mizutani, Editors, pp.186-197*, U.N. Singh and K. Mizutani, Editors., 2005, SPIE: Lidar Remote Sensing for Industry and Environmental Monitoring V, Upendra N. Singh; Kohei Mizutani, Editors, pp.186-197. p. 186-197.

43. Frehlich, R., *Doppler lidar measurements of winds and turbulence in the boundary layer.* IOP Conference Series: Earth and Environmental Science, 2008. **1**(1): p. 012017.

44. Koch, G.J., *Using a Doppler light detection and ranging (lidar) system to characterize an atmospheric thermal providing lift for soaring raptors.* Journal of Field Ornithology, 2006. **77**(3): p. 315-318.

45. Emeis, S., *Surface-Based Remote Sensing of the Atmospheric Boundary Layer*. Earth and Environmental Science. Vol. 40. 2011: Springer Netherlands.

46. NikNaks, *A diagram of the atmospheric boundary layer*, Atmospheric_boundary_layer.svg, Editor, 2012, http://commons.wikimedia.org/wiki/File:Atmospheric_boundary_layer.svg. p. A diagram of the atmospheric boundary layer.

47. National Weather Service, A., SD Weather Forecast Office. *Facts About Weather Balloons*. 2012 [cited 2012; Available from: http://www.crh.noaa.gov/abr/?n=wxballoonfacts.php.

48. National Weather Service Upper-air Observations Program. *NOAA National Weather Service, Radiosonde Observations* 2012; Available from: http://www.ua.nws.noaa.gov/factsheet.htm.

49. Office of Air Quality Planning and Standards, *Meteorological Monitoring Guidance for Regulatory Modeling Applications*, O.o.A.a.R. U.S. Environmental Protection Agency, Office of Air Quality Planning and Standards, Editor, 2000, http://www.epa.gov/scram001/guidance/met/mmgrma.pdf: Research Triangle Park, NC 27711.

50. Kühn, S., *Anemometer*, Anemometer.jpg, Editor, 2004, Wikimedia Commons.

51. NOAA Photo Library - NOAA Central Library and OAR/ERL/National Severe Storms Laboratory (NSSL), *Wind speed and direction instrument - NOAA.jpg*, W.s.a.d.i.-.

NOAA.jpg, Editor, 1994, NOAA Photo Library. p. Project Vortex mesonet vehicle aerovane for wind speed and direction.

52. Kühn, S., *Wind Vane*, W.V. Windrichtungsgeber.jpg, Editor, 2004, Wikimedia Commons.

53. NOAA, *A portable SODAR wind profiling system*, U.S.D.o.C. National Oceanic & Atmospheric Administration (NOAA), Editor, 2005, Wikimedia Commons, NOSA.

54. U.S. Department of Energy, *A radar wind profiler installation*, 2006, US Department of Energy.

55. Kovalev, V.A. and W.E. Eichinger, *Elastic lidar : theory, practice, and analysis methods,*, 2004, Hoboken, N.J.: Wiley.

56. Weitkamp, C., *Lidar, Range-Resolved Optical Remote Sensing of the Atmosphere*. Springer Series in Optical Sciences. Vol. 102. 2005, New York: Springer Science.

57. Gan, C.-M., "Utilizing Operational and Improved Remote Sensing Measurements to Assess Air Quality Monitoring Model Forecasts," Doctor of Philosophy, Electrical Engineering, City University of New York, New York, 2011.

58. Gill Instruments. *WindMaster 3D Sonic Anemometer*. 2012 [cited 2012 March]; Available from: http://www.gill.co.uk/products/anemometer/windmaster.htm.

59. Huang, H.-s., et al. *The application of edge technique in wind lidar*. in *Radar (Radar), 2011 IEEE CIE International Conference on*. 2011.

60. Gentry, B.M., et al., *Wind Profiles Obtained with a Molecular Direct Detection Doppler LIDAR during IHOP_2002*, in *22nd Internation Laser Radar Conference (ILRC 2004)*, G. Pappalardo and A. Amodeo, Editors., 2004: Matera, Italy. p. 731.

61. Huffaker, R.M. and R.M. Hardesty, *Remote sensing of atmospheric wind velocities using solid-state and CO2 coherent laser systems.* Proceedings of the IEEE, 1996. **84**(2): p. 181-204.

62. American National Standards Laser Institute of America, *American National Standard for safe use of lasers - ANSI Z136.1-2000*, 2000, Orlando, FL: Laser Institute of America.

63. American National Standards Laser Institute of America, *American national standard for the safe use of lasers outdoors - ANSI Z136.6-2000*, 2000, Orlando, FL: Laser Institute of America.

64. Dutton, H.J.R., *Understanding optical communications*, 1998, Prentice Hall PTR: Upper Saddle River, N.J.

65. Agrawal, G.P., *Nonlinear Fiber Optics*, 2006, San Diego: Elsevier Science Publishing Co Inc.

66. Suhara, T. and M. Fujimura, *Waveguide nonlinear-optic devices*, 2003, Berlin; New York: Springer.

67. Agrawal, G.P. *Applications of Nonlinear Fiber Optics*. 2008; Available from: http://www.myilibrary.com?id=234510&ref=toc.

68. Ferreira, M.F.S., *Nonlinear effects in optical fibers*, 2011, Hoboken, N.J.; [Washington, D.C.]: Wiley ; Optical Society of America.

69. Furukawa, R.A., A. Tagaya, and Y. Koike, *Modal analysis of a multimode polarization-maintaining plastic optical fiber fabricated using poly(methyl methacrylate/benzyl methacrylate) copolymer.* Applied Physics Letters, 2008. **93**(10): p. 103303-3.

70. Oz Optics, *Standard Table: fibers, cables, connectors, lenses, and laser head adapters* O. Optics, Editor, 2012.

71. Mortensen, N.A., et al., *Numerical aperture of single-mode photonic crystal fibers.* Photonics Technology Letters, IEEE, 2002. **14**(8): p. 1094-1096.

72. Anderson, W. and D. Philen, *Spot size measurements for single-mode fibers - A comparison of four techniques.* Lightwave Technology, Journal of, 1983. **1**(1): p. 20-26.

73. JDSU Actema Saint-Etienne, *Guide to Fiber Optic Measurement*, 2001. p. 108.

74. Kirnehkrib, *Beamprofile SM+MM-Fiber*, 2011, Wikimedia. p. Beam profile of a single-mode and a multi-mode fiber.

75. Kirnehkrib, *Beamprofile Multi-Mode-Fiber*, 2011, Wikimedia. p. Different beam profiles of one and the same multi-mode fiber, depending on the light in-coupling characteristics and the bending of the fiber.

76. Hui, R. and M.S. O'Sullivan. *Fiber optic measurement techniques*. 2009; Available from: http://www.knovel.com/knovel2/Toc.jsp?BookID=2779.

77. Sezerman, O. and G. Best, *Accurate Alignment Preserves Polarization*, in *Laser Focus World*, 1997, PennWell Corporation/Technology Group: Nashua, NH. p. 3.

78. Oz Optics, *Polarization Measurements- Application Note APN0005*, 1999, Oz Optics: Ottawa, Ontario, Canada p. 10.

79. Noda, J., K. Okamoto, and Y. Sasaki, *Polarization Maintaining Fibers and Their Applications.* Lightwave Technology, Journal of, 1986. **4**(8): p. 1071-1089.

80. Emslie, C., *Polarization Maintaining Fiber Handbook - Excerpt Specialty Optical Fibers Handbook*, in *Specialty Optical Fibers Handbook*, 2007, Academic Press: Burlington. p. 243-277.

81. Kumar, A. and A. Ghatak, *Polarization of Light With Applications in Optical Fibers*, 2010, Bellingham, Washington: SPIE Press.

82. Derickson, D., *Fiber optic test and measurement*, 1998, Upper Saddle River, N.J.: Prentice Hall PTR.

83. Diamond Losone, *Polarization Extinction Ratio Measurement - Technical Note 003*, Diamond SA: Losone, Switzerland.

84. Diamond Losone, *Assembly and measuring technology for fibre optic polarization maintaining components*

85. Reeves, F., *Polarization Maintaining Alignment within Optical Fiber Systems*, 2009, PhotonCom, a division of Simbol Test Systems Inc.

86. Masuda, S. and T. Iwama, *Single-mode fiber-optic directional coupler.* Appl. Opt., 1982. **21**(19): p. 3484-3488.

87. Malitson, I.H., *Interspecimen Comparison of the Refractive Index of Fused Silica.* J. Opt. Soc. Am., 1965. **55**(10): p. 1205-1208.

88. International Telecommunication Union, *Optical Fibres, Cables and systems International Telecommunication Union ITU-T Manual 2009*, 2009, ITU Telecommunication Standardization Sector: Geneva, Switzerland. p. 324.

89. International Telecommunication Union. *About ITU - Overview*. 2012 [cited 2012; Available from: http://www.itu.int/en/about/Pages/default.aspx.

90. International Telecommunication Union, *ITU-T Series G Transmission Systems and Media Digital Systems and Networks Recomendation G.957- Optical interfaces for equipments and systems relating to the synchronous digital hierarchy*, 2006, International Telecommunication Union. p. 28.

91. Siegel, P.H., *Terahertz technology.* Microwave Theory and Techniques, IEEE Transactions on, 2002. **50**(3): p. 910-928.

92. Rohde, R.A., *Atmospheric Absorption Bands*, 2007, Global Warming Art. p. This figure shows the absorption and scattering of direct light in the the Earth's Atmosphere. This image was constructed by Robert A. Rohde using data from the Spectral Calculator. http://www.globalwarmingart.com/wiki/File:Atmospheric_Absorption_Bands_png.

93. Holl, J., *Man Sentenced to Probation For Shining Laser at a Plane*, in *New York Times*, 2006, New York Times Publishing Company: New York, N.Y., United States. p. B3.

94. Austen, I., *Laser Pointers Draw Attention (to Themselves)*, in *New York Times*, 2005, New York Times Publishing Company: New York, N.Y., United States. p. G8.

95. New York Times, *Laser Beam Pointed at a Police Helicopter*, in *New York Times*, 2005, New York Times Publishing Company: New York, N.Y., United States. p. B4.

96. Armental, M. *Providence man arrested on charges he pointed laser beam at drivers on Route 95*. 2012 [cited 2012 03/28/12]; Available from: http://news.providencejournal.com/breaking-news/2012/01/providence-man-73.html.

97. Negroni, C., *High-Powered Laser Pointers Pose Risk to Pilots*, in *New York Times*, 2011, The New York Times Company: New York.

98. The Associated Press State & Local Wire, *Laser pointed at San Diego Harbor Police officers*, 2012, The Associated Press State. p. STATE AND REGIONAL.

99. This is Gloucestershire. *Laser pen attack lands train driver in hospital*. 2012 [cited 2012 03/29/2012]; Available from: http://www.thisisgloucestershire.co.uk/Laser-pen-attack-lands-train-driver-hospital/story-15198344-detail/story.html.

100. Nakagawara, V., et al., *The Effects of Laser Illumination on Operational and Visual Performance of Pilots During Final Approach*, F.F.T.P.D. FAA Civil Aerospace Medical Institute, 3US Air Force Research Laboratory, SAE-G-10 Committee,, Editor, 2004, Office of Aerospace Medicine, Federal Aviation Administration: Washington, DC. p. 18.

101. Federal Aviation Administration, *FAA Flight Simulator Images from "The Effects of Laser Illumination on Operational and Visual Performance of Pilots During Final Approach."*, 2004, Federal Aviation Administration: Oklahoma City,OK p. No laser exposure: baseline for a “normal” approach. 0.5 μW/cm²: corresponds to a 5 mW (milliwatt) laser pointer at 3,700 feet, or a 50 mW pointer at 2.2 miles. 5.0 μW/cm²: corresponds to a 5 mW pointer at 1,200 feet, or a 50 mW pointer at 3,800 feet. 50 μW/cm²: corresponds to a 5 mW pointer at 350 feet, or a 50 mW pointer at 1,100 feet. .

102. Barat, K., *Laser safety: tools and training*, 2009, Boca Raton: CRC Press.

103. Silfvast, W.T., *Laser Fundamentals*, 2004, Cambridge, England: Cambridge University Press.

104. Azadeh, M., *Fiber Optic Engineering*. Optical Networks, ed. B. Mukherjee, 2009, New York: Springer. 376.

105. Canadian Instrumentation and Research. *The Wave Guide, Modes and Evanescent Tails, b matching and over coupling*. [Web page] 2012 2009]; Available from: http://www.cirl.com/evanescent.php.

106. Korpel, A., *Acousto-optics*, 1988, New York: M. Dekker.

107. Neos Technologies, *INTRODUCTION TO ACOUSTO-OPTIC MODULATORS AND DEFLECTORS*, Neos Technologies. p. 22.

108. Neos Technologies, *Acusto-Optic Products Catalog- Introduction to AO Modulation*, N. Technologies, Editor, 2000. p. 5-8.

109. Guo, Y., *Nonlinear photonics: Nonlinearities in optics, optoelectronics, and fiber communications*, 2002, Hong Kong; Berlin; New York: Chinese University Press ; Springer.

110. Becker, P.C., N.A. Olsson, and J.R. Simpson, *Erbium-doped fiber amplifiers : fundamentals and technology*, 1999, San Diego: Academic Press.

111. Texas Instruments, *ADS5474 14 bit, 400 MSPS Analog-to-Digital Converter*, 2012, Texas Instruments,.

112. Ding, C., L. Pan, and B. Lü, *Changes in the state of polarization of apertured stochastic electromagnetic modified Bessel–Gauss beams in free-space propagation.* Applied Physics B: Lasers and Optics.

113. Horowitz, P. and W. Hill, *The art of electronics*, 1989, Cambridge [England]; New York: Cambridge University Press.

114. Rubiola, E., *Phase noise and frequency stability in oscillators*, 2009, Cambridge, UK; New York: Cambridge University Press.

115. Maas, S.A., *Noise in linear and nonlinear circuits*, 2005, Boston, MA: Artech House.

116. Oppenheim, A.V., A.S. Willsky, and I.T. Young, *Signals and systems*, 1983, Englewood Cliffs, N.J.: Prentice-Hall.

117. Frederick, D.K. and A.B. Carlson, *Linear systems in communication and control*, 1971, New York: Wiley.

118. Gaskill, J.D., *Linear systems, Fourier transforms, and optics*, 1978, New York: Wiley.

119. Johnson, J.B., *Thermal Agitation of Electricity in Conductors.* Physical Review, 1928. **32**(1): p. 97-109.

120. Kester, W. and Analog Devices Inc, *Data conversion handbook*, 2005, Amsterdam; Boston: Elsevier : Newnes. 976.

121. Innovative Intergration, *X5-400M datasheet*, 2011, Innovative Intergration: Simi Valley, CA.

122. IEEE, *IEEE Standard for Digitizing Waveform Recorders - Redline.* IEEE Std 1057-2007 (Revision of IEEE Std 1057-1994) - Redline, 2008: p. 1-210.

123. *IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters.* IEEE Std 1241-2010 (Revision of IEEE Std 1241-2000), 2011: p. 1-139.

124. Moxfyre, *Numerical_aperture_for_a_lens.svg*, 2009, Wiki Media, Public Domain. p. Numerical_aperture_for_a_lens.svg, Better illustration of numerical aperture in the case of a thin lens, in order to compare with the concept of angular aperture. Based on Angular aperture.svg.

125. Sakurambo, *Airy-pattern.svg* 2007, Wikipedia. p. Vectorized version of Image: Airy-pattern.png.

126. Sakurambo, *Airy-3d.svg*, 2008, Wikipedia. p. 3D representation of an Airy disk pattern.

127. Inductiveload, *Circular aperture variables.svg*, 2009, Wikimedia.org. p. A diagram showing the basic variables of an optical system containing a circular aperture and an observation screen.

128. Inductiveload, *Airy Pattern Intensity and Encircled Power.svg*, 2009, Wikimedia.org. p. A graph showing the intensity and encircled power of an Airy pattern.

129. Cao, G. and X. Yu, *Accuracy analysis of a Hartmann-Shack wavefront sensor operated with a faint object.* Optical Engineering, 1994. **33**(7): p. 2331-2335.

130. CVI Melles Griot, *Gaussian Beam Optics - All Things Photonics*, 2009.

131. Friedman, E. and J.L. Miller, *Photonics rules of thumb: Optics, electro-optics, fiber optics, and lasers*, 2004, New York; Bellingham, Wash., USA: McGraw-Hill; SPIE Press.

132. Wallner, O., P.J. Winzer, and W.R. Leeb, *Alignment Tolerances for Plane-Wave to Single-Mode Fiber Coupling and Their Mitigation by Use of Pigtailed Collimators.* Appl. Opt., 2002. **41**(4): p. 637-643.

133. Majumdar, A.K. and J.C. Ricklin, *Free-Space Laser Communications: Principles and Advances*. Optical and Fiber Communications Reports. Vol. 2. 2008, New York: Springer. 391.

134. Andrews, L.C., *Field guide to atmospheric optics*, 2004, Bellingham, WA: SPIE Press.

135. Andrews, L.C. and R.L. Phillips, *Laser beam propagation through random media, 2nd Ed.*, 2005, Bellingham, WA: SPIE Optical Engineering Press.

136. Tunick, A., *CN2 model to calculate the micrometeorological influences on the refractive index structure parameter.* Environmental Modelling & Software, 2003. **18**(2): p. 165-171.

137. Stotts, L.B., et al., *Free-space optical communications link budget estimation.* Appl. Opt., 2010. **49**(28): p. 5333-5343.

138. Stotts, L.B., et al., *Hybrid Optical RF Airborne Communications.* Proceedings of the IEEE, 2009. **97**(6): p. 1109-1127.

139. Miller, M.G. and P.L. Zieske, *Turbulence environmental characterization*, 1979, Rome Air Development Center, AVCO Everette Research Lab Inc.: Griffiss Air Force Base, NY. p. 135.

140. Parenti, R.R. and R.J. Sasiela, *Laser-guide-star systems for astronomical applications.* J. Opt. Soc. Am. A, 1994. **11**(1): p. 288-309.

141. Accetta, J.S. and D.L. Shumaker, *Propagation through atmospheric optical turbulence*, in *The Infrared and electro-optical systems handbook*, F.G. Smith, Editor, 1993, Infrared Information Analysis Center ; SPIE Optical Engineering Press: Ann Arbor, Mich.; Bellingham, Wash.

142. Kameyama, S., et al., *Compact all-fiber pulsed coherent Doppler lidar system for wind sensing.* Appl. Opt., 2007. **46**(11): p. 1953-1962.

143. Baars, H., et al., *Continuous monitoring of the boundary-layer top with lidar.* Atmospheric Chemistry and Physics, 2008. **8**(12): p. 16.

144. Davis, K.J., et al., *An Objective Method for Deriving Atmospheric Structure from Airborne Lidar Observations.* Journal of Atmospheric and Oceanic Technology, 2000. **17**(11): p. 1455-1468.

145. Brooks, I.M., *Finding Boundary Layer Top: Application of a Wavelet Covariance Transform to Lidar Backscatter Profiles.* Journal of Atmospheric and Oceanic Technology, 2003. **20**(8): p. 1092-1105.

146. Cohn, S.A. and W.M. Angevine, *Boundary Layer Height and Entrainment Zone Thickness Measured by Lidars and Wind-Profiling Radars.* Journal of Applied Meteorology, 2000. **39**(8): p. 1233-1247.

147. Environmental Software and Services. *Pasquill Stability Classes, AirWare On-line Reference Manual* 2012 [cited 2012 08/10/12]; Available from: http://www.ess.co.at/MANUALS/AIRWARE/stability_class.html.

148. Air Resoures Laboratory. *Pasquill Stability Classes*. 2012 [cited 2012 08/10/12]; Pasquill Stability Classes Table]. Available from: http://ready.arl.noaa.gov/READYpgclass.php.

149. Redfern Integrated Optics Inc., *NASA Awards Redfern Integrated Optics (RIO) with Contract for Development of Laser for Space LIDAR*, 2011, Redfern Integrated Optics Inc. .

150. Yang, F., et al., *100-mW linear polarization single-frequency all-fiber seed laser for coherent Doppler lidar application.* Optics Communications, 2012. **285**(2): p. 149-152.