DEVELOPMENT OF A MICRO PULSED LIDAR AND A SINGLY-RESONANT OPTICAL PARAMETRIC OSCILLATOR FOR CO_2 DIAL FOR USE IN ATMOSPHERIC STUDIES

by

Chat Chantjaroen

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

Doctor of Philosophy

in

Physics

MONTANA STATE UNIVERSITY
Bozeman, Montana

April 2017

©COPYRIGHT

by

Chat Chantjaroen

2017

All Rights Reserved

DEDICATION

I dedicate this thesis to my father: Major General Chin Chantjaroen, and to my mother: Puangthong Chantjaroen, for their care, love, and raising me to be who I am today. To my elder brother who I look up to: Major Chart Chantjaroen. To my wife: Lawan Chantjaroen, for her love and support throughout our marriage and many years of graduate school. To my sons: Pukit (Luke) and Pat (Rock) Chantjaroen.

ACKNOWLEDGEMENTS

I would like to thank and to express my special appreciation for both of my advisors, Dr. Kevin Repasky and Dr. John Carlsten for being the best mentors, for all the great knowledge and skills they have taught, and for their assistance and guidance throughout my Ph.D. program. I am very much grateful for your invaluable support. I would also like to thank my sponsors—Chulachomklao Royal Military Academy of Thailand and Montana Research and Economic Development Initiative for their financial support. For without them, my graduate work would neither have existed nor been completed.

In addition, I would like to thank the rest of my graduate committee: Dr. Yves Idzerda, Dr. Joseph Shaw, and Dr. Shannon Willoughby for their support. Thanks to Dr. Gregory Francis, Dr. Bennett Link, and other professors for passing on their knowledge. Thanks to the physics administrative staff: Margaret, Sarah, and Norm for all their help.

I would like to give special thanks to Briana Jones for her help with difficult problems in the OPO project. Special thanks to Drew Moen for his help in the lab especially in the beginning of the MPL project. Special thanks to Tianbo Liu and Cooper McCann for their useful tips for presentating technical research and for all other help. Special thanks to Ryan Snow for intellectual conversations regarding general science and life philosophy. Special thanks to my physics classmates: Stephen Crouch, Orion Bellorado, Ben Rosemeyer, Shane Atwood, Frank Borawski, and Mark Stalnaker for creating a great study group that enhanced my physics understanding which allowed me to pass the brutal comprehensive exam. Thanks to all other friends for their help in all aspects.

TABLE OF CONTENTS

1. INTRODUCTION 1

Aerosol Effect on Climate Change 1
Technologies on Aerosol Measurement...4
Carbon Dioxide Effect on Climate Change 7
Technologies on Carbon Dioxide Measurement...8
Dissertation Organization 10

2. THEORY I 12

Principle of Lidar 12
Lidar Equation 14
Aerosol Inversion..18
Overlap Function 19
Matching Technique between Horizontal Overlap and Vertical Returns...............27
AERONET AOD and Lidar Ratio 30
Surface Temperature and Pressure...33
Synchronization of Lidar, AERONET, and Weather Station Data 33
Lapse Rate and Temperature Model 38
Pressure Model...41
Molecular Back Scatter and Extinction Coefficient 42
Calibration Constant 44
Inversion 46
Principle of DIAL 55
DIAL Equation...55
CO_2 Line Selection for DIAL 58
Doppler-Broadened Line Shape...60
Pressure-Broadened Line Shape 61
Voigt Profile..62
Temperature Sensitivity Line-Strength...63
Absorption Cross-Section 63
Laser Transmitter Needed for CO_2 DIAL...65

3. THEORY II..66

Introduction to Nonlinear Optics 66
Parametric and Nonparametric Process 67
Mathematical Expression of Nonlinear Polarization 67
Second-Harmonic Generation...68
Second-Order Frequency Mixing Processes...70
Sum-Frequency Generation 71

TABLE OF CONTENTS – CONTINUED

Difference-Frequency Generation 72
Optical Parametric Generation, Amplification, and Oscillation 73
Mathematical Description of Nonlinear Optics 75
Derivation of the Wave Equation for Nonlinear Material 75
Derivation of Coupled Wave Equation for Plane Waves in Second-Order Frequency Mixing Processes 77
Coupled Wave Equations for Lossless Material 81
Non-Depleted Pump Solution 81
Phase-Matching Condition 82
Angle Tuning 85
Temperature Tuning 85
Quasi-Phase-Matching 86
Depleted Pump Solution 88
Gaussian Beam 91
SNLO Model 92

4. MICRO-PULSED LIDAR 94

Instrument Design 94
Transmitter 95
Receiver 96
Data Acquisition and Control Unit 96
Tripod Mount Control 98
Methods and Setup 101
Transmitted Beam Collimation 101
Transmitter-Receiver Alignment 103
Horizontal Data Acquisition 105
Vertical Data Acquisition 108
Results 110
20 July 2015 113
31 July 2015 114
6 September 2016 115
30 September 2016 116
4 November 2016 117
9 November 2016 118
10 November 2016 119

5. OPTICAL PARAMETRIC AMPLIFICATION 120

Instrument Design 120
Methods and Setup 122

TABLE OF CONTENTS – CONTINUED

Pump-Seed Laser Alignment 122
Crystal Alignment 127
Crystal Selection 129
Results 131
Output Spectrum 131
Output Energy 136
SNLO Comparison 137

6. OPTICAL PARAMETRIC OSCILLATION 141

Instrument Design 141
Dither Locking Theory 143
Methods and Setup 150
Optical Cavity 150
Dither Locking 154
Results 157
Output Spectrum 157
Output Energy 159
SNLO Comparison 162

7. DISCUSSION OF INITIAL DIAL DESIGN 165

DIAL Performance Model 165
Initial DIAL Transmitter Design 169
Initial DIAL Design 172

8. CONCLUSIONS 175

MPL 175
OPO 177

REFERENCES CITED 179

APPENDICES 184

APPENDIX A: Labview Program for MPL Data Acquisition 185
APPENDIX B: Labview Program for MPL Scan 192
APPENDIX C: Matlab Code for Temperature and Pressure Models 196
APPENDIX D: Matlab Code for MPL Overlap Retreival 200
APPENDIX E: Matlab Code for Aerosol Inversion 211
APPENDIX F: Matlab Code for OPA Non-Depleted and Depleted Pump 251
APPENDIX G: Solidworks Drawings for PPLN Crystal Housing 255

LIST OF TABLES

Table Page

3.1 Phase-matching methods for uniaxial crystals. (Boyd 2003 Table 2.7.2) [36].........85

4.1 MPL transmitter parameters 95

4.2 MPL receiver parameters.........96

7.1 Transmitter and receiver parameters used for the DIAL performance modeling 167

7.2 Atmospheric parameters used for the DIAL performance modeling.........167

LIST OF FIGURES

Figure Page

1.1 Scanning electron microscope images (not at the same scale) show the wide variety of aerosol types. From left to right: volcanic ash, pollen, sea salt, and soot. (Micrographs courtesy by Chere Petty, University of Maryland, and Peter Buseck, Arizona State University) 1

1.2 Map of global distribution of aerosols. Intense colors indicate a thick layer of aerosols. Yellow areas are predominantly coarse particles, like dust, and red areas are mainly fine aerosols, like smoke or pollution. Gray indicates areas with no data. (NASA map by Robert Simmon, based on MODIS data from NASA Earth Observations) 2

1.3 Current AERONET sites around the world (as of November 17, 2016). [11] 6

1.4 CIMEL sun sky multiband photometer model CE318-N at Montana State University, Bozeman site 45.67°N 111.05°W. (Picture courtesy by Joseph Shaw, Montana State University) 6

1.5 Total annual anthropogenic GHG emissions ($GtCO_2$ eq/yr) by groups of gases 1970–2010: CO_2 from fossil fuel combustion and industrial processes; CO_2 from Forestry and Other Land Use (FOLU); methane (CH_4); nitrous oxide (N_2O); fluorinated gases covered under the Kyoto Protocol (F-gases). At the right side of the figure GHG emissions in 2010 are shown again broken down into these components with the associated uncertainties (90% confidence interval) indicated by the error bars. Global CO_2 emissions from fossil fuel combustion are known within 8% uncertainty (90% confidence interval). CO_2 emissions from FOLU have very large uncertainties attached in the order of ±50%. Uncertainty for global emissions of CH_4, N_2O and the F-gases has been estimated as 20%, 60% and 20%, respectively. 2010 was the most recent year for which emission statistics on all gases as well as assessment of uncertainties were essentially complete at the time of data cut-off for this report. [5] 7

LIST OF FIGURES – CONTINUED

Figure Page

1.6 The record of global average temperatures compiled by NASA's Goddard Institute for Space Studies [13]. The "zero" on this graph corresponds to the mean temperature from 1961-1990, as directed by IPCC. 8

1.7 Current NOAA's Global Greenhouse Gas Reference Network active sites around the world (As of November 18, 2016). Air samples are collected by surface flask (red circle), airborne flask (purple airplane), in situ tower (green triangle), in situ observatory (blue square), and surface in situ (pink star). [16] 9

2.1 Schematic of basic architecture of a lidar 12

2.2 Basic configurations of lidar 13

2.3 Schematic of a lidar system with the illustrating components in the lidar equation 17

2.4 Flow chart showing key steps in the aerosol inversion....18

2.5 Beams from different locations focused by the fiber coupler lens onto the fiber result in different overlaps 20

2.6 MPL time averaged horizontal profiles of actual and raw counts with background subtraction. Data was taken on August 9, 2016, 22:50:19-22:52:04. Horizontal data for the overlap calculation is usually taken at nighttime to minimize the returned signal noise. The peak signal at 8.6 km is the return from the mountain south of Bozeman 22

2.7 Plot of the range corrected return as a function of range (blue) on a log scale. The full overlap begins where the plot becomes linear starting at 1.7 km indicating by the left green line. The straight fit line (red) within the two green lines portion provides the slope and intercept which are necessary for retrieving the overlap function 24

2.8 Overlap function (blue) and modified overlap function (red) 25

LIST OF FIGURES – CONTINUED

Figure Page

2.9 Comparison of returns with and without overlap correction 26

2.10 Plots of horizontal overlaps for telescope focus adjusted from near to far 28

2.11 Matching between the horizontal (green and red) and vertical returns (blue). On the plot, H51 data is plotted in the far left, then H52 and so on. Based on the rising slope, the best match with the vertical returns is H54ar..29

2.12 Time averaged vertical range corrected returns with (blue) and without (green) horizontal overlap applied. The near ground returns where the overlap gives the most uncertainty is modified and plotted in red. Peak at 6 km indicates clouds that floated by during that time window 29

2.13 AERONET AOD download page. [20] 31

2.14 AERONET phase function and single scattering albedo download page. [21]...32

2.15 MSU ECE weather station website. [22] 33

2.16 Available data points on June 30, 2015 34

2.17 Weather station surface temperature data (blue) and synchronized data (pink)..35

2.18 Weather station surface pressure data (blue) and synchronized data (pink)..36

2.19 AERONET AOD measured at 500 nm (blue) and 675 nm (red), linearly interpolated AOD for 532 nm (green), and synchronized AOD (pink)..37

2.20 AERONET single scattering albedo measured at 440 nm (blue) and 675 nm (red), linearly interpolated SSA for 532 nm (green), and synchronized SSA (pink)..37

LIST OF FIGURES – CONTINUED

Figure Page

2.21 AERONET 180° phase function measured at 440 nm (blue) and 675 nm (red), linearly interpolated PFN for 532 nm (green), and synchronized PFN (pink) 38

2.22 Temperature profiles from radiosonde (green) launched at Bozeman (N 45°39'43", W 111°02'42"), from November 23-26, 2013, twice daily and temperature model for Bozeman (blue) 40

2.23 Pressure profiles from radiosonde launched at Bozeman (N 45°39'43", W 111°02'42"), from November 23-26, 2013, twice daily (green) and pressure model for average condition (red) for Bozeman (blue) 42

2.24 Molecular back scatter and extinction coefficient 43

2.25 Calibration constant as a function of altitude 46

2.26 Total and molecular back scatter profile. With the right lidar ratio, their slopes above the boundary layer (~3.2 km) match 48

2.27 Aerosol back scatter profile results from the subtraction between the total and molecular back scatter profile in Figure 2.26 49

2.28 Percent error in lidar AOD as compare to AERONET AOD when scanning the value of S_a from 0 to 100 Sr) 50

2.29 Plots show good agreement between the lidar AOD as a function of range (blue) and the total AERONET AOD (red) when $S_a = 50.9\,\mathrm{Sr}$ 51

2.30 Range corrected returns with overlap applied over Bozeman on June 30, 2015 09:23:38-11:23:24 52

2.31 Total back scatter coefficient over Bozeman on June 30, 2015 09:23:38-11:23:24 52

2.32 Molecular back scatter coefficient over Bozeman on June 30, 2015 09:23:38-11:23:24 53

LIST OF FIGURES – CONTINUED

Figure Page

2.33 Aerosol back scatter coefficient over Bozeman on June 30, 2015 09:23:38-11:23:24 53

2.34 AERONET and lidar total AOD over Bozeman on June 30, 2015 09:23:38-11:23:24 54

2.35 Aerosol lidar ratio over Bozeman on June 30, 2015 09:23:38-11:23:24. The S_a AERONET Raw Data is calculated from the SSA and PFN as in equation (2.27) 54

2.36 Locations of online and offline wavelength in absorption feature of a molecule (left). Online and offline returns from a DIAL (right) 55

2.37 Plot of the atmospheric CO_2 concentration of 390 ppm as a function of wavelength for a 10 km path length, 296 K temperature, and 0.85 atm pressure 59

2.38 Line profiles: A is Lorentz, B is Doppler, and C is Voigt profile resulting from convolution of A and B. [32] 62

2.39 Absorption cross section of CO_2 as a function of wavelength at 296 K temperature and 0.85 atm pressure 64

3.1 In linear optics, a light wave acting on a molecule causes it to vibrate and then emits a wave with the same frequency 66

3.2 In nonlinear optics, a more intense light wave sends electrons to a higher energy state and thus emits a wave with different frequency 66

3.3 Geometry of SHG (left). Energy level diagram describing SHG (right) 70

3.4 Geometry of SFG (left). Energy level diagram describing SFG (right) 72

3.5 Geometry of DFG (left). Energy level diagram describing DFG (right) 72

LIST OF FIGURES – CONTINUED

Figure Page

3.6 OPO geometry (left). Optical cavity is absent for OPA. Both optical cavity and seed laser are absent for OPG. Energy level diagram (right) 74

3.7 Effect of phase mismatch factor Δk on the output signal intensity in OPA....82

3.8 Geometry of the ordinary (perpendicular), extraordinary (parallel), and optical axis of a uniaxial crystal. [40] 84

3.9 Refractive indices of a negative uniaxial crystal. (Boyd 2003 Figure 2.7.2) [36] 84

3.10 Schematic of a periodically poled crystal in QPM technique. [42] 86

3.11 Comparison of the spatial variation field intensity between three different phase-matching conditions. 87

3.12 Input parameters for coupled wave equations' numerical integration in Matlab program. 89

3.13 Comparison of the OPA output signal energy between depleted and non-depleted pump approximation for the case of plane waves in lossless system 90

3.14 Gaussian beam profile....91

3.15 Gaussian intensity distribution. [43] 92

4.1 Schematic of the Micro-Pulsed Lidar (MPL) 94

4.2 Front panel of the Labview program for MPL DAQ and control....97

4.3 MPL system (left), transmitter (top right), and receiver (bottom right) 99

4.4 Demonstration of weight counter balance in the MPL design....100

4.5 Front panel of the slant path program for MPL mount control....101

LIST OF FIGURES – CONTINUED

Figure Page

4.6 MPL transmitted beam collimating procedure performed in the lab 102

4.7 Interference pattern of the transmitted beam from the shear-plate in vertical (left) and horizontal (right) axis. The degree of collimation is indicated by how straight and parallel the fringe pattern is to the reference line on the plate 102

4.8 Transmitter-receiver aligning procedure..105

4.9 MPL horizontal configuration...106

4.10 Real-time ln(N) plot in the Labview DAQ program used for the peak-up process in the final transmitter-receiver alignment........................107

4.11 MCS settings (left column), data-point reading (middle column), path and filename settings (right column top), and DAQ settings (right column bottom) 107

4.12 Comparison plots between real-time normalized vertical data (white) and normalized horizontal data (red) used in the vertical-horizontal alignment matching 109

4.13 Lidar inversion results 20 July 2015 113

4.14 Lidar inversion results 31 July 2015 114

4.15 Lidar inversion results 6 September 2016 115

4.16 Lidar inversion results 30 September 2016 116

4.17 Lidar inversion results 4 November 2016..117

4.18 Lidar inversion results 9 November 2016..118

4.19 Lidar inversion results 10 November 2016..119

5.1 OPG/OPA setup (The seed laser is turned off for OPG) 121

LIST OF FIGURES – CONTINUED

Figure Page

5.2 Image of the pump beam at 305 mm from the laser source (top) and image of the seed beam at 600 mm from the laser head of the optical amplifier (bottom) 122

5.3 Gaussian profile of the pump beam (top) and seed beam (bottom). The laser source for both beams are located at $z = 0$ in each plot. The beam waist is indicated by the vertical blue line 123

5.4 Gaussian beam through thin lens diagram 124

5.5 Pump beam through lens design with various lens focal length. The lens is represented by the black vertical line 125

5.6 Optimization of 250-mm lens locations for pump beam 126

5.7 PPLN crystal (2×2×32 mm) in the aluminum heater (left). The bottom piece is the heat sink, the middle and top piece are the heater. Between the heat sink and the heater is the Thoslabs thermoelectric cooler (TEC). To ensure minimal temperature gradients within the crystal, the top piece is made to perfectly enclose the crystal with along with a thermistor for temperature response. The rotation and translation stages are used for crystal alignment (right) 128

5.8 Pump-seed-crystal alignment (left) and information needed for SNLO input (right)...129

5.9 Crystal temperature vs OPG output signal wavelength. In the legend is crystal poling period in μm unit 130

5.10 Crystal poling period vs their average peak output wavelength 131

5.11 OPG/OPA output spectra with poor spectral overlap (top), poor spatial overlap (middle), good spectral and spatial overlap (bottom)................132

5.12 OPG/OPA output spectra with different input pump energies 135

LIST OF FIGURES – CONTINUED

Figure Page

5.13 OPG/OPA output energy (top) and conversion efficiency (bottom). In the measurements, crystal temperature was 28.8 °C and the seed power was 160 mW 136

5.14 Input in SNLO model for OPA output energy 138

5.15 Conversion efficiency comparison between OPA and SNLO model with different values of d_{eff} and Δk 139

6.1 Schematic of the OPO with the locking system 142

6.2 Image of the OPO with the locking system 142

6.3 Optical cavity transmission 144

6.4 Dither signal as a function of time. This is the output of the AOM and the input of the optical cavity 146

6.5 Cavity transmission output when the cavity is on-resonant at λ_0 147

6.6 Cavity transmission output when the cavity is off-resonant. In the case where $\lambda > \lambda_0$, the cavity transmission is in phase with the dither signal 148

6.7 Cavity transmission output when the cavity is off-resonant. In the case where $\lambda < \lambda_0$, the cavity transmission is out of phase with the dither signal 148

6.8 Error signal as a function of cavity transmission wavelength. The cavity is on-resonant when $\lambda = \lambda_0$ where the error signal is zero. To maintain resonance, the cavity transmission wavelength must stay within shaded region which indicates the range of the dithering wavelength. Plot courtesy by Briana Jones, Montana State University [49] 149

LIST OF FIGURES – CONTINUED

Figure Page

6.9 Optical cavity design for the location of the curved mirror. To resonate at the seed wavelength, each mirror must be placed at where its radius of curvature (ROC) matches that of the seed laser. From the measurements, the waist of the seed beam is located at 186 mm (dashed black line). This is where the flat mirror should be located since their ROC of infinity matches. The seed and pump beam ROCs calculated based on the Gaussian fits in Figure 5.8 are plotted in blue and red, respectively. The intercept between the curved mirror ROC (green line) and the seed ROC (solid blue line) is where the curved mirror should be located as indicated by the dashed blue line at 678 mm 151

6.10 Schematic of the OPO cavity. The rectangle represents the crystal with its actual dimension. The left vertical black line represents the flat mirror. The right vertical black line represents the curved mirror. Note that the mirror locations are exact but the mirror diameters and thicknesses are not to scale....................152

6.11 Image of the OPO cavity....................152

6.12 OPO cavity transmission from the PZT scan (left) with the zoom-in (right). In the right plot, the solid pink line represents the FSR. The green lines represent the FWHM of each peak. The amplitude of the green lines is determined as the half way between the dashed and solid pink lines. The measured values of the FSR and FWHM are used to calculate the finesse as shown in the box of the right plot....................153

6.13 Hold circuit diagram (left) and image (right) designed to ignore the analog signals of high-intensity pulses. A Schmitt trigger (SN74121) and a sample and hold (LF398N) are used in conjunction to time the arrival of the large signal pulses and to set how long to ignore the pulses. A variable resistor is used to fine-tune the hold time. [49] 155

6.14 OPG/OPA/OPO spectra (left) and the corresponding FWHM measurements (right). FWHMs are carefully measured at -3dBm down from the peaks....................159

LIST OF FIGURES – CONTINUED

Figure Page

6.15 OPG/OPA/OPO output energy (top) and conversion efficiency (bottom). In the measurements, crystal temperature was 27.9 °C and the seed power was 160 mW 160

6.16 Predictions of conversion efficiency and output energy for seeded-locked OPO 162

6.17 Input in SNLO model for OPO output energy 163

6.18 Output energy and conversion efficiency comparisons between OPO and SNLO model with $d_{eff} = 13$ pm/V and $\Delta k = 0$ 164

7.1 The percent error in the DIAL retrieval of CO_2 as a function of range for temporal averaging ranging from 1 minute to 15 minutes and spatial averaging ranging from 75 m to 300 m.............................168

7.2 The percent error in the DIAL retrieval of CH_4 as a function of range for temporal averaging ranging from 1 minute to 15 minutes and spatial averaging ranging from 75 m to 300 m.............................169

7.3 Schematic of the laser transmitter design based on a singly-resonant OPO 170

7.4 Schematic of the DIAL design...173

ABSTRACT

According to the Fifth Assessment Report (AR5) from the Intergovernmental Panel on Climate Change (IPCC), aerosols and CO_2 are the largest contributors to anthropogenic radiative forcing—net negative for aerosols and positive for CO_2. This relates to the amount of impact that aerosols and CO_2 can have on our atmosphere and climate system. CO_2 is the predominant greenhouse gas in the atmosphere and causes great impacts on our climate system. Recent studies show that a less well known atmospheric component—aerosols, which are solid particles or liquid droplets suspended in air, can cause great impact on our climate system too. They can affect our climate directly by absorbing and scattering sunlight to warm or cool our climate. They can also affect our climate indirectly by affecting cloud microphysical properties. Typically sulfate aerosols or sea salts act as condensation nuclei for clouds to form. Clouds are estimated to shade about 60% of the earth at any given time. They are preventing much of the sunlight from reaching the earth's surface and are helping with the flow of the global water cycle. These are what permit lifeforms on earth. In the IPCC report, both aerosols and CO_2 also have the largest uncertainties and aerosols remains at a low level of scientific understanding. These indicate the need of more accurate measurements and that new technologies and instruments needs to be developed.

This dissertation focuses on the development of two instruments—a scannable Micro Pulsed Lidar (MPL) for atmospheric aerosol measurements and an Optical Parametric Oscillator (OPO) for use as a transmitter in a Differential Absorption Lidar (DIAL) for atmospheric CO_2 measurements. The MPL demonstrates successful measurements of aerosols. It provides the total aerosol optical depth (AOD) and aerosol lidar ratio (S_a) that agree well with an instrument used by the Aerosol Robotic Network (AERONET). It also successfully provides range-resolved information about aerosols that AERONET instrument is incapable of. The range-resolved information is important in the study of the sources and sinks of aerosols. The OPO results show good promise for its use as a DIAL transmitter.

INTRODUCTION

Aerosol Effect on Climate Change

An aerosol is a colloidal suspension of particles dispersed in air or “atmospheric particulate” such as fog, steam, haze, dust, particulate air pollutants, and smoke [1]. Aerosols can be found in the atmosphere from the earth’s surface up to the stratosphere. They can scale from a few nanometers to a few microns. Despite their small size, they have major impacts on our climate system. The largest source of natural aerosols is an ocean which exchanges the air-sea particulate matters that contribute to the global cycles of carbon, nitrogen, and sulfur aerosols. In addition, the sea salt is left suspended in the atmosphere as the ocean water evaporates in the process. Other major sources of natural aerosols are volcanic eruptions, forest exudates, geyser steams, and wildfires. Major sources of anthropogenic aerosols are coal, biomass burning, industrial pollution, and emissions from road vehicles which, for the next 40 years, is estimated to have the largest global warming impacts of all human activities [2].

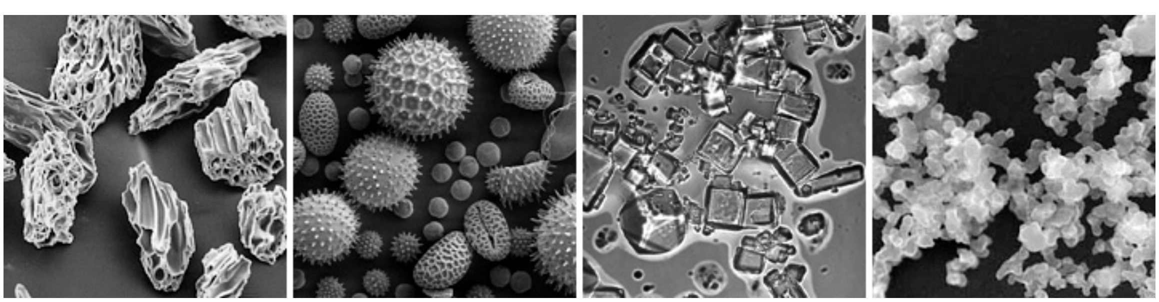


Figure 1.1 Scanning electron microscope images (not at the same scale) show the wide variety of aerosol types. From left to right: volcanic ash, pollen, sea salt, and soot. (Micrographs courtesy by Chere Petty, University of Maryland, and Peter Buseck, Arizona State University)

The map in Figure 1.2 shows the global distribution of aerosols detected by satellites. It indicates a thin layer of evenly distributed aerosols which is primarily salt, covering most of the world's oceans. Thick aerosol regions in Eastern United States, urban areas in Europe, and northeastern China are mostly man-made which is typically from coal power plants, industrial emissions, and vehicle traffic in large cities. Whereas the thick layers of aerosols in areas such as the Amazon, central Africa, Canada, and western United States are mostly the result of slash-and-burn agriculture and wildfires [3].

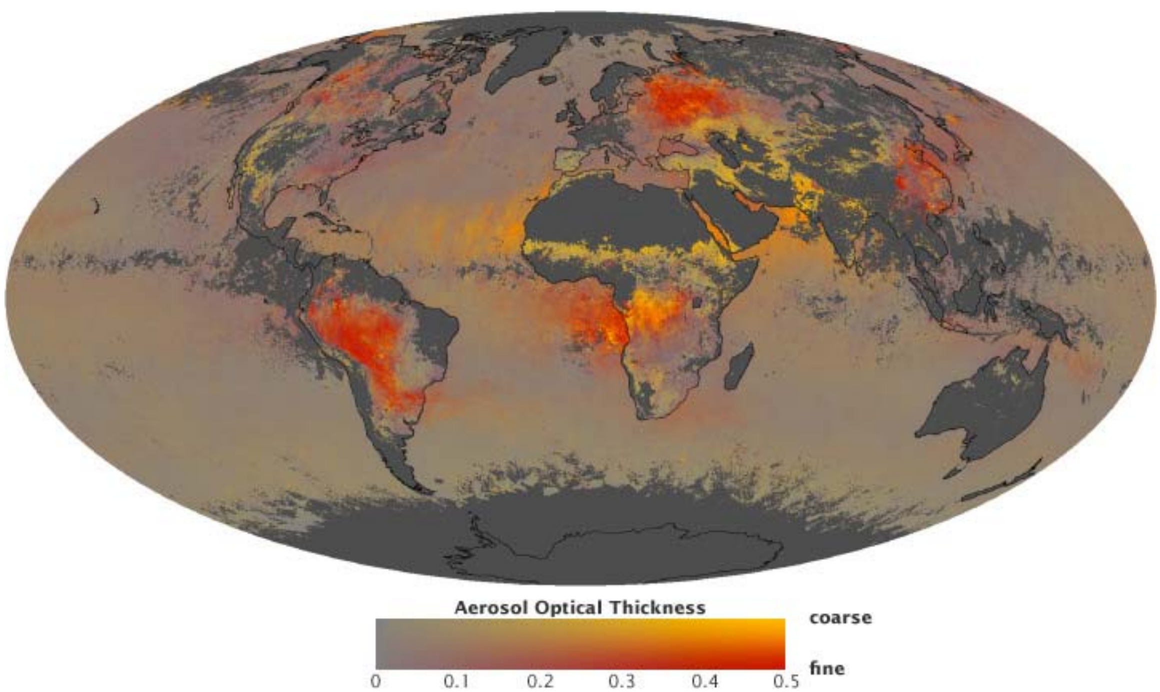


Figure 1.2 Map of global distribution of aerosols. Intense colors indicate a thick layer of aerosols. Yellow areas are predominantly coarse particles, like dust, and red areas are mainly fine aerosols, like smoke or pollution. Gray indicates areas with no data. (NASA map by Robert Simmon, based on MODIS data from NASA Earth Observations)

Although most aerosols only remain suspended in the atmosphere for less than a week, they can still travel long distances. It is not uncommon that dust particles from the

Sahara frequently cross the Atlantic and reach the Caribbean. Wind can blow pollution from eastern China and Japan to the mid Pacific Ocean. Smoke from Canadian and Siberian wildfires usually reaches the Arctic [4].

According to the Fifth Assessment Report (AR5) of the Intergovernmental Panel on Climate Change (IPCC), aerosols dominate the uncertainty in the total anthropogenic radiative forcing (RF) which is the key aspect in understanding of our climate change model [5]. This is likely because of the vast variety of aerosols and their different properties, their short lifetime and long distance travel, as well as their complex interaction with clouds.

The "direct effect" of aerosols on the earth's climate system is the scattering and absorbing of sunlight. This effect can cause major impact on climate especially when a large amount of aerosols scatter light. A recent example is the eruption of Mount Pinatubo in the Philippines in 1991 that released more than 20 million tons of sulfur dioxide into the atmosphere. As a result, the sulfate aerosols settled above the clouds where they cannot be reduced by the rain. They remained there for several years, reflecting much of sunlight and causing an abrupt drop in global temperature by 0.6 °C [6]. Although most aerosols reflect sunlight which causes cooling or negative radiative forcing (RF), some aerosols also absorb sunlight and causes warming or positive RF.

Aerosols also have an "indirect effect" on climate change by altering global reflectivity or albedo via cloud interaction. Typically, clouds are formed because natural aerosols, often sulfates or sea salt, serve as condensation nuclei. Similar to aerosols, the overall absorption and reflection of sunlight by clouds results in cooling. Clouds are estimated to shade about 60% of the earth at any given time [3], preventing much of the sunlight from reaching the earth's surface.

Technologies on Aerosol Measurement

The effect of aerosols on climate change became clear only about 40 years ago. Some of the oldest records on aerosol measurement can be traced back to 1905 from the Smithsonian Institute solar observatories. However, only seasonal and volcanic eruptions aerosols data were apparent. During the first half of the twentieth century, the observatories used an instrument called the "spectrobolometer" which is only located at 13 sites [7].

A larger network was developed in 1961 using an "analog Sun photometer" developed by Volz (1959) as a basis instrument [8]. There were 29 stations across the United States from 1961 to 1966 [9] and 30 stations across Europe from 1963 to 1967 [10]. No accuracy assessment was made for this network, however, the U.S. data are generally consistent with current measurements [8]. Later in 1977, the most ambitious attempt in history to monitor the global aerosol optical depth (AOD) was officially launched by the World Meteorological Organization Background Air Pollution Monitoring Network (WMO BAPMoN). The network composed of 95 sites operated by several member countries. Due to the diversity of instrument, expertise, analysis methods, and quality control, the network was shut down in 1992. Other attempts were made by smaller networks such as the Global Atmospheric Watch (GAW) network and the NASA's Global Learning and Observations to Benefit the Environment (GLOBE) program, but its results were either unpublished or unavailable [8].

Today, the largest and most successful program that has been monitoring aerosols and their properties at a global scale since 1993 is the Aerosol Robotic Network (AERONET) run by NASA. Currently, the network is providing aerosol data using the same instrument, calibration technique, and data analysis algorithm from approximately 400 sites in 50 countries. The latest instrument model used in the network is the CIMEL sun sky multiband photometer CE318-T [11]. Alongside the photometer, ground- and aircraft-based remote sensing instruments are being deployed at field sites through other programs such as the Atmospheric Radiation Measurement (ARM) or through multi-investigator field campaigns. Due to the spatial and temporal variation nature of atmospheric aerosols, a remote sensing instrument such as lidar has become very useful and popular. Unlike the sun sky photometer, it has the ability to rapidly record data, does not depend on the presence of the sun in the background, provides spatial AOD distribution, and in some cases can even classify aerosol species. However, there are still some challenges with lidar instruments such as their accuracy and consistency. The goal of the first project in this dissertation is to improve the accuracy and efficiency of the aerosol Micro-Pulsed Lidar (MPL) built at Montana State University and to demonstrate it in conjunction with the AERONET's sun-sky photometer.

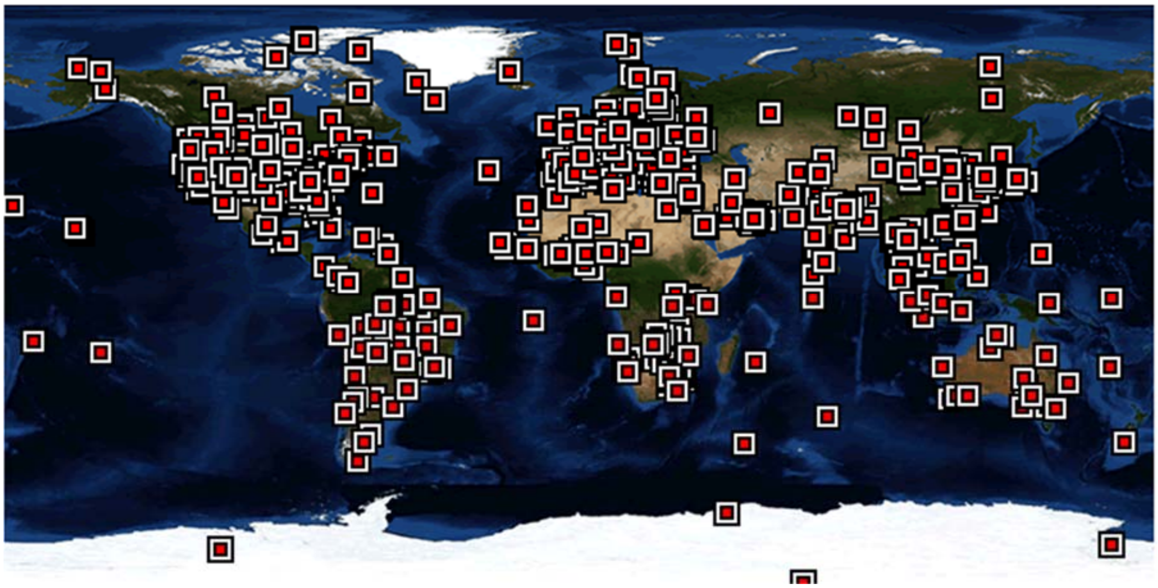


Figure 1.3 Current AERONET sites around the world (as of November 17, 2016). [11]

Figure 1.4 CIMEL sun sky multiband photometer model CE318-N at Montana State University, Bozeman site 45.67°N 111.05°W. (Picture courtesy by Joseph Shaw, Montana State University)

<u>Carbon Dioxide Effect on Climate Change</u>

As widely known nowadays, CO_2 is the most prevailing greenhouse gas that plays the largest role in global warming. According to the IPCC 2014 report, CO_2 made up about 76% of the total greenhouse gases emission in 2010. This greenhouse gas emission has almost doubled since 1970 [5] and has increased by a factor of 20 since 1900 [12].

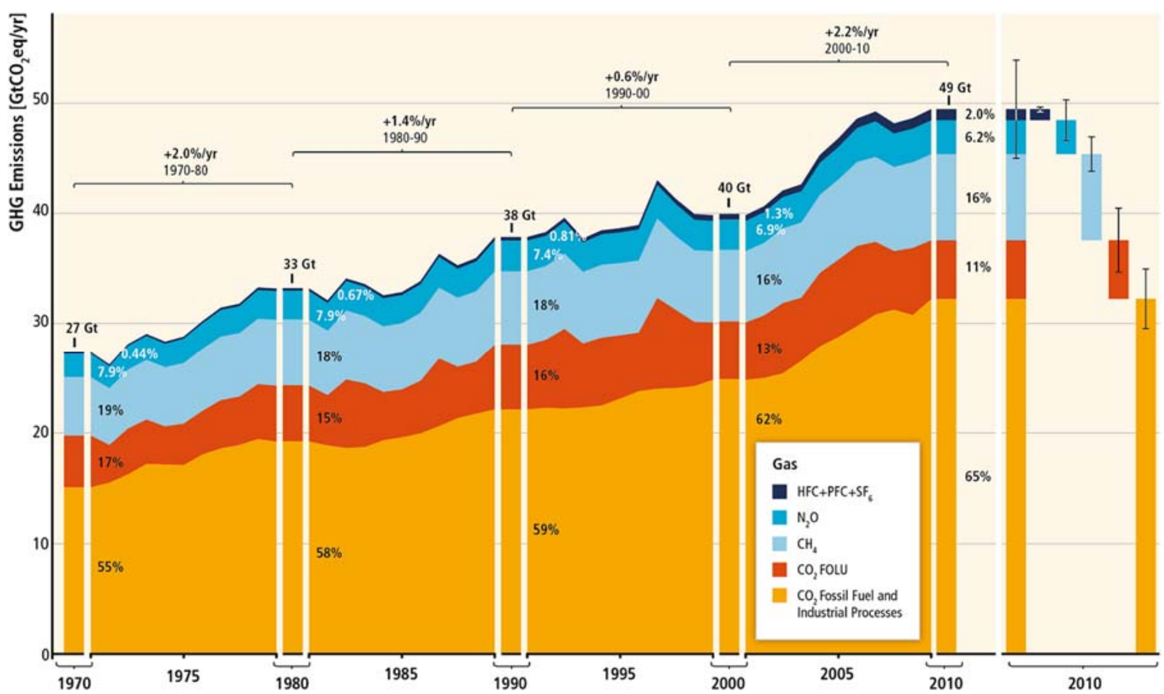


Figure 1.5 Total annual anthropogenic GHG emissions ($GtCO_2$ eq/yr) by groups of gases 1970–2010: CO_2 from fossil fuel combustion and industrial processes; CO_2 from Forestry and Other Land Use (FOLU); methane (CH_4); nitrous oxide (N_2O); fluorinated gases covered under the Kyoto Protocol (F-gases). At the right side of the figure GHG emissions in 2010 are shown again broken down into these components with the associated uncertainties (90% confidence interval) indicated by the error bars. Global CO_2 emissions from fossil fuel combustion are known within 8% uncertainty (90% confidence interval). CO_2 emissions from FOLU have very large uncertainties attached in the order of ±50%. Uncertainty for global emissions of CH_4, N_2O and the F-gases has been estimated as 20%, 60% and 20%, respectively. 2010 was the most recent year for which emission statistics on all gases as well as assessment of uncertainties were essentially complete at the time of data cut-off for this report. [5]

The global average temperature record since 1880 is shown in Figure 1.6. We can see the corresponding trend between an increasing in global temperature and increasing in greenhouse gases emission from 1970 to 2010. The plot demonstrates that the global temperature increased by 0.6 °C [13] in just a 40-year. At this rate, global warming has a dramatic effect on the earth that includes rising sea levels due to the melting of the polar ice caps, and more severe storms as well as other events such as El Niño.

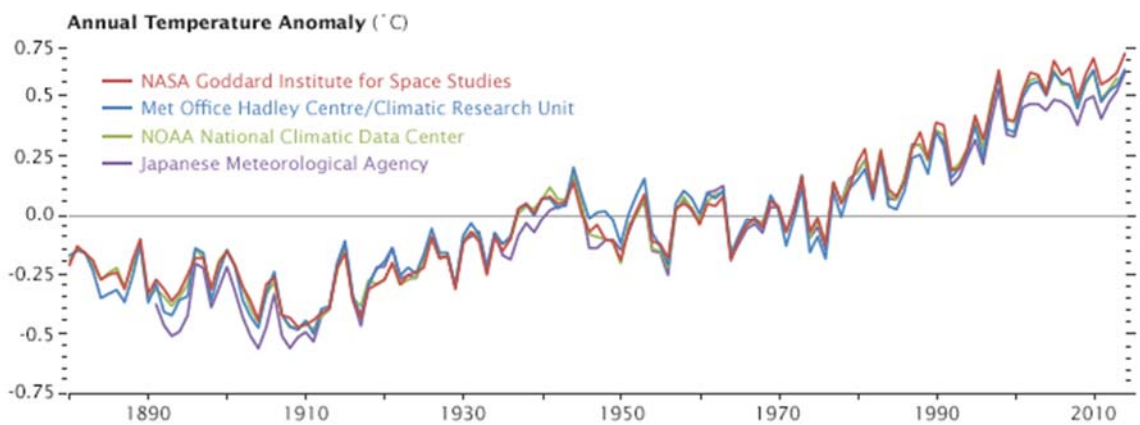


Figure 1.6 The record of global average temperatures compiled by NASA's Goddard Institute for Space Studies [13]. The "zero" on this graph corresponds to the mean temperature from 1961-1990, as directed by IPCC.

Technologies on Carbon Dioxide Measurement

The current largest network that is monitoring global greenhouse gases including CO_2 is run by National Oceanic and Atmospheric Administration (NOAA). This network, embraced by World Meteorological Organization (WMO), is not only providing up-to-date information on CO_2 and other gases, but is also raising and maintaining international collaboration on global warming issues [14]. Currently, the network consists of 96 active sites in 38 countries. Their methods of collecting air samples are via aircraft, balloon, tower, observatory, and surface flask. The air samples are returned to NOAA's Earth System Research Laboratory (ESRL) in Boulder, Colorado for analysis where up to 55 gases are traceable. This facility also maintains WMO's calibration scales for CO_2, CH_4, CO, N_2O, and SF_6 [15].

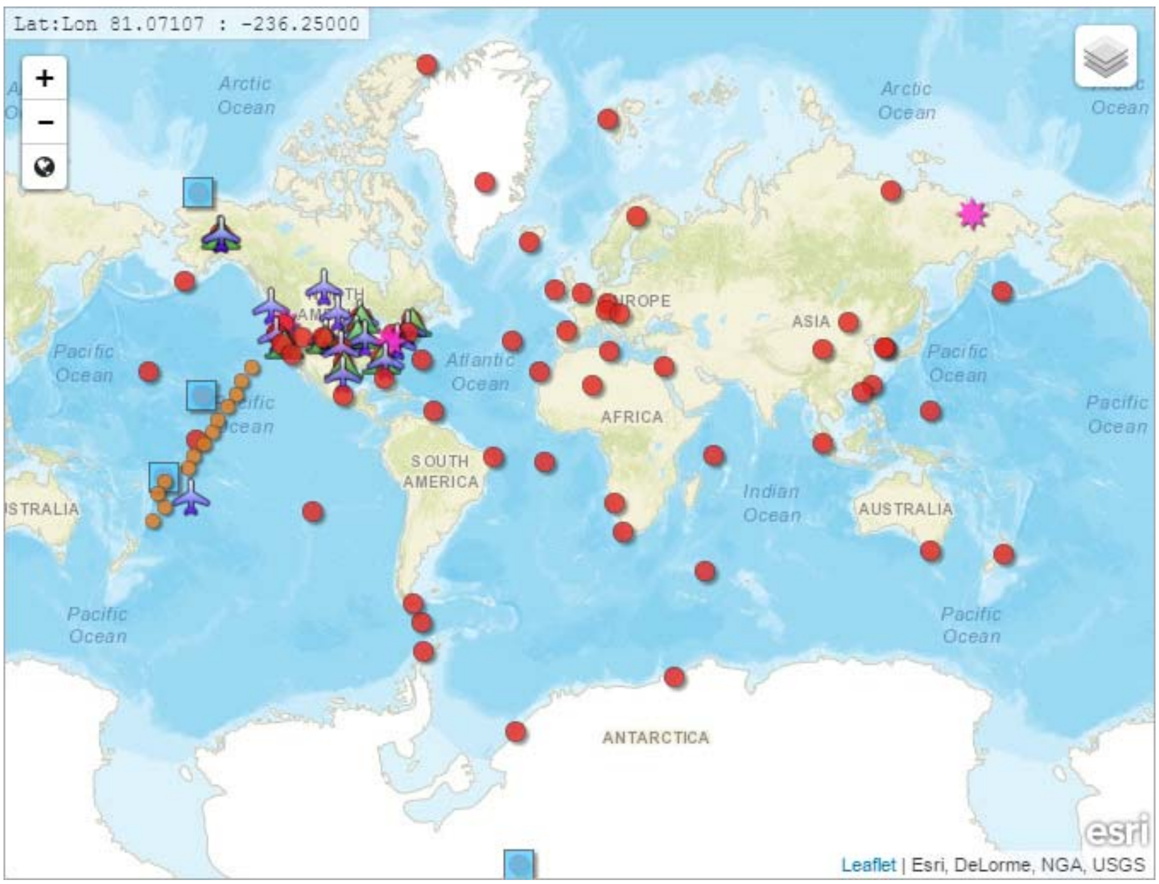


Figure 1.7 Current NOAA's Global Greenhouse Gas Reference Network active sites around the world (As of November 18, 2016). Air samples are collected by surface flask (red circle), airborne flask (purple airplane), in situ tower (green triangle), in situ observatory (blue square), and surface in situ (pink star). [16]

Although the NOAA network can provide accurate measurement of atmospheric CO_2 from direct samples, it is still insufficient for precisely understanding the sources and sinks of CO_2. The major problem with direct sample method is the lack of information on the spatial distribution of CO_2 and that it is not cost effective. Another method of monitoring CO_2 on a global scale is with passive satellites, which can provide the map of spatial distribution; however, their column-integrated measurements cannot retrieve detail of the concentration near the earth's surface. In addition, the spatial resolution from passive satellites can be very poor especially in the lower atmosphere where measurements are needed most for determining sources and sinks. Improved understanding of the spatial distribution of CO_2 and other greenhouse gases is necessary to quantify their anthropogenic climate impact. Another potential method of measuring CO_2 is with the Differential Absorption Lidar (DIAL) which is capable of providing spatial distribution with good resolution. While extensive research has been done on DIAL instrumentation for CO_2 measurements, accuracy of less than 2% error needed for carbon cycle studies remains a challenge. The goal of the second research project in this dissertation is to explore the potential of a singly resonant optical parametric oscillator (OPO) to be used in a CO_2 DIAL laser transmitter in order to improve its accuracy.

Dissertation Organization

This dissertation consists of two research projects—the first is on the development of the aerosol Micro-Pulsed Lidar (MPL) and the second is on the development of the CO_2 DIAL laser transmitter using the nonlinear optical parametric oscillator (OPO). The theories of lidar and DIAL in general along with the detailed lidar aerosol inversion technique and detailed CO_2 line selection for DIAL are described in Chapter 2. The theory of nonlinear processes is described in Chapter 3. The design and setup of the MPL as well as its measurement results are presented in Chapter 4. Design, setup, and results of the optical parametric amplification are presented in Chapter 5. The optical cavity locking technique and results from the optical parametric oscillation are presented in Chapter 6. The initial DIAL design is discussed in Chapter 7. Conclusions of the MPL and OPO projects with final thoughts and some insight towards future work are discussed in Chapter 8.

THEORY I

Principle of Lidar

Lidar is a form of optical remote sensing that uses laser light as its signal. By measuring the time of a returned signal, the distance of an object of interest can be retrieved. For this reason, lidar is widely used in high resolution mapping, 3D imaging, and object detection, which is involved in many applications such as geography, geology, forestry, seismology, archaeology, ecology, and engineering survey. In atmospheric science, lidar is used to retreive range-resolved profiles of clouds, water vapor, aerosols, and gases such as carbon dioxide and methane.

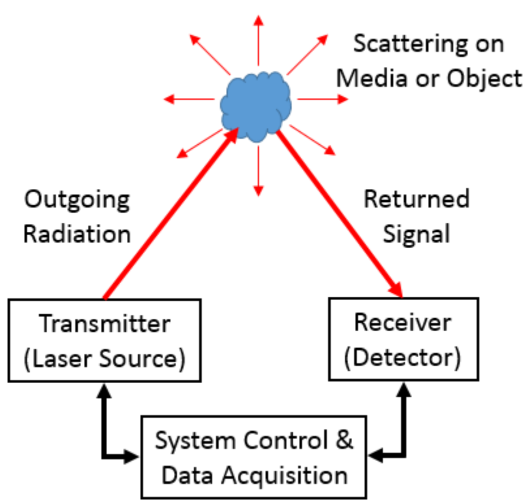


Figure 2.1 Schematic of basic architecture of a lidar.

As demonstrated in Figure 2.1, a typical lidar system consists of three main components—a transmitter, a receiver, and a system control and data acquisition (DAQ) unit. A transmitter is generally designed to provide laser pulses that meet certain requirements depending on application needs. Typical requirements include wavelength which can range from the UV to the NIR, pulse energy, pulse duration, repetition rate, bandwidth, and divergence angle. A transmitter usually comprises of a laser source(s), modulation/frequency shift device, beam splitter, beam combiner, amplifying optics, collimating optics, and diagnostic equipment. A receiver is used to collect the returned photons, filter out any unwanted wavelengths, compress background noises, and then converts them to electric signal via a detector. It usually comprises of an optical telescope, field stop, collimating optics, filter, and detector. A system control and DAQ unit is primarily used to record data and coordinate the time-of-flight between the outgoing signal and the returned signal in order to retrieve information for range resolution. In many cases, it is also designed to provide mechanical control of the system. A system control and DAQ unit usually comprises of a trigger control, multi-channel scaler card, and computer hardware and software.

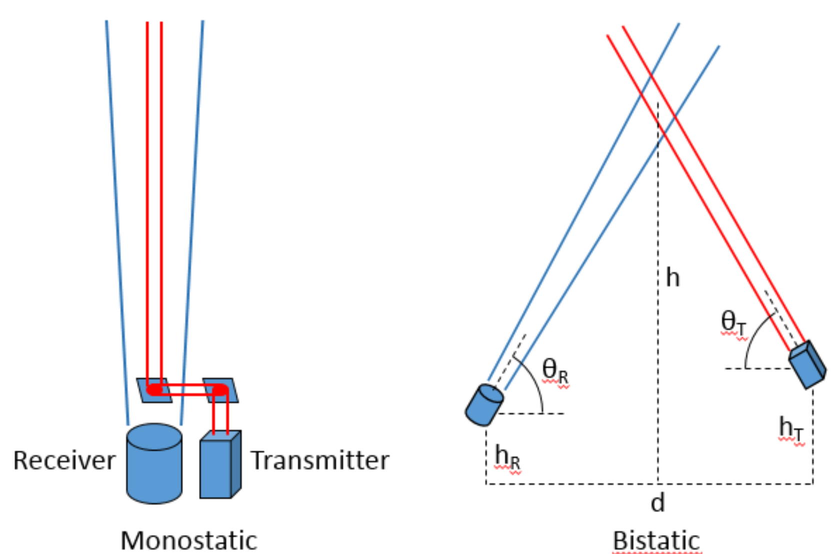


Figure 2.2 Basic configurations of lidar.

As shown in Figure 2.2, there are two basic lidar configurations—monostatic and bistatic. A monostatic configuration has the transmitter and receiver at the same location where both of their optical axes are often collinear. In this configuration, the range information is determined by

$$r = c \times \frac{\Delta t}{2} \tag{2.1}$$

where c is the speed of light and Δt is the time-of-flight of a laser pulse from exiting the transmitter to entering the receiver. A bistatic configuration has the transmitter and receiver at considerably different location which allows the use of geometry to determine the range information. In Figure 2.2, the height of an object can be calculated by

$$h = \frac{d \tan(\theta_T)\tan(\theta_R) + h_T \tan(\theta_R) + h_R \tan(\theta_T)}{\tan(\theta_T) + \tan(\theta_R)} \tag{2.2}$$

For this reason, the bistatic configuration typically uses a continuous wave (CW) laser rather than a pulse laser.

Lidar Equation

Consider an outgoing laser beam from a lidar propagating through space. The power of the outgoing beam at the distance $r + dr$ is

$$p(r + dr) = p(r)\left[1 - \sigma(r)dr\right] \tag{2.3}$$

where $p(r)$ is the power at the distance r and $\sigma(r)$ is the total atmospheric extinction coefficient taking into account both molecular and aerosol contributions where

$$\sigma(r) = \sigma_m(r) + \sigma_a(r) \tag{2.4}$$

where subscripts m and a denote molecular and aerosol, respectively, and the term "extinction" includes both scattering and absorption.

To solve for $p(r)$, equation (2.3) must first be rewritten in a differential form as

$$\frac{1}{p(r)}\frac{dp(r)}{dr} = -\sigma(r) \tag{2.5}$$

This first order nonlinear ordinary differential equation is easily seen to be separable and can be solved by

$$\int_{p(0)}^{p(r)} \frac{1}{p}dp = -\int_0^r \sigma(r')dr' \tag{2.6}$$

yielding

$$\ln\left[\frac{p(r)}{p(0)}\right] = -\int_0^r \sigma(r')dr' \tag{2.7}$$

If $p(0) = P_t$ is the transmitted power at the origin $(r = 0)$, then the power at the distance r is

$$p(r) = P_t e^{-\int_0^r \sigma(r')dr'} \tag{2.8}$$

Now consider the scattered light returning to the telescope whose field of view is wider than the beam. If the laser is pulsed at time $t = 0$, the scattered light from range r will return to the telescope at time t where

$$t = \frac{2r}{c} \tag{2.9}$$

where $c = 3\times10^8$ m/s, the speed of light in air. Also at a range between r and $r + \Delta r$, a fraction $\beta(r)\Delta r$ of the laser beam power $p(r)$ at distance r is scattered by some scattering density. Therefore, the power of the returned signal received by the telescope can be expressed as

$$P(r) = p(r)\frac{A}{r^2}\beta(r)\Delta r e^{-\int_0^r \sigma(r')dr'} \tag{2.10}$$

where $P(r)$ is the power of the returned signal received by the telescope.

$p(r)$ is the incident power at range r.

A is the area of the telescope collecting mirror.

r is the range where the laser is being scattered.

$\beta(r)$ is the back scatter coefficient.

Δr is the thickness of scattering volume or the range bin resolution.

$e^{-\int_0^r \sigma(r')dr'}$ is the attenuation along the return trip which is obtained from equation (2.8).

To express equation (2.10) in terms of the power of the outgoing beam, equation (2.8) can be used to substitute for $p(r)$ to get

$$P(r) = P_t \frac{A}{r^2}\Delta r \beta(r) e^{-2\int_0^r \sigma(r')dr'} \tag{2.11}$$

This is for the ideal case, where in reality the efficiency of the telescope and the overlap between the outgoing laser beam and telescope field of view must also be taken into account. In addition to that, many of the variables are actually functions of wavelength; so that for a lidar system, the actual power received by the detector is

$$P(\lambda, r) = P_t(\lambda)\frac{A}{r^2}\Delta r \beta(\lambda, r) T^2(\lambda, r) O(\lambda, r)\varepsilon_R(\lambda)\varepsilon_D(\lambda) \tag{2.12}$$

which is a general from of the lidar equation. See Figure 2.3 below for detail.

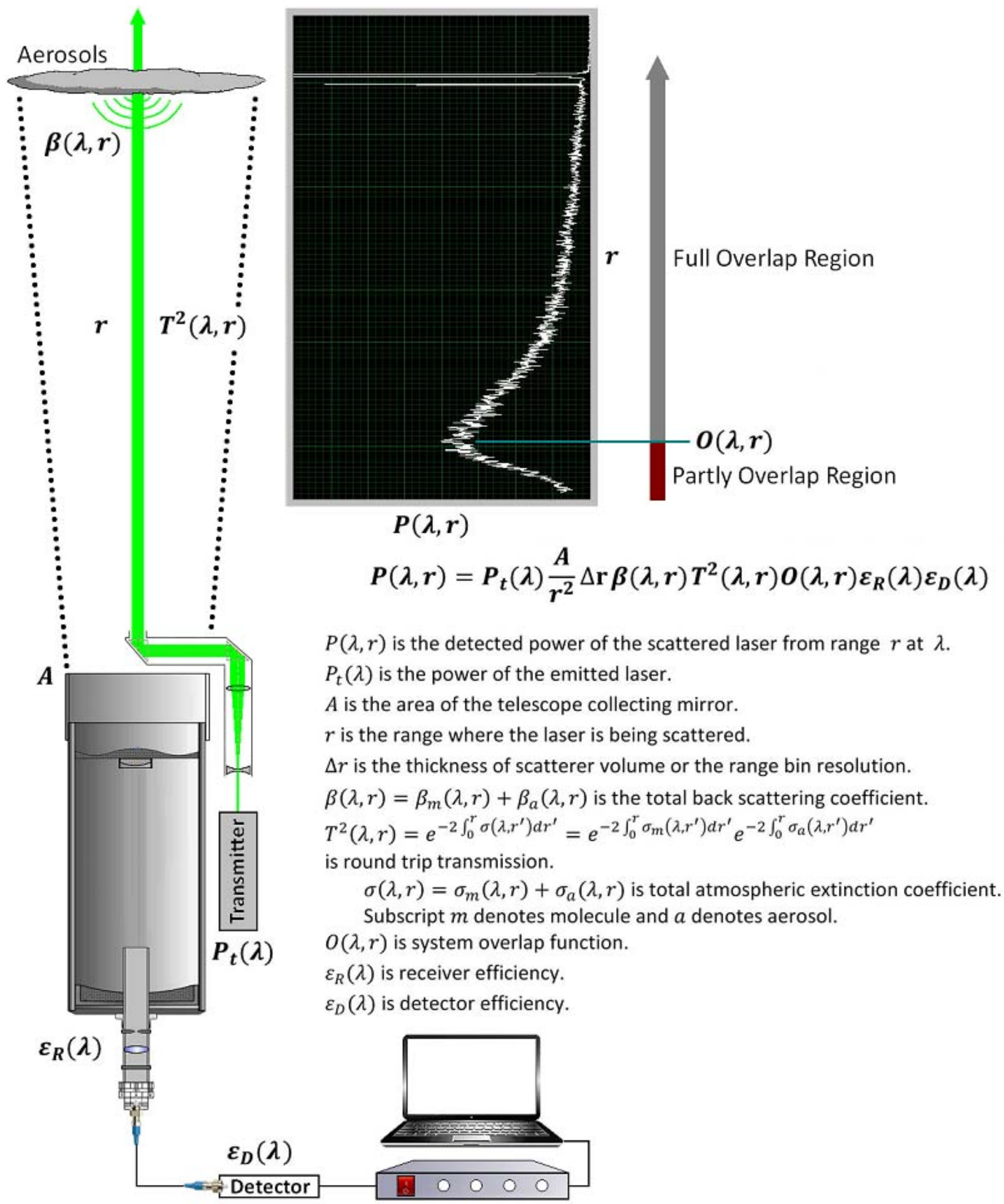


Figure 2.3 Schematic of a lidar system with the illustrating components in the lidar equation.

Aerosol Inversion

The lidar equation cannot be solved directly because it consists of two unknowns—the aerosol backscatter, $\beta_a(\lambda, r)$, and the aerosol extinction, $\sigma_a(\lambda, r)$. Because of the range resolution nature of lidar data, the equation must be solved numerically. To resolve these difficult issues, the "lidar inversion" or "aerosol inversion" technique [17, 18] is deployed for retrieving these aerosol optical properties.

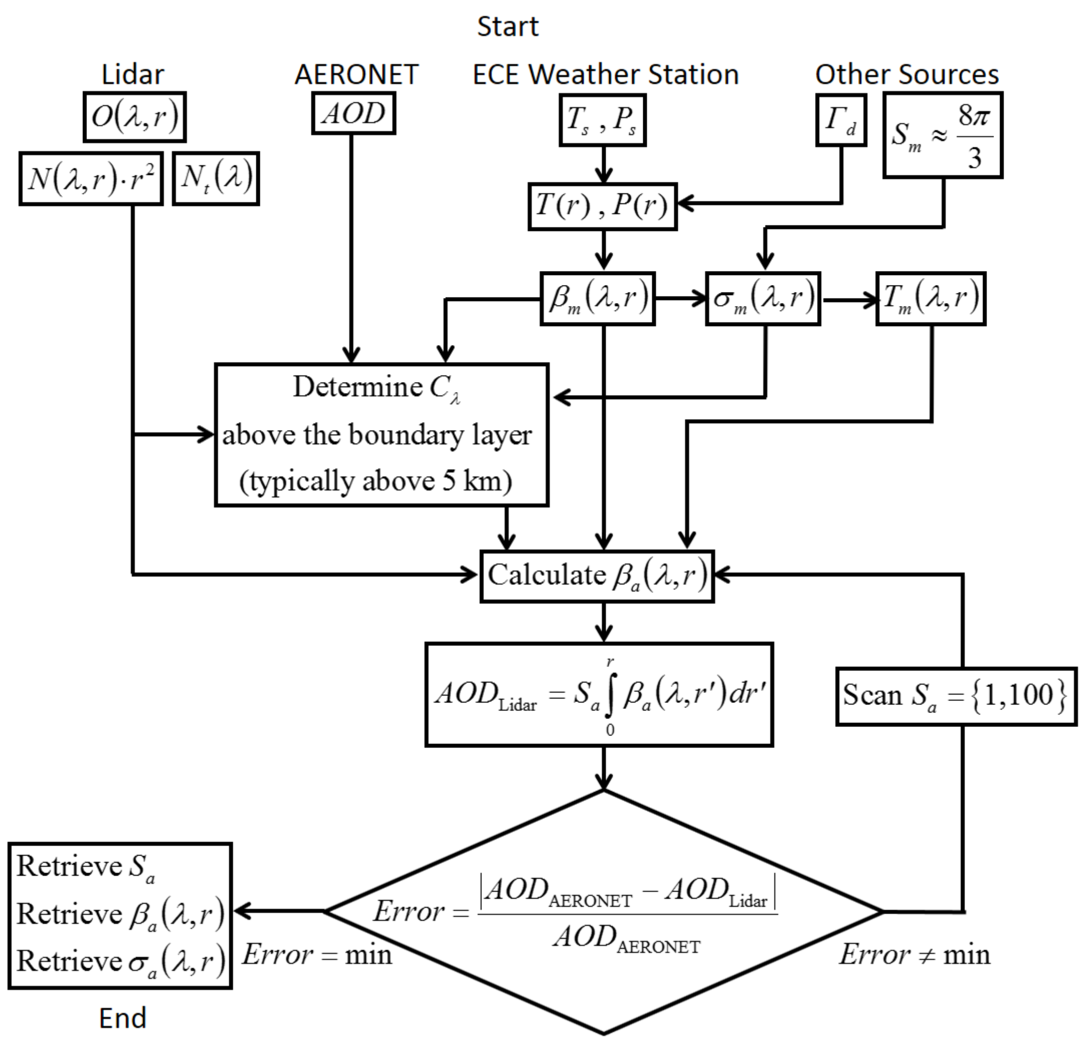


Figure 2.4 Flow chart showing key steps in the aerosol inversion.

The aerosol inversion technique follows the steps displayed in Figure 2.4. By establishing a variable, $S_a = \frac{\sigma_a(\lambda, r)}{\beta_a(\lambda, r)} = \text{constant}$, called the "aerosol lidar ratio", one of the unknowns in the lidar equation can be eliminated and therefore the lidar equation can be solved. In general, the only unknown left for solving is the spatial distribution of aerosol backscatter, $\beta_a(\lambda, r)$. Although only one unknown remians, solving the lidar equation is still not trivial. It requires a total aerosol optical depth (AOD) acquired from AERONET and a slope matching between the total backscatter, $\beta_{total}(\lambda, r)$, and the molecular backscatter, $\beta_m(\lambda, r)$, above the aerosol boundary layer. Further detail of the slope matching method will be discussed within this section of the dissertation; however, for simplicity, other variables in each step of the flow chart shall be discussed first.

<u>Overlap Function</u>

The overlap function, $O(r)$, in the lidar equation is often called the geometric factor because it purely depends on the geometry of the receiver optics. In Figure 2.5, the overlap function as a function of range $O(r) = 1$ when the size of the focused beam is smaller than the size of the fiber as demonstrated by the green beam. The red and blue beams shown in Figure 2.5 demonstrate when the system is partly overlaping because only part of the focused beam gets into the fiber, causing $O(r) < 1$. Note that $O_{red}(r) > O_{blue}(r)$ and the partially overlap function can be calculated by

$$O(r) = \frac{\text{Size of the fiber}}{\text{Size of the focused beam at the fiber location}} \tag{2.13}$$

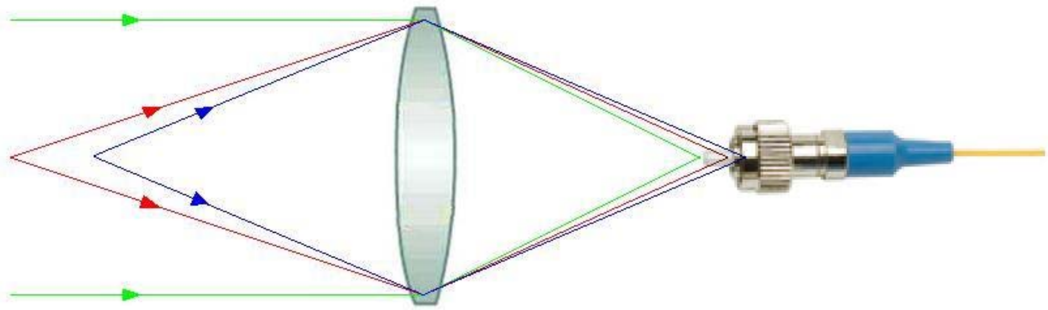


Figure 2.5 Beams from different locations focused by the fiber coupler lens onto the fiber result in different overlaps.

In the design, the overlap function of the system can be estimated from an optical design software such as Zemax; however, in practice, because of the complication from many receiver optical components and their alignment imperfection, the overlap function is usually acquired experimentally by retrieving the horizontal returns of the lidar. For this to be valid, the assumption must be made that the atmosphere is homogenous and well mixed. Under this assumption, the lidar light beam is directed parallel to the flat and uniform horizontal areas of the earth's surface where neither atmospheric disturbances nor local sources and plumes occur such that the extinction coefficient, $\sigma(\lambda, r)$, remains constant at the same altitude [19]. Although in reality, no absolute homogenous atmosphere exists, this assumption provides the most convenient and probably the most accurate way to obtain the overlap function of a lidar system.

Before we can calculate the overlap function from the lidar horizontal returns, we must first relate the lidar equation to the number of photon counts from the actual lidar returns. In other words, the lidar equation can be written as

$$N_{Raw}(r) = N_t \frac{A}{r^2} \Delta r \beta(r) T^2(r) O(r) \varepsilon_R \varepsilon_D \tag{2.14}$$

where N_{Raw} is the raw number of photons detected by the lidar receiver and N_t is the number of the outgoing photons from the lidar transmitter. Note that the power terms in equation (2.12) are replaced by the number of photons to yield equation (2.14) which still makes the lidar equation true because they are proportional to one another. Also note that the wavelength-dependent nature of variables in equation (2.12) can be omitted here in equation (2.14) if the lidar operates with only one wavelength which is the case for the Micro-Pulsed Lidar (MPL) in this dissertation. To further simplify the lidar equation, equation (2.14) can be expressed as

$$N(r) = N_t \frac{A}{r^2} \Delta r \beta(r) T^2(r) O(r) \varepsilon_R \tag{2.15}$$

where
$$N(r) = \frac{(C_R \times \text{Correction Factor}) - \text{Dark Count}}{\text{Photon Detection Efficiency Module}} - \text{Background Light} \tag{2.16}$$

and
$$\text{Correction Factor} = \frac{1}{1-(t_d \times C_R)} \tag{2.17}$$

Here $N(r)$ is called "the actual counts with background light subtraction" where the "background light" refers to other scattered lights in the sky (aside from the transmitted laser) that the lidar detector detects. Since $N(r)$ falls off as $1/r^2$, it should fall off very quickly and approach zero as r becomes very large; however, that is not the case. Instead of approaching zero, it approaches some constant value which has the magnitude depending on the brightness of the sky. This tells us that the lidar does not only detect the return counts of the aerosol scattered laser, but also the background light in the sky. We then must subtract these unwanted counts from the return signal. In practice, the background light can be determined by averaging the constant portion of the return signal at far range such as the signal in the last 750 m (15.75-16.50 km) in the MPL case.

The first term on the right hand side of equation (2.16) is the modified version of how the detector efficiency, ε_D , in equation (2.14) is being applied. The expression is based on the efficiency calculation in the spec sheet of the avalanche photodiode detector (APD) used in the MPL where C_R is the output module count rate and t_d is the module dead time. The photon detection efficiency module is estimated to be 0.54 for the MPL wavelength of 532 nm. All of these calculations including the background light subtraction are done within the Labview program that is used for the MPL control and DAQ. An example of recorded actual and raw counts of a horizontal profile is shown in Figure 2.6.

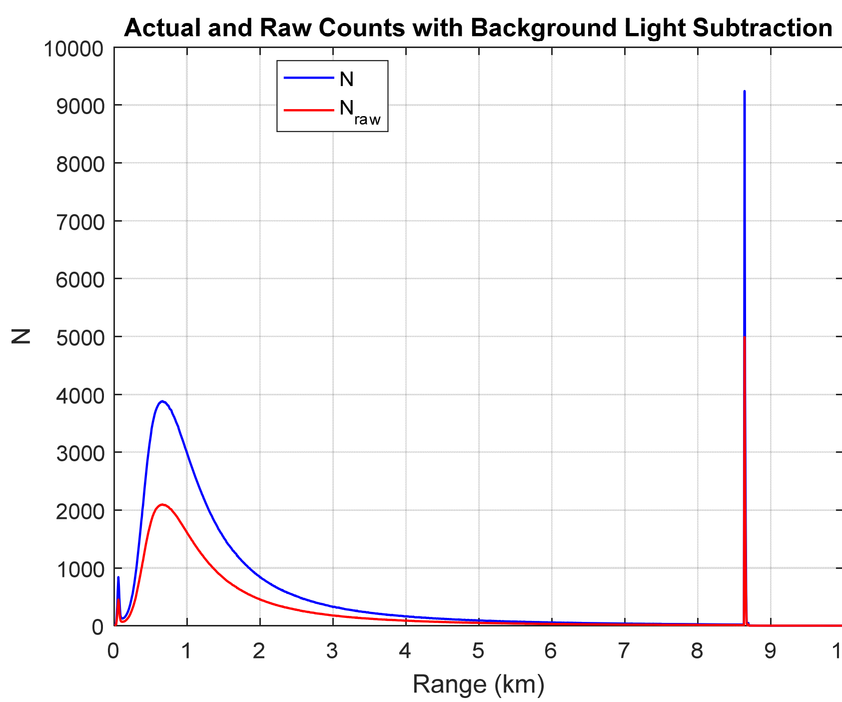


Figure 2.6 MPL time averaged horizontal profiles of actual and raw counts with background subtraction. Data was taken on August 9, 2016, 22:50:19-22:52:04. Horizontal data for the overlap calculation is usually taken at nighttime to minimize the returned signal noise. The peak signal at 8.6 km is the return from the mountain south of Bozeman.

Generally when solving the lidar equation, equation (2.15) is expressed in terms of the "range corrected return", $L(r)$, and the calibration constant, C_0, as

$$L(r) = C_0 \beta(r) T^2(r) O(r) \tag{2.18}$$

where
$$L(r) = N(r) \times r^2 \tag{2.19}$$

and
$$C_0 = N_t A \Delta r \varepsilon_R \tag{2.20}$$

Next we substitute $T^2(r) = e^{-2\int_0^r \sigma(r')dr'}$ into equation (2.18) and treat $\sigma(r)$ and $\beta(r)$ as constant under the assumption of homogenous and well mixed atmosphere [19]. Then

$$L(r) = C_0 \beta e^{-2\sigma r} O(r) \tag{2.21}$$

Taking the natural log on both sides,

$$\ln\left[L(r)\right] = \ln\left[C_0 \beta\right] - 2\sigma r + \ln\left[O(r)\right] \tag{2.22}$$

As discussed earlier, at a range where the system is in full overlap, $O(r) = 1$, making the third term in equation (2.22) disappear, reducing the equation to

$$\ln\left[L(r)\right] = \ln\left[C_0 \beta\right] - 2\sigma r \tag{2.23}$$

which describes a linear relationship between $\ln\left[L(r)\right]$ and r as shown in the linear portion of the range corrected return plot in Figure 2.7.

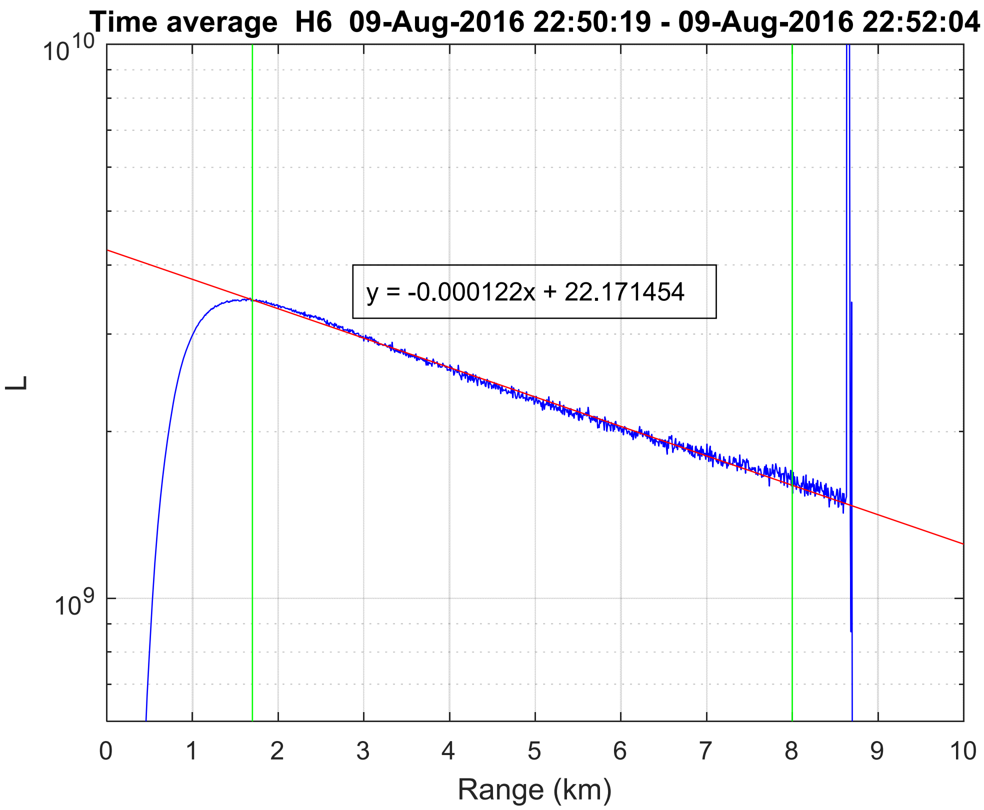


Figure 2.7 Plot of the range corrected return as a function of range (blue) on a log scale. The full overlap begins where the plot becomes linear starting at 1.7 km indicating by the left green line. The straight fit line (red) within the two green lines portion provides the slope and intercept which are necessary for retrieving the overlap function.

The term -2σ and $\ln\left[C_0\beta\right]$ in equation (2.23) are determined from the slope and intercept of the straight-fit line in Figure 2.7 plot. With this information, the overlap function can now be determined by solving equation (2.22) as follow

$$\ln\left[O(r)\right]=\ln\left[L(r)\right]-\ln\left[C_0\beta\right]+2\sigma r \tag{2.24}$$

and thus

$$O(r)=e^{\ln\left[L(r)\right]-\ln\left[C_0\beta\right]+2\sigma r} \tag{2.25}$$

or

$$O(r)=e^{\ln\left[L(r)\right]-22.171454+0.000122r} \tag{2.26}$$

based on the result in Figure 2.7.

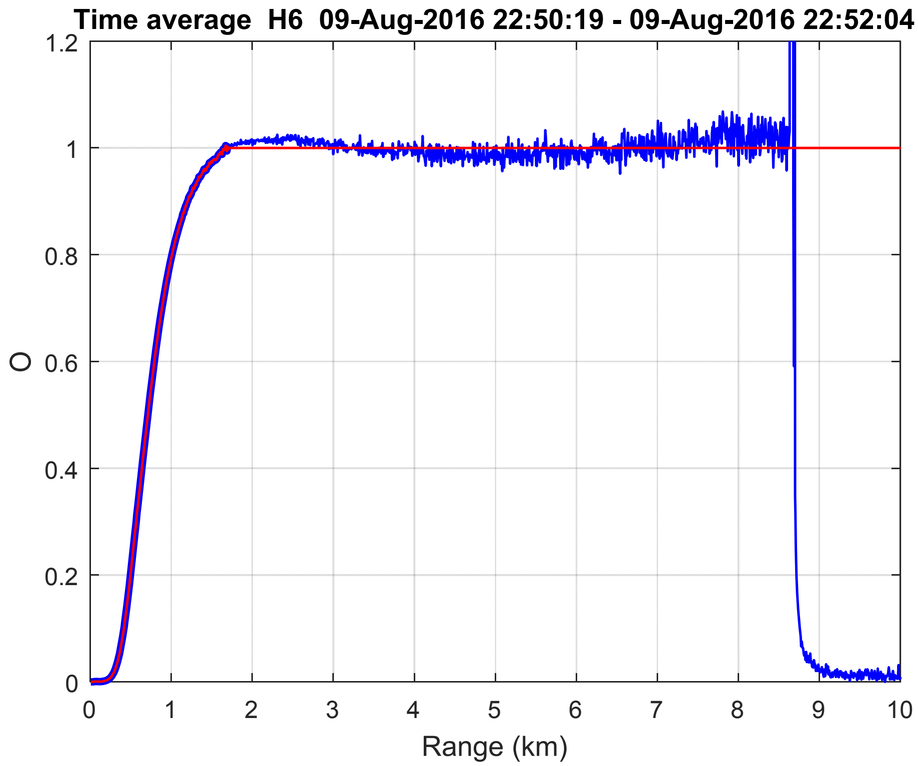


Figure 2.8 Overlap function (blue) and modified overlap function (red).

The retrieved system overlap function plot is shown in blue in Figure 2.8. At range beyond 1.7 km, the system is in full overlap, and therefore the overlap function can be set to 1 as shown in the modified overlap function plot in red. In comparison, in the range from 1.7 to 8.5 km, the modified overlap (red) is within 5% error from the retrieved overlap (blue) which is within the acceptable error, and thus will be used for further calculation in the aerosol inversion.

To demonstrate how to apply the overlap function, we divide the actual returns, $N(r)$, and the range corrected returns, $L(r)$, by the overlap function, to correct the returns in the partially overlapping region near the ground. As an example, the overlap function has been applied to the horizontal data itself. The data with and without the overlap correction are plotted in Figure 2.9. By applying the overlap function, data in the partially overlap region (below 1.7 km) is corrected as if the system is in full overlap. In this region, the $N(r)/O(r)$ curve in black extends the $1/r^2$ fall off nature of $N(r)$ plotted in green, and the $L(r)/O(r)$ plotted in red extends the linear slope nature of $L(r)$ plotted in blue. In the region where the system is in full overlap (above 1.7 km), $O(r)=1$, there is no correction done on $N(r)$ and $L(r)$. In other words, $N(r)$ and $L(r)$ are what the lidar sees while $N(r)/O(r)$ and $L(r)/O(r)$ are what the actual returns should be. The overlap function retrieved from the horizontal returns will be used as a correction for the vertical returns of the MPL when retrieving for atmospheric aerosol.

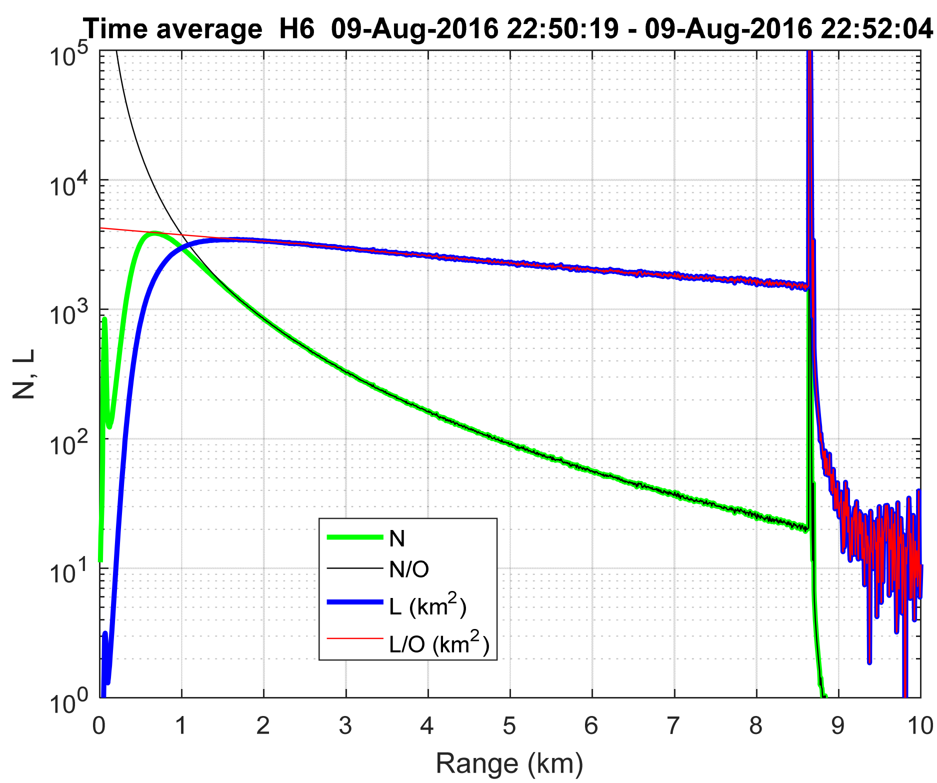


Figure 2.9 Comparison of returns with and without overlap correction.

Matching Technique between Horizontal Overlap and Vertical Returns

The hardest problem with the MPL aerosol measurement is to achieve good accuracy. Even with careful measurement on every single step, the retrieved lidar ratio, S_a, of the lidar can be very different from that of the AERONET's sun sky photometer. One of the biggest causes of error is a change in the alignment of the optical components within the system when going from horizontal to vertical orientation. Even a small shift or tilt from one component can cause a big effect on the returns going into the receiver fiber as shown in Figure 2.3. To put this into perspective, the fiber size used in the MPL is only 105 μm in diameter and thus only takes about this much shift to cause the returns to completely miss the fiber. This problem cannot be solved by merely getting a bigger fiber. Increasing the size of fiber means increasing the amount of unwanted light while still not being able to solve other issues such as the shifting of focus.

The component that seems to have caused the biggest misalignment when going from horizontal to vertical is the primary mirror of the receiving telescope. The gravity sag on the telescope mirror's vertical orientation causes the telescope's focus to shift from its horizontal orientation. To minimize this shifting, every component in the MPL must be carefully aligned and tightened in place. Even with our best effort, the shifting still delivers noticeable error on the data.

The matching technique between the horizontal overlap and vertical returns is introduced to help alleviate the error. First, the horizontal overlaps are determined for various telescope focuses as shown in the plots of Figure 2.10. Then the corresponding horizontal data for these overlaps are normalized and compare with the normalized vertical data as shown in Figure 2.11. The matching technique matches the rising slope between the horizontal and vertical data in the partially overlap region. After the matching is completed, the calculated overlap from the selected horizontal data is applied to the vertical returns. An example of the vertical range corrected returns with horizontal overlap correction is demonstrated in Figure 2.12. In the plot, $L(r)$ is plotted in green, $L(r)/O(r)$ in blue, and the near ground modified version of $L(r)/O(r)$ in red. Since $O(r)$ provides the most uncertainty near ground (0~300 m), sometimes it is appropriate to estimate and modify $L(r)/O(r)$ in this region.

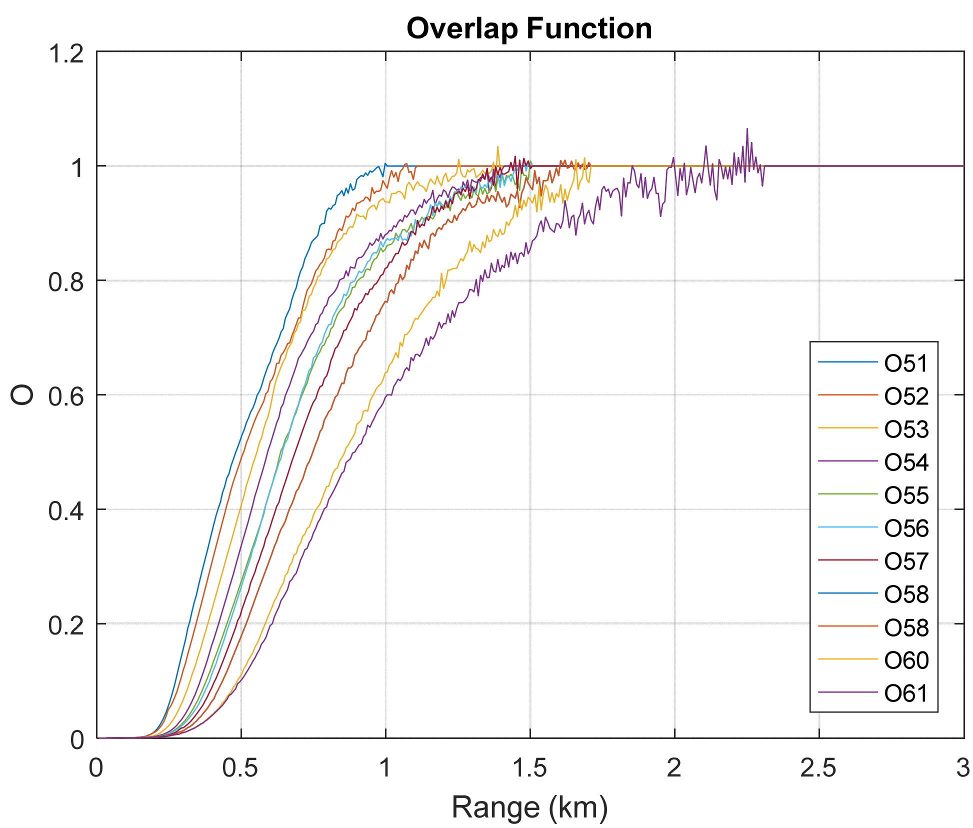


Figure 2.10 Plots of horizontal overlaps for telescope focus adjusted from near to far.

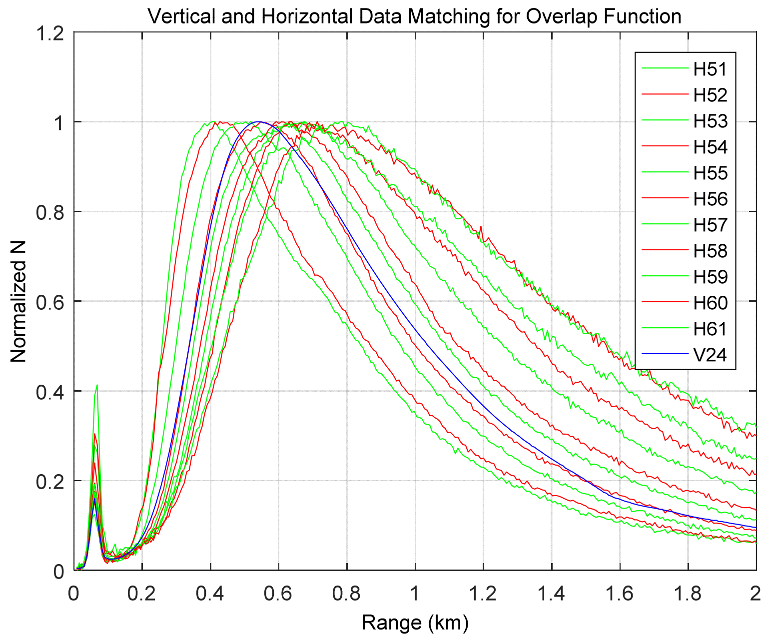


Figure 2.11 Matching between the horizontal (green and red) and vertical returns (blue). On the plot, H51 data is plotted in the far left, then H52 and so on. Based on the rising slope, the best match with the vertical returns is H54.

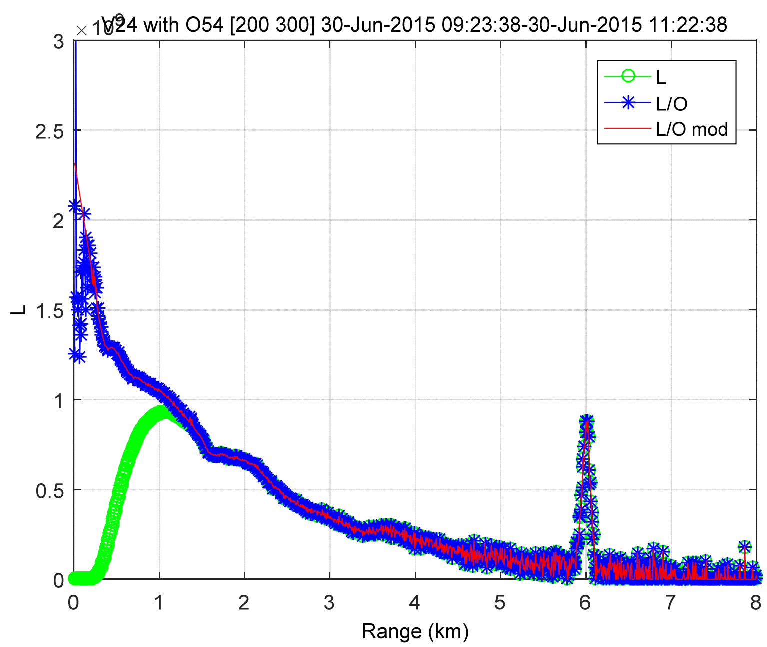


Figure 2.12 Time averaged vertical range corrected returns with (blue) and without (green) horizontal overlap applied. The near ground returns where the overlap gives the most uncertainty is modified and plotted in red. Peak at 6 km indicates clouds that floated by during that time window.

AERONET AOD and Lidar Ratio

As shown in the Figure 2.4 flow chart, a total aerosol optical depth (AOD) is needed as a reference in the aerosol inversion technique. This information can be obtained from NASA's AERONET website where the sun sky photometer data are processed and published. The needed data is listed in the "data display" under "aerosol optical depth (V2)" section, "Bozeman" site, and "AOD level 1.5" as shown in Figure 2.13.

Other useful information from AERONET are the phase function (PFN) and the single scattering albedo (SSA). Although these are not required in the inversion, they can be used to estimate the aerosol lidar ratio, S_a, beforehand as

$$S_a = \frac{4\pi}{\mathrm{PFN}\left(180^\circ\right) \times \mathrm{SSA}} \tag{2.27}$$

where $\mathrm{PFN}\left(180^\circ\right)$ is the phase function at 180°.

The download page for PFN and SSA is listed in "download tool" under "aerosol inversions (V2)", "Bozeman" site as shown in Figure 2.14.

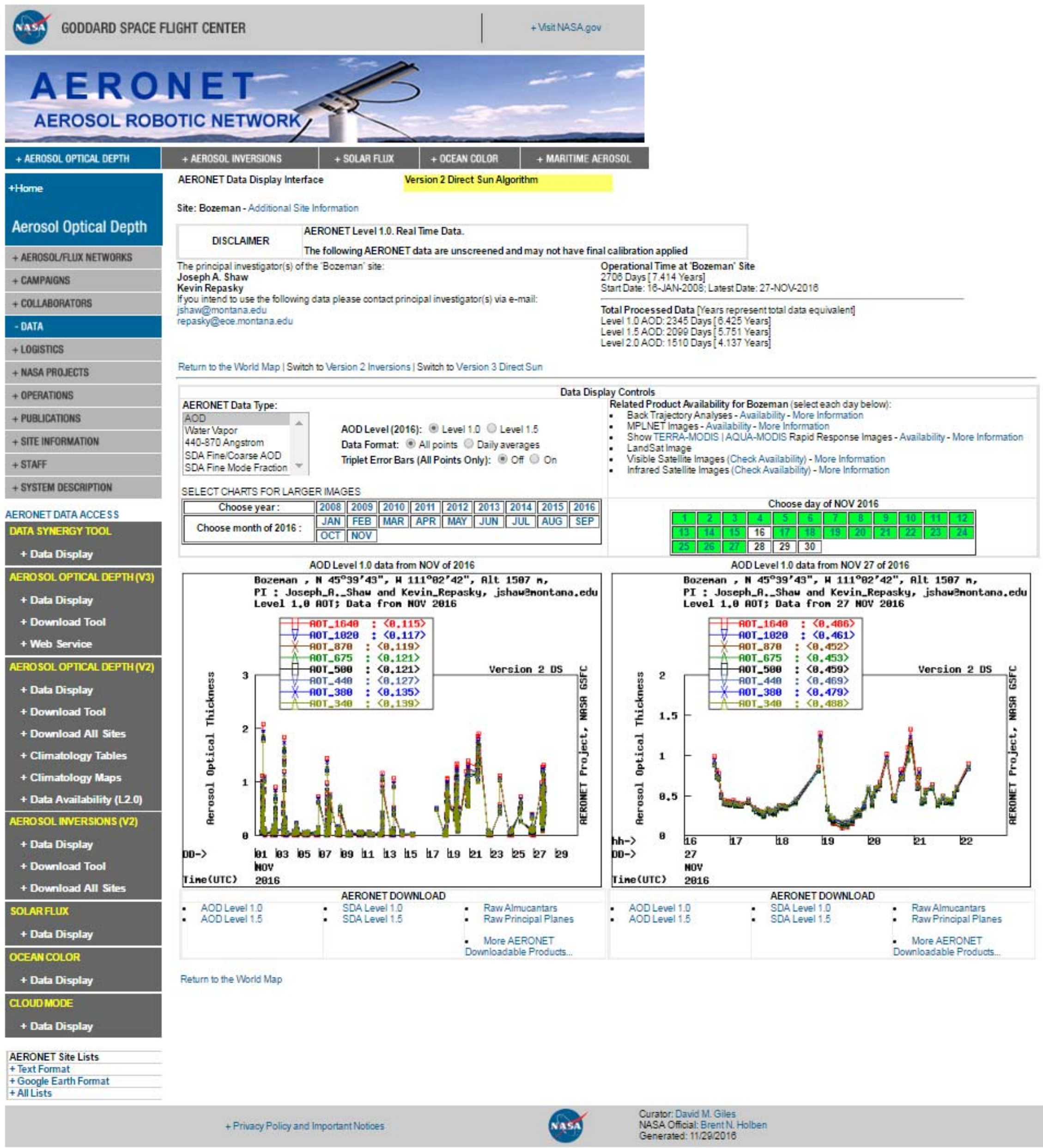


Figure 2.13 AERONET AOD download page. [20]

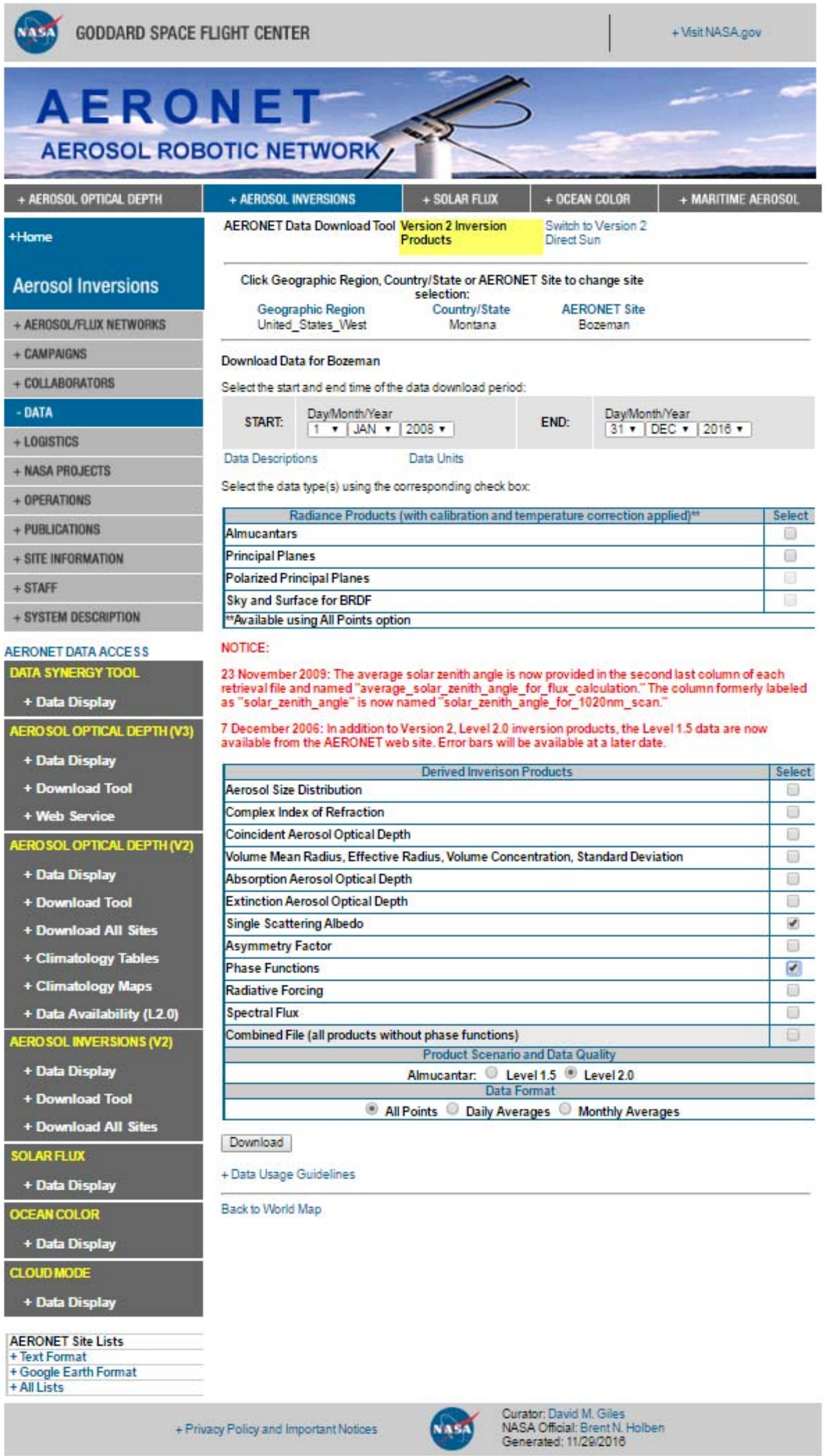


Figure 2.14 AERONET phase function and single scattering albedo download page. [21]

Surface Temperature and Pressure

Other information needed in the aerosol inversion is the surface temperature and pressure, T_s and P_s. They can be obtained from the MSU ECE weather station website [22]. The station's barometer, temperature probe, humidity probe, and wind monitor are located on top of Cobleigh Hall, Montana State University, Bozeman, Latitude 45.67° N, Longitude 111.05° W, elevation 1524 m. A screen shot of the weather station website is shown in Figure 2.15.

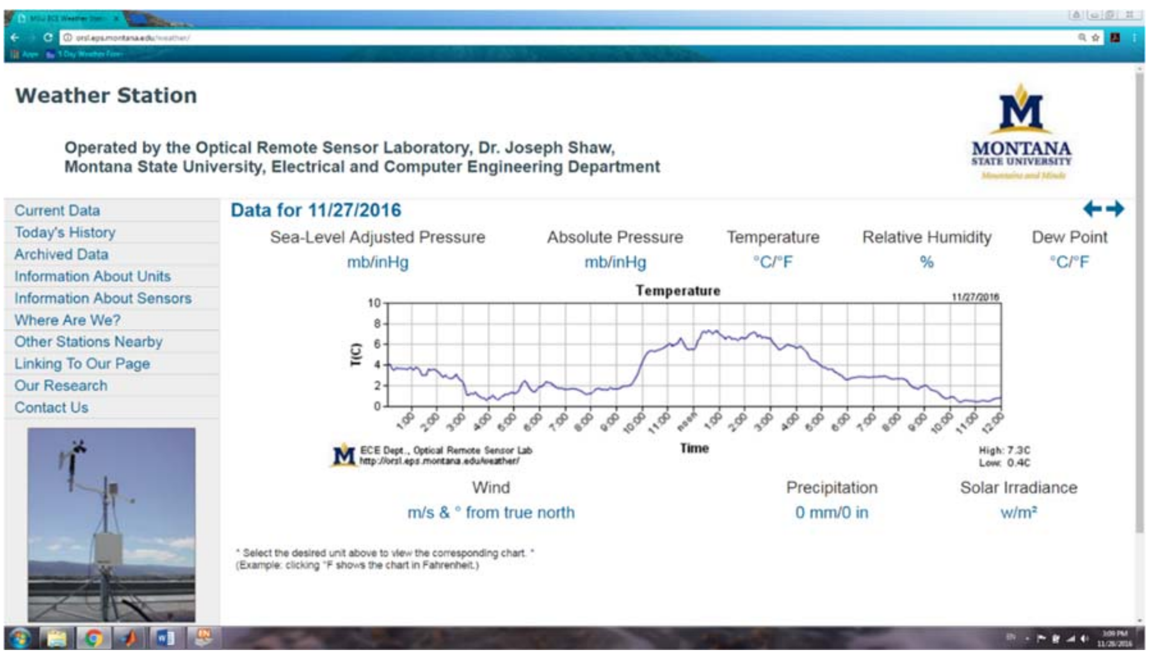


Figure 2.15 MSU ECE weather station website. [22]

Synchronization of Lidar, AERONET, and Weather Station Data

As shown in Figure 2.4 flow chart, the aerosol inversion data come from three sources—the lidar, AERONET, and MSU ECE weather station. These sources record data at their own different time and time intervals. Time synchronization for these data must be executed before proceeding onto the next step. First, the AERONET's GMT time is converted to lidar's and weather station's MDT time. Then, the interpolation approach is used for time synchronization.

To put this into perspective, the available data points from each source on June 30, 2015 are shown in Figure 2.16. The synchronized time window is selected based on the availability of lidar, AERONET AOD, and weather station data. In Figure 2.16, the synchronized data (pink) must be within existing period of lidar (green), AERONET AOD (red), and weather station (blue) data. Noting that AERONET SSA and PFN are only for the comparison of final results and thus are not considered when selecting the synchronized time window.

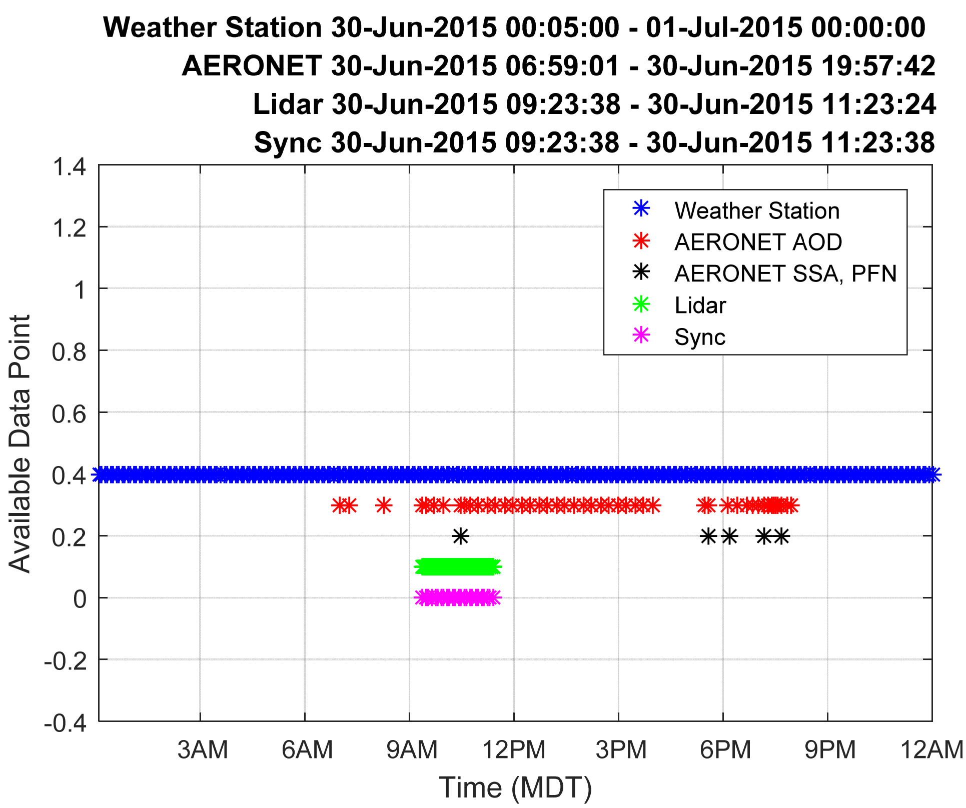


Figure 2.16 Available data points on June 30, 2015.

As shown in Figure 2.16, weather station records data every 5-minute interval while the typically recorded time interval for lidar is set to 10-11 seconds. The AERONET AOD recorded time interval can range from about 10 minutes up to several hours and sometimes may not even be available. This is because the sun sky photometer cannot take rapid measurements and it also depends on the background sunlight. If thick enough clouds present, it will result in unsuccessful measurement because that particular set of data is discarded by the instrument. With these differences in recorded time intervals, it is important to accurately interpolate the data when synchronizing them. The synchronized time interval can be set to anything depending on the resolution wanted. Example plots in this section have it set at 5 minutes.

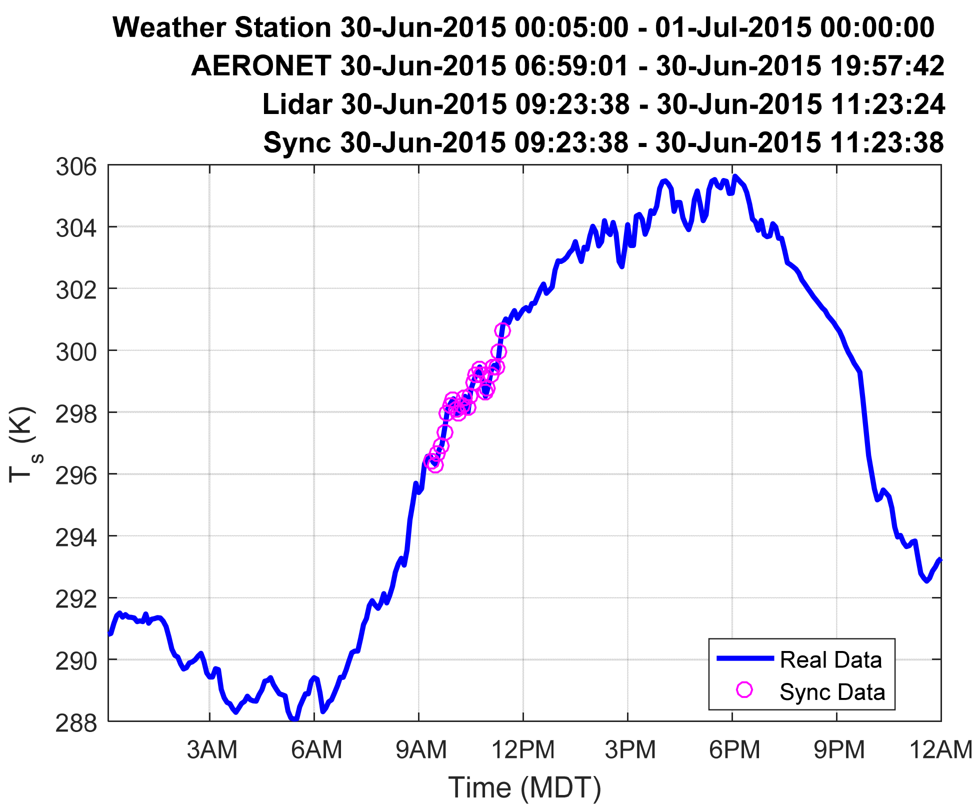


Figure 2.17 Weather station surface temperature data (blue) and synchronized data (pink).

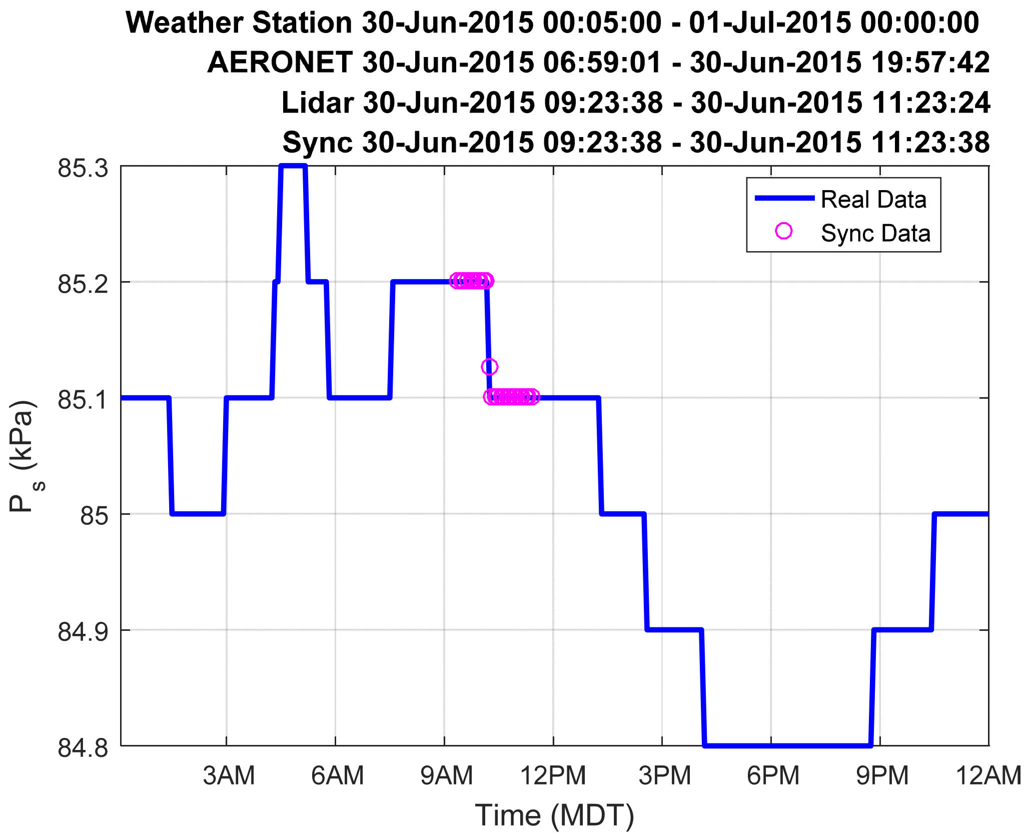


Figure 2.18 Weather station surface pressure data (blue) and synchronized data (pink).

The surface temperature and pressure from the weather station and their synchronized values are plotted in Figure 2.17 and Figure 2.18, respectively. Both show good agreement between the real and synchronized data, meaning that the interpolation technique works well for synchronization.

Data from AERONET are plotted in Figure 2.19, Figure 2.20, and Figure 2.21. AOD data are available at 500 and 675 nm while the SSA and PFN data are available at 440 and 675 nm. Since the MPL operates at 532 nm, these data must be linearly interpolated for this wavelength first as shown in green. Then time synchronization is performed and plotted in pink.

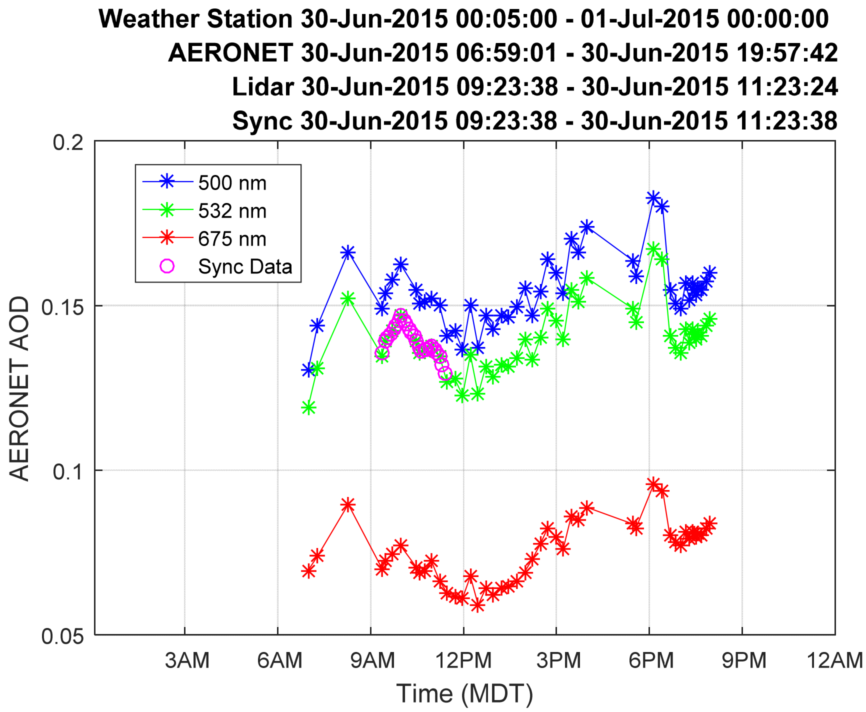


Figure 2.19 AERONET AOD measured at 500 nm (blue) and 675 nm (red), linearly interpolated AOD for 532 nm (green), and synchronized AOD (pink).

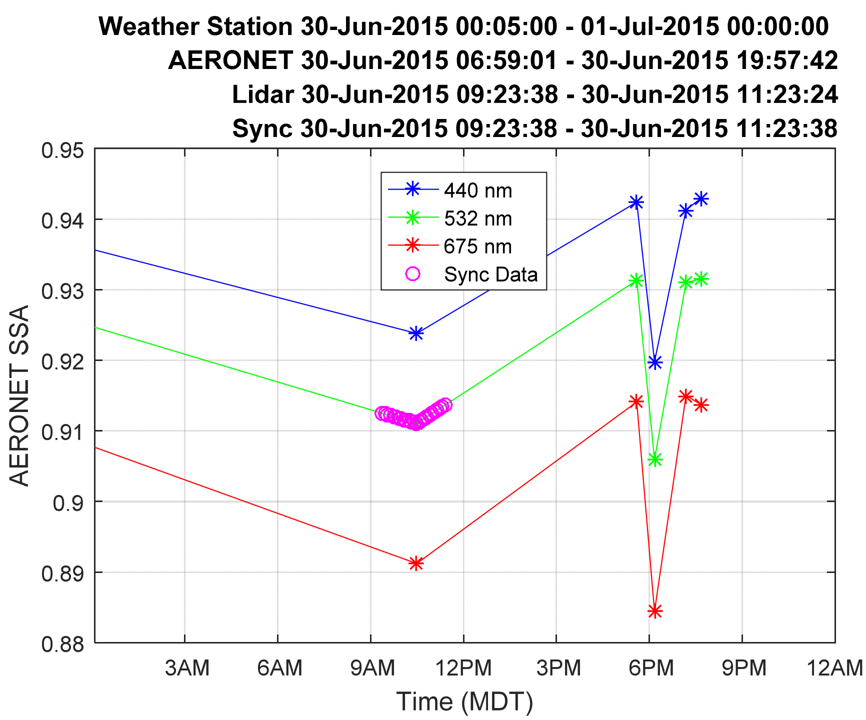


Figure 2.20 AERONET single scattering albedo measured at 440 nm (blue) and 675 nm (red), linearly interpolated SSA for 532 nm (green), and synchronized SSA (pink).

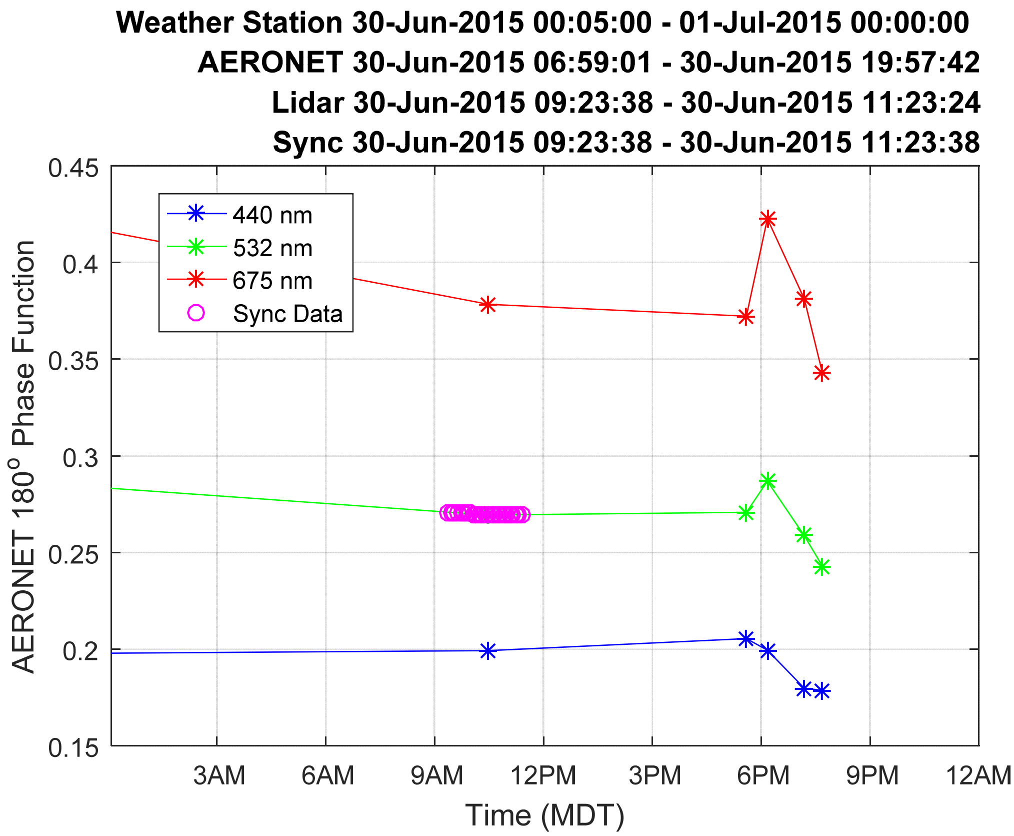


Figure 2.21 AERONET 180° phase function measured at 440 nm (blue) and 675 nm (red), linearly interpolated PFN for 532 nm (green), and synchronized PFN (pink).

After data from the lidar, AERONET, and MSU ECE weather station are time synchronized, the next step in the Figure 2.4 flow chart is to look at the lapse rate and atmospheric temperature model.

Lapse Rate and Temperature Model

Under the following assumptions [23]

1. The earth's atmosphere is adiabatic in which the physical state is described by pressure, volume, and temperature without any heat being added or withdrawn from it.

2. The earth's atmosphere is in hydrostatic equilibrium in which the air remains the same pressure at the same altitude.
3. An air parcel moves slowly enough such that its macroscopic kinetic energy is negligible compare to its total energy.

The change in temperature with respect to altitude for a dry air parcel is given by

$$-\frac{dT}{dz} = \frac{g}{C_p} = \Gamma_d \tag{2.28}$$

where T is the air temperature in Kelvin, z is the altitude in meter, $g = 9.81\ \text{m/s}^2$ is the gravitational acceleration, $C_p = 1004\ \text{J/K}\cdot\text{kg}$ is the specific heat at constant pressure for dry air, and Γ_d is called the dry adiabatic lapse rate which has the value of $\frac{g}{C_p} = 9.8\ \text{K/km}$.

It is important to emphasize that Γ_d is the rate of change of temperature of a parcel of dry air that is being raised or lowered adiabatically in the atmosphere. The actual lapse rate $\Gamma = \frac{\partial T}{\partial z}$ as measured by radiosonde for moist air has a value between 5.0-9.8 K/km in the troposphere. By rearranging equation (2.28), replacing Γ_d with Γ, and then integrating both sides

$$\int_{T_s}^{T} dT = -\int_{z_s}^{z} \Gamma dz$$

we get

$$T = T_s - \Gamma\left(z - z_s\right) = T_s - \Gamma h \tag{2.29}$$

where T_s and z_s are the surface temperature and altitude, and h is the vertical height from the surface.

To acquire the lapse rate at the Bozeman location where this research takes place, the radiosondes were launched from time to time to collect the atmospheric temperature and pressure data. As an example, Figure 2.22 shows radiosonde's temperature data plots in green during November 23-26, 2013 launches. For this group of data, the linear fit line in blue between h = 1.9-10.2 km indicates that the temperature decays with the slope of 7.7 K/km, which is the lapse rate for those days. Based on these results, the temperature model for Bozeman is

$$\begin{aligned} T &= T_s && , 0-1.9 \text{ km} \\ &= T_s - \Gamma h && , 1.9-10.2 \text{ km} \\ &= T(10.2 \text{ km}) && , 10.2-20 \text{ km} \end{aligned} \tag{2.30}$$

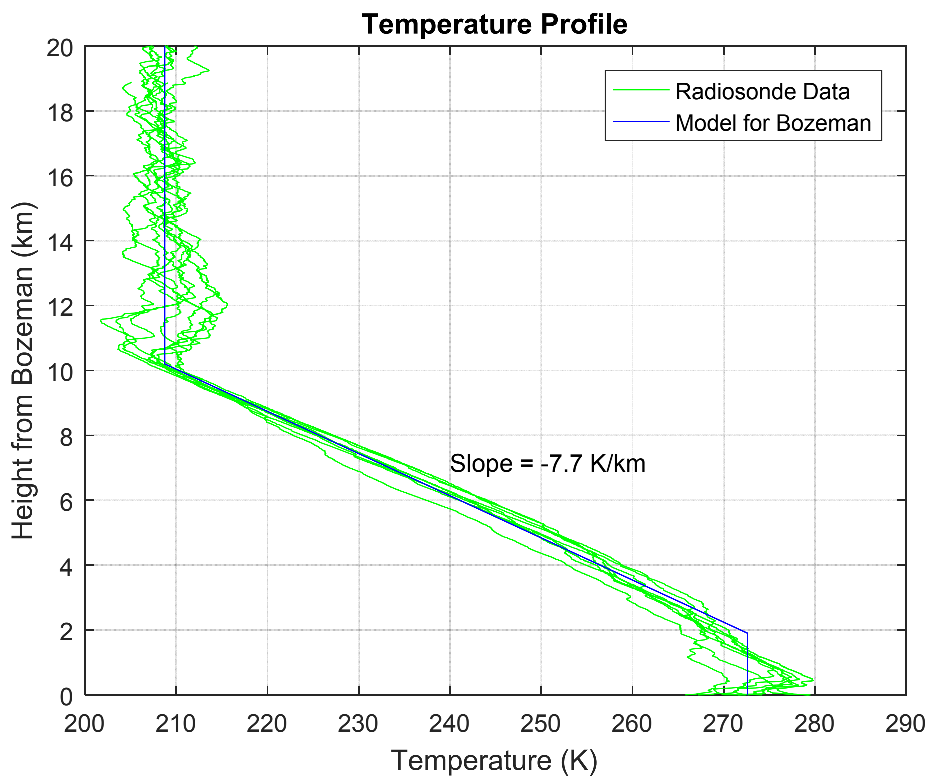


Figure 2.22 Temperature profiles from radiosonde (green) launched at Bozeman (N 45°39'43", W 111°02'42"), from November 23-26, 2013, twice daily and temperature model for Bozeman (blue).

Pressure Model

The pressure P in Pascal as a function of altitude h in the first 11 km of the atmosphere can be predicted by [19]

$$\begin{aligned} P &= P_s\left(\frac{T_s}{T_s-\Gamma h}\right)^{-\left(\frac{0.034164}{\Gamma}\right)} && ,\ 0-11\ \text{km} \\ &= P(11\ \text{km})\times e^{\frac{-0.034164(h-11000)}{T(11\ \text{km})}} && ,\ 11-20\ \text{km} \end{aligned} \tag{2.31}$$

where P_s and T_s are the surface pressure and temperature, and Γ is the lapse rate.

Although this model can provide a good approximation for average conditions of the atmosphere, it is considered somewhat different from the actual conditions at Bozeman. As shown in Figure 2.23, the average condition model plotted in red deviates from the radiosonde data plotted in green at a large maximum error of about 9% at 11 km. In comparison, the Bozeman model, calculated and plotted in blue, brings the maximum error down to only about 1% at 9 km. The equation describing the Bozeman model is

$$\begin{aligned} P &= P_s\left(\frac{T_s}{T_s-\Gamma h}\right)^{-\left(\frac{0.0326}{\Gamma}\right)} && ,\ 0-9\ \text{km} \\ &= P(9\ \text{km})\times e^{\frac{-0.0353(h-9000)}{T(9\ \text{km})}} && ,\ 9-20\ \text{km} \end{aligned} \tag{2.32}$$

For time and cost efficiency, instead of launching radiosonde twice daily, models in equation (2.30) and (2.32) are used to predict the atmospheric temperature and pressure based only on $T_s, P_s,$ and $\Gamma \approx 7.7$ K/km.

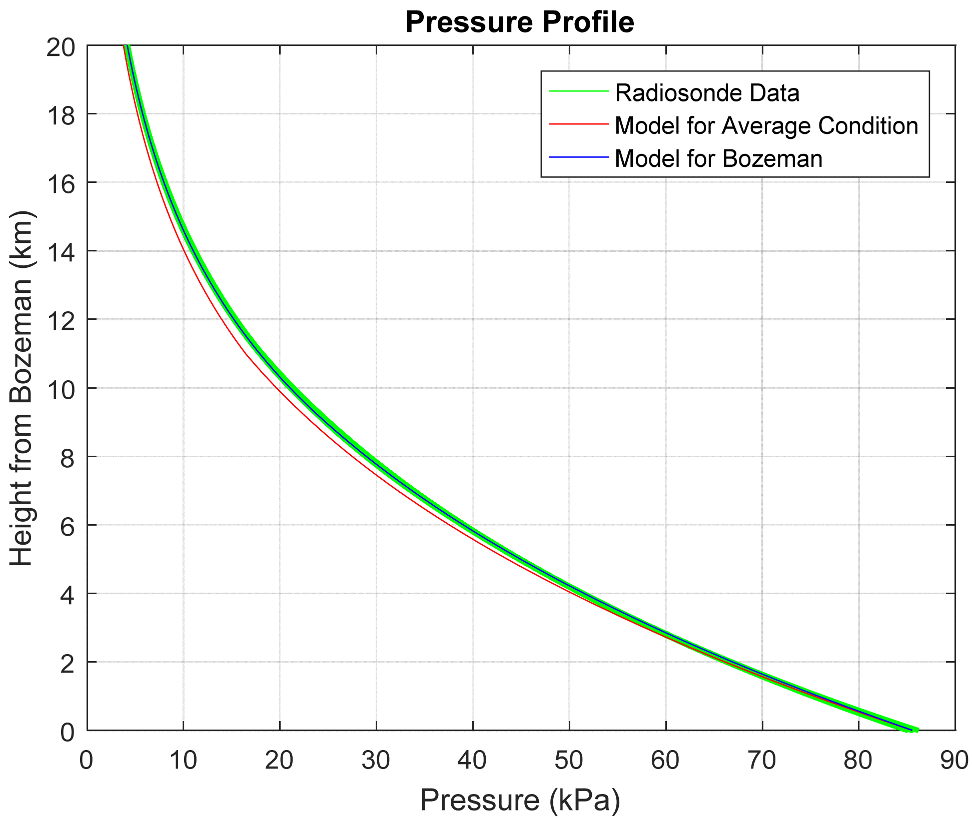


Figure 2.23 Pressure profiles from radiosonde launched at Bozeman (N 45°39’43”, W 111°02’42”), from November 23-26, 2013, twice daily (green) and pressure model for average condition (red) for Bozeman (blue).

Molecular Back Scatter and Extinction Coefficient

The molecular back scatter coefficient, $\beta_m(\lambda, r)$, which is also known as Rayleigh back scatter can be modeled as [19]

$$\beta_m(\lambda, r)[\mathrm{m}\times\mathrm{steradian}]^{-1} = 374280\left(\frac{P(r)[\mathrm{kPa}]}{T(r)[\mathrm{K}]}\right)\left(\frac{1}{\lambda^4[\mathrm{nm}]}\right) \qquad (2.33)$$

where $T(r)$ and $P(r)$ are atmospheric temperature and pressure which can be predicted using the models in equation (2.30) and (2.32), and λ is the operating wavelength of a lidar which is 532 nm for the MPL.

After $\beta_m(\lambda,r)$ is determined, the molecular extinction coefficient, $\sigma_m(\lambda,r)$, can easily be calculated from the definition of the molecular lidar ratio, S_m, which is defined by [18]

$$S_m \equiv \frac{\sigma_m(\lambda,r)}{\beta_m(\lambda,r)} \tag{2.34}$$

where from the scattering theory [24], $S_m \approx \frac{8\pi}{3}$, therefore $\sigma_m(\lambda,r)$ can be modeled as

$$\sigma_m(\lambda,r) = \frac{8\pi}{3}\beta_m(\lambda,r) \tag{2.35}$$

An example of molecular back scatter and extinction profile calculated based on the data taken on June 30, 2015 is shown in Figure 2.24.

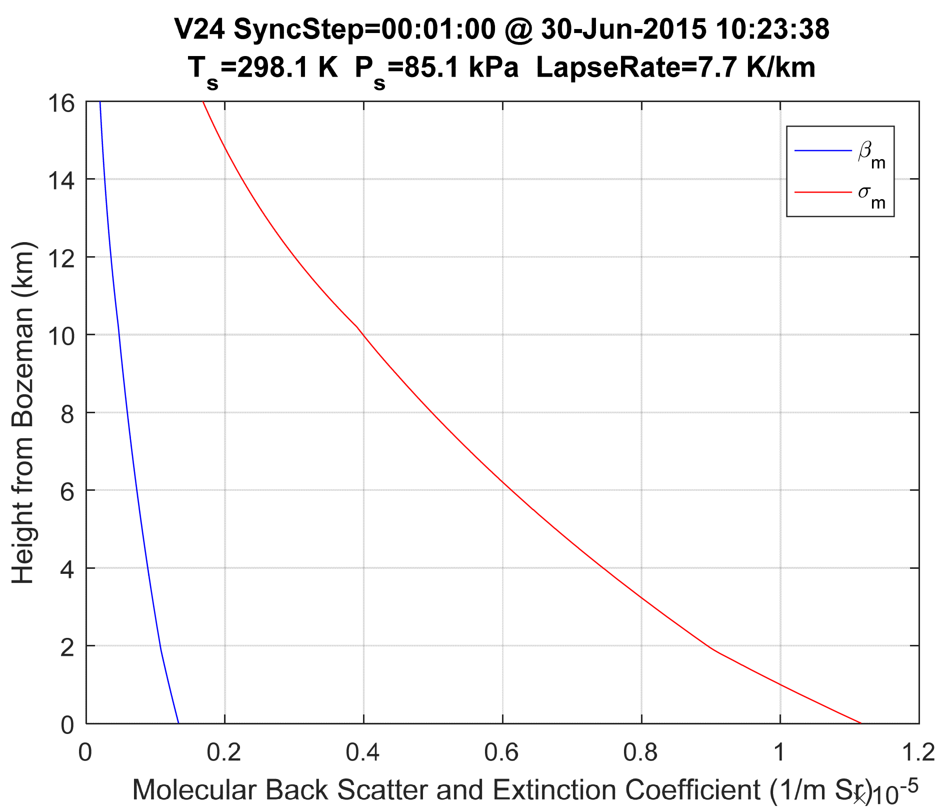


Figure 2.24 Molecular back scatter and extinction coefficient.

Calibration Constant

The calibration constant can be retrieved by first expressing the lidar equation, equation (2.12) as

$$P(\lambda,r)=P_t(\lambda)\frac{C_\lambda(r)}{r^2}\left[\beta_m(\lambda,r)+\beta_a(\lambda,r)\right]T^2(\lambda,r) \tag{2.36}$$

with
$$C_\lambda(r)=A\Delta rO(\lambda,r)\varepsilon_R(\lambda)\varepsilon_D(\lambda) \tag{2.37}$$

where $C_\lambda(r)$ is called the calibration constant. At an altitude above 5 km, $\beta_a(\lambda,r)$ typically approaches zero, so then equation (2.36) can be written as

$$C_\lambda(r)=\frac{P(\lambda,r)\cdot r^2}{P_t(\lambda)\beta_m(\lambda,r)e^{-2\cdot AOD}e^{-2\int_0^r\sigma_m(\lambda,r')dr'}} \tag{2.38}$$

With the modification where the power ratio is replaced with the number of photon ratio, $\frac{P(\lambda,r)}{P_t(\lambda)}=\frac{N(\lambda,r)}{N_t(\lambda)}$, and the range corrected returns, $L(\lambda,r)=N(\lambda,r)\cdot r^2$ are used in the expression, we get

$$C_\lambda(r)=\frac{L(\lambda,r)}{N_t(\lambda)\beta_m(\lambda,r)e^{-2\cdot AOD}e^{-2\int_0^r\sigma_m(\lambda,r')dr'}} \tag{2.39}$$

where AOD is the total thickness of AOD acquired from AERONET and $N_t(\lambda)$ is the number of the outgoing photons from the MPL transmitter which can be calculated as

$$N_t(\lambda)=\frac{E_{pulse}\times\text{Number Accumulates}}{E_{photon}} \tag{2.40}$$

where E_{pulse} is the energy per pulse of an outgoing laser. It is estimated to be 3 μJ which is slightly less than the laser spec sheet value, which is acceptable because the transmittion of the optical system is taken into account.

The value of the number accumulates can be set in the Labview program used for MPL control. More details of the program are discussed in Chapter 4. In the program, it is set to 5000, meaning that the program counts and adds up the number of returned photons from 5000 pulses of an outgoing laser.

The energy of a photon is given by

$$E_{photon} = \frac{hc}{\lambda} = \frac{\left(6.63\times10^{-34}\ \text{m}^2\cdot\text{kg/s}\right)\left(3\times10^{8}\ \text{m/s}\right)}{\left(532\times10^{-9}\ \text{m}\right)} = 3.74\times10^{-19}\ \text{J} \tag{2.41}$$

From this information, equation (2.40) estimates that $N_t\left(\lambda\right) = 4.01\times10^{16}$ photons.

Figure 2.25 shows the retrieved calibration constant as a function of altitude as described by equation (2.39) from the data taken on June 30, 2015. Note that the calibration constant becomes constant at the altitude above the aerosol boundary layer which is about 3.2 km in the case of the plot. This constant value of the calibration constant, C_λ, can be determined by averaging $C_\lambda\left(r\right)$ above the boundary layer. It will then be used in further steps of the aerosol inversion calculation.

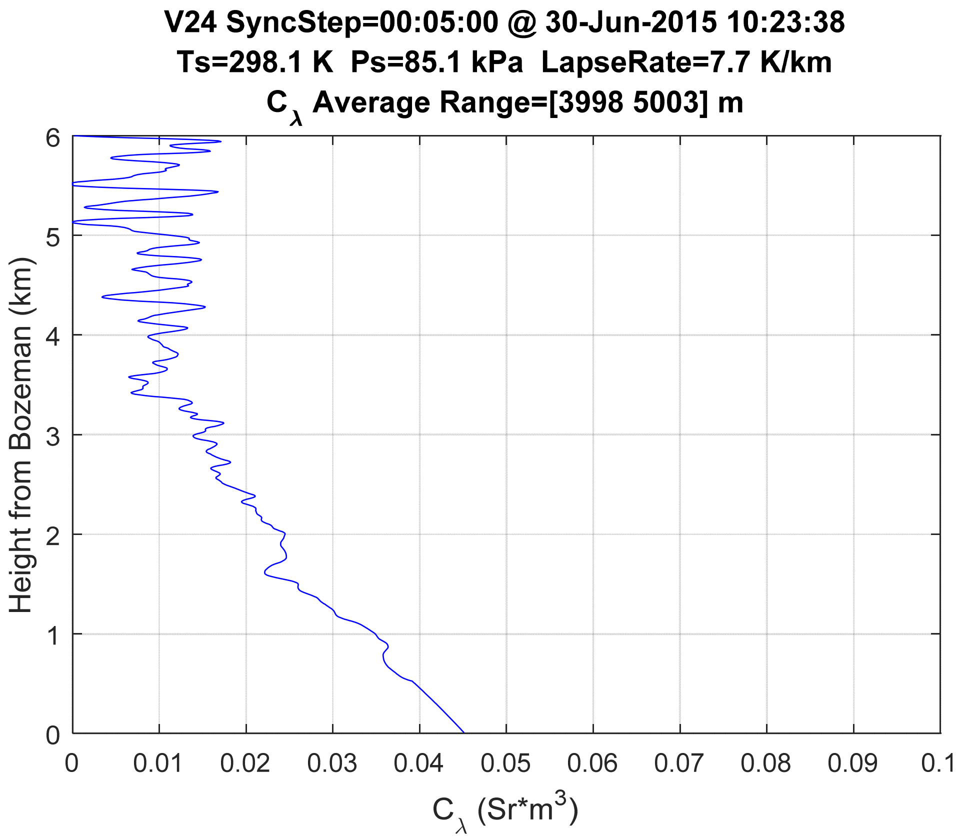


Figure 2.25 Calibration constant as a function of altitude.

<u>Inversion</u>

Once C_λ above the boundary layer is determined, the inversion begins with the calculation of the aerosol back scatter coefficient, $\beta_a(\lambda, r)$. To solve for $\beta_a(\lambda, r)$, we go back to the expression in equation (2.36), but with N substituting for P

$$N(\lambda, r) = N_t(\lambda)\frac{C_\lambda(r)}{r^2}\left[\beta_m(\lambda, r) + \beta_a(\lambda, r)\right]T^2(\lambda, r) \tag{2.42}$$

in which the total round trip atmospheric transmission can be expressed as

$$T^2(\lambda, r) = T_a^2(\lambda, r)T_m^2(\lambda, r) = e^{-2\int_0^r \sigma_a(\lambda, r')dr'}e^{-2\int_0^r \sigma_m(\lambda, r')dr'} \tag{2.43}$$

Because there are two unknowns in the lidar equation, $\beta_a(\lambda, r)$ and $\sigma_a(\lambda, r)$, it requires the assumption of a constant aerosol lidar ratio [18]

$$S_a(\lambda) = \frac{\sigma_a(\lambda, r)}{\beta_a(\lambda, r)} = \text{constant} \tag{2.44}$$

Then $T_a^2(\lambda, r)$ can be written in terms of $\beta_a(\lambda, r)$ and $S_a(\lambda)$ as

$$T_a^2(\lambda, r) = e^{-2S_a(\lambda)\int_0^r \beta_a(\lambda, r')dr'} \tag{2.45}$$

Differentiate both sides with respect to r and rearrange the equation, get

$$\beta_a(\lambda, r) = -\frac{1}{2S_a(\lambda)T_a^2(\lambda, r)} \frac{dT_a^2(\lambda, r)}{dr} \tag{2.46}$$

Next we substitute equation (2.43) and (2.46) back into equation (2.42) as well as the relationship in equation (2.19) and with a little rearranging, yield

$$\frac{dT_a^2(\lambda, r)}{dr} - 2S_a(\lambda)\beta_m T_a^2(\lambda, r) = -\frac{2S_a(\lambda)L(\lambda, r)}{N_t(\lambda)C_\lambda T_m^2(\lambda, r)} \tag{2.47}$$

This equation is a linear first order differential equation [25] which yields the solution

$$T_a^2(\lambda, r) = e^{2S_a(\lambda)\int_0^r \beta_m(\lambda, r')dr'} \left[-\frac{2S_a(\lambda)}{N_t(\lambda)C_\lambda} \int_0^r \frac{L(\lambda, r')}{T_m^2(\lambda, r')} e^{-2S_a(\lambda)\int_0^{r'} \beta_m(\lambda, r'')dr''} dr' \right] \tag{2.48}$$

Substitute this result back into equation (2.42) and solve for $\beta_a(\lambda, r)$, yield

$$\beta_a(\lambda, r) = \frac{L(\lambda, r)e^{2[S_m(\lambda) - S_a(\lambda)]\int_0^r \beta_m(\lambda, r')dr'}}{N_t(\lambda)C_\lambda - 2S_a(\lambda)\int_0^r \frac{L(\lambda, r')}{T_m^2(\lambda, r')} e^{-2S_a(\lambda)\int_0^{r'} \beta_m(\lambda, r'')dr''} dr'} - \beta_m(\lambda, r) \tag{2.49}$$

with the first term on the right hand side being $\beta_{total}(\lambda, r)$.

In practice, numerical integration is required when solving for $\beta_{total}(\lambda, r)$. To put this into perspective, when performing the inversion, $\beta_{total}(\lambda, r)$ and $\beta_m(\lambda, r)$ are usually plotted together as shown in Figure 2.26. With the appropriate lidar ratio, their slopes above the boundary layer should match since there are only air molecules and no aerosol present above the boundary layer.

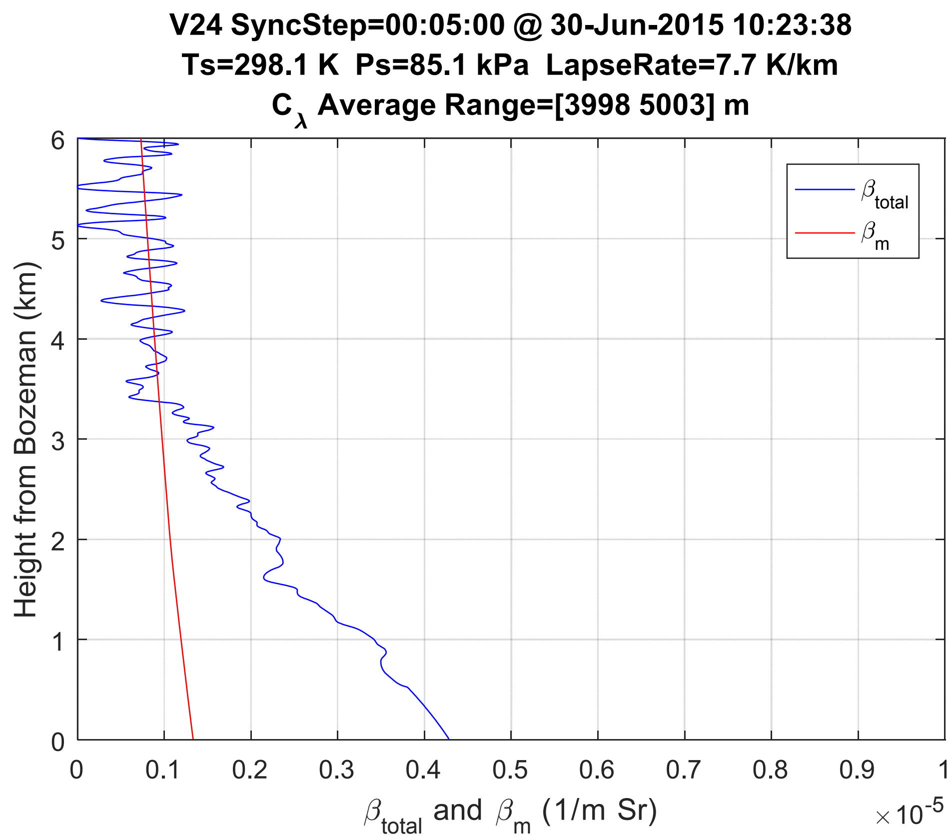


Figure 2.26 Total and molecular back scatter profile. With the right lidar ratio, their slopes above the boundary layer (~3.2 km) match.

$\beta_a(\lambda, r)$ can be determined by just subtracting $\beta_m(\lambda, r)$ from $\beta_{total}(\lambda, r)$ as shown in Figure 2.27.

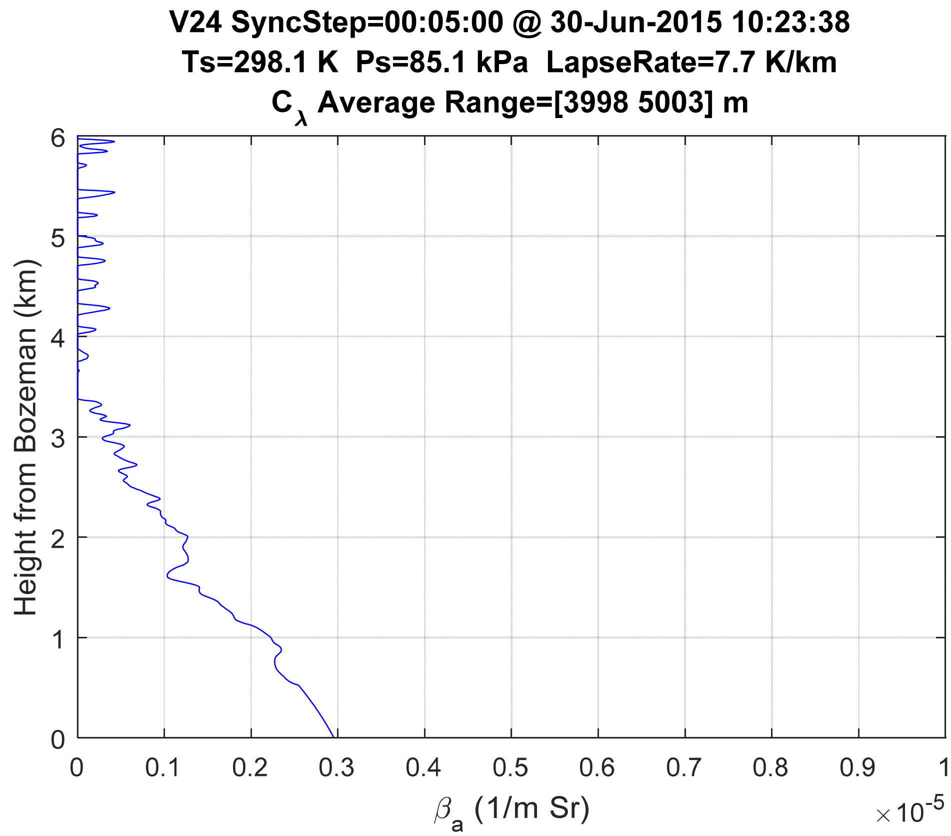


Figure 2.27 Aerosol back scatter profile results from the subtraction between the total and molecular back scatter profile in Figure 2.26.

After retrieving $\beta_a(\lambda,r)$, the lidar AOD can be calculated from

$$AOD_{Lidar} = S_a(\lambda)\int_0^r \beta_a(\lambda,r')dr' \tag{2.50}$$

To best match the lidar AOD to that from AERONET, the aerosol lidar ratio, $S_a(\lambda)$, value is scanned from 1 to 100 Sr to find the value that yields the minimum error in AOD as shown in Figure 2.4 flow chart where

$$\text{Percent Error} = \frac{\left|AOD_{AERONET} - AOD_{Lidar}\right|}{AOD_{AERONET}} \times 100\% \tag{2.51}$$

An example plot of the AOD error from $S_a(\lambda)$ scanning at a specific data time is shown in Figure 2.28. It indicates that the lidar AOD best agrees with the AERONET AOD of 0.141 at a minimum error of 0.116% when $S_a(\lambda) = 50.9$ Sr.

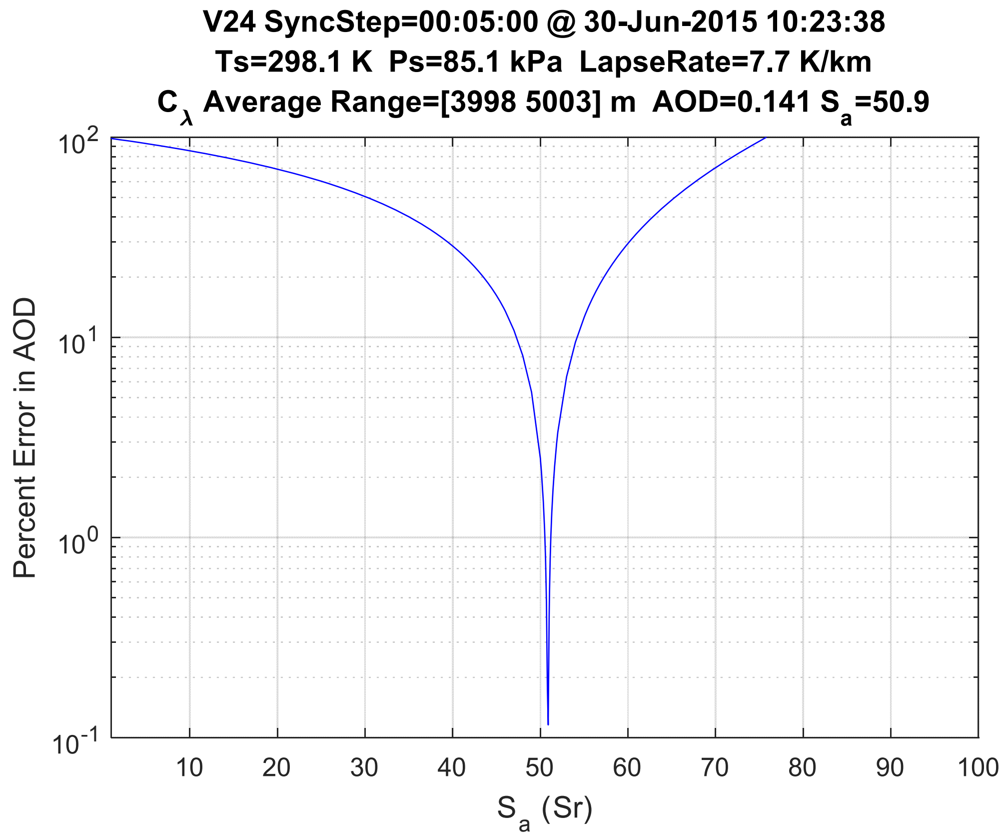


Figure 2.28 Percent error in lidar AOD as compare to AERONET AOD when scanning the value of S_a from 0 to 100 Sr.

Giving the appropriate value of S_a , the lidar AOD as a function of range can be numerically calculated as in equation (2.50). An example plot of the range dependent lidar AOD is shown in Figure 2.29.

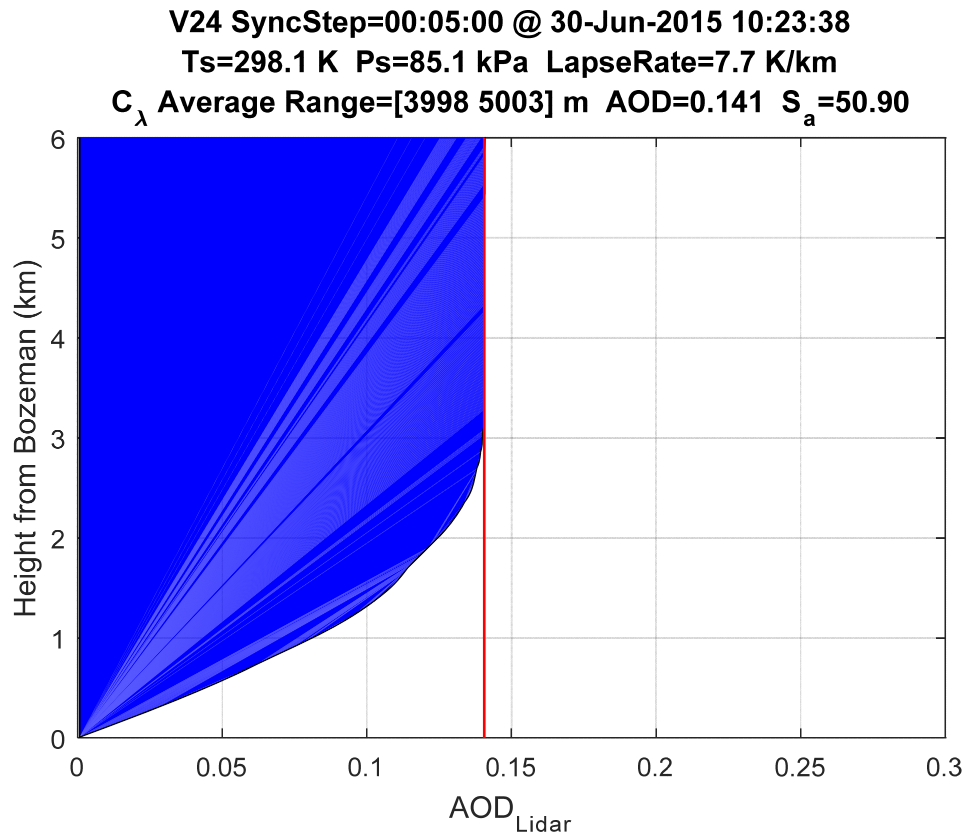


Figure 2.29 Plots show good agreement between the lidar AOD as a function of range (blue) and the total AERONET AOD (red) when $S_a = 50.9$ Sr.

Examples of lidar retrievals on June 30, 2015 including the range corrected return with the overlap applied, L/O, the total back scatter, β_{total}, molecular back scatter, β_m, and aerosol back scatter, β_a, as a function of range and time are plotted in Figure 2.30 to Figure 2.33, consecutively, while the AOD and S_a as a function of time are plotted in Figure 2.34 and Figure 2.35. Note that the offset plot near 11:00 is the result from APD oversaturation due to clouds presented at 6 km.

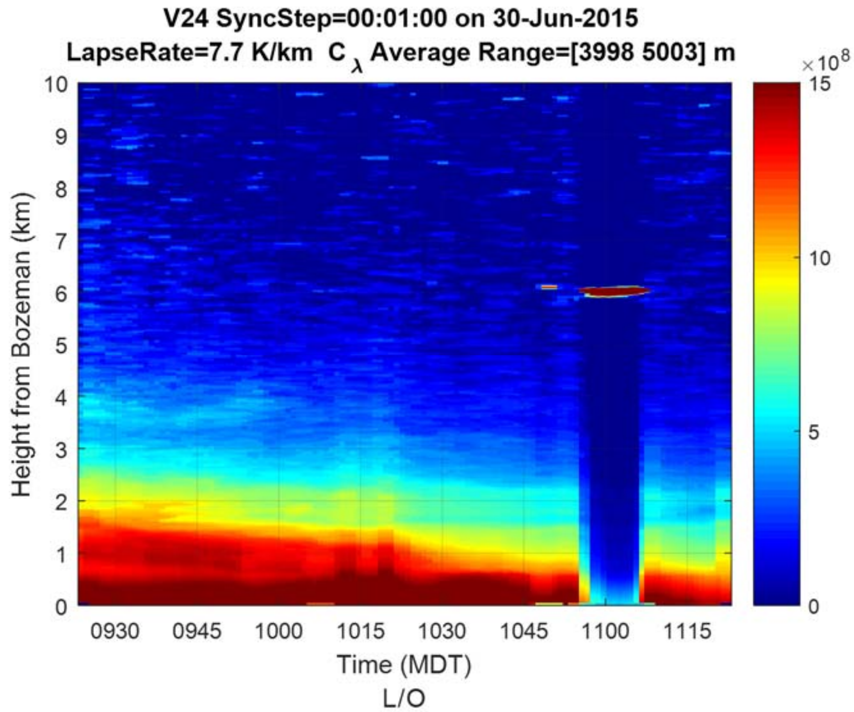


Figure 2.30 Range corrected returns with overlap applied over Bozeman on June 30, 2015 09:23:38-11:23:24.

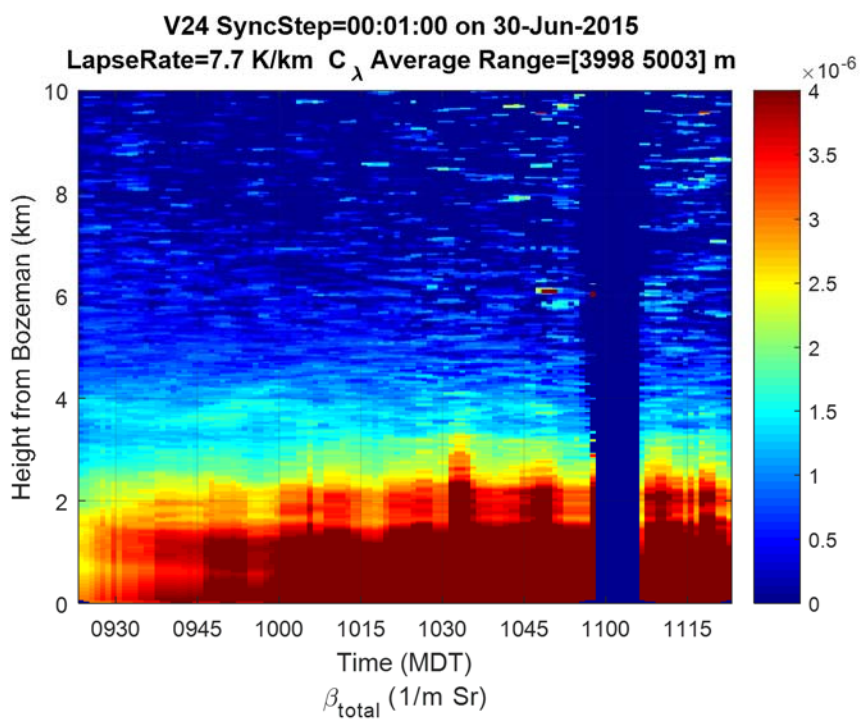


Figure 2.31 Total back scatter coefficient over Bozeman on June 30, 2015 09:23:38-11:23:24.

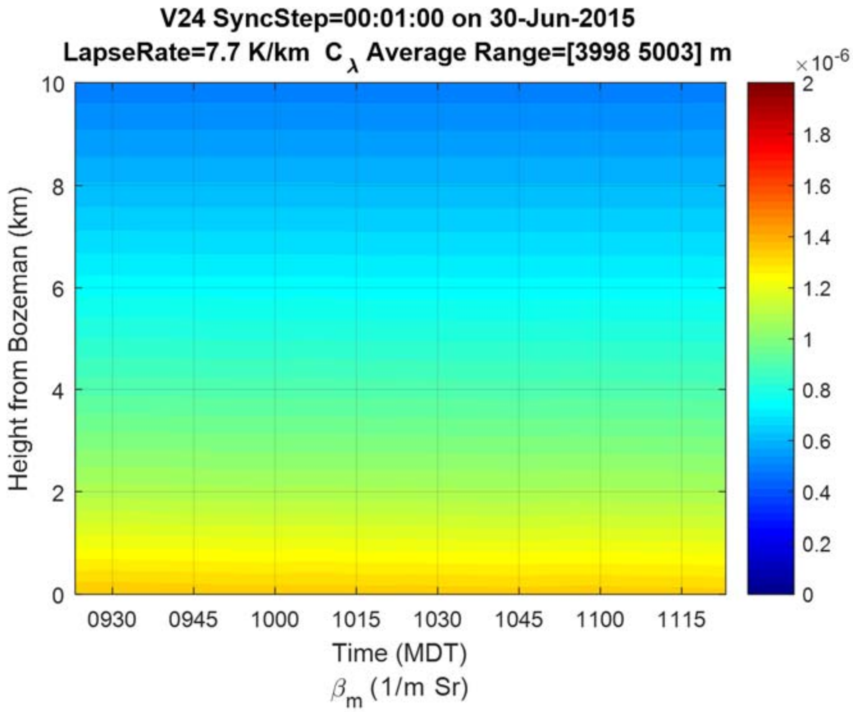


Figure 2.32 Molecular back scatter coefficient over Bozeman on June 30, 2015 09:23:38-11:23:24.

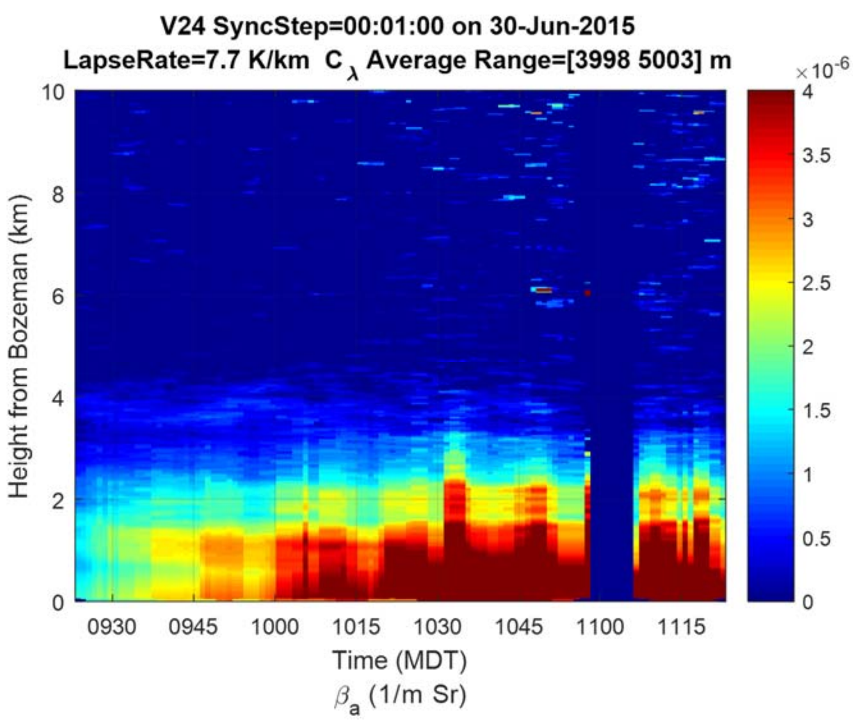


Figure 2.33 Aerosol back scatter coefficient over Bozeman on June 30, 2015 09:23:38-11:23:24.

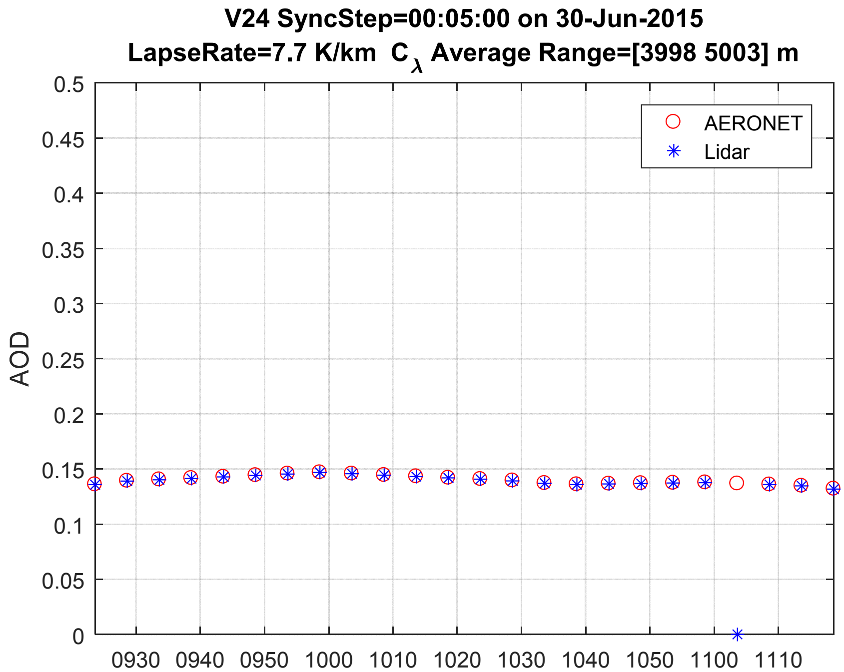


Figure 2.34 AERONET and lidar total AOD over Bozeman on June 30, 2015 09:23:38-11:23:24.

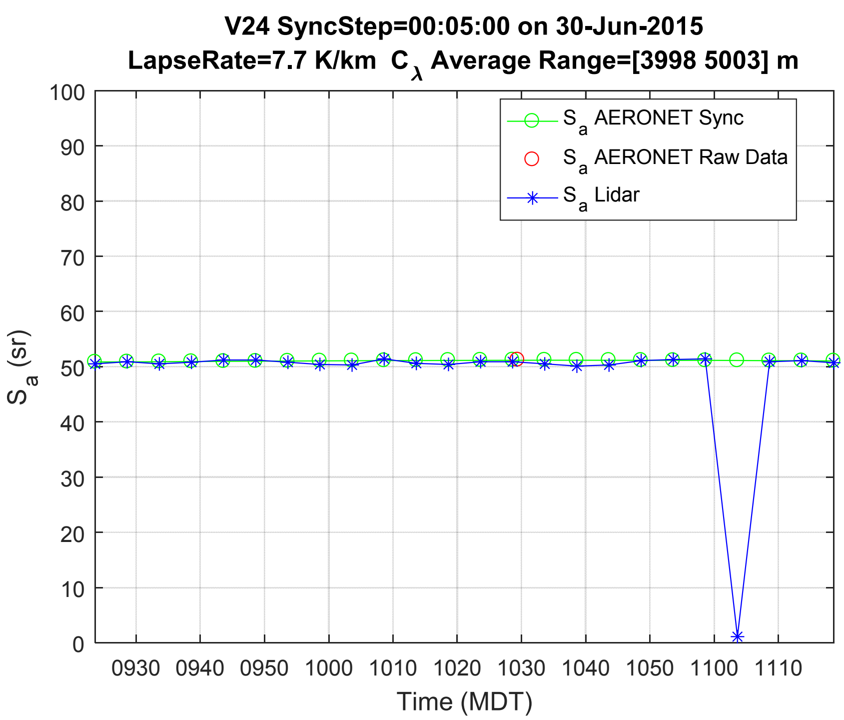


Figure 2.35 Aerosol lidar ratio over Bozeman on June 30, 2015 09:23:38-11:23:24. The S_a AERONET Raw Data is calculated from the SSA and PFN as in equation (2.27).

Principle of DIAL

DIAL is a special class of lidar that uses two closely-spaced laser wavelengths—one centered on the absorption line of the molecule of interest called the "online wavelength" and another located just off that same absorption line called the "offline wavelength" as shown in Figure 2.36 (left). With a small separation between the online and offline wavelengths, the scattering effect by molecules is considered equal. As a result, the difference in returns between the two wavelengths is then due entirely to absorption by the molecules. The plot in Figure 2.36 (right) demonstrates the difference between the online and offline returns from a DIAL system. Their ratio can be used to calculate the concentration profile of the molecule of interest.

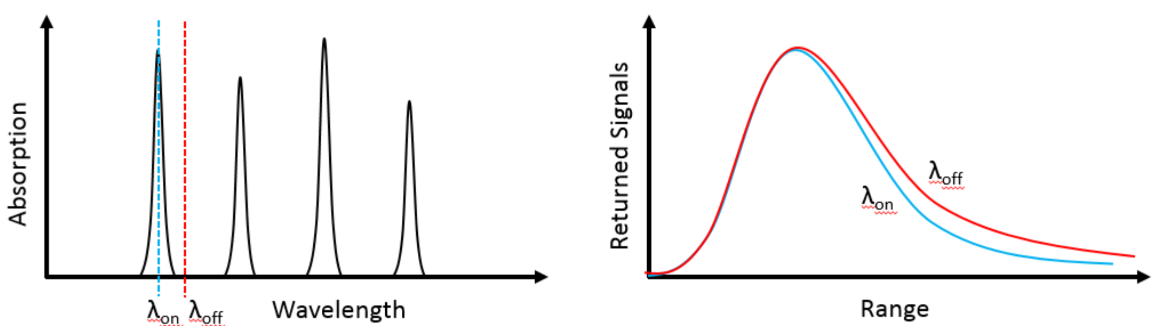


Figure 2.36 Locations of online and offline wavelength in absorption feature of a molecule (left). Online and offline returns from a DIAL (right).

DIAL Equation

The DIAL equation can be derived from the lidar equation, equation (2.12). First, the total (molecular and particulate) extinction coefficient, $\sigma(r)$, is expressed as the sum of the scattering and absorbing constituents [19].

$$\sigma_{\text{on}}(r) = \beta_{\text{on}}(r) + \kappa_{\text{on}} n(r) \tag{2.52}$$

and
$$\sigma_{\text{off}}(r) = \beta_{\text{off}}(r) + \kappa_{\text{off}} n(r) \tag{2.53}$$

where "on" denotes parameter at the online and "off' denotes parameter at the offline wavelength, β represents the total scattering coefficient, and κ represents the absorption cross sections of the particle of interest. n is the number of the absorbing molecules in a unit volume at range r. Then equation (2.12) can be expressed as

$$P_{\text{on}}(r) = C_{\text{on}} \frac{\beta_{\pi,\text{on}}(r)}{r^2} \exp\left[-2\int_{r_1}^{r} \left[\beta_{\text{on}}(r') + \kappa_{\text{on}} n(r') \right] dr' \right] \tag{2.54}$$

where
$$C_{\text{on}} = P_{t,\text{on}} A \Delta r \varepsilon_R \varepsilon_D T_{\text{on}}^2(0, r_1) \tag{2.55}$$

for the online wavelength. Here, the back scattering coefficient omits the subscript π. The range r_1 indicates the starting point of the examined path along which the gas concentration is measured, which is usually where the full overlap of the lidar begins. C is the calibration constant of a lidar system which includes the transmitted power, the area of the collecting mirror, lidar resolution, lidar efficiencies, and the transmission term outside of the examined path. It should be noted that the receiver and detector efficiency can be treated the same for both the online and offline wavelength since the two wavelengths are spatially close by. The term $T_{\text{on}}^2(0, r_1)$ is the two-way transmission over the range of incomplete overlap from 0 to r_1.

$$T_{\text{on}}^2(0, r) = \exp\left[-2\int_{0}^{r_1} \left[\beta_{\text{on}}(r') + \kappa_{\text{on}} n(r') \right] dr' \right] \tag{2.56}$$

Equation (2.54), (2.55), and (2.56) are the same for the case of the offline wavelength. The gas concentration $n(r)$ can be derived from the ratio of the returned signal $P_{\text{on}}(r)$ and $P_{\text{off}}(r)$

$$\frac{P_{\text{on}}(r)}{P_{\text{off}}(r)}=\frac{C_{\text{on}}}{C_{\text{off}}}\frac{\beta_{\pi,\text{on}}(r)}{\beta_{\pi,\text{off}}(r)}\exp\left[-2\int_{r_1}^{r}\left[\beta_{\text{on}}(r')-\beta_{\text{off}}(r')+(\kappa_{\text{on}}-\kappa_{\text{off}})n(r')\right]dr'\right] \quad (2.57)$$

The selection of online and offline wavelengths is important in DIAL measurements. To yield the most accurate results, the wavelengths must be selected such that the difference in scattering coefficients term, $\beta_{\text{on}}(r')-\beta_{\text{off}}(r')$, is small while the difference in the absorption cross sections term, $\kappa_{\text{on}}+\kappa_{\text{off}}$, is large, so that the exponential term in equation (2.57) is primarily related to differential absorption and not to differential scattering. If we take the natural log and differentiate both sides of equation (2.57) with respect to r, the absorbing gas concentration $n(r)$ can be solved as

$$n(r)=-\frac{1}{2\Delta\kappa}\frac{d}{dr}\left[\ln\frac{P_{\text{on}}(r)}{P_{\text{off}}(r)}\right]+\frac{1}{2\Delta\kappa}\frac{d}{dr}\left[\ln\frac{\beta_{\pi,\text{on}}(r)}{\beta_{\pi,\text{off}}(r)}\right]-\frac{\Delta\beta}{\Delta\kappa} \quad (2.58)$$

where $\Delta\kappa=\kappa_{\text{on}}-\kappa_{\text{off}}$ is the differential absorption cross section and $\Delta\beta=\beta_{\text{on}}-\beta_{\text{off}}$ is the differential scattering of the measured gas. In a general DIAL data analysis, the first term containing the logarithm of $P_{\text{on}}(r)/P_{\text{off}}(r)$ usually provides an initial estimate of the gas concentration while the second term containing the logarithm of $\beta_{\pi,\text{on}}(r)/\beta_{\pi,\text{off}}(r)$ and the third term containing $\Delta\beta/\Delta\kappa$ are considered as correction terms. For more accurate results, the second and third term must be determined in some way in the final analysis.

However, often both parameters cannot be accurately calculated, especially in the lower troposphere, because they depend largely on atmospheric conditions.

The most important result in DIAL inversion is the first term in equation (2.58) which provides the initial estimate of the absorbing gas concentration. If this term cannot be accurately obtained, the remaining corrections are useless. To solve for the first term, a numerical differentiation is generally executed. This can be used to derive the gas concentration by calculating the logarithmic differences of $P_{on}(r)/P_{off}(r)$ for range increments Δr that are relatively large compare with the lidar range resolution. By averaging discrete data points at range r and $r+\Delta r$ and neglecting the second and third term in equation (2.58), the absorbing gas concentration can be expressed in the simplest form as

$$n(r) = \frac{1}{2\Delta\kappa\Delta r}\left[\ln\frac{P_{on}(r)}{P_{on}(r+\Delta r)} - \ln\frac{P_{off}(r)}{P_{off}(r+\Delta r)}\right] \tag{2.59}$$

This equation is referred to as the DIAL equation. It should be noted that Δr in equation (2.59) refers to range increment and is different from Δr in equation (2.12) which refers to the lidar range resolution. In DIAL measurements, the range increment is usually tens or even hundreds of meters while the range resolution is much smaller.

CO_2 Line Selection for DIAL

The appropriate CO_2 absorption line for the DIAL is selected based on three key factors. First, the absorption-line strength must be strong enough to attenuate the online wavelength by a sufficient amount, yet weak enough that the online signal can be measured beyond 6 km. Second, the absorption cross-section must have as little influence from temperature changes as possible. And third, the absorption feature must be well isolated from other interfering absorption features [26]. In addition, the supporting technologies such as laser sources, optical components, and detectors must be considered as well when choosing an absorption feature. CO_2 has distinct absorption features near 1.6, 2.0, 2.8, and 4.4 μm, with the line strength of the bands increasing at the longer wavelengths. Therefore, it would make sense to select the longest wavelength line for the strongest line strength. However, the availability of indium gallium arsenide (InGaAs) avalanche photodiodes (APD) with high quantum efficiencies is limited to only approximately 1.7 μm. Furthermore, the lack of suitable commercially available detectors at longer wavelengths limits the choice of absorption features to near the 1.6 μm band shown in Figure 2.37 below.

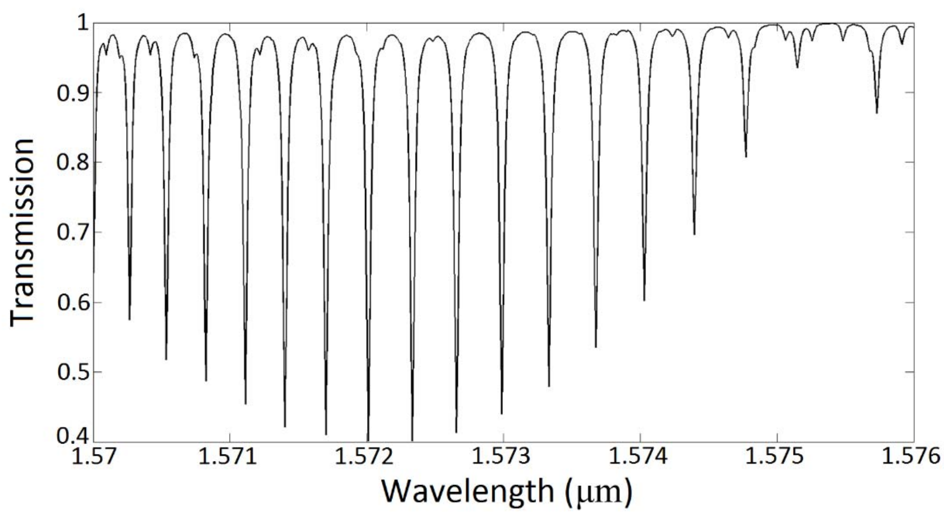


Figure 2.37 Plot of the atmospheric CO_2 concentration of 390 ppm as a function of wavelength for a 10 km path length, 296 K temperature, and 0.85 atm pressure.

To select a specific line near the 1.6 μm in Figure 2.37, the consideration for each individual line absorption is broken into two parts—shape and strength. The line strength is mainly subject to only temperature change while the line shape is subject to both temperature and pressure changes which cause natural broadening effects and shifting. Two primary types of line broadening in the atmosphere are Doppler broadening which tends to dominate the line shape at low pressures, < 0.01 atm, and pressure broadening which tends to dominate the line shape at high pressures, > 0.1 atm [26].

Doppler-Broadened Line Shape

Doppler broadening generates a Gaussian line shape which depends on atmospheric temperature. This dominates in altitudes above approximately 50 km and is a result of the variety of velocities and directions of the atoms and molecules interacting with the light. Doppler frequency shifts from the distribution of molecular velocities, causes absorption line broadening with a Doppler wavenumber shift of $\nu' = \nu\left(1 - \dfrac{v}{c}\right)$. The Maxwell-Boltzmann distribution of velocities along x is

$$p(v_x) = \frac{1}{\sqrt{\pi} v_0} e^{-\left(\frac{v_x}{v_0}\right)^2} \tag{2.60}$$

where $v_0 = \sqrt{2k_B T / m}$ and m is the mass of molecule. This results in a Gaussian line shape for Doppler broadening [27]

$$f_D(\nu - \nu_0) = \frac{1}{\alpha_D \sqrt{\pi}} \exp\left[-\frac{(\nu - \nu_0)^2}{\alpha_D^2} \right] \tag{2.61}$$

where
$$\alpha_D = \nu_0 \sqrt{\frac{2k_B T}{mc^2}} \tag{2.62}$$

which yield the half-width at half-maximum (HWHM), $\gamma_D = \alpha_D \sqrt{\ln 2}$ as
$$\gamma_D = \nu_0 \sqrt{\frac{1.3863 k_B T}{mc^2}} \tag{2.63}$$

where ν_0 is the center frequency of the absorption feature and the mass, m, in this case refers to the mass of the CO_2 molecule.

Pressure-Broadened Line Shape

Pressure broadening generates a Lorentzian line shape which depends on both atmospheric pressure and temperature. This dominates in altitudes below approximately 2 km and is the result of collisions among the molecules that are interacting with the light. The Lorentzian line shape is described by [28]

$$f_L(\nu - \nu_0) = \frac{\gamma_L}{\pi\left[(\nu - \nu_0)^2 + \gamma_L^2\right]} \tag{2.64}$$

The Lorentzian HWHM, γ_L, evaluated at pressure P and temperature T is described by [26, 28, 29]

$$\gamma_L = \gamma_0 \frac{P}{P_0}\left(\frac{T_0}{T}\right)^n \tag{2.65}$$

where γ_0 is the HWHM at the reference pressure P_0 and reference temperature T_0, and n is the linewidth temperature dependence parameter which is determined empirically. All of these reference variables are provided by the Hitran database [30].

Voigt Profile

When the pressure broadened Lorentz half-width becomes comparable to the Doppler half-width, the broadening effects must be convolved to get what is called the Voigt line shape. This is very useful in the calculation of the final line shape because it accounts for both primary line broadening mechanisms found in the atmosphere [31]. The Voigt profile as a function of wavenumber can be described as

$$g\left(\nu-\nu_0\right)=\frac{\gamma_L}{\gamma_D^2\pi^{3/2}}\int_{-\infty}^{+\infty}\frac{e^{-t^2}}{\left(\frac{\gamma_L}{\gamma_D}\right)^2+\left(\frac{\nu-\nu_0}{\gamma_D}-t\right)^2}dt \tag{2.66}$$

There is no simple analytical function for the Voigt profile, so various approximations are used. As shown in Figure 2.38, the Voigt profile has a Lorentz shape in the line wings and it has a Doppler behavior at the line center.

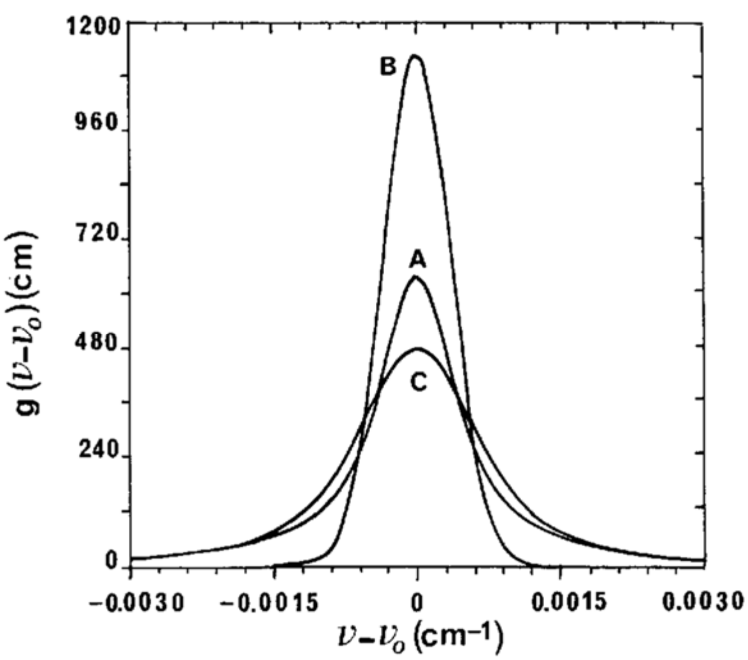


Figure 2.38 Line profiles: A is Lorentz, B is Doppler, and C is Voigt profile resulting from convolution of A and B. [32]

Temperature Sensitivity Line-Strength

The temperature dependent line strength or line intensity, $S(T)$, with the availability of the line strength at reference temperature, T_0, from a spectroscopic database, can be described as [33]

$$S(T)=S(T_0)\frac{Q_T(T_0)}{Q_T(T)}\left[\frac{1-\exp(-hc\nu_0/k_BT)}{1-\exp(-hc\nu_0/k_BT_0)}\right]\exp\left[\frac{hcE_L}{k_B}\left(\frac{1}{T_0}-\frac{1}{T}\right)\right] \tag{2.67}$$

where $S(T_0)$ is the reference line strength for an absorption feature found at the center wavenumber, ν_0, and at a temperature, T_0. $Q_T(T)$ and $Q_T(T_0)$ are the total internal partition function at temperature T and T_0 repectively. E_L is the lower of the two energy states involved in the absorbing process. Values for these mentioned variables can be found from the Hitran database [30]. T is the temperature at which the line strength is being calculated, and the remaining constants, h, c, k_B, are Planck's constant, the speed of light, and the Boltzmann constant, respectively.

Absorption Cross-Section

The molecular absorption cross section, σ (or κ used in the DIAL equation section), can be found from the convolution integral between the line strength, $S(T)$, and the normalized line shape, $g(\nu-\nu_0)$, based on the Voigt profile as [34]

$$\sigma = S * g \tag{2.68}$$

which results in [29]

$$\sigma(\nu)=\frac{ky}{\pi}\int_{-\infty}^{+\infty}\frac{e^{-t^2}}{y^2+(x-t)^2}dt \tag{2.69}$$

where $x=0.3466\frac{\nu-\nu_0}{\gamma_D}$, $y=0.3466\frac{\gamma_L}{\gamma_D}$, $k=0.4697\frac{S(T)}{\gamma_D}$, ν is the frequency, ν_0 is the center frequency of the absorption feature, $S(T)$ is the absorption line strength, γ_D is the Doppler broadened linewidth as described in equation (2.63), and γ_L is the pressure broadened linewidth as described in equation (2.65). Using the information about the CO_2 and water vapor lines between the wavelengths 1.571 and 1.572 µm in the Hitran database, the absorption cross section as a function of wavelength calculated at temperature 296 K and pressure 0.85 atm is plotted in Figure 2.39 below.

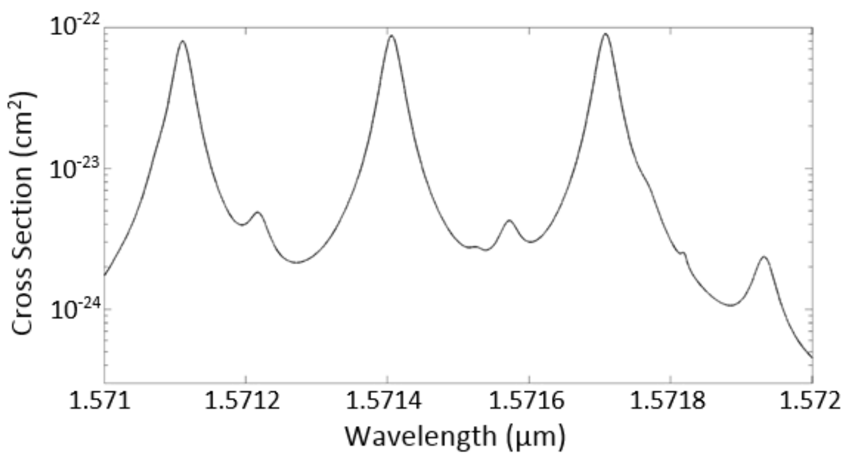


Figure 2.39 Absorption cross section of CO_2 as a function of wavelength at 296 K temperature and 0.85 atm pressure.

Based on CO_2 absorption features, the availability of commercial optics, and the cross section calculation as well as its temperature dependence analysis [35], the on-line and off-line wavelengths chosen for the CO_2 DIAL are 1.5714060 and 1.5712585 μm respectively.

Laser Transmitter Needed for CO_2 DIAL

Based on the CO_2 line selection, the most suitable wavelength for the DIAL laser transmitter online and offline wavelengths are 1571.406 and 1571.2585 nm, where CO_2 has strong absorption features that do not overlap with other common absorption lines. While these absorption features are already narrow, on the order of 3 MHz at their full-width half-maximum (FWHM), the DIAL data processing requires that the transmitter output laser must be even narrower, ideally on the order of MHz. From modeling to be discussed in Chapter 7, the laser transmitter must also have a high repetition rate on the order of 1 kHz for fast integration times. In addition, to meet the accuracy of below 2% error needed for carbon cycle studies as discussed in Chapter 7, the laser transmitter needs a high output pulse energy of about 3 mJ. Unfortunately, a laser transmitter with these requirements is not commercially available. The alternative is to employ the nonlinear optics with commercially available components to develop the laser source that meets these specifications. As mentioned earlier in the Dissertation Organization section, the theory of nonlinear processes is discussed in the next chapter, Chapter 3.

THEORY II

Introduction to Nonlinear Optics

Nonlinear optics is a branch of optics that describes the phenomena of light in nonlinear media in which its polarization responds nonlinearly to the electric field of light. In contrast with linear optics where light is generally weak and is deflected or delayed without frequency change, nonlinear optics involves sufficiently intense light that can modify the optical properties of a medium and induce a frequency change. This allows one to develop a new laser source by converting one laser frequency to another that suits one's application. In nature, only a small subset of materials has the properties that allow nonlinear interactions to occur.

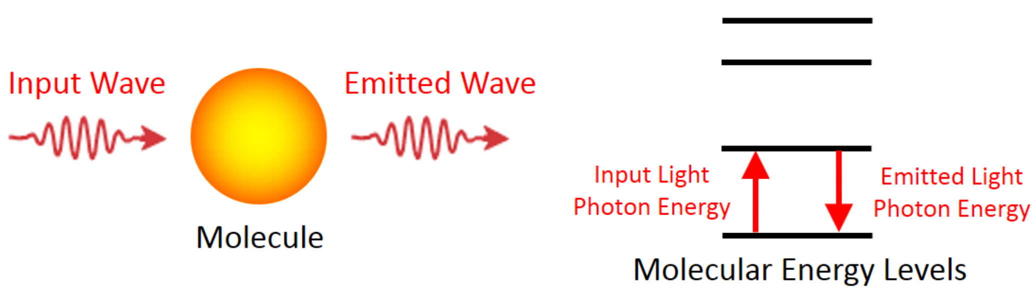


Figure 3.1 In linear optics, a light wave acting on a molecule causes it to vibrate and then emits a wave with the same frequency.

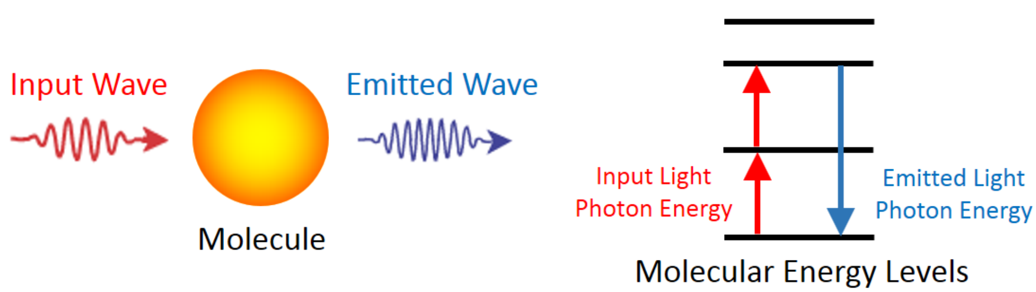


Figure 3.2 In nonlinear optics, a more intense light wave sends electrons to a higher energy state and thus emits a wave with different frequency.

Parametric and Nonparametric Process

Nonlinear processes fall into two main categories—parametric and nonparametric. The parametric process is one in which the final quantum state of the system remains unchanged from the initial quantum state. In other words, a parametric population can be removed from the ground state only for a brief period of time when it resides in the virtual state. Conversely, the nonparametric process involves the transfer of population from a real state to another. Mathematically, a parametric process is described by a real susceptibility while a nonparametric process is described by a complex one. Another difference is that photon energy must be conserved in a parametric process and need not be conserved in the nonparametric process because energy can be transferred to or from the media. Some examples of the parametric process are Second-Harmonic Generation (SHG), Third-Harmonic Generation (THG), High-Harmonic Generation (HHG), Sum-Frequency Generation (SFG), Difference-Frequency Generation (DFG), Optical Parametric Generation (OPG), Optical Parametric Amplification (OPA), Optical Parametric Oscillation (OPO), and Optical Rectification (OR). Some examples of the nonparametric process are saturable absorption, two-photon absorption, and stimulated Raman scattering. [36]

Mathematical Expression of Nonlinear Polarization

To help understand optical nonlinearity, we begin by mathematically expressing the dipole moment per unit volume or polarization $\vec{P}(\vec{r},t)$ of a material as [36, 37]

$$\begin{aligned} \vec{P}(\vec{r},t) &= \varepsilon_0\left[\chi^{(1)}\vec{E}(\vec{r},t)+\chi^{(2)}\vec{E}^2(\vec{r},t)+\chi^{(3)}\vec{E}^3(\vec{r},t)+\textit{Higher Order Terms}\right] \\ &= \vec{P}^{(1)}(\vec{r},t)+\vec{P}^{(2)}(\vec{r},t)+\vec{P}^{(3)}(\vec{r},t)+\ldots \end{aligned} \tag{3.1}$$

where $\vec{E}(\vec{r},t)$ is the applied electric field, $\chi^{(1)}$ is linear susceptibility, and $\chi^{(2)}$ and $\chi^{(3)}$ are second- and third-order nonlinear susceptibility. Likewise, $\vec{P}^{(1)}(\vec{r},t)=\varepsilon_0\chi^{(1)}\vec{E}(\vec{r},t)$ is referred as linear polarization. The terms $\vec{P}^{(2)}(\vec{r},t)=\varepsilon_0\chi^{(2)}\vec{E}^2(\vec{r},t)$ and $\vec{P}^{(3)}(\vec{r},t)=\varepsilon_0\chi^{(3)}\vec{E}^3(\vec{r},t)$ are referred as second- and third-order nonlinear polarization. It should be noted that the electric susceptibility χ, which is a property of the material, is a proportionality constant that indicates the degree of polarization of a material in response to an applied electric field. The greater the electric susceptibility, the greater the ability of a material to polarize in response to the field, and thereby reduce the total electric field inside the material. While the susceptibility can be complex, only real susceptibility is discussed in this dissertation since it involves only parametric processes. To give an idea, most nonlinear materials have $\chi^{(1)}$, $\chi^{(2)}$, and $\chi^{(3)}$ on the order of 1, 10^{-12} m/V, and 10^{-24} m^2/V^2 respectively.

Second-Harmonic Generation

Following the previous section, let us next consider the simplest case where a plane wave of frequency ω is incident upon a nonlinear material in which the second-order susceptibility $\chi^{(2)}$ is nonzero. Here the magnitude of the electric field of an incident plane wave traveling in the *z*-direction can be described as

$$E(z,t)=E(z)e^{-i\omega t}+c.c. \tag{3.2}$$

where $E(z)$ is the field's complex amplitude and $c.c.$ is the complex conjugate term.

The second-order nonlinear polarization created in a material according to equation (3.1) is

$$P^{(2)}(z,t) = 2\varepsilon_0 \chi^{(2)} E(z) E^*(z) + \left[\varepsilon_0 \chi^{(2)} E^2(z) e^{-2\omega t} + c.c. \right] \tag{3.3}$$

As described in equation (3.3), the second-order polarization consists of a contribution at zero frequency (the first term) and a contribution at frequency 2ω (the second term). This time varying polarization of the material in response to an applied electric field can act as a new source that generates a new electromagnetic field according to the wave equation in nonlinear material as described below

$$\nabla^2 \vec{E} = \mu_0 \sigma \frac{\partial \vec{E}}{\partial t} + \mu_0 \varepsilon \frac{\partial^2 \vec{E}}{\partial t^2} + \mu_0 \frac{\partial^2 \vec{P}^{(NL)}}{\partial t^2} \tag{3.4}$$

which is to be derived later in this chapter. Here $\vec{E}$ is the generated electric field, μ_0 is permeability of free space, σ is material's conductivity, ε is permittivity, and $\vec{P}^{(NL)}$ is the nonlinear polarization that includes any order.

Based on equation (3.4), the first term in equation (3.3) does not lead to the generation of a new field because its second time derivative vanishes. The process resulting from this first term is known as Optical Rectification (OR) in which a static field is created within the nonlinear material. The second term in equation (3.3) leads to the generation of a new field at frequency 2ω . For this reason, the process resulting from this second term is known as Second-Harmonic Generation (SHG).

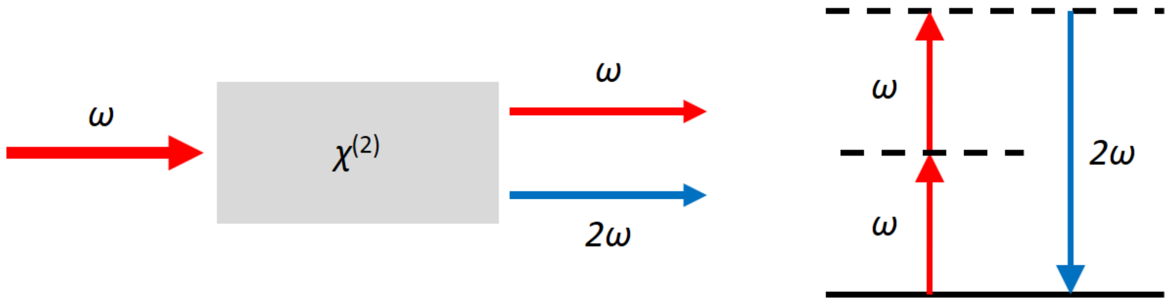


Figure 3.3 Geometry of SHG (left). Energy level diagram describing SHG (right).

Second-Order Frequency Mixing Processes

In the previous section, we explored the case of a monochromatic plane wave incident upon a nonlinear material with the susceptibility $\chi^{(2)}$. In this section, we consider the case of an incident plane wave traveling in the *z*-direction that consists of two distinct frequency components, ω_1 and ω_2. The magnitude of the electric field of this plane wave is described by

$$E(z,t) = E_1(z)e^{-i\omega_1 t} + E_2(z)e^{-i\omega_2 t} + c.c. \tag{3.5}$$

Substituting this into the second-order polarization $\vec{P}^{(2)}(\vec{r},t) = \varepsilon_0 \chi^{(2)} \vec{E}^2(\vec{r},t)$ yields

$$\begin{aligned} P^{(2)}(t) = \varepsilon_0 \chi^{(2)} \Big[& E_1^2 e^{-2i\omega_1 t} + E_2^2 e^{-2i\omega_2 t} + 2E_1E_2 e^{-i(\omega_1+\omega_2)t} + 2E_1E_2^* e^{-i(\omega_1-\omega_2)t} + c.c. \Big] \\ & + 2\varepsilon_0 \chi^{(2)} \left[E_1E_1^* + E_2E_2^* \right] \end{aligned} \tag{3.6}$$

where, for simplicity, the *z*-dependence is omitted from the expression. Equation (3.6) shows that the second-order nonlinear polarization results in multiple frequencies; some were not even present in the original plane wave. They can be written separately for each frequency as

$$\begin{aligned} &P(2\omega_1) = \varepsilon_0 \chi^{(2)} E_1^2 \quad \text{(SHG)} \\ &P(2\omega_2) = \varepsilon_0 \chi^{(2)} E_2^2 \quad \text{(SHG)} \\ &P(\omega_1 + \omega_2) = 2\varepsilon_0 \chi^{(2)} E_1 E_2 \quad \text{(SFG)} \\ &P(\omega_1 - \omega_2) = 2\varepsilon_0 \chi^{(2)} E_1 E_2^* \quad \text{(DFG)} \\ &P(0) = 2\varepsilon_0 \chi^{(2)} \left(E_1 E_1^* + E_2 E_2^* \right) \quad \text{(OR)} \end{aligned} \tag{3.7}$$

where the $c.c.$ part for negative frequencies need not be written explicitly. Here each expression is labeled by name of the physical process—Second-Harmonic Generation (SHG), Sum-Frequency Generation (SFG), Difference-Frequency Generation (DFG), and Optical Rectification (OR). One might expect that all of these processes can occur at the same time, but in reality only one process can occur at a time because the efficiency of the nonlinear interaction depends on the phase-matching condition which permits only one frequency to occur.

Sum-Frequency Generation

The process of Sum-Frequency Generation (SFG) is described by a nonlinear polarization of the form

$$P(\omega_1 + \omega_2) = 2\varepsilon_0 \chi^{(2)} E_1 E_2 \tag{3.8}$$

A main application of SFG is the development of a frequency-tunable UV laser by mixing a frequency-fixed and a frequency-tunable visible laser. The diagram illustrating the SFG is shown in Figure 3.4 below.

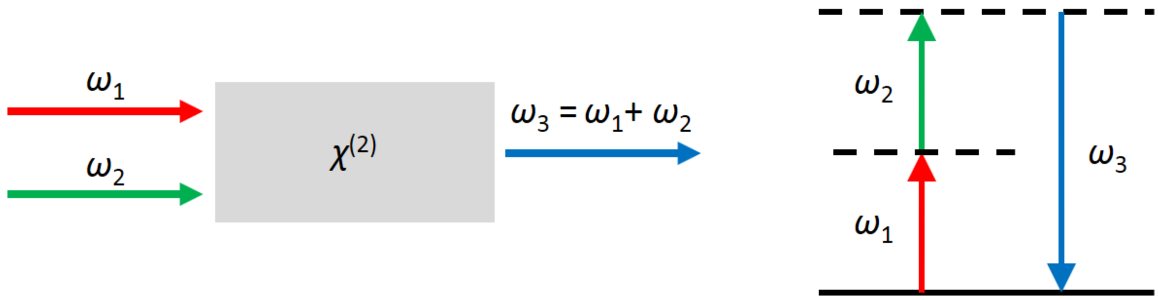


Figure 3.4 Geometry of SFG (left). Energy level diagram describing SFG (right).

Difference-Frequency Generation

The process of Difference-Frequency Generation (DFG) is described by a nonlinear polarization of the form

$$P(\omega_1 - \omega_2) = 2\varepsilon_0 \chi^{(2)} E_1 E_2^* \tag{3.9}$$

The main application of DFG is that it is used in the development of a frequency-tunable IR laser by mixing a frequency-fixed and a frequency-tunable visible laser. The diagram illustrating the SFG is shown in Figure 3.5 below.

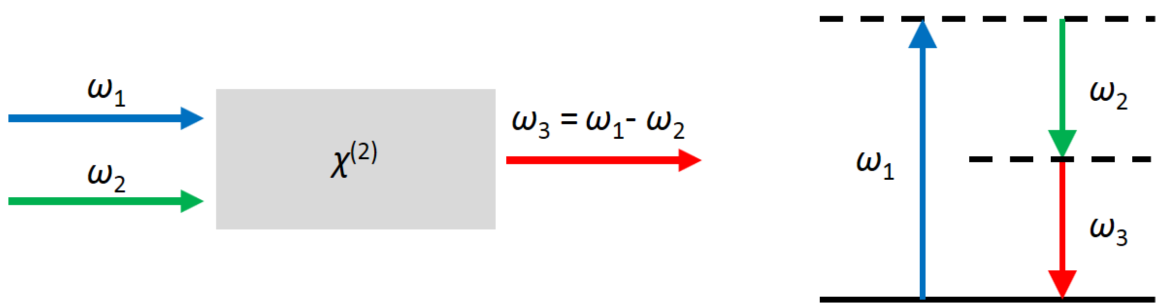


Figure 3.5 Geometry of DFG (left). Energy level diagram describing DFG (right).

The significant difference between the SFG and DFG is shown in their energy level diagrams which can be explained based on the conservation of energy. In SFG, the molecule absorbs energy from two lower frequency photons ω_1 and ω_2 to create a higher frequency photon ω_3. Conversely in DFG, the molecule absorbs energy from a high frequency photon ω_1 to jump to the highest virtual level, then decays and creates two lower frequency photons ω_2 and ω_3. Although the ω_2 input field in DFG does not play a role in the molecular energy level jump, its presence still helps increase the ω_2 output field by stimulating the molecule to vibrate at that frequency. For this reason, DFG is also known as Optical Parametric Amplification (OPA) in the sense that the output at frequency ω_2 is amplified via stimulated emission. If the input ω_2 field is not applied, the generated output at frequency ω_2 will be much weaker since it is created by spontaneous emission. Such processes are known as Optical Parametric Generation (OPG). If the nonlinear material or crystal is placed inside an optical resonator, the output fields at ω_2 and/or ω_3 can be enhanced to become even stronger than in the case of OPA. Such a process is known as Optical Parametric Oscillation (OPO).

Optical Parametric Generation, Amplification, and Oscillation

The second project in this dissertation is focused on the development of the CO_2 DIAL laser transmitter based on the use of the nonlinear processes called Optical Parametric Generation (OPG), Optical Parametric Amplification (OPA), and Optical Parametric Oscillation (OPO). A diagram for OPG/OPA/OPO is shown in Figure 3.6 where the highest frequency in the process ω_3 is referred as pump, the ω_1 input field is referred as seed, the ω_1 output field is referred as signal, and the ω_2 output field is referred as idler.

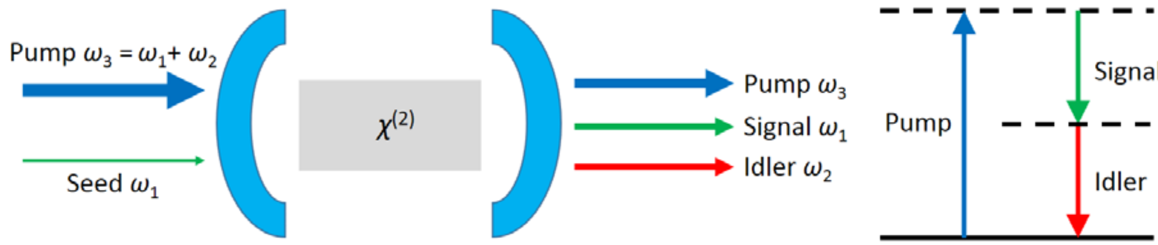


Figure 3.6 OPO geometry (left). Optical cavity is absent for OPA. Both optical cavity and seed laser are absent for OPG. Energy level diagram (right).

As shown in the Figure 3.6 energy level diagram, a pump photon incident upon a nonlinear material is converted to signal and idler phonons. Both signal and idler photons have lower frequency and thus longer wavelength than the pump photon which satisfies the conservation of energy as described in equation (3.10) below.

$$\frac{1}{\lambda_{pump}} = \frac{1}{\lambda_{signal}} + \frac{1}{\lambda_{idler}} \tag{3.10}$$

As mentioned in the previous section, OPG relies on spontaneous emission. For this reason, the frequency spectrum for both the signal and idler being generated are very broad. OPA on the other hand, uses a seed laser at the desired signal wavelength to amplify the output signal which is resulting in a stronger and narrower spectrum. To further enhance the signal, OPO uses an optical cavity which is usually designed to resonate at the seed/signal wavelength. An OPO can be seeded or unseeded and is frequently used at infrared wavelengths where laser sources are not readily available. In the case of the second research project in this dissertation, the OPO is used to produce the signal at the proposed CO_2 DIAL wavelength of 1571 nm from a 1064-nm pump and a 1571-nm seed laser. In the process of developing the OPO, OPG and OPA have also been studied.

Mathematical Description of Nonlinear Optics

To fully understand nonlinear optics, we must understand the mathematics behind it. Starting from Maxwell's equations in matter, we can attain the wave equation for a nonlinear material. Then we shall use that to derive a set of coupled differential wave equations for the case of plane waves in second-order frequency mixing processes. Next we shall explore the solution to these coupled wave equations in cases of non-depleted and depleted pump approximation for OPA. Along the way, the lossless approximation and phase-matching condition shall also be discussed. Afterwards, we shall consider the mathematical description of a Gaussian beam, which is used to describe the actual nature of a laser beam. In the end, nonlinear system modeling for cases of OPA and OPO involving Gaussian beams shall be discussed.

Derivation of the Wave Equation for Nonlinear Material

Following the derivation in Yariv [38], we start with Maxwell's equations in matter [39]

$$\nabla \cdot \vec{D} = \rho_f \tag{3.11}$$

$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \tag{3.12}$$

$$\nabla \cdot \vec{B} = 0 \tag{3.13}$$

$$\nabla \times \vec{H} = \vec{J}_f + \frac{\partial \vec{D}}{\partial t} \tag{3.14}$$

with the definitions

$$\vec{D} = \varepsilon_0 \vec{E} + \vec{P} \tag{3.15}$$

and
$$\vec{H}=\frac{1}{\mu_0}\vec{B}-\vec{M} \tag{3.16}$$

We are primarily interested in the solution of these equations in regions of space that contain no free charge, which makes

$$\rho_f=0 \tag{3.17}$$

and nonmagnetic material, so that

$$\vec{M}=0 \tag{3.18}$$

However, we allow the material to be nonlinear where its induced polarization $\vec{P}$ by an applied electric field $\vec{E}$ takes the form in equation (3.1), which can also be expressed as

$$\vec{P}=\vec{P}_L+\vec{P}_{NL} \tag{3.19}$$

where $\vec{P}_L=\varepsilon_0\chi^{(1)}\vec{E}$ is linear polarization and $\vec{P}_{NL}=\varepsilon_0\left[\chi^{(2)}\vec{E}^2+\chi^{(3)}\vec{E}^3+...\right]$ is nonlinear polarization.

To derive the vector wave equation, we take the curl of equation (3.12) and use $\nabla\times\left(\nabla\times\vec{E}\right)=\nabla\left(\nabla\cdot\vec{E}\right)-\nabla^2\vec{E}$ to obtain

$$\nabla\left(\nabla\cdot\vec{E}\right)-\nabla^2\vec{E}=-\frac{\partial}{\partial t}\left(\nabla\times\vec{B}\right) \tag{3.20}$$

Although $\nabla\cdot\vec{D}=0$, equation (3.15) implies that $\nabla\cdot\vec{E}$ is not necessary zero. However, in nonlinear optics, $\nabla\cdot\vec{E}$ in the first term of equation (3.20) can usually be dropped under many circumstances. For example, $\nabla\cdot\vec{E}=0$ in the case of an infinite plane wave. More generally, $\nabla\cdot\vec{E}$ can be shown to be very small, especially under the slowly-varying amplitude approximation [36]. Therefore by neglecting $\nabla\cdot\vec{E}$, substituting equation (3.16) into (3.14) to replace $\nabla \times \vec{B}$, and substituting equation (3.15) for $\vec{D}$, equation (3.20) becomes

$$\nabla^2 \vec{E} = \mu_0 \frac{\partial \vec{J}_f}{\partial t} + \mu_0 \varepsilon_0 \frac{\partial^2 \vec{E}}{\partial t^2} + \mu_0 \frac{\partial^2 \vec{P}}{\partial t^2} \tag{3.21}$$

Using the definition $\vec{J}_f = \sigma \vec{E}$ where σ is conductivity, equation (3.19), and the definition $\varepsilon = \varepsilon_0 \left(1 + \chi^{(1)}\right)$ for permittivity, equation (3.21) can be rewritten as

$$\nabla^2 \vec{E} = \mu_0 \sigma \frac{\partial \vec{E}}{\partial t} + \mu_0 \varepsilon \frac{\partial^2 \vec{E}}{\partial t^2} + \mu_0 \frac{\partial^2 \vec{P}_{NL}}{\partial t^2} \tag{3.22}$$

which is known as the wave equation for nonlinear material.

Derivation of Coupled Wave Equations for Plane Waves
<u>in Second-Order Frequency Mixing Processes</u>

We now consider a specific case of a second-order frequency mixing process that deals with only three distinct frequencies ω_1, ω_2, and ω_3. The process can be SHG, SFG or DFG depending on how ω_1, ω_2, and ω_3 are defined i.e. $\omega_1 + \omega_2 = \omega_3$ for DFG, $\omega_1 = \omega_2 = \frac{\omega_3}{2}$ for SHG. For simplicity, we also assume traveling plane waves in the *z*-direction for all three frequencies and that their electric field components for each frequency can be described by [38]

$$\begin{aligned}
E_i^{(\omega_1)}(z,t) &= \frac{1}{2}\left[E_{1i}(z)e^{i(\omega_1 t - k_1 z)} + c.c.\right] \\
E_j^{(\omega_2)}(z,t) &= \frac{1}{2}\left[E_{2j}(z)e^{i(\omega_2 t - k_2 z)} + c.c.\right] \\
E_k^{(\omega_3)}(z,t) &= \frac{1}{2}\left[E_{3k}(z)e^{i(\omega_3 t - k_3 z)} + c.c.\right]
\end{aligned} \tag{3.23}$$

where subscripts *i, j, k* refers to Cartesian coordinates which can take on values *x* and *y*.

Oftentimes only the second-order term is being considered for $\vec{P}_{NL}$ in equation (3.22) since the higher-order terms are negligible. In such case, the components of $\vec{P}_{NL}$ are normally expressed as

$$\left(P_{NL}\right)_i = d_{ijk} E_j E_k \tag{3.24}$$

where $d_{ijk} = \chi_{ijk}^{(2)}$ is the material-dependent tensor. With this notation, the *i* component of $\vec{P}_{NL}$ at the frequency $\omega_1 = \omega_3 - \omega_2$, the *j* component of $\vec{P}_{NL}$ at the frequency $\omega_2 = \omega_3 - \omega_1$, and the *k* component of $\vec{P}_{NL}$ at the frequency $\omega_3 = \omega_1 + \omega_2$ are described in equation (3.25) below.

$$\begin{aligned}
\left[P_{NL}^{(\omega_1)}(z,t)\right]_i &= \frac{d_{ikj}}{2} E_{3k}(z) E_{2j}^*(z) e^{i\left[(\omega_3-\omega_2)t-(k_3-k_2)z\right]} + c.c. \\
\left[P_{NL}^{(\omega_2)}(z,t)\right]_j &= \frac{d_{jki}}{2} E_{3k}(z) E_{1i}^*(z) e^{i\left[(\omega_3-\omega_1)t-(k_3-k_1)z\right]} + c.c. \\
\left[P_{NL}^{(\omega_3)}(z,t)\right]_k &= \frac{d_{kij}}{2} E_{1i}(z) E_{2j}(z) e^{i\left[(\omega_1+\omega_2)t-(k_1+k_2)z\right]} + c.c.
\end{aligned} \tag{3.25}$$

We can now proceed toward solving the wave equation by substituting equation (3.23) for $\vec{E}$ and equation (3.25) for $\vec{P}_{NL}$ into equation (3.22). Because the number of interacting frequencies is finite, equation (3.22) can be satisfied separately by each frequency component as

$$\nabla^2 E_i^{(\omega_1)} = \mu_0 \sigma \frac{\partial E_i^{(\omega_1)}}{\partial t} + \mu_0 \varepsilon \frac{\partial^2 E_i^{(\omega_1)}}{\partial t^2} + \mu_0 \frac{\partial^2 \left[\vec{P}_{NL}^{(\omega_1)}\right]_i}{\partial t^2} \tag{3.26}$$

with similar expressions for ω_2 and ω_3. For succinctness, the following only shows the frequency ω_1 calculation. Starting with taking the Laplacian of $E_i^{(\omega_1)}$ yields

$$\begin{aligned}\nabla^2 E_i^{(\omega_1)}(z,t) &= \frac{1}{2}\frac{\partial^2}{\partial z^2}\left[E_{1i}(z)e^{i(\omega_1 t-k_1 z)}+c.c.\right]\\ &= -\frac{1}{2}\left[k_1^2 E_{1i}(z)+2ik_1\frac{dE_{1i}(z)}{dz}\right]e^{i(\omega_1 t-k_1 z)}+c.c.\end{aligned} \tag{3.27}$$

where the variation of the complex field amplitudes with z was assumed to be small enough such that

$$\frac{dE_{1i}}{dz}k_1 \gg \frac{d^2E_{1i}}{dz^2} \tag{3.28}$$

By substituting the result in equation (3.27) into (3.26) and working out the time derivative of $E_i^{(\omega_1)}$, we arrive at the wave equation for the ω_1 frequency as

$$\begin{aligned}&\left[\frac{k_1^2}{2}E_{1i}(z)+ik_1\frac{dE_{1i}(z)}{dz}\right]e^{i(\omega_1 t-k_1 z)}+c.c.=\\ &\left(-i\omega_1\mu_0\sigma+\omega_1^2\mu_0\varepsilon\right)\left[\frac{E_{1i}(z)}{2}e^{i(\omega_1 t-k_1 z)}+c.c.\right]-\mu_0\frac{\partial^2}{\partial t^2}\left[P_{NL}^{(\omega_1)}(z,t)\right]_i\end{aligned} \tag{3.29}$$

Replacing $\left[P_{NL}^{(\omega_1)}(z,t)\right]_i$ by equation (3.25) and using the fact that $\omega_1^2\mu_0\varepsilon=k_1^2$ as well as $\omega_3-\omega_2=\omega_1$, we obtain

$$ik_1\frac{dE_{1i}}{dz}e^{-ik_1z}=-\frac{i\omega_1\mu_0\sigma}{2}E_{1i}e^{-ik_1z}+\frac{\mu_0\omega_1^2}{2}d_{ikj}E_{3k}E_{2j}^{*}e^{-i(k_3-k_2)z} \tag{3.30}$$

Next we divide the last equation by $ik_1e^{-ik_1z}$, allow σ and ε to be a function of frequency, and do similar calculations for ω_2 and ω_3 to get a set of coupled differential wave equations

$$\begin{aligned}
\frac{dE_{1i}}{dz} &= -\frac{\sigma_1}{2}\sqrt{\frac{\mu_0}{\varepsilon_1}}E_{1i} - \frac{i\omega_1}{2}d_{ikj}E_{3k}E_{2j}^{*}e^{-i(k_3-k_2-k_1)z} \\
\frac{dE_{2j}^{*}}{dz} &= -\frac{\sigma_2}{2}\sqrt{\frac{\mu_0}{\varepsilon_2}}E_{2j}^{*} + \frac{i\omega_2}{2}d_{jik}E_{1i}E_{3k}^{*}e^{-i(k_1-k_3+k_2)z} \\
\frac{dE_{3k}}{dz} &= -\frac{\sigma_3}{2}\sqrt{\frac{\mu_0}{\varepsilon_3}}E_{3k} + \frac{i\omega_3}{2}d_{kij}E_{1i}E_{2j}e^{-i(k_1+k_2-k_3)z}
\end{aligned} \tag{3.31}$$

To help simplify the equation above, we define

the new field $$A_l = \sqrt{\frac{n_l}{\omega_l}}E_l \qquad l = 1,2,3 \tag{3.32}$$

with intensity $$I_l = \frac{P_l}{A} = \frac{1}{2}\sqrt{\frac{\varepsilon_0}{\mu_0}}n_l\left|E_l\right|^2 = \frac{1}{2}\sqrt{\frac{\varepsilon_0}{\mu_0}}\omega_l\left|A_l\right|^2 \tag{3.33}$$

the phase-matching condition $$\Delta k = k_3 - (k_1 + k_2) \tag{3.34}$$

the coupling parameter $$\kappa \equiv d_{123}\sqrt{\left(\frac{\mu_0}{\varepsilon_0}\right)\frac{\omega_1\omega_2\omega_3}{n_1 n_2 n_3}} \tag{3.35}$$

and the parameter $$\alpha_l \equiv \sigma_l\sqrt{\frac{\mu_0}{\varepsilon_l}} \qquad l = 1,2,3 \tag{3.36}$$

where l=1, 2, 3 are the polarization directions of $\vec{E}_1$, $\vec{E}_2$, $\vec{E}_3$. Then the expression in (3.31) becomes

$$\begin{aligned}
\frac{dA_1}{dz} &= -\frac{1}{2}\alpha_1 A_1 - \frac{i}{2}\kappa A_2^{*}A_3 e^{-i(\Delta k)z} && \textit{Signal} \\
\frac{dA_2^{*}}{dz} &= -\frac{1}{2}\alpha_2 A_2^{*} + \frac{i}{2}\kappa A_1 A_3^{*} e^{i(\Delta k)z} && \textit{Idler} \\
\frac{dA_3}{dz} &= -\frac{1}{2}\alpha_3 A_3 - \frac{i}{2}\kappa A_1 A_2 e^{i(\Delta k)z} && \textit{Pump}
\end{aligned} \tag{3.37}$$

which is known as the coupled wave equations for nonlinear processes.

Coupled Wave Equations for Lossless Material

In general, crystals used in nonlinear processes are transparent at the pump, signal, and idler wavelengths, so that there are negligible losses due to absorption. Therefore, the loss term $\alpha_l \approx 0$ is usually dropped from equation (3.37), which yields the coupled wave equations for lossless material

$$\begin{aligned} \frac{dA_1}{dz} &= -\frac{i}{2}\kappa A_2^* A_3 e^{-i(\Delta k)z} && \textit{Signal} \\ \frac{dA_2^*}{dz} &= \frac{i}{2}\kappa A_1 A_3^* e^{i(\Delta k)z} && \textit{Idler} \\ \frac{dA_3}{dz} &= -\frac{i}{2}\kappa A_1 A_2 e^{i(\Delta k)z} && \textit{Pump} \end{aligned} \tag{3.38}$$

Non-Depleted Pump Solution

In the non-depleted pump approximation, the pump beam is negligibly depleted in the nonlinear material. Under this assumption, $A_3(z) = A_3(0) = \text{constant}$, which makes $\frac{dA_3}{dz} = 0$. With these simplifications and for the general case in which both the signal and idler with field amplitudes $A_1(0)$ and $A_2(0)$ present at the input, the solution for $A_1(z)$ and $A_2(z)$ in equation (3.38) are

$$\begin{aligned} A_1(z)e^{i(\Delta k)z/2} &= A_1(0)\left[\cosh(bz) - \frac{i\Delta k}{2b}\sinh(bz)\right] - \frac{i\kappa}{2b}A_2^*(0)A_3(0)\sinh(bz) \\ A_2^*(z)e^{-i(\Delta k)z/2} &= A_2^*(0)\left[\cosh(bz) - \frac{i\Delta k}{2b}\sinh(bz)\right] - \frac{i\kappa}{2b}A_1(0)A_3(0)\sinh(bz) \end{aligned} \tag{3.39}$$

with $$b = \frac{1}{2}\sqrt{\left[\kappa A_3(0)\right]^2 - (\Delta k)^2}$$

In the case of OPA, only the input seed at the signal wavelength is used, so that only $A_1(0)$ is present while $A_2(0)$ vanishes making

$$\begin{aligned} A_1(z)e^{i(\Delta k)z/2} &= A_1(0)\left[\cosh(bz) - \frac{i\Delta k}{2b}\sinh(bz)\right] \\ A_2^*(z)e^{-i(\Delta k)z/2} &= -\frac{i\kappa}{2b}A_1(0)A_3(0)\sinh(bz) \end{aligned} \tag{3.40}$$

Using equation (3.33) to convert to output intensities at the end of crystal length $z = L$, yields

$$\begin{aligned} I_1 &= \frac{1}{2}\sqrt{\frac{\varepsilon_0}{\mu_0}}\omega_1\left|A_1(0)\right|^2\left[\cosh^2(bL) + \frac{(\Delta k)^2}{4b^2}\sinh^2(bL)\right] \\ I_2 &= \frac{1}{2}\sqrt{\frac{\varepsilon_0}{\mu_0}}\omega_1\left|A_1(0)\right|^2\left|A_3(0)\right|^2\frac{(\Delta k)^2}{4b^2}\sinh^2(bL) \end{aligned} \tag{3.41}$$

Phase-Matching Condition

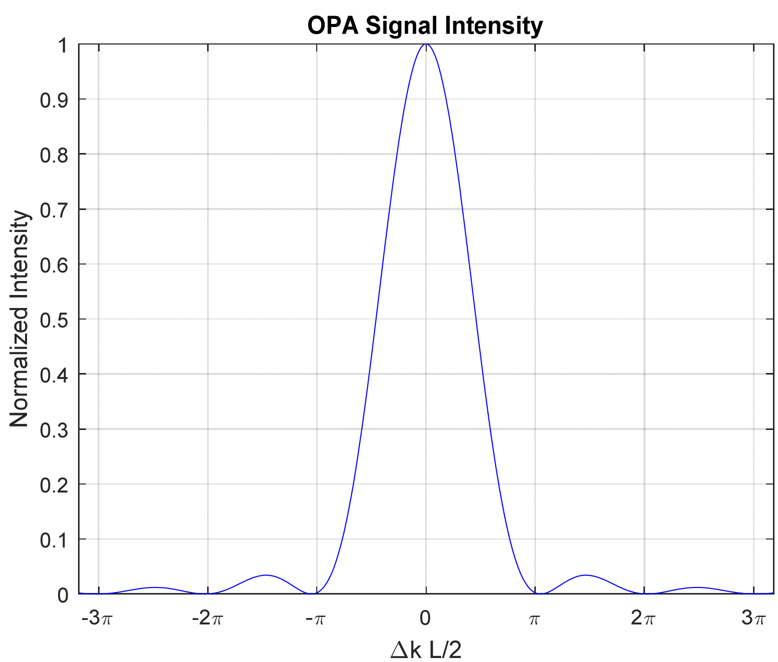


Figure 3.7 Effect of phase mismatch factor Δk on the output signal intensity in OPA.

Following the results of the previous section, it should be noted that the output signal intensity I_1 has the maximum value when $\Delta k = 0$ as shown in Figure 3.7. The condition at which $\Delta k = 0$ resulting in the highest possible output in a nonlinear process is called "perfect phase-matching". This makes $\Delta k = k_3 - k_1 - k_2 = 0$ for OPA, and thus the following equation must be satisfied.

$$n_1\omega_1 + n_2\omega_2 = n_3\omega_3 \tag{3.42}$$

However, conservation of energy requires that

$$\omega_1 + \omega_2 = \omega_3 \tag{3.43}$$

which makes equation (3.42) impossible to satisfy because, in most materials, the index of refraction $n(\omega)$ increases with ω. Nevertheless, phase-matching can be achieved in a certain class of crystals that are birefringent, meaning the refractive index depends on the polarization direction of light. The most common birefringent material is a uniaxial crystal in which light whose polarization is perpendicular to the optical axis experiences an "ordinary" refractive index n_o and light whose polarization is parallel to the optical axis experiences an "extraordinary" refractive index n_e. The geometry of the ordinary, extraordinary, and optical axis is shown in Figure 3.8 below.

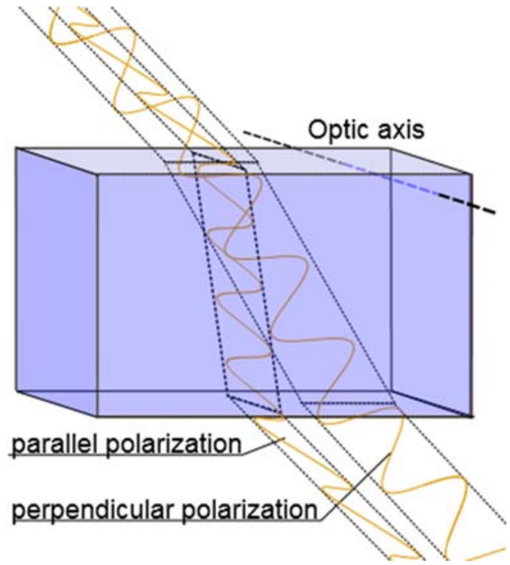


Figure 3.8 Geometry of the ordinary (perpendicular), extraordinary (parallel), and optical axis of a uniaxial crystal. [40]

Two types of uniaxial crystal are positive uniaxial where $n_e > n_o$ and negative uniaxial where $n_e < n_o$. An example of the refractive index plots as a function of frequency for the case of negative uniaxial is shown in Figure 3.9. With the crystal's property of polarization-dependent refractive indices, two types of phase-matching can be achieved for each of the positive and negative uniaxial crystal [41]. These phase-matching methods are summarized in Boyd 2003. [36]

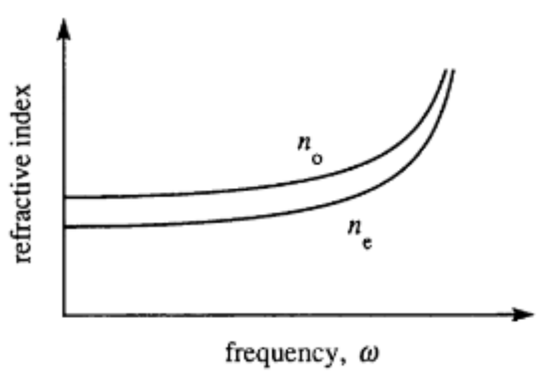


Figure 3.9 Refractive indices of a negative uniaxial crystal. (Boyd 2003 Figure 2.7.2) [36]

Table 3.1 Phase-matching methods for uniaxial crystals. (Boyd 2003 Table 2.7.2) [36]

	Positive uniaxial	Negative uniaxial
Type I	$n_3^o\omega_3 = n_1^e\omega_1 + n_2^e\omega_2$	$n_3^e\omega_3 = n_1^o\omega_1 + n_2^o\omega_2$
Type II	$n_3^o\omega_3 = n_1^o\omega_1 + n_2^e\omega_2$	$n_3^e\omega_3 = n_1^e\omega_1 + n_2^o\omega_2$

<u>Angle Tuning</u>

This method is mostly applied to the case of a uniaxial crystal where the value of the extraordinary refractive index depends on, not only the direction of polarization, but also the direction of propagation of light. The angular dependent extraordinary refractive index $n_e(\theta)$ depends on the angle θ between the optical axis and the direction of propagation as

$$\frac{1}{n_e^2(\theta)} = \frac{\sin^2(\theta)}{\bar{n}_e^2} + \frac{\cos^2(\theta)}{n_o^2} \tag{3.44}$$

where $\bar{n}_e$ is the principle value of the extraordinary refractive index. For this reason, the perfect phase-matching can be achieved by precisely adjusting the angular orientation of the crystal with respect to the propagation direction of the incident light as well as its polarization direction.

<u>Temperature Tuning</u>

For some crystals, the amount of birefringence is strongly dependent on temperature. This allows the possible phase-matching by keeping θ fixed at 90° while varying only the temperature of the crystal. With proper crystal selection, temperature tuning can provide an advantage over angle tuning. This is because, in angle tuning, the ordinary and extraordinary waves have their overlap reduced when θ has a value other than 0° or 90°. As a result, the efficiency of the nonlinear process decreases.

<u>Quasi-Phase-Matching</u>

Under many circumstances, the material may possess either no birefringence or insufficient birefringence to compensate for the dispersion of the linear refractive index. As a result, neither the angle tuning nor temperature tuning can be used effectively. Another specific circumstance where both tuning methods do not work is when the application requires the use of the d_{33} nonlinear coefficient, which tends to be much larger than the off-diagonal coefficients, and is the largest coefficient in the case of lithium niobate. This coefficient can be accessed only if all the interacting waves have the same polarization direction, which is impossible to compensate for dispersion.

A technique known as quasi-phase-matching (QPM) is generally used when the angle and temperature tuning cannot be implemented. As shown in Figure 3.10, this technique divides the nonlinear crystal into periodic polarization domains where the sign of the polarization is inverted at regular intervals. Such crystal is said to be "periodically poled" which also periodically inverts the sign of $\chi^{(2)}$ and thus the sign of the nonlinear coupling coefficient d_{eff}. As a result, the nonzero mismatch value of Δk caused by dispersion can be compensated. Although QPM does not provide the perfect phase-matching, its increased gain is still the most efficient for many materials.

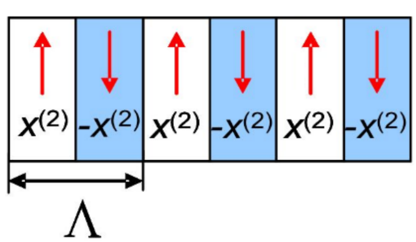


Figure 3.10 Schematic of a periodically poled crystal in QPM technique. [42]

The intensity comparison between the three phase-matching conditions is shown in Figure 3.11. The perfect phase-matching intensity (green) is quadratic dependent on propagation distance. QPM intensity (blue) is smaller than perfect PM by a factor of $4/\pi^2$. The non-phase-matching (red) just causes the intensity to oscillate periodically with distance.

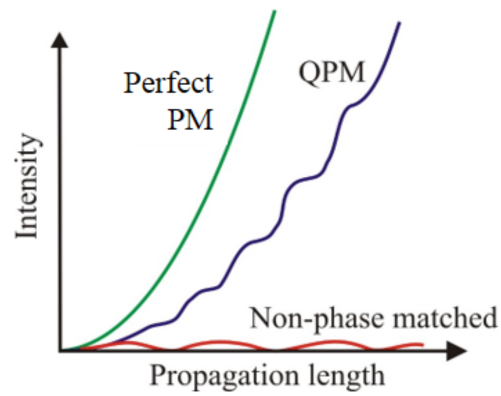


Figure 3.11 Comparison of the spatial variation field intensity between three different phase-matching conditions.

The spatially-dependent nonlinear coupling coefficient $d(z)$ in QPM, which is a square wave function, can be described mathematically as [36]

$$d(z) = d_{eff}\,\mathrm{sign}\left[\cos\left(\frac{2\pi z}{\Lambda}\right)\right] \tag{3.45}$$

where $\mathrm{sign}(x)$ is a function that outputs −1, 0, or 1 depending on the value of input x whether it is negative, zero, or positive and Λ is the poling period as shown in Figure 3.10. To make further approximation, it is useful to express equation (3.45) as a Fourier series

$$d(z) = d_{eff} \sum_{m=-\infty}^{m=+\infty} G_m e^{ik_m z} \tag{3.46}$$

where $k_m = 2\pi m/\Lambda$ is the grating vector associated with the m^{th} order in the Fourier series and the amplitude G_m is given by

$$G_m = \frac{2}{m\pi}\sin\left(\frac{m\pi}{2}\right) \tag{3.47}$$

Since the first-order dominates the amount of output intensity, $d(z)$ in equation (3.46) can estimated to be

$$d(z) = d_{eff}\frac{2}{\pi}e^{i\frac{2\pi}{\Lambda}z} \tag{3.48}$$

If we substitute this result into the coupled wave equations, the field amplitude will change by a factor of $2/\pi$ and the phase by $2\pi/\Lambda$, making the phase-matching condition for QPM as

$$\Delta k_Q = k_1 + k_2 - k_3 - \frac{2\pi}{\Lambda} \tag{3.49}$$

In crystals, where the index of refraction is temperature dependent, such as lithium niobate, the value of Λ can be slightly adjusted by heating or cooling the crystal to satisfy phase-matching. Because of its good potential, lithium niobate and thus QPM technique are used in this research.

Depleted Pump Solution

In the case of depleted pump, there is no analytical solution to the coupled wave equations and therefore they must be solved numerically. For the simplest case of plane waves with a lossless system in OPA, numerical integration can be performed on equation

(3.38) to yield the results. A program written in Matlab (see Appendix) is used to perform the task. To implement the calculation, the program needs to adopt values for many input parameters as shown in Figure 3.12. It should be noted that the coupled wave equations do not assume the input pulsed laser, therefore the energy of the input pulsed pump laser is converted into power using

$$P_{peak} = 0.94 \times \frac{E_{pump}}{\Delta t} \tag{3.50}$$

where Δt is the pulse width, which is set at a typical value of 10 ns in this example calculation. Also to simplify the calculation, Δk is set to zero and the periodically poled lithium niobate (PPLN) crystal with a constant refractive index is assumed in the program.

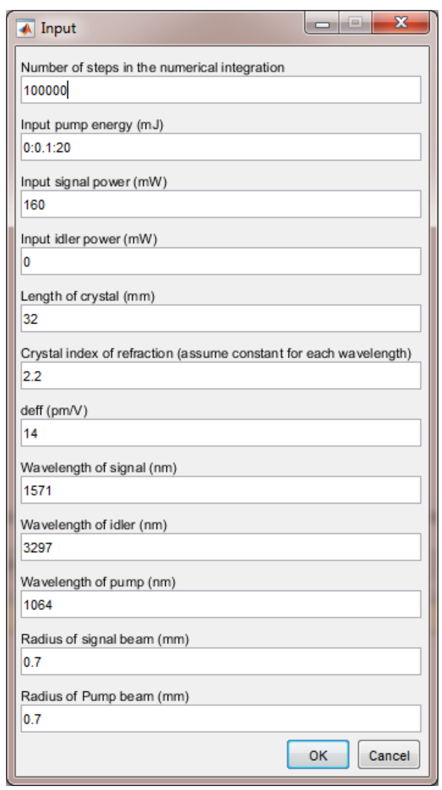


Figure 3.12 Input parameters for coupled wave equations' numerical integration in Matlab program.

The comparison of the OPA output signal energy between the depleted and non-depleted pump approximation is shown in Figure 3.13. As expected, the non-depleted pump approximation is only valid for small input pump energy, where in this particular example, the two results begin to diverge from each other near 8 mJ of input pump energy. While the non-depleted pump output signal energy increases to unrealistically high values (it reaches 20 mJ with only 12 mJ input pump energy), it is normal that the depleted pump output signal energy reaches a peak and then drops down. This phenomenon occurs when the signal and idler become over saturated in the crystal, resulting in the back-conversion, in which the conversion process happens in reverse, i.e. signal and idler converts to pump. Depleted pump approximation should be used whenever possible since it provides a more accurate result.

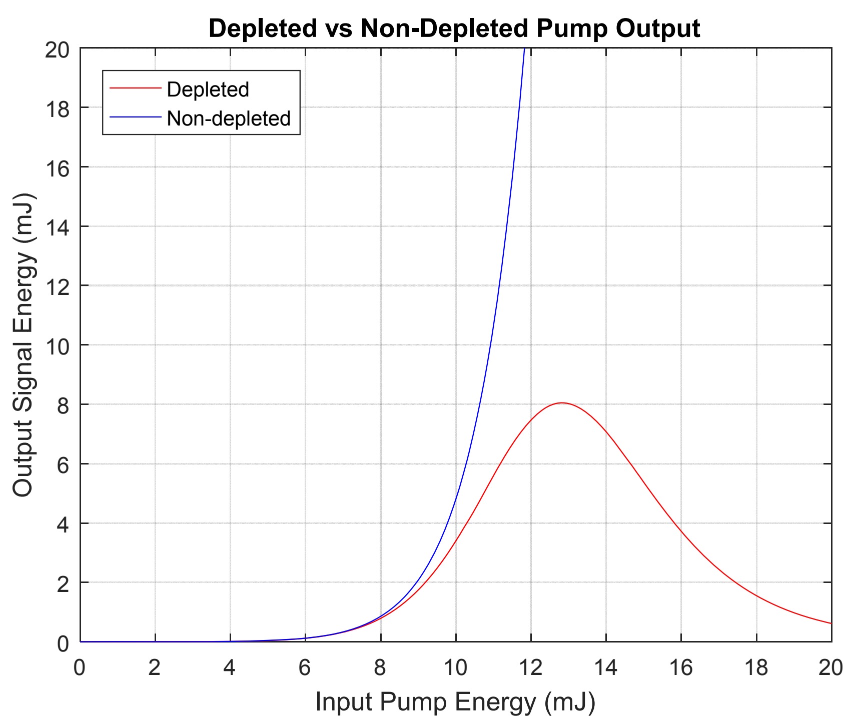


Figure 3.13 Comparison of the OPA output signal energy between depleted and non-depleted pump approximation for the case of plane waves in lossless system.

Gaussian Beam

The Gaussian shape nature of a laser beam can be described by

$$E(r,z) = E_0 \underbrace{\frac{w_0}{w(z)}}_{\text{On axis field strength}} \underbrace{e^{-\frac{r^2}{w^2(z)}}}_{\text{Gaussian profile}} e^{-ikz-i\frac{kr^2}{2R(z)}+i\Phi(z)} \tag{3.51}$$

Plane wave carrier

Wavefront curvature

Excess phase

with $\frac{1}{e^2}$ radius $$w(z) = w_0\sqrt{1+\left(\frac{z}{z_0}\right)^2} \tag{3.52}$$

wavefront radius of curvature $$R(z) = z\left[1+\left(\frac{z}{z_0}\right)^2\right] \tag{3.53}$$

spatial variation of the phase $$\Phi(z) = \arctan\left(\frac{z}{z_0}\right) \tag{3.54}$$

and Rayleigh range $$z_0 = \frac{\pi w_0^2}{\lambda} \tag{3.55}$$

where w_0 represents the beam waist radius.

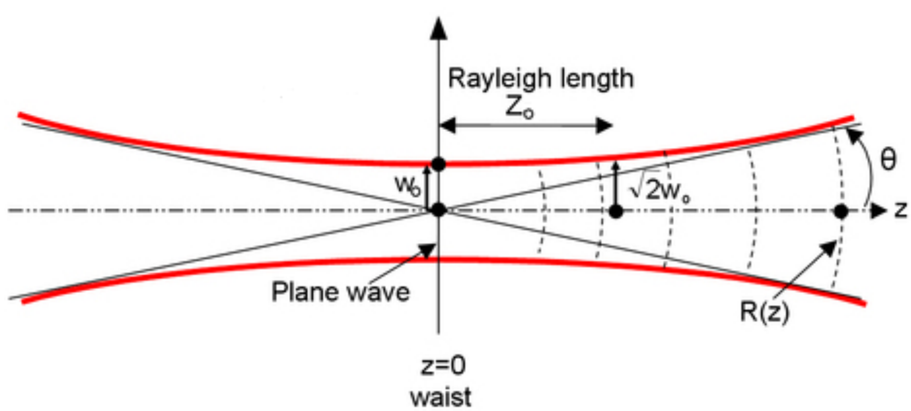


Figure 3.14 Gaussian beam profile.

A Gaussian beam profile with described parameters is demonstrated in Figure 3.14. At the waist location $z = 0$, the beam radius $w(z)$ has the smallest value of w_0, and the wavefront radius of curvature $R(z)$ has the largest value of $+\infty$ (plane wave). At the Rayleigh range of $z = z_0$, $w(z) = \sqrt{2}w_0$ and $R(z)$ has the smallest value. At $z \gg z_0$, $R(z) \approx z$ and $w(z) \approx \theta z$, where θ is the half-angle beam divergence.

The normalized Gaussian intensity shape distribution is demonstrated in Figure 3.15. It displays an important feature that the $1/e^2$ diameter $2w$ or the "spot size", which is commonly used in laser calculation, contains about 86% of beam intensity.

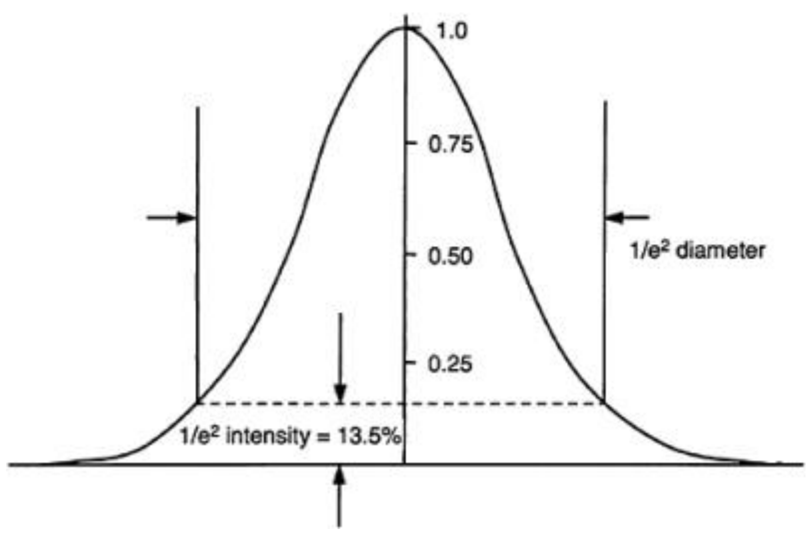


Figure 3.15 Gaussian intensity distribution. [43]

SNLO Model

To accurately model the OPA, the wave equation involving the Gaussian beam described in equation (3.51) for all three frequencies needs to be solved. In addition, approximations such as lossless material and non-depleted pump must be omitted. A better method to account for the pulsed laser, which is needed for a DIAL, is also required in the calculation. Nonlinear process modeling becomes even more complex for the OPO where there are many more factors coming into play including cavity length, cavity mirrors' reflectivity/transmission, mirrors' curvature, and relative crystal location inside the cavity.

Within the scope and time frame of this research, SNLO [44] which is an advanced program developed at Sandia National Laboratory, is deployed for modeling the OPA and OPO. The program is designed particularly for nonlinear processes calculations. It also contains necessary information of most nonlinear crystals such as wavelength-dependent index of refraction along ordinary and extraordinary axis, nonlinear coefficients, and temperature data. More detail about SNLO and its results are discussed in chapter 5 and 6.

MICRO-PULSED LIDAR

Instrument Design

A schematic of the Micro-Pulsed Lidar (MPL) described in this dissertation is shown in Figure 4.1 below.

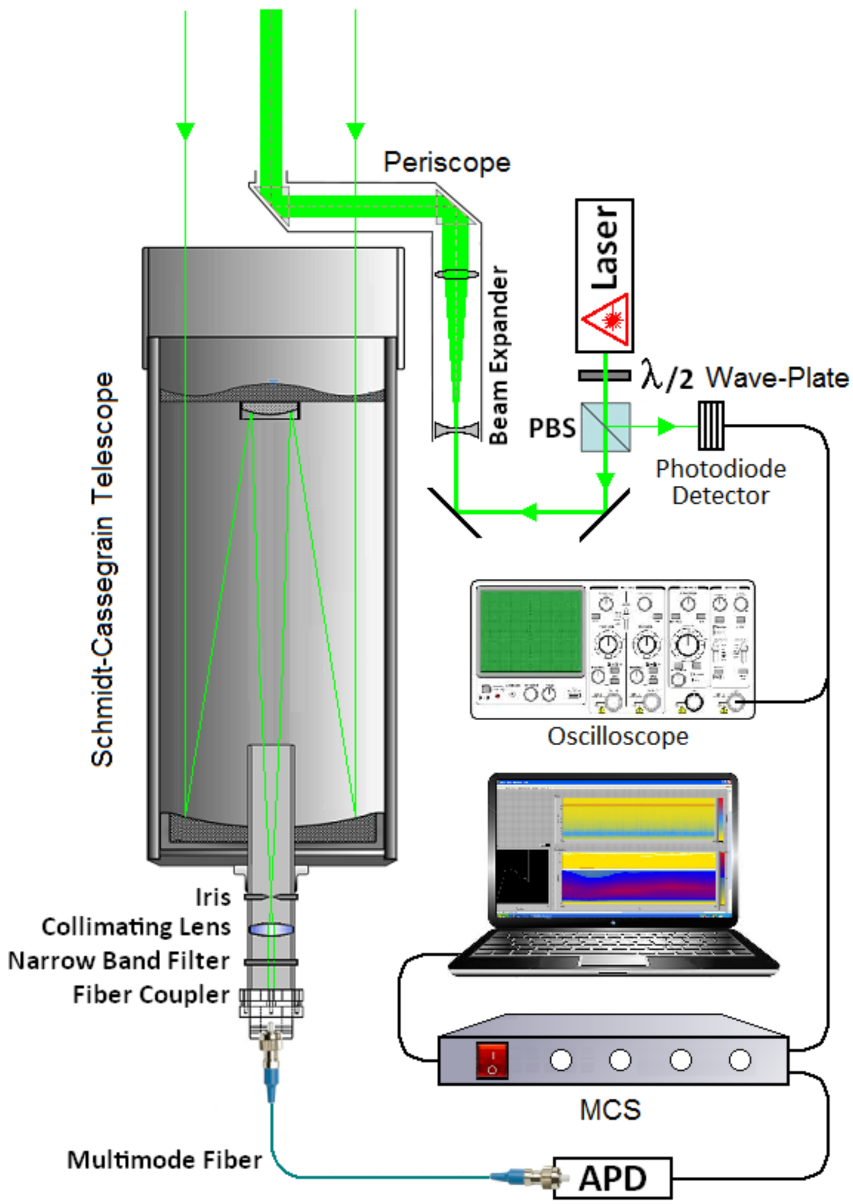


Figure 4.1 Schematic of the Micro-Pulsed Lidar (MPL).

<u>Transmitter</u>

The MPL transmitter uses a Team Photonics Nd:YAG frequency doubled 532-nm passive Q-switched laser which has a pulse repetition frequency (PRF) of 7.2 kHz with a 1-ns pulse duration and a 3-μJ pulse energy. A half-wave-plate (λ/2) is used for changing the polarization direction of the laser beam, which provides control over the amount of light that is transmitted/reflected at the polarizing beam splitter (PBS). When the MPL is operating, we want to have the majority of the beam transmitting through the PBS to provide the maximum possible signal for the system. However, 10% of the beam is redirected at the PBS to act as a triggering signal for the scaler card. Two silver coated mirrors are used to guide the PBS transmitted beam to a Thorlabs BE10M-A beam expander, where the beam is collimated and expanded at 10X magnification for eye-safe operation. A periscope is used for creating a monostatic configuration for the MLP. The MPL transmitter parameters are listed in Table 4.1.

Table 4.1 MPL transmitter parameters.

Parameter	Value
Wavelength	532 nm
Laser energy per pulse	> 3 μJ
Laser average power	25.57 mW
Pulse width	1 ns
Pulse repetition frequency	7.2 kHz
Outgoing beam diameter	40 mm
Outgoing beam average power	19.7 mW
Beam expander	10x

Receiver

The MPL receiver uses a Celestron 11-inch Schmidt-Cassegrain telescope with an f/# (focal length to diameter ratio) of f/10 to collect the back scattered returns. Other components in the receiver are 250-mm focal length collimating lens, a narrow band filter with 532-nm center wavelength, and a Thorlabs PAF-X-11-PC-A fiber coupler. A multimode optical fiber is used to deliver the coupled light to an avalanche photodiode detector (APD) where the light is converted into electrical signal. The MPL receiver parameters are listed in Table 4.2.

Table 4.2 MPL receiver parameters.

Parameter	**Value**
Telescope diameter	278 mm
Telescope f/#	f/10
Collimation lens focal length	250 mm
Narrow band filter center wavelength	532±0.2 nm
Narrow band filter FWHM	1 nm
Fiber coupler lens focal length	11 mm
Fiber coupler lens diameter	1.8 mm
Optical fiber core diameter	105 µm
APD efficiency at 532 nm	54%
APD dark count rate	150 counts/s
APD saturation count rate	12 MHz

Data Acquisition and Control Unit

The data acquisition (DAQ) and control unit uses a multi-channel scaler card (MCS) developed by Sigma Space. The MCS is capable of scaling the range resolution of the data by synchronizing the APD returns with the triggering signal which allows it to perform the counts in each bin based on their time-of-flight.

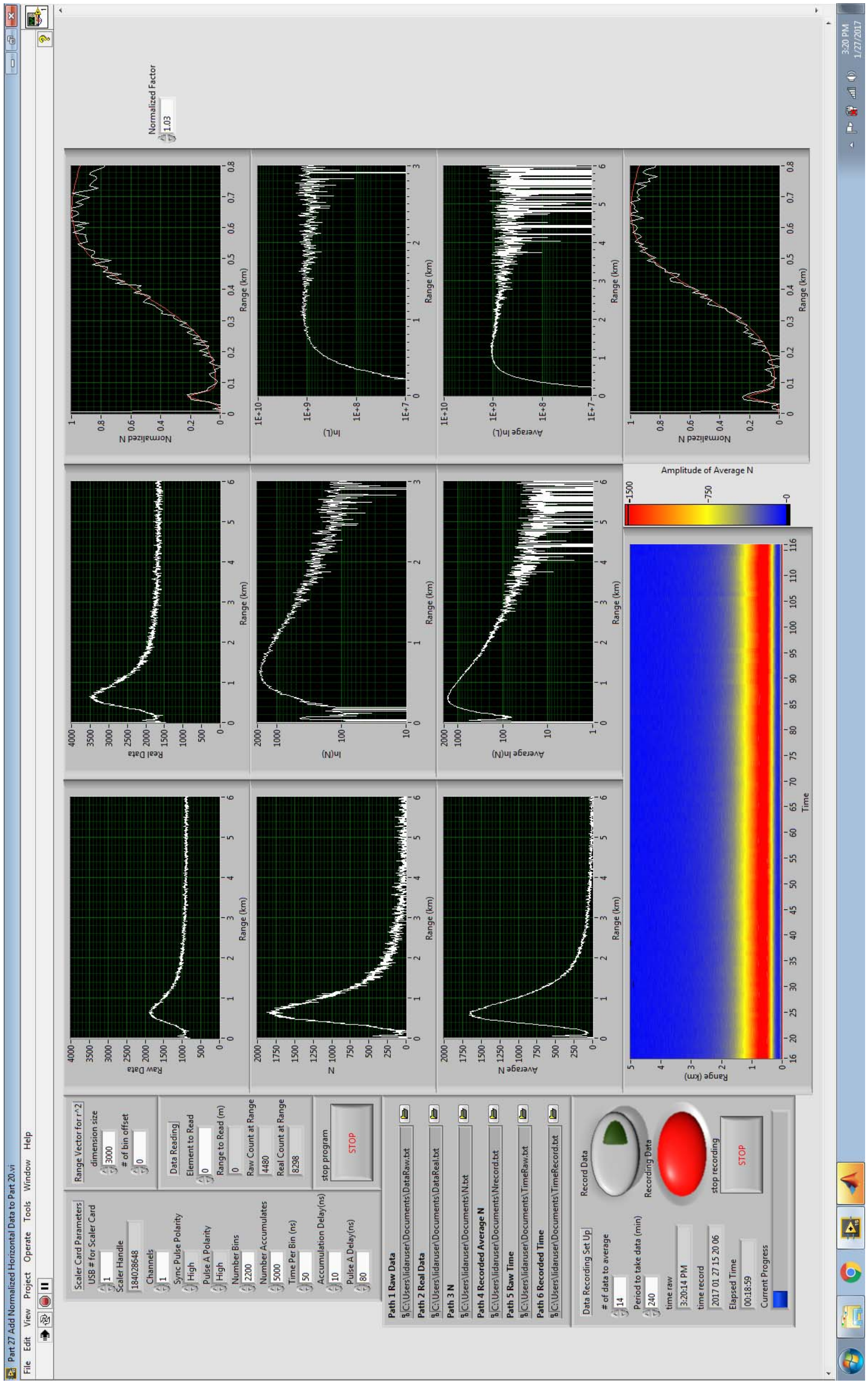


Figure 4.2 Front panel of the Labview program for MPL DAQ and control.

A program written in Labview (see Appendix A) is used as the DAQ and control of the MPL. As shown in Figure 4.2, the program's front panel consists of five main sections: the MCS settings (top left first column), range setting and real-time data-point reading (top left second column), data-saving paths and filenames (center left), data-recording settings (bottom left), and data plots (plot region). Each of the plots in the program displays the measurement in real time which is very useful when adjusting to maintain the system alignment. The detail on how to align the MPL is discussed later in the methods and setup section. Aside from controlling the MCS, recording data, and displaying real-time plots, the program also simultaneously performs preliminary calculations including APD efficiency correction, background light averaging, background light subtraction to get N, range correction (the $1/r^2$ factor in the lidar equation) to get L, $\ln(N)$ calculation, $\ln(L)$ calculation, and time averaging of data. All of these calculations are shown in the block diagram of the Labview program (see Appendix A).

Tripod Mount Control

The MPL is designed to be a compact system such that it can be attached to the Celestron mount and tripod as shown in Figure 4.3 (left). This provides the MPL with a moving and rotating mechanism. Due to the weight limit on the tripod, the MPL is designed to comprise of only essential components. All of the transmitting optics are fitted on just an 8″×10″ optical breadboard as shown in Figure 4.3 (top right). All of the receiving optics attached in the back of the telescope are fitted in a 5-in long tube of 1-in diameter as shown in Figure 4.3 (bottom right).

Figure 4.3 MPL system (left), transmitter (top right), and receiver (bottom right).

To electronically rotate the mount, the system weight must be counter-balanced. As shown in Figure 4.4, the laser power supply is attached in the back of the telescope to counter balance the weight of the transmitter that is attached in the front. A counter balanced weight of 20 pounds is also attached to the rod extended on the opposite side of the telescope. The slant path program written in Labview (see Appendix) is used for controlling the mount which allows a precise shooting direction. It also introduces a scanning capability to the system for the first time. Although a majority of data was taken in the horizontal and vertical directions, the programming has been developed and tested for future scanning use. The front panel of the slant path program is shown in Figure 4.5.

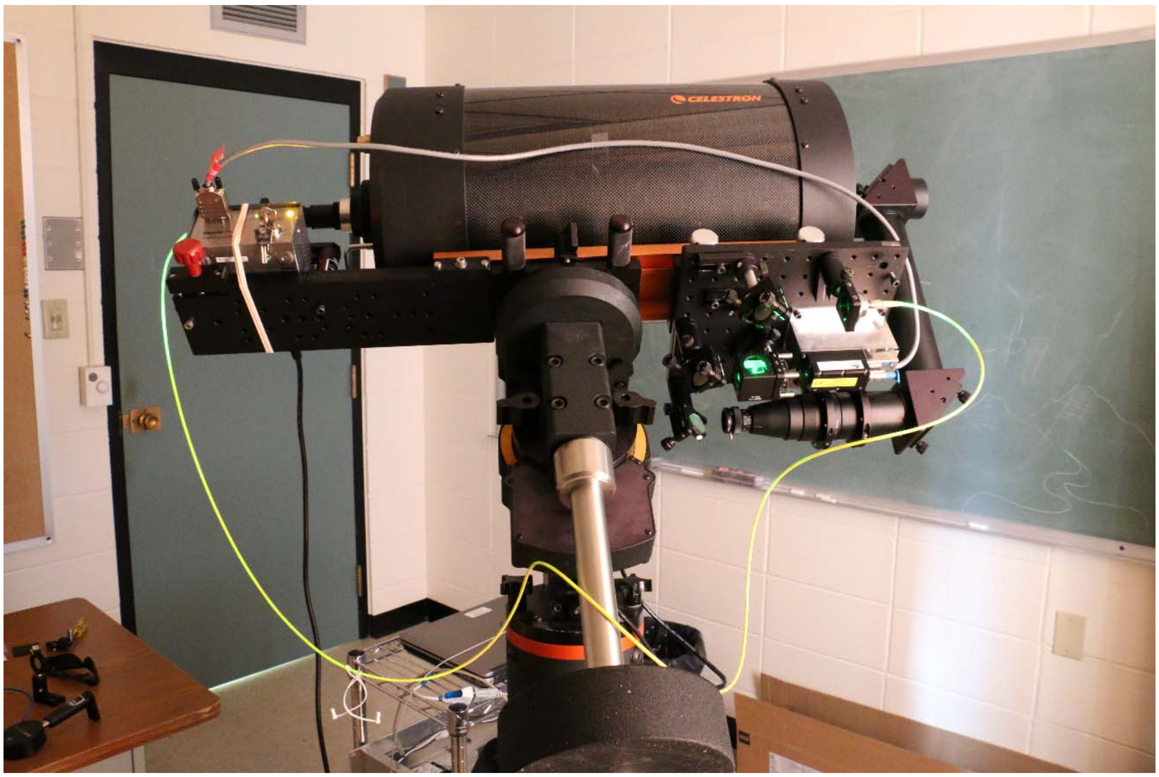


Figure 4.4 Demonstration of weight counter balance in the MPL design.

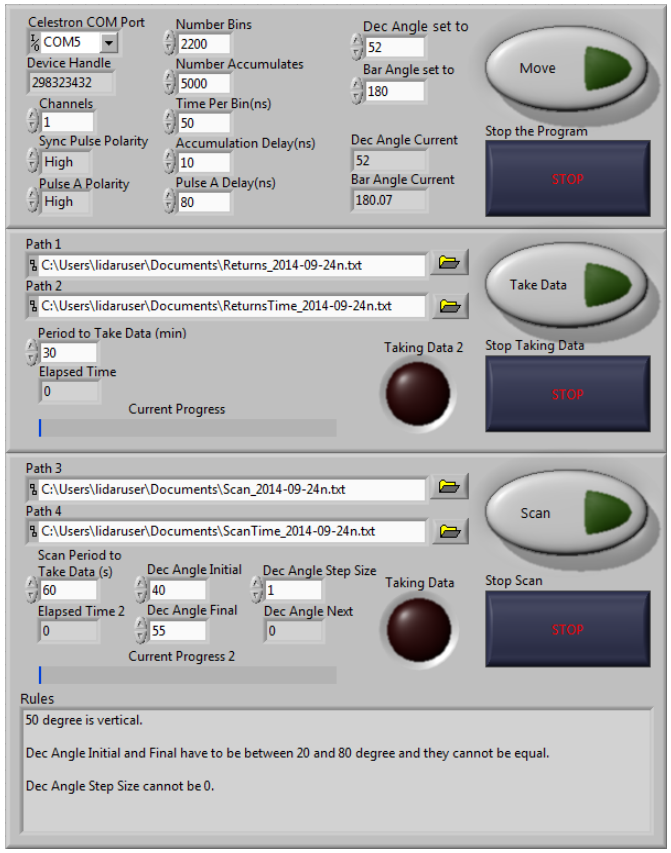


Figure 4.5 Front panel of the slant path program for MPL mount control.

Methods and Setup

Transmitted Beam Collimation

One of the key difficulties of a lidar system is the proper collimation of the transmitted beam, especially with long-range measurements on small particles as found by the MPL. In such measurements, even a slightly diverging beam can cause significant losses in detecting the backscattered light. For this reason, the MPL transmitter is carefully aligned in the lab as shown in Figure 4.6. To get the best collimation, the laser beam must perfectly overlap the optical axis of the beam expander, at which point its length can be adjusted to collimate the beam.

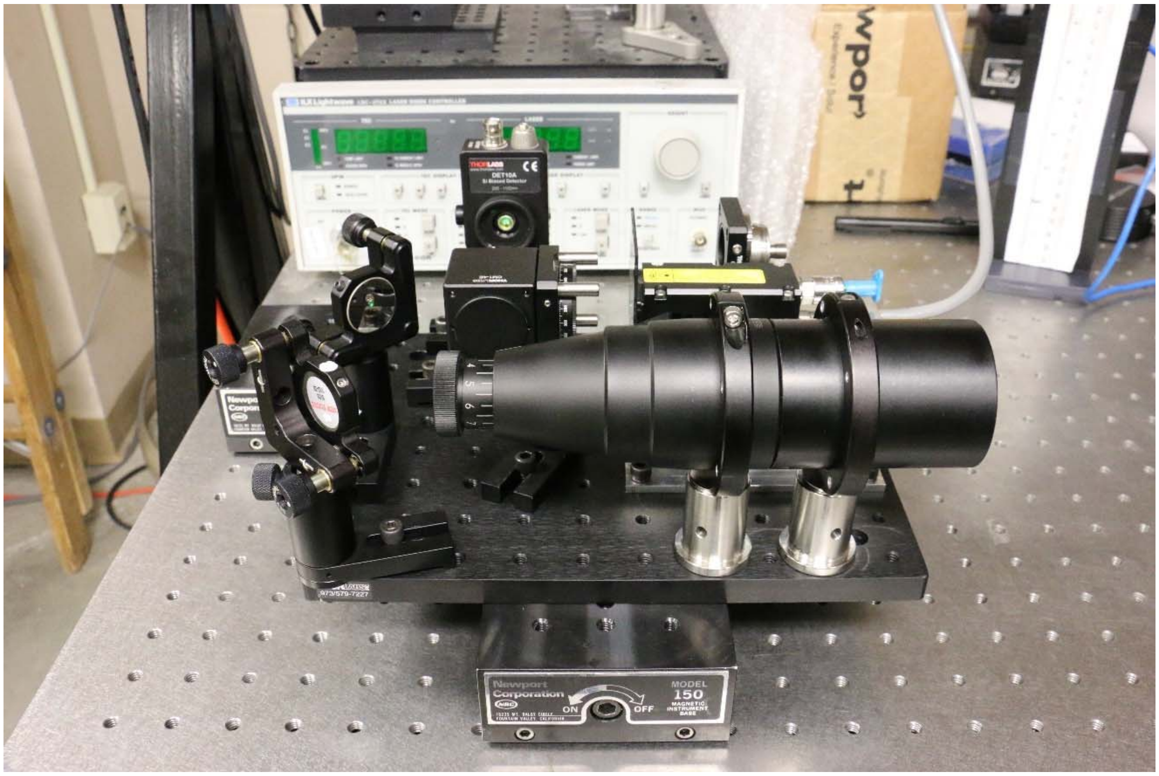


Figure 4.6 MPL transmitted beam collimating procedure performed in the lab.

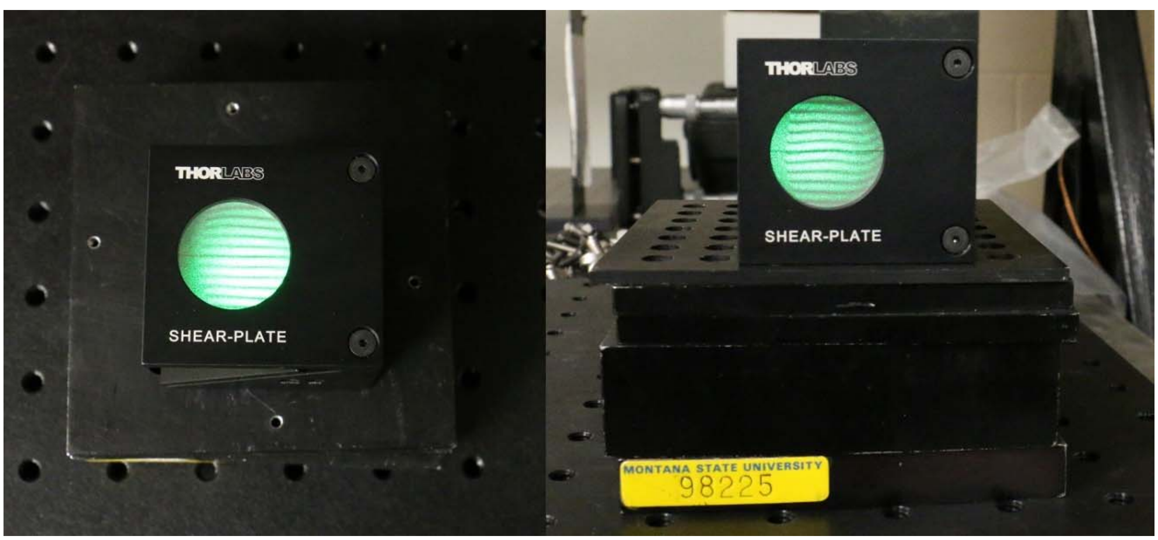


Figure 4.7 Interference pattern of the transmitted beam from the shear-plate in vertical (left) and horizontal (right) axis. The degree of collimation is indicated by how straight and parallel the fringe pattern is to the reference line on the plate.

A Thorlabs shear-plate is used to measure the degree of collimation of the transmitted beam. For a well-collimated beam, the interference fringes are straight and parallel to the reference line on the plate as shown in Figure 4.7. The beam is either converging or diverging if the fringes are straight but tilted with some angle relative to the reference line. In addition to the degree of collimation, the fringes are also sensitive to spherical aberration, coma, and astigmatism, as they can be indicated by the curvature of the fringes. More information about the shear plate can be found on Thorlabs website [45]. It should be noted that the collimation measurement should be done on two orthogonal axes i.e. vertical and horizontal as shown in Figure 4.7, since some of the results can be independent from one another on different axis.

Transmitter-Receiver Alignment

As shown in Figure 4.1 diagram, a periscope is used for making the MPL monostatic in which the transmitted beam is in complete overlap with the optical axis of the receiving telescope. For accuracy, the periscope length is best adjusted such that the transmitted beam exits the periscope mirror on the central axis of the telescope. Once this is achieved, only the transmitted beam angle needs to be adjusted. To set this position correctly, we need an outgoing beam coming out of the telescope for a reference. The quickest and easiest way to get a reference beam is to redirect and use the PBS reflected beam shown in the Figure 4.1 diagram. While this beam is normally used as a triggering signal when the MPL is in operation, it can instead be used as a reference beam through the back of the telescope when the MPL is not in operation. As shown in Figure 4.4, this beam is fiber coupled into a single mode fiber. It can be delivered and injected into the fiber coupler in the back of the telescope as shown in Figure 4.8. This beam will then come out of the front of the telescope as shown in the ring pattern on the white screen in the figure. If this reference beam is not strong enough to be seen, the half-wave-plate in the transmitter can be rotated to increase its intensity. Similar to the transmitted beam, the reference beam also needs to be collimated. To collimate the reference beam, the receiving optics must be carefully aligned in such a way that the beam going through the narrow band filter is collimated, the focusing lens focuses the beam at the center of the iris, then the beam is made to come out collimated by adjusting the telescope focal length. This can be demonstrated by Figure 4.1, but with a thought that the beam is going in the backward direction from the back to the front of the telescope. Once the receiver is well aligned and the reference beam is collimated, the direction of the transmitted beam can be adjusted to overlap the central axis of the reference beam. Beginning at the short distance, the transmitted beam must be set to the center of the reference beam as shown on the screen in Figure 4.8. Then a similar adjustment must be made at longer distances on a hard target. This process is usually done at night when the laser beams are easily seen over longer distances.

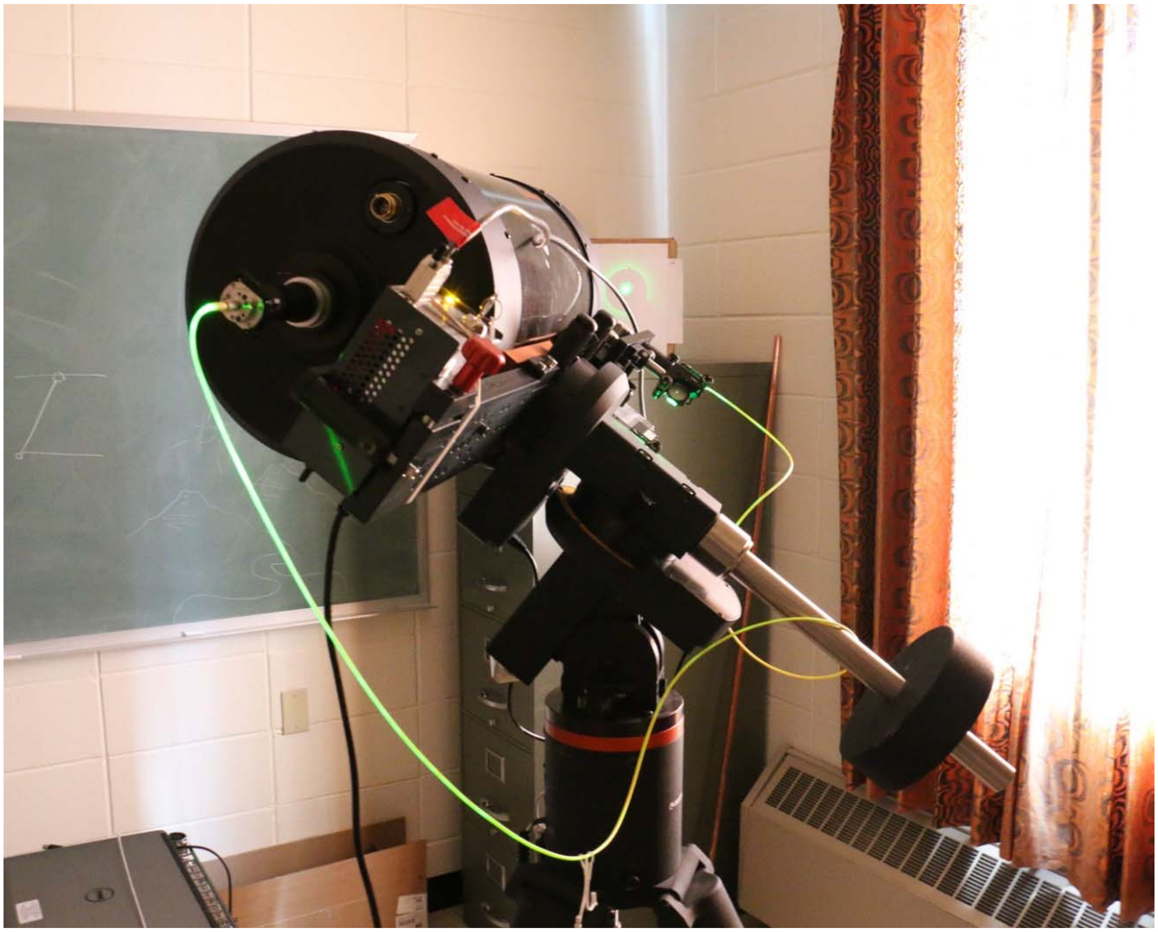


Figure 4.8 Transmitter-receiver aligning procedure.

Horizontal Data Acquisition

The horizontal configuration of the MPL is shown in Figure 4.9. To point the MPL horizontally, run the slant path program shown in Figure 4.5, set dec angle to 180° and bar angle to 90°, then press “move”. After the MPL reaches its position, turn off the mount power switch and stop the program. To acquire the data, run the DAQ program shown in Figure 4.2. Although not required, it is useful to shoot the MPL at a hard target, which can help making the final transmitter-receiver alignment easier. The hard target used in this experiment was the edge of the mountains south of Bozeman.

Figure 4.9 MPL horizontal configuration.

Before taking data, it is always a good idea to perform a final transmitter-receiver alignment to make sure that it is at its optimal alignment. To do this, the periscope exit mirror is adjusted to peak up the real-time ln(N) plot in Figure 4.2 which is enlarged and shown in Figure 4.10. The peak-up location is usually at 2 km where the MPL is in full overlap and the noise level is low enough that the value can still be easily read. If data is taken at night when the noise level is very low, locations beyond 2 km can also be used for the final alignment.

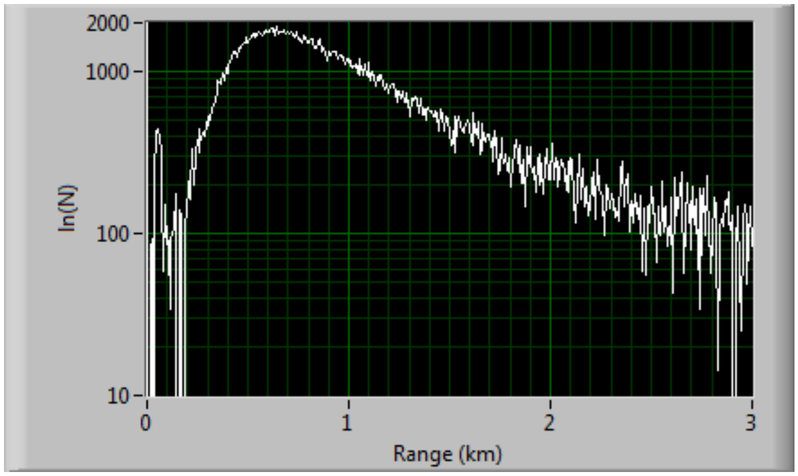


Figure 4.10 Real-time ln(N) plot in the Labview DAQ program used for the peak-up process in the final transmitter-receiver alignment.

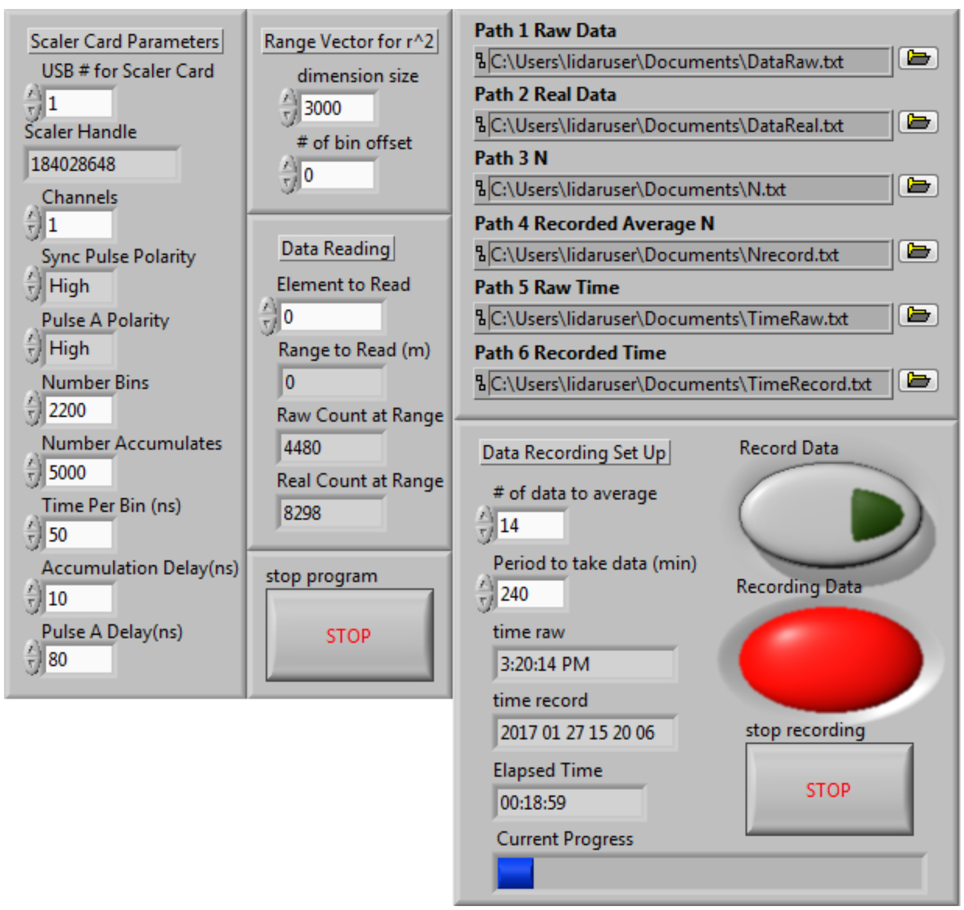


Figure 4.11 MCS settings (left column), data-point reading (middle column), path and filename settings (right column top), and DAQ settings (right column bottom).

After performing the final transmitter-receiver alignment, data recording can begin. Figure 4.11 is an enlarged version of the setting section in Figure 4.2. To record data, we first assign paths and filenames as shown in the figure top-right section. We must make sure that there is no similar filename in the assigned directory. Next we can set the averaging time and recording period as shown in the figure bottom-right section. For example, when the "# of data to average" is set to 14, the program averages the data for every 10 or 11 s before recording it. By pressing the "Record Data" button, the program will start recording data and light up the "Recording Data" bulb to solid red. This will activate the bottom half of the plots in Figure 4.2 which display the time-averaged data that is being recorded in real time. The program will automatically stop recording data once the recording period is met. This is indicated by a full progress bar and an off red indicator bulb. The recording can also be stopped manually by pressing the "stop recording" button. Note that the program is still running even after the program stops recording data. The program can be stopped by pressing the "stop program" button in addition to the stop button on the front panel.

Vertical Data Acquisition

The vertical configuration of the MPL is shown in Figure 4.3 (left). To point the MPL to the vertical direction, run the slant path program shown in Figure 4.5, set dec angle to 52° and bar angle to 180°, then press "move". The rest of the steps are similar to the previous section except for one extra. As mentioned earlier in Chapter 2, when going from the horizontal to the vertical orientation, the gravity sag on the telescope primary mirror has a big effect on the MPL alignment. To account for that, an additional plot used for comparing between the normalized horizontal and vertical data is added to the program as shown in Figure 4.2 top-right plot which is enlarged and shown in Figure 4.12. In the figure, the normalized horizontal data is plotted in red and the real-time normalized vertical data is plotted in white. To help with the comparison, the “normalized factor”, which is the multiplication factor on the normalized vertical data, can be set at the value between 1 to 1.05 depending on the noise level. To ensure that the MPL alignment of the vertical configuration matches that of the horizontal, the rising slope of the white must match that of the red plot. For this to happen, the telescope focal length is incrementally adjusted with the periscope mirror readjusted for the vertical signal in white to peak back up. Out of many plot configurations for various telescope focal lengths, an optimal vertical alignment is selected when the plot in Figure 4.10 is peaked up at 2 km and the white and red plot rising slopes in Figure 4.12 match. To record the vertical data, perform the same steps as described in the previous section.

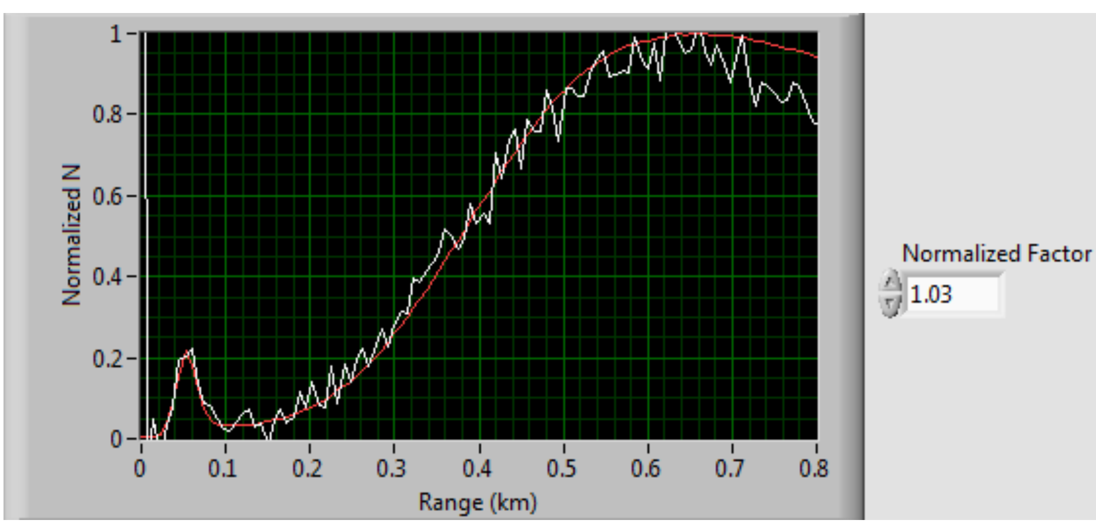


Figure 4.12 Comparison plots between real-time normalized vertical data (white) and normalized horizontal data (red) used in the vertical-horizontal alignment matching.

Results

MPL measurements were conducted at Bozeman, Montana, U.S.A., coordinate N 45° 39′ 43″, W 111° 02′ 42″, elevation 1507 m. Nearly 100 sets of data were taken during the year of 2015 and 2016. 7 data sets—2 in July 2015, 2 in September 2016, and 3 November 2016, are presented in this dissertation. Data sets in 2015 are examples when the MPL is exposed to outdoor temperatures while data sets in 2016 are examples when the MPL is temperature controlled in a room isolated from outside weather. The months of July 2015 and September 2016 data sets are examples of atmospheric pollution due to wildfire smoke over the area. Data from the month of November 2016 provide examples of clean continental atmospheric conditions. Data from two or three days in each of these months is presented to demonstrate the consistency of the MPL.

All of the data in 2015 was taken in an opened roof port room where temperature fluctuations normally cause the MPL to wander off its alignment. It therefore required frequent checkups and signal peak-ups to maintain the alignment. If the system alignment was to wander off too much before the adjustment was made, a discontinuity of data would evidently appear such as in the L/O, β_{total}, and β_a plots on 20 July 2015 around 1015 MDT and 31 July 2015 around 1100 MDT. To eliminate the problem, the system was later moved to a temperature-controlled room with a roof port window where all of the 2016 data was taken. It should be noted that discontinuities such as the one on 30 September 2016 around 1140 MDT is the result of the detector saturation due to the presence of clouds, as opposed to temperature fluctuations.

Lidar AODs retrieved from the MPL measurements are usually in good agreement with AERONET AODs in general except for when the detector is oversaturated such as on 30 September 2016. Examples of lidar and AERONET AOD comparisons are displayed in the AOD plots shown in Figure 4.13 through Figure 4.19. Regardless of whether the atmosphere is clean or smoky, the AODs agree. In smoky conditions such as in Figure 4.13 through Figure 4.16, relatively high values of AOD ranging between 0.05 and 0.12 can be observed, while in clean winter conditions, low values of AOD ranging between 0.01 and 0.04 are observed such as shown in Figure 4.17 through Figure 4.19. In correlation with the amount of AOD, β_a plots indicate thick aerosol boundary layers of about 3-4 km for the smoky conditions, while the layer can be as thin as 1-2 km in clean winter conditions.

Sometimes the MPL provides good agreement between the lidar and AERONET S_a, but oftentimes it does not. Figure 4.13 and Figure 4.14 are examples when the S_a values match. This usually occurs around an S_a value of 50-60 Sr. Other than that, the lidar S_a tends to have values lower than what is expected for the given atmospheric conditions. In smoky conditions such as in Figure 4.15 and Figure 4.16 where the expected S_a is around 70 [46], the lidar S_a is only 30-40 Sr and the AERONET S_a is 80-90 Sr. In clean conditions such as in Figure 4.17 through Figure 4.19 where the expected S_a is around 35 Sr [46], the lidar S_a is only 20-25 Sr on average and the AERONET S_a exceeds a possible value of 100 Sr. Although there is evidence that the AERONET sun-sky photometer is incapable of providing the accurate S_a when the atmosphere is very clean, there is no correlation between the lidar S_a accuracy and the atmospheric conditions. Despite this issue, the MPL still proved to be reliable in terms of total AODs, which agree with AERONET's and correlate well with atmospheric conditions. In addition, it is capable of providing other useful information that the sun-sky photometer is incapable of, such as range resolved AOD, range resolved β_a, and a direct measurement of the aerosol boundary layer.

20 July 2015

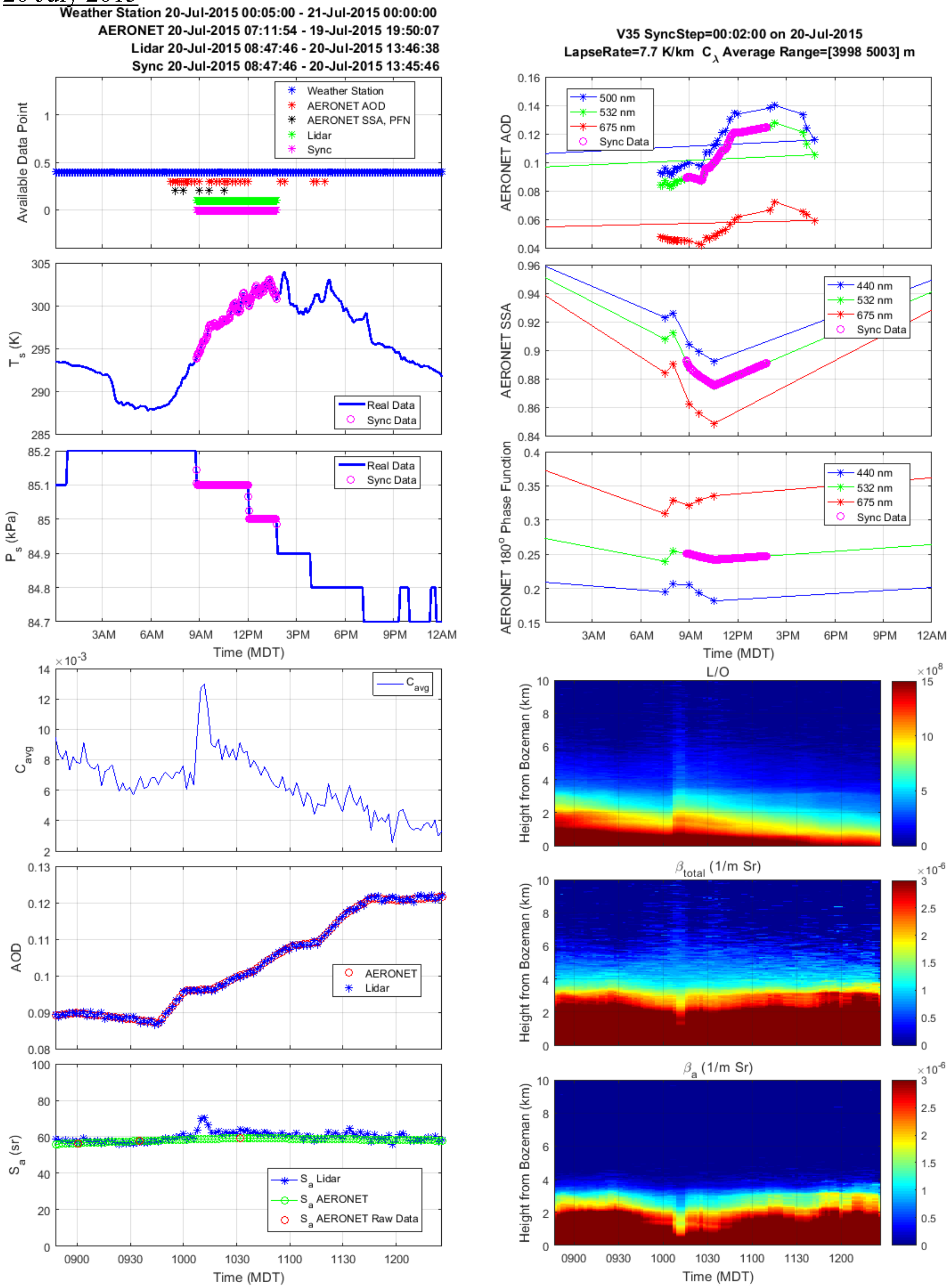


Figure 4.13 Lidar inversion results 20 July 2015.

31 July 2015

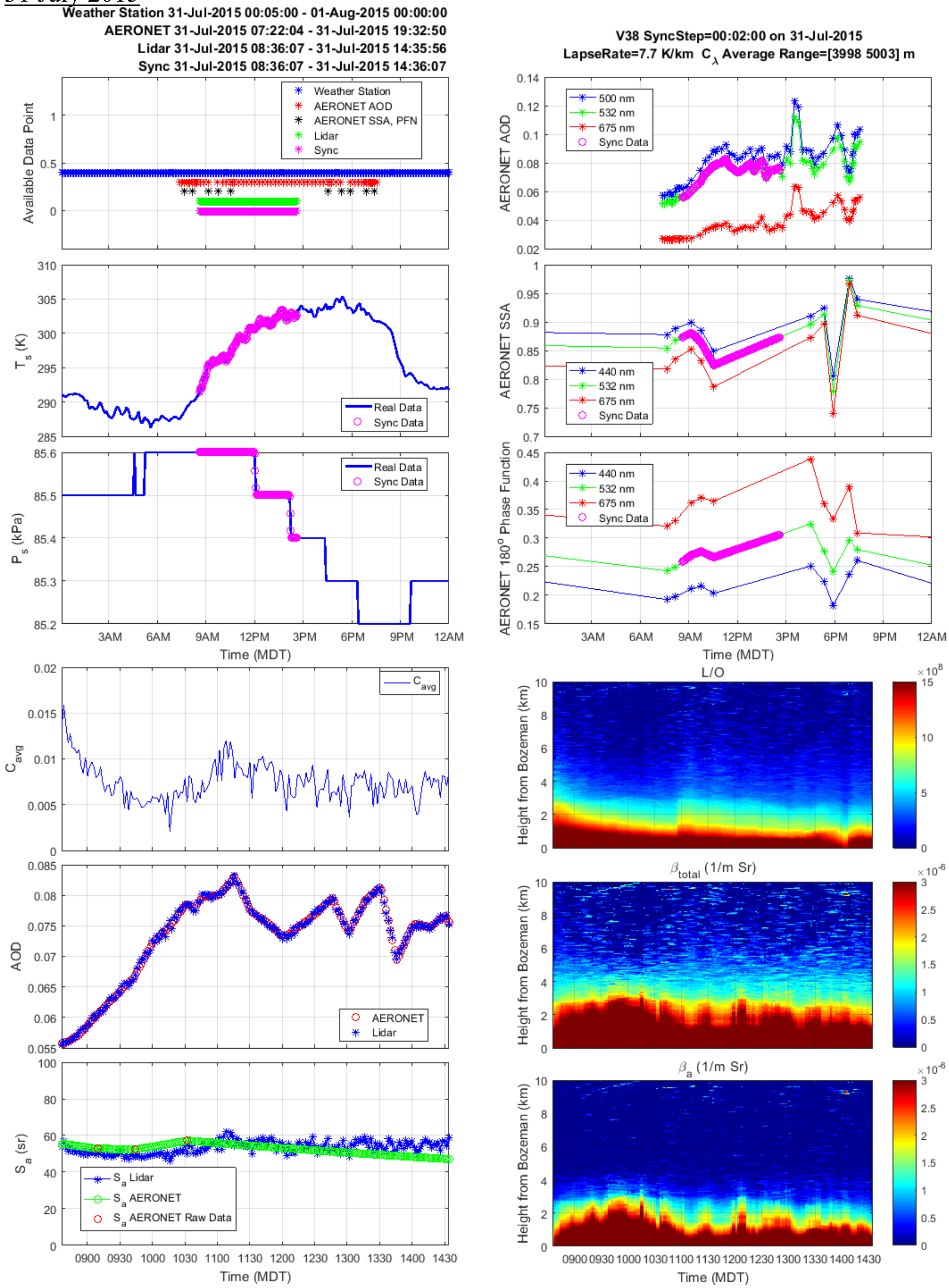


Figure 4.14 Lidar inversion results 31 July 2015.

6 September 2016

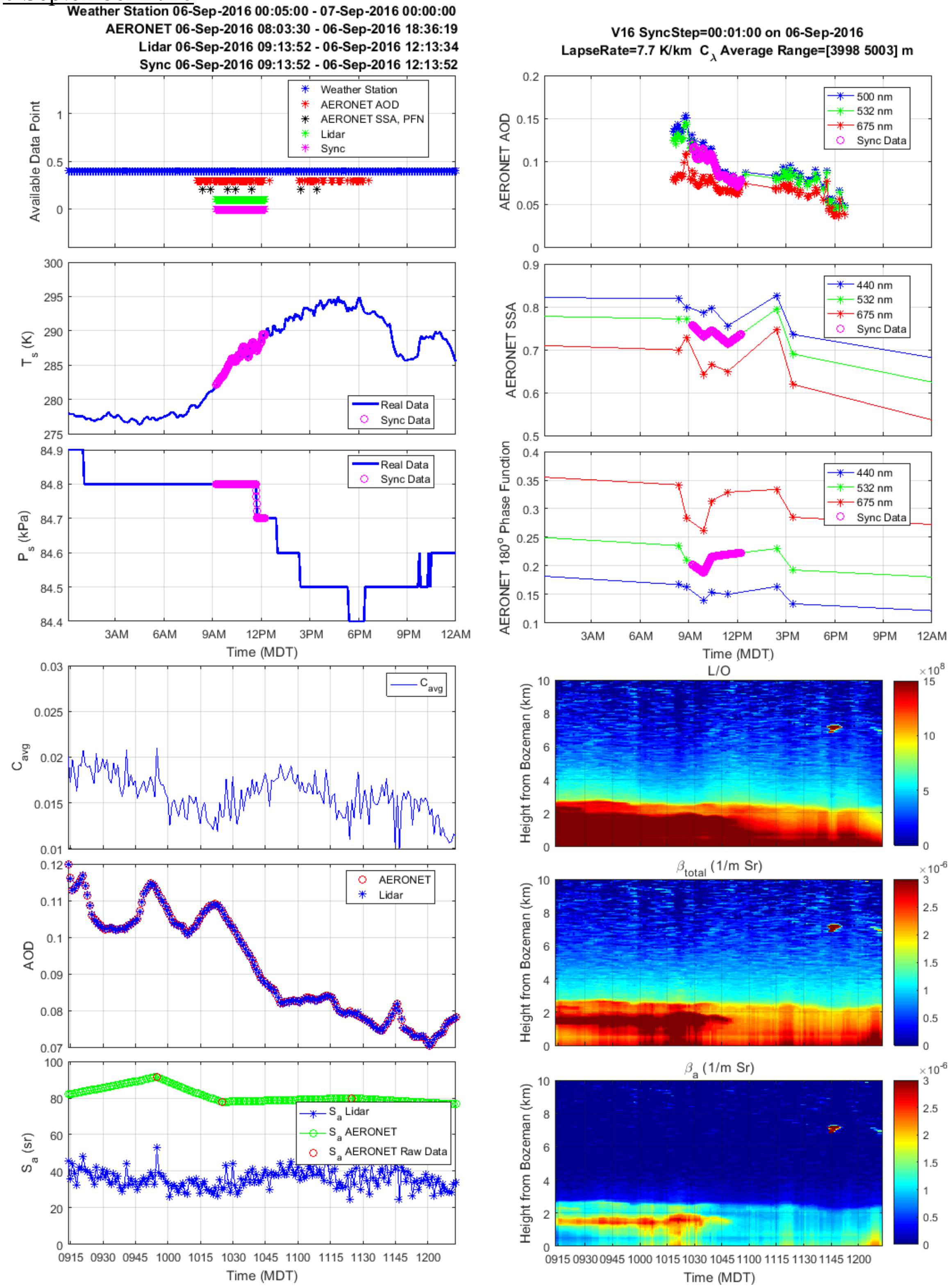


Figure 4.15 Lidar inversion results 6 September 2016.

30 September 2016

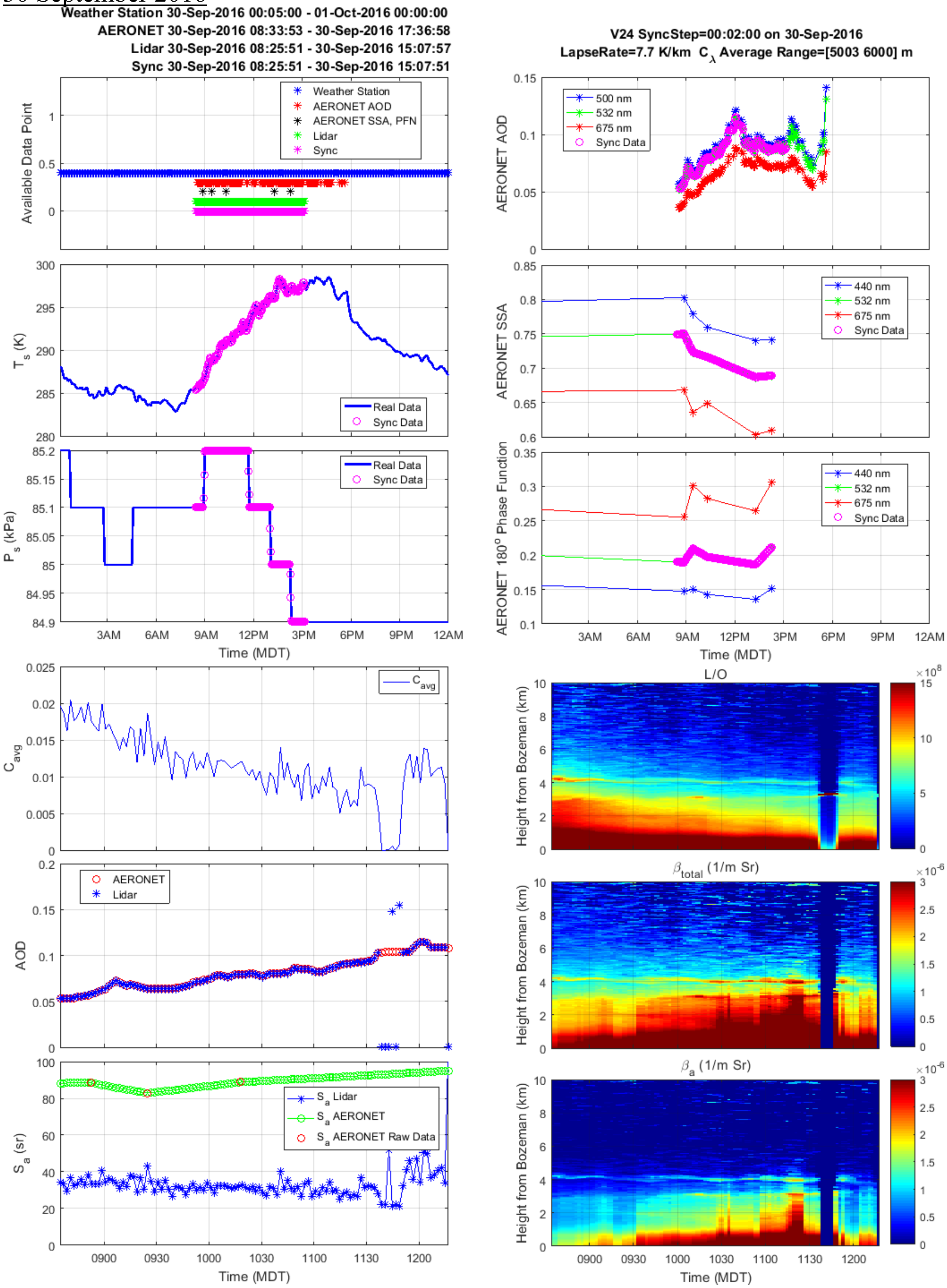


Figure 4.16 Lidar inversion results 30 September 2016.

4 November 2016

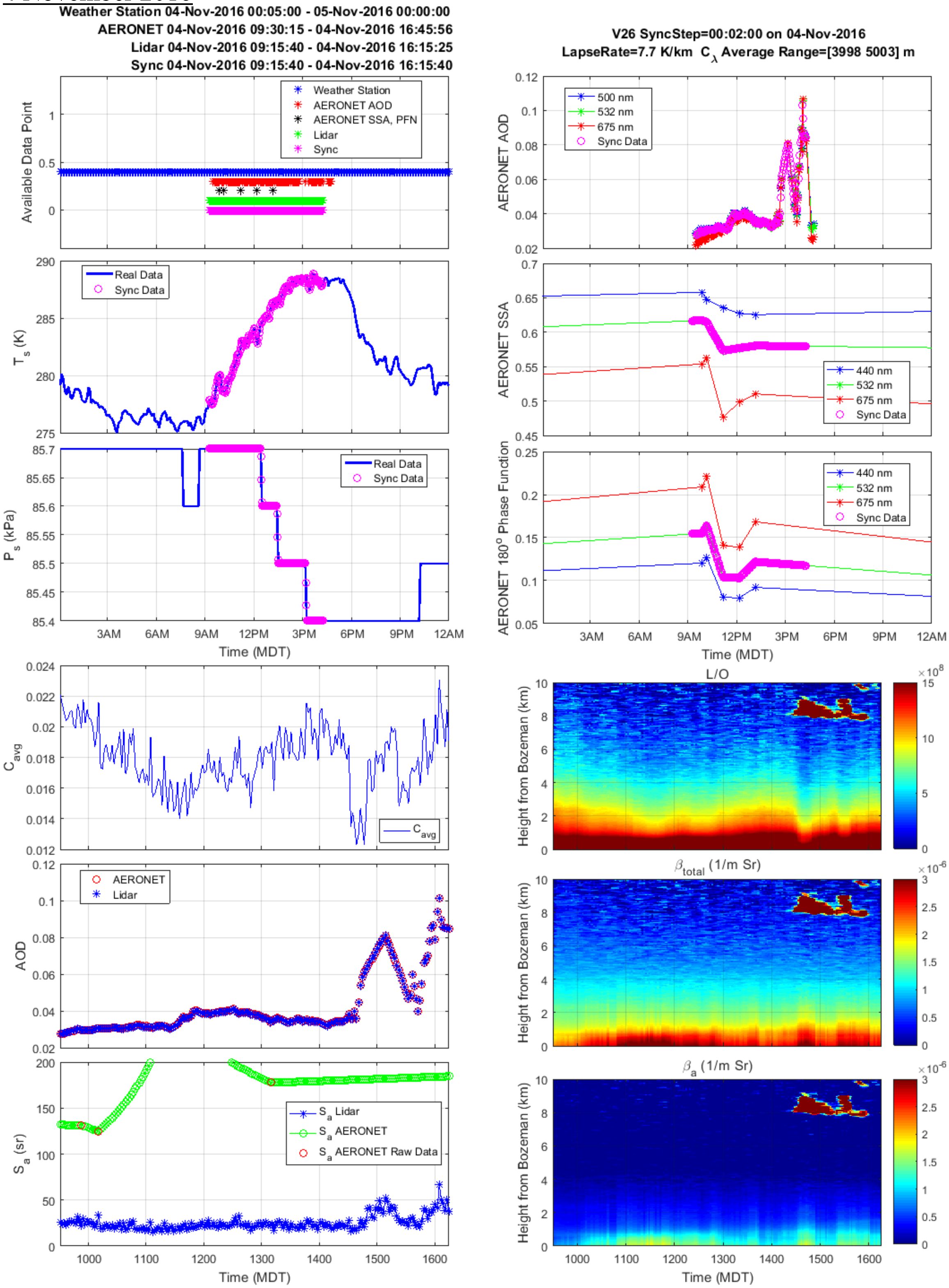


Figure 4.17 Lidar inversion results 4 November 2016.

9 November 2016

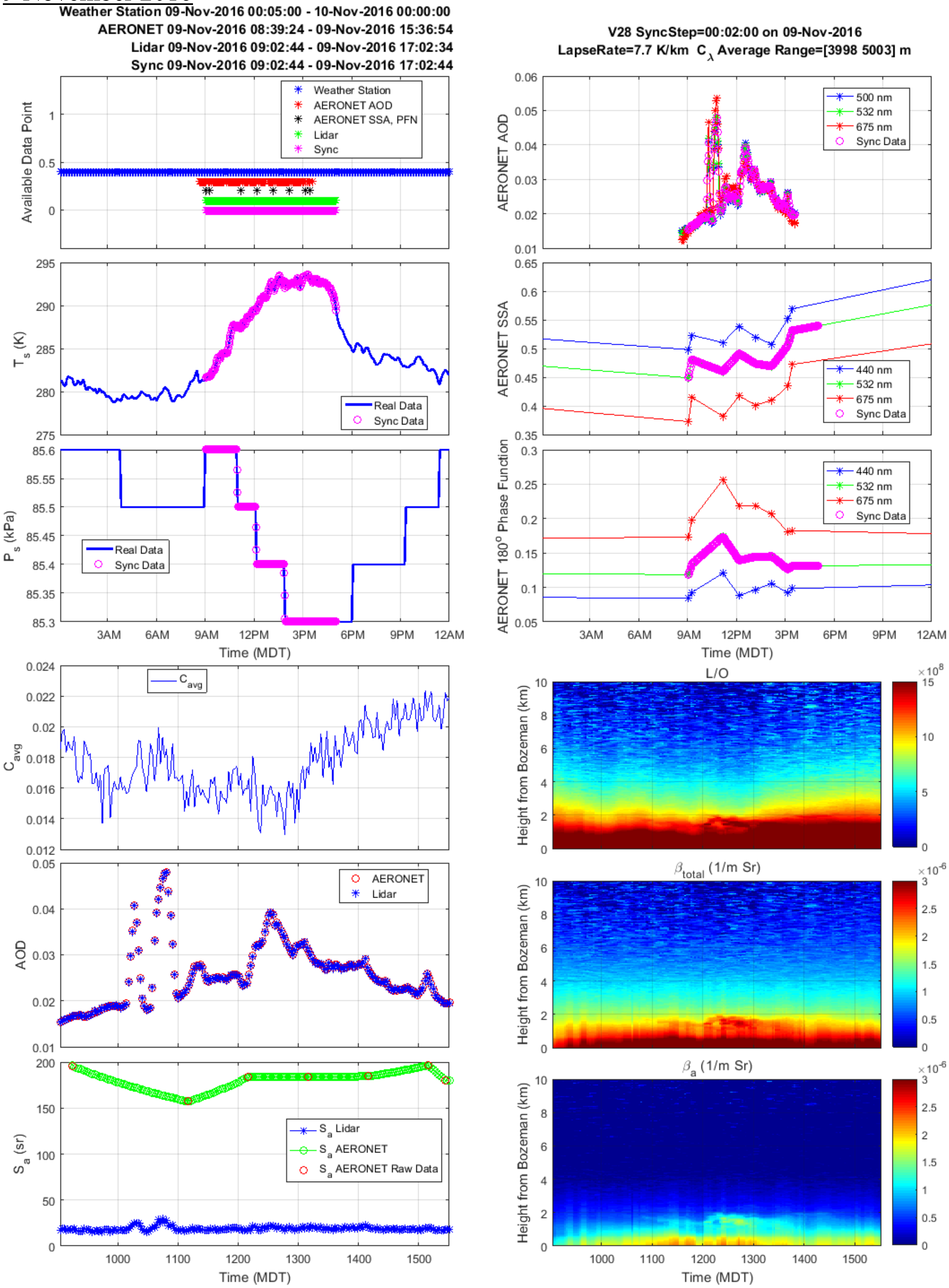


Figure 4.18 Lidar inversion results 9 November 2016.

10 November 2016

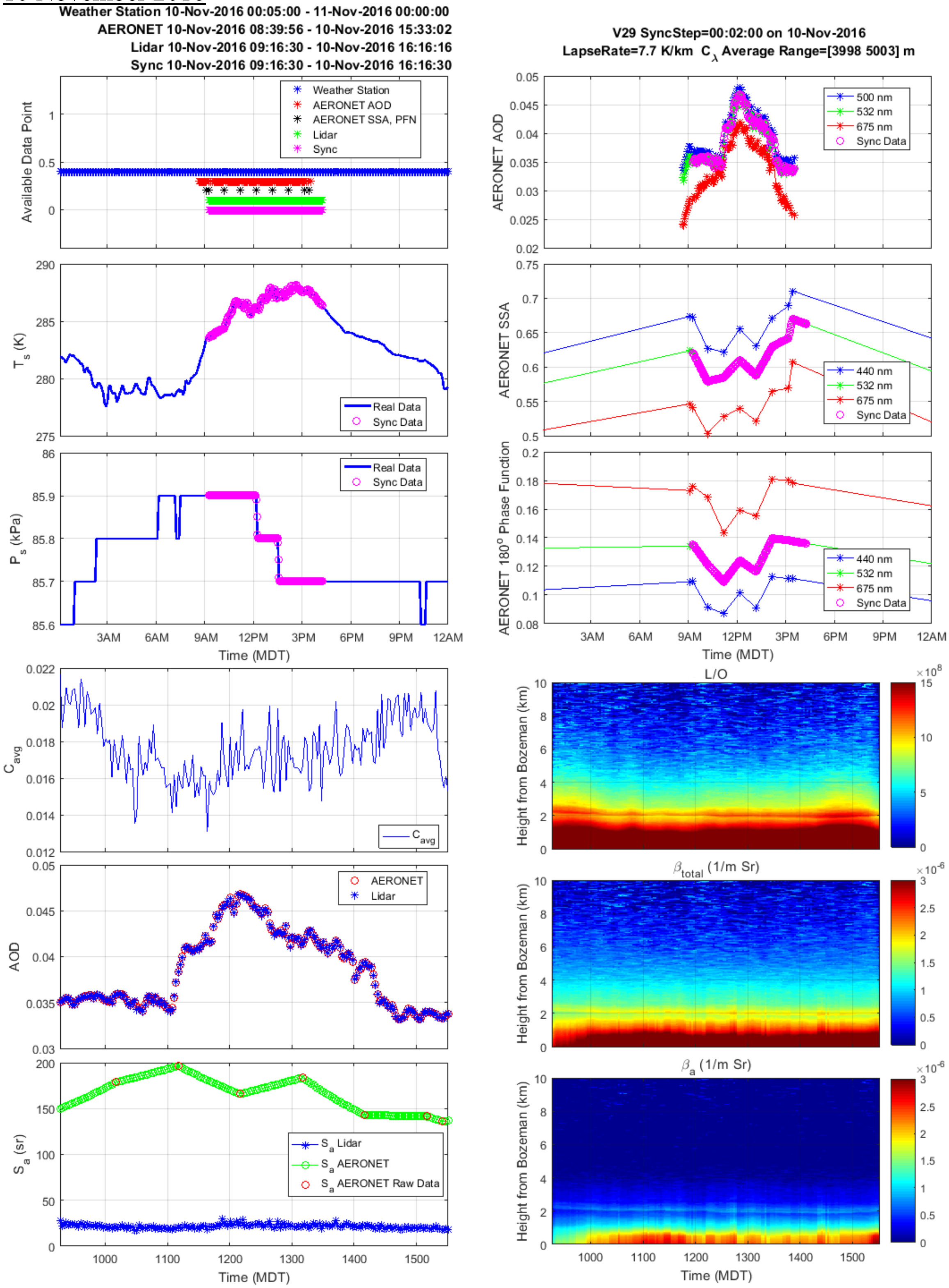


Figure 4.19 Lidar inversion results 10 November 2016.

OPTICAL PARAMETRIC AMPLIFICATION

In the second project of this dissertation, we explore the potential use of the OPO in the CO_2 DIAL transmitter. In the process of building the OPO, the OPA is built and studied which at the same time also allows OPG study. While the goal is focused on the OPO, the understandings of OPG and OPA are also important. Both OPG and OPA measurements help in the crystal selection and QPM process on which the OPO efficiency depends. In addition, by comparing the OPA measurement with the SNLO theoretical model, one can estimate the value of d_{eff} which acts as a gain term, and Δk which acts a loss term, in the system. In addition to describing crystal property, d_{eff} and Δk are also needed as inputs in the SNLO model when trying to predict the OPO performance.

Instrument Design

The experimental setup of the OPG/OPA is shown in Figure 5.1. A Big Sky Laser Ultra series flash-lamp-pumped Q-switch Nd:YAG laser of 1064-nm wavelength with 10-ns pulse duration, 20-Hz PRF, and a maximum average energy of 25 mJ is used as a pump laser. A Q-photonics distributed feedback (DFB) continuous wave (CW) laser of 1571-nm wavelength with 10-mW maximum power is used as a seed laser. The IPG Photonics fiber amplifier is used to amplify the seed power to about 160 mW. By applying a half-wave-plate and a Glan polarizer on both lasers, we can adjust the pump energy and seed power without changing their Gaussian beam properties. If the pump energy or seed power adjustment were done on their controller boxes, the beam's Gaussian properties i.e. shape and wavefront, would change and thus make the system alignment inconsistent. To maximize the system efficiency, the pump and seed beams are designed to be collinear and focused at the same spot within the crystal. In addition to the design, another half-wave-plate is installed before the focusing lens in each beam path to increase the efficiency by allowing us to rotate and match the polarization direction of the beam to the optical axis of the crystal. Two Pellin-Broca prisms are used to separate the output beam by wavelengths. The third focusing lens is designed to focus the output beam near the detector location, otherwise the highly-divergent beam is impossible to detect by the detector. A Thorlabs energy meter model PM100D with an ES111C power detector is used for the measurement of output energy. An optical spectrum analyzer (OSA), Agilent model 86142B, is used for the measurement of output spectra. For measurement accuracy, the detector and OSA are blocked from wavelengths other than 1571 nm by surrounding them with a black box (not shown in the diagram) which has an entrance only through the iris and the band pass filter.

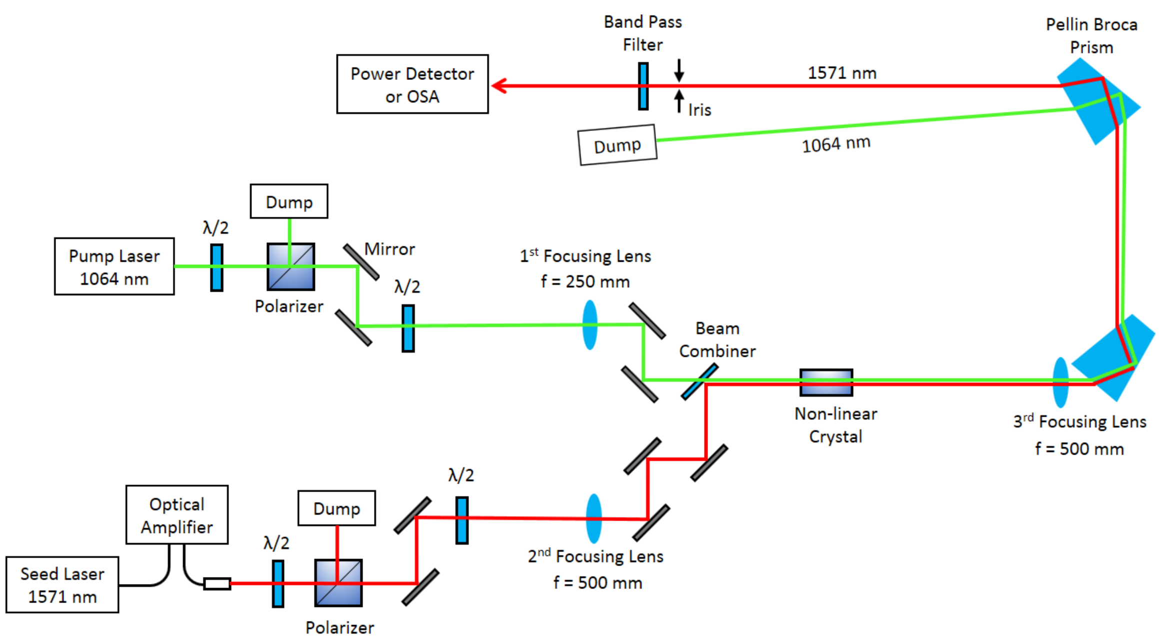


Figure 5.1 OPG/OPA setup (The seed laser is turned off for OPG).

Methods and Setup

Pump-Seed Laser Alignment

Two key successes of the OPA depend on how well the pump and seed beams are collinear and on how well their focuses overlap within the crystal. To select a proper focusing lens for each beam, their Gaussian shapes must be characterized. A Spiricon beam profiler model SP503U-1550 is used for the measurements of beam shape, centroid, and diameter at several locations during the alignment. An example measurement at one location for each beam is shown in Figure 5.2. It should be noted that different filters were used for the pump and seed measurements, therefore they are not intended for intensity comparison, but rather for plotting their Gaussian profiles as shown in Figure 5.3.

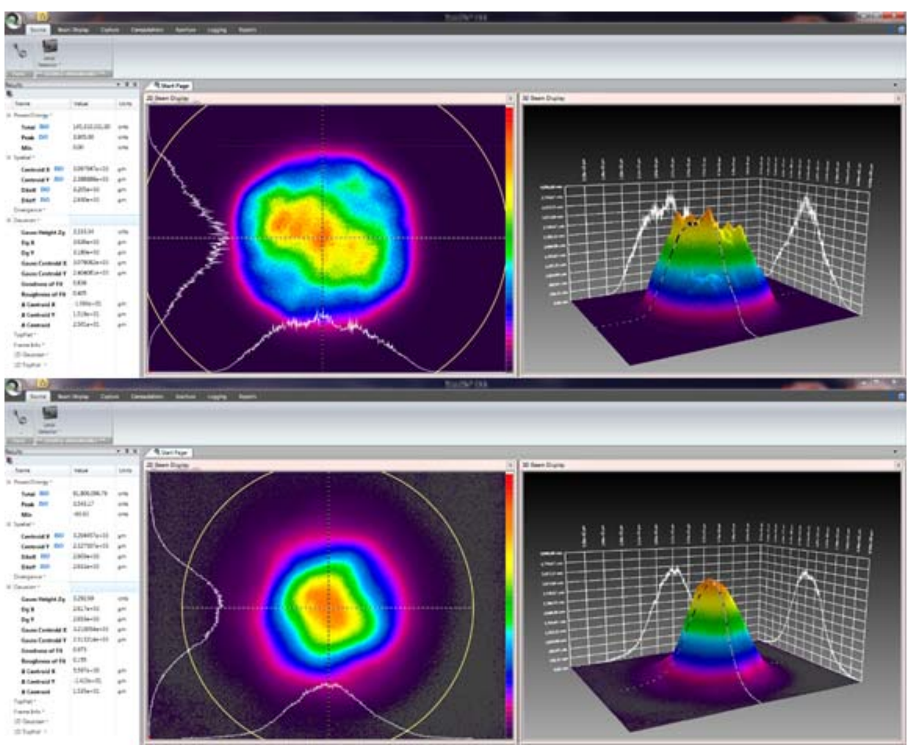


Figure 5.2 Image of the pump beam at 305 mm from the laser source (top) and image of the seed beam at 600 mm from the laser head of the optical amplifier (bottom).

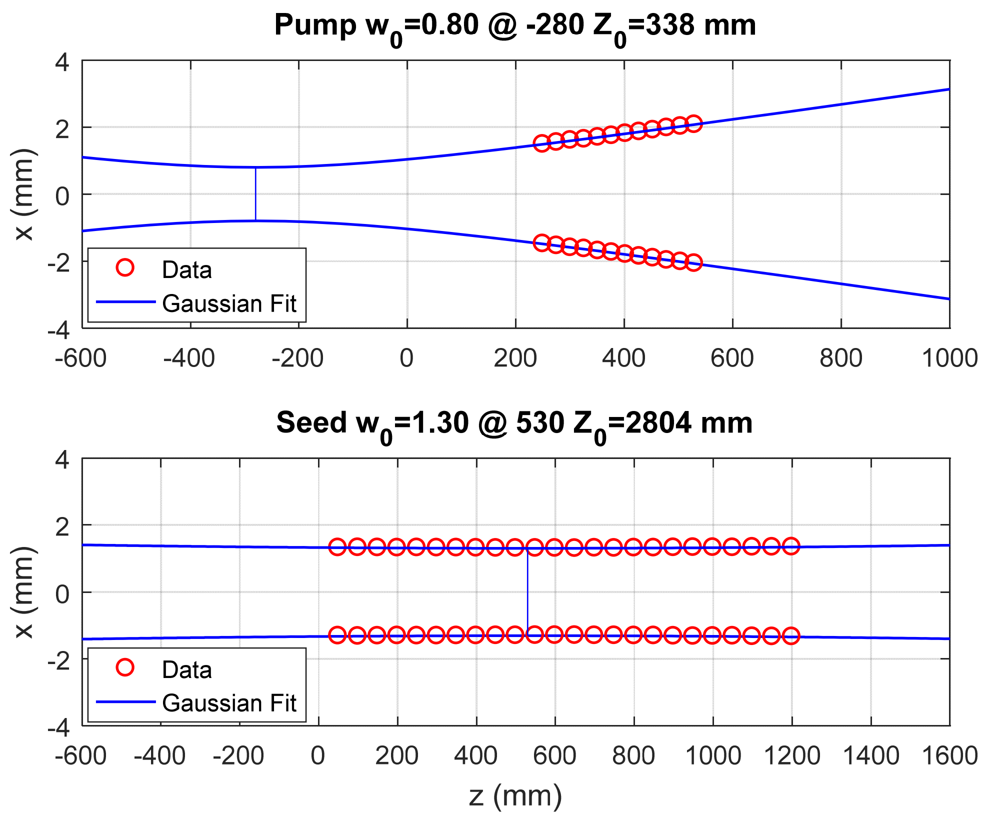
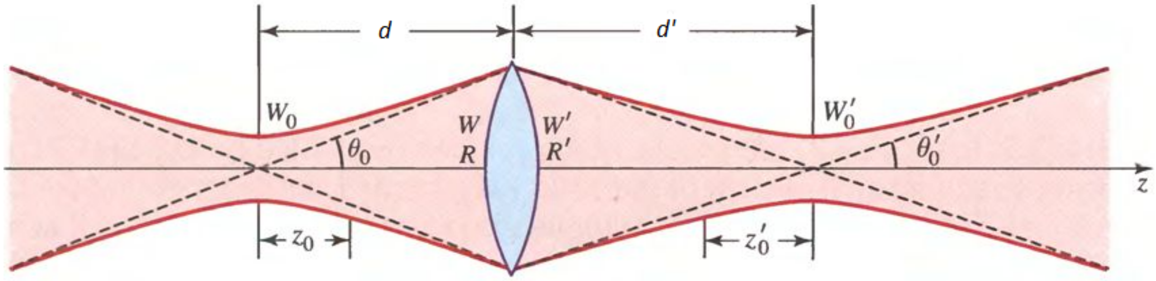


Figure 5.3 Gaussian profile of the pump beam (top) and seed beam (bottom). The laser source for both beams are located at $z = 0$ in each plot. The beam waist is indicated by the vertical blue line.

Because the seed laser is well collimated as shown in Figure 5.3 (bottom), it makes the design for the seed focusing lens very simple, i.e. the beam will be focused at the distance approximately equals to the lens focal length. However, the very Gaussian nature of the pump beam shown in Figure 5.3 makes the design for its focusing lens rather difficult, i.e. it requires more complex calculations than in the case of collimated beam to locate where the beam will be focused. To select the appropriate focusing lens for the pump beam, we must consider all aspects of the Gaussian beam through a lens as described in Figure 5.4.



Waist radius	$W_0' = MW_0$	Magnification	$M = \dfrac{M_r}{\sqrt{1+r^2}}$
Waist location	$(d' - f) = M^2(d - f)$		
Depth of focus	$2z_0' = M^2(2z_0)$		$M_r = \left\lvert \dfrac{f}{z-f} \right\rvert$
Divergence angle	$2\theta_0' = \dfrac{2\theta_0}{M}$		$r = \dfrac{z_0}{z-f}$

Figure 5.4 Gaussian beam through thin lens diagram.

To maximize the system efficiency, we want the new beam to have the optimal waist [47] possible which can be achieved by using an appropriate short focal lens, however we must locate the new waist at a long enough distance for the insertion of other necessary optic components, which requires using a long focal lens. At the same time, we must also ensure that the new beam in not converging or diverging too fast such that the beam size becomes larger than the crystal entrance or exit area which reduces the system efficiency. This is an optimization problem with many variables and constraints on new waist size, new waist location, new divergence angle, and new beam size at the crystal entrance/exit.

Fortunately, the availability of the commercial lens focal length limits the number of lens choices to five. Calculations based on equations in Figure 5.4 for each of the lens focal lengths are performed in a program written in Matlab (see Appendix). In these calculations, the lens is placed at a fixed reasonable location of 800 mm away from the pump laser source. Based on the calculation results plotted in Figure 5.5, the best choice of the lens focal length is 250 mm because its waist remains very small $w_0' = 0.22$ mm while still providing just enough room $d' = 315$ mm for other necessary optic components.

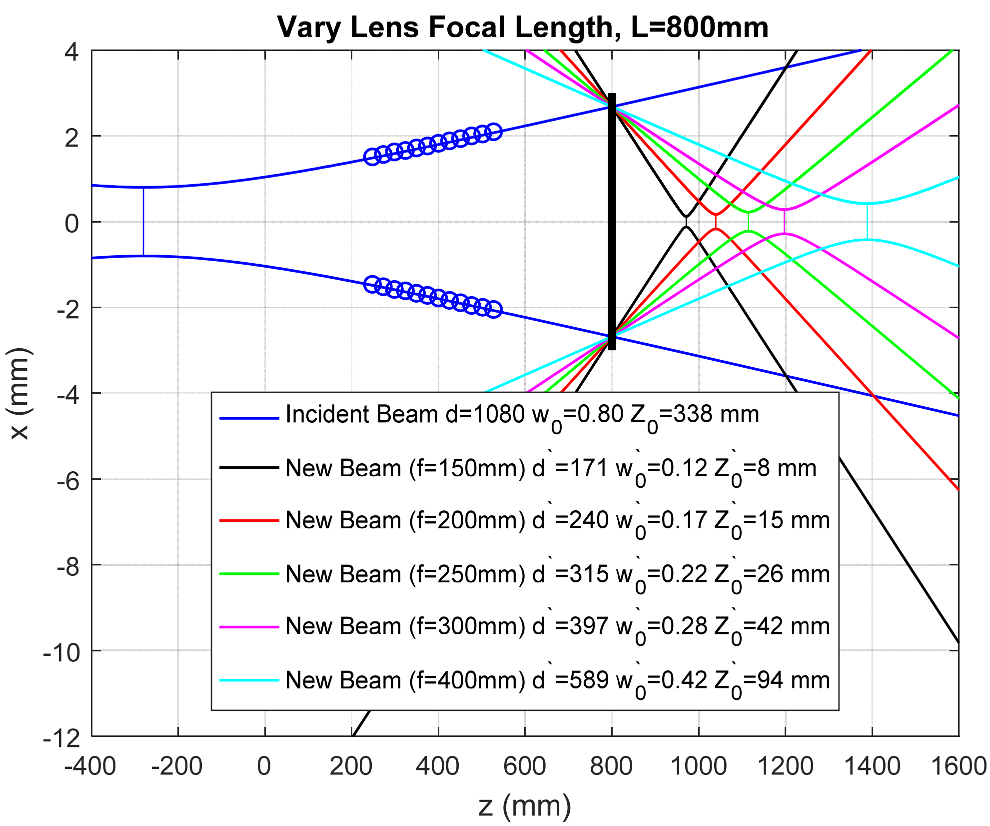


Figure 5.5 Pump beam through lens design with various lens focal length. The lens is represented by the black vertical line.

Once the lens focal length is selected, the lens location should be optimized next. Another Matlab program is written to calculate the Gaussian profile through the 250-mm lens for various lens locations. The results plotted in Figure 5.6 indicate the optimal lens location at 800 mm from the laser source, where the beam waist w_0' and beam size w' at crystal entrance/exit remain small while the system is not too large (with size $L+d'$) and leaving just enough distance d' for other necessary components including two guiding mirrors and a beam combiner as shown in Figure 5.1 and for a future OPO optical cavity mirror. The Rayleigh range z_0' in the top-right plot implies how much the beam diverges i.e. the divergence angle is inversely proportional to z_0'. Ideally, we want the beam divergence to be small enough that the beam size does not become larger than the crystal entrance/exit and yet large enough that the beam waist remains small to maximize system efficiency.

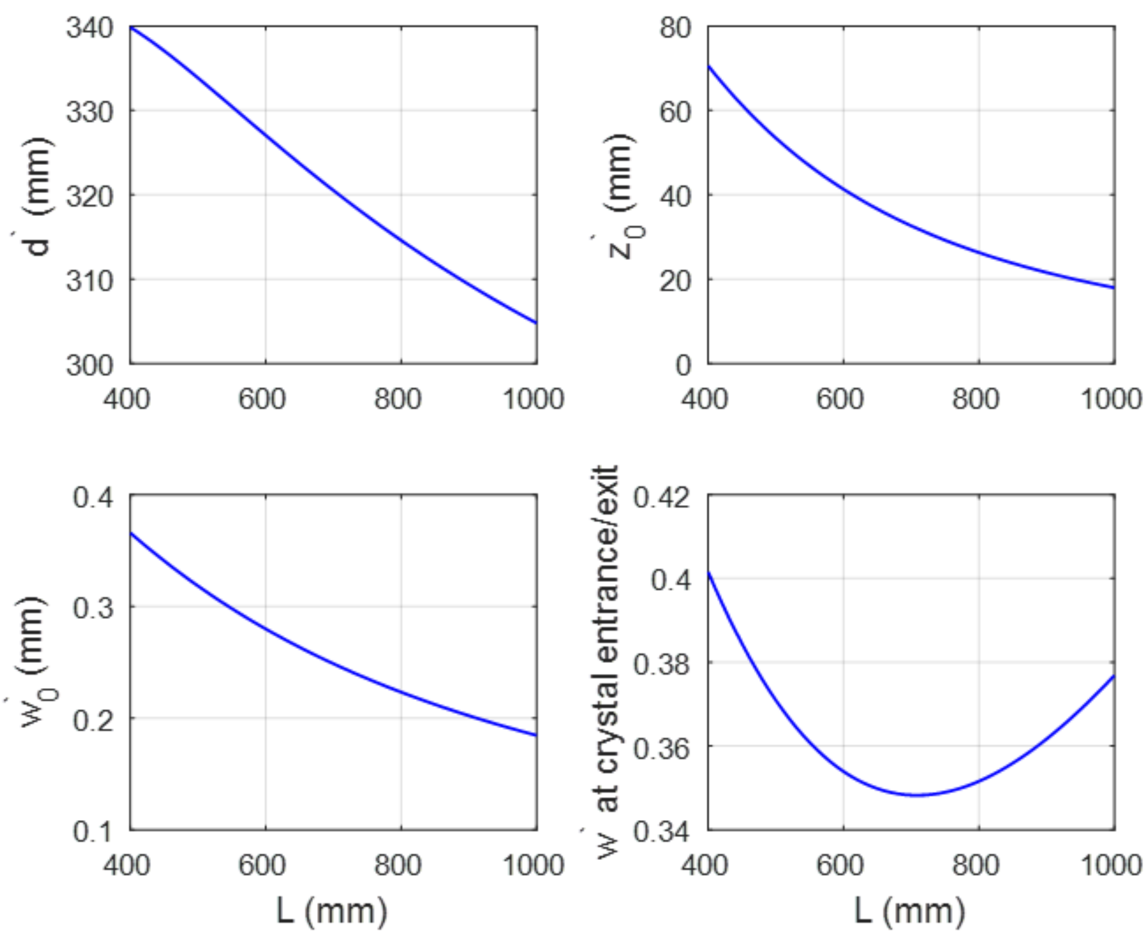


Figure 5.6 Optimization of 250-mm lens locations for pump beam.

After installing the focusing lens for the pump beam and locating its new beam focus, we can now design the focusing lens for the seed beam. In an ideal situation, we want to focus the pump and seed beams at the same location. Therefore, we can use the already-found pump focal location as a reference to back track the calculation for the seed focusing lens. Although the calculation can be easily done backwards, the alignment which needs to be done forwards is still quite a challenge. In the final design, a 500-mm focal lens is selected for the seed beam and is located at an approximately distance of 500 mm from where both beams are focused. The alignment for the pump-seed collinearity and co-focus was done by simultaneously adjusting the seed focusing lens location, three seed guiding mirrors, two pump guiding mirrors, and the beam combiner (shown in Figure 5.1) while keeping the pump focusing lens location fixed.

Crystal Alignment

The nonlinear crystal used in this experiment is a periodically poled lithium niobate (PPLN) crystal which has 5% magnesium-oxide doped with lithium niobate ($Mg:LiNbO_3$) made by AdvR. Since the crystal needs to be temperature controlled as explained in the quasi-phase-matching section, it is put inside a custom built aluminum heater as shown in Figure 5.7 (left). The heater is then attached to the rotation and translation stage which provides 6 degrees of freedom for the alignment as shown in Figure 5.7 (right).

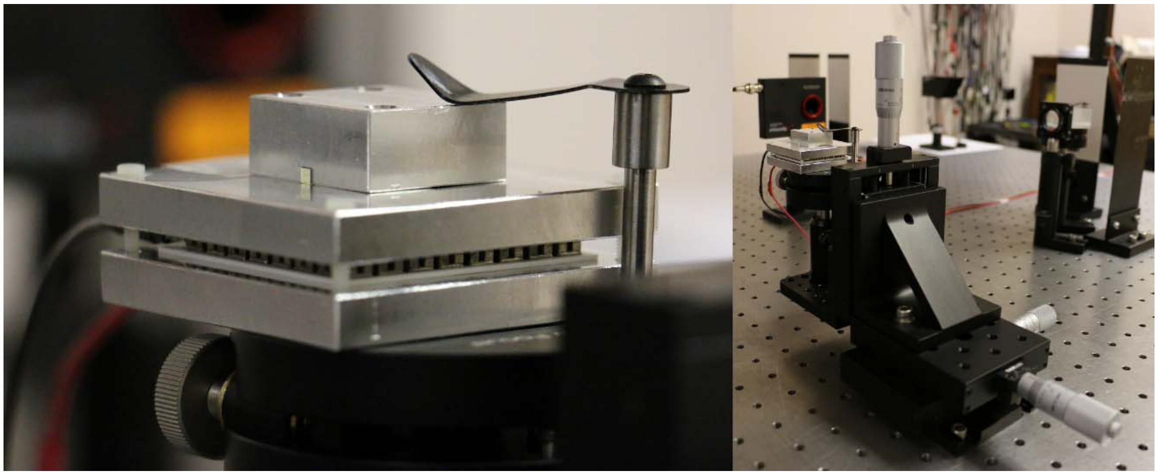


Figure 5.7 PPLN crystal (2×2×32 mm) in the aluminum heater (left). The bottom piece is the heat sink, the middle and top piece are the heater. Between the heat sink and the heater is the Thoslabs thermoelectric cooler (TEC). To ensure minimal temperature gradients within the crystal, the top piece is made to perfectly enclose the crystal with along with a thermistor for temperature response. The rotation and translation stages are used for crystal alignment (right).

The crystal is aligned in such a way that its central axis overlaps with the pump-seed collinear beam and its optical axis is in same orientation as the polarization direction of both beams. The actual measurements of the pump and seed beams in the overlap segment with the Gaussian fits and the crystal position are demonstrated in Figure 5.8. A very good alignment is observed in the figure as the waist locations between the pump and seed beam are only 3 mm apart. It should be noted that the crystal is not placed at the beam waist because the location is reserved for future use when building the OPO cavity which is to be discussed later in the next chapter.

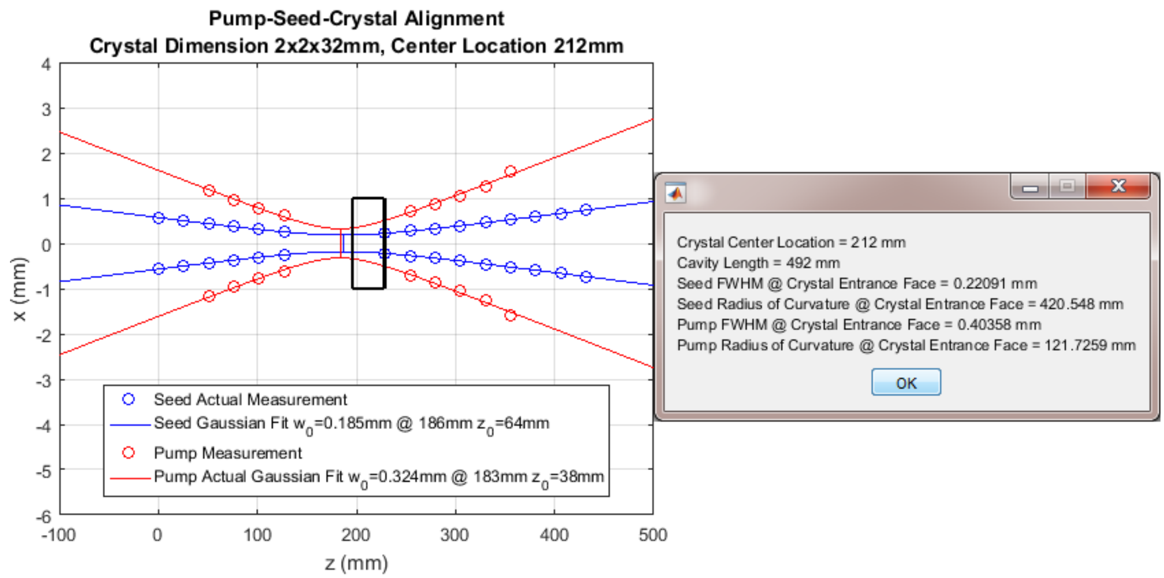


Figure 5.8 Pump-seed-crystal alignment (left) and information needed for SNLO input (right).

Crystal Selection

Three PPLN crystals with poling periods of 31.153, 31.250, and 31.546 μm were available to use for this project. To select a proper crystal that will produce the output signal wavelength of 1571 nm, all three were temperature scanned by tuning a temperature controller that controls the TEC in the heater shown in Figure 5.7 (left). It should be noted that these crystals have different sizes and each has their own custom-built heater top piece shown in Figure 5.7 which is designed to be replaceable without destroying the alignment of the crystal. The scanning uses the setup shown in Figure 5.1 for OPG with the seed laser turned off, and the output signal wavelength is measured by the optical spectrum analyzer (OSA). The scanning results are shown in red, green, and pink in Figure 5.9.

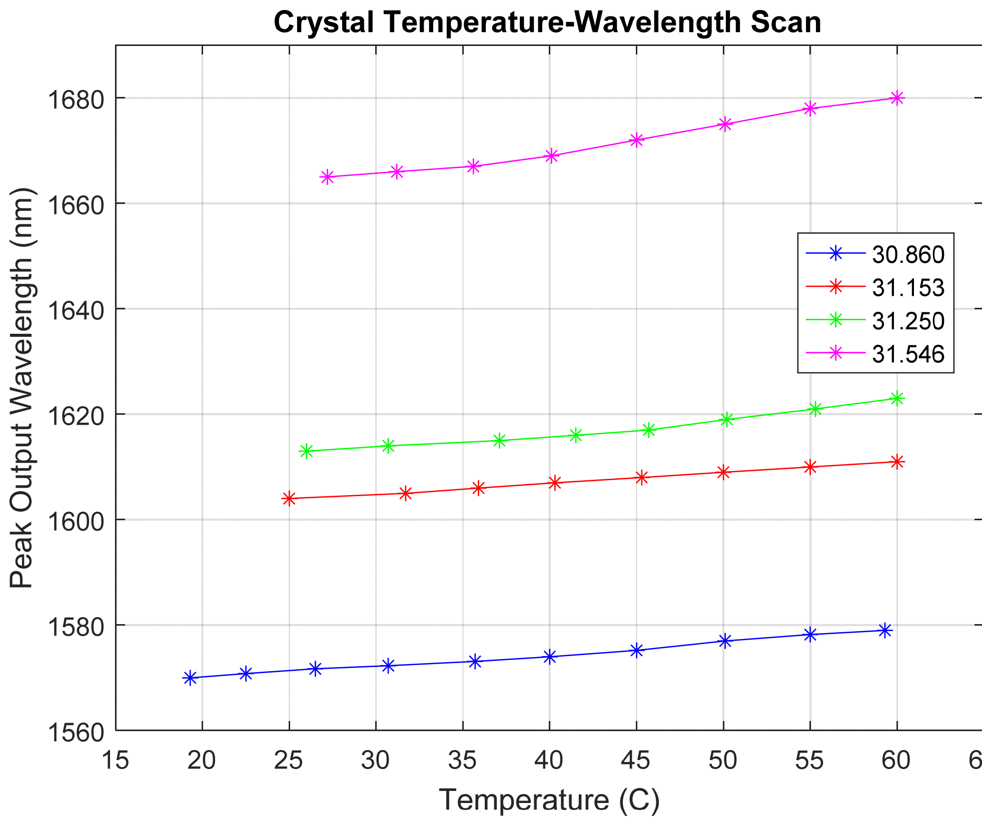


Figure 5.9 Crystal temperature vs OPG output signal wavelength. In the legend is crystal poling period in μm unit.

Although none of these three crystals (red, green, and pink plot) emits an output at 1571 nm, their measurements can be used to estimate the value of the poling period needed for the job. An estimate is found by plotting the poling period of these three crystals against their average peak output wavelength as shown in Figure 5.10. By substituting 1571.406 nm for λ in the linear fit line equation shown in the plot, the poling period Λ is calculated to be 30.943 μm. Based on the commercial availability, the selected crystal has the closest poling period of 30.860 μm with the size 2×2×32 mm and is AR coated. It is characterized as shown in Figure 5.9 blue plot.

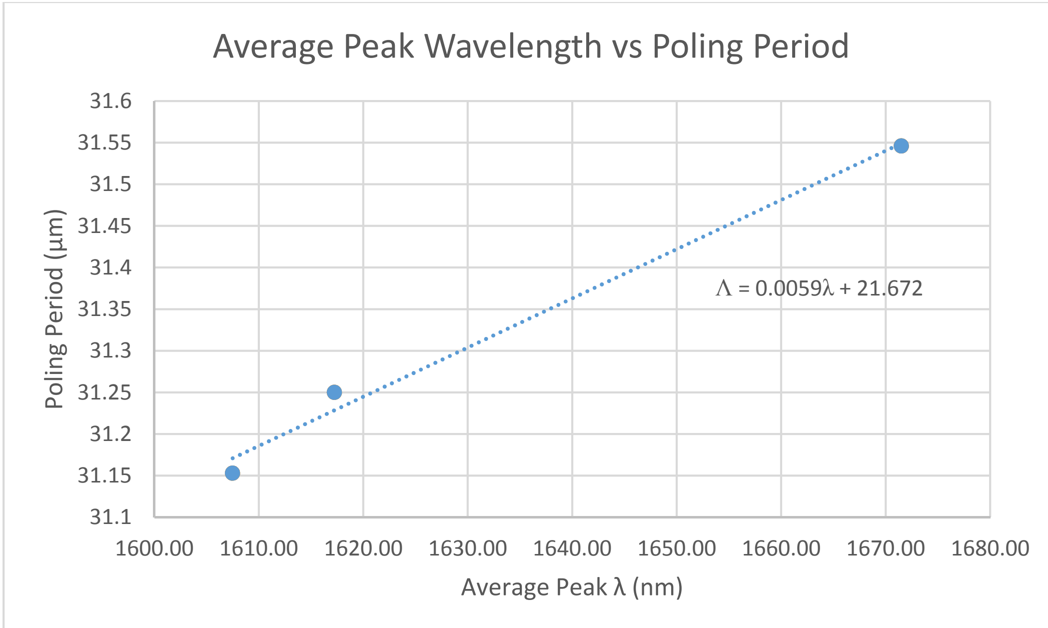


Figure 5.10 Crystal poling period vs their average peak output wavelength.

Results

Output Spectrum

We first take a look at how the system alignment relates to OPG/OPA output spectra. Figure 5.11 demonstrates three cases of output spectra that we can consider—poor spectral overlap, poor spatial overlap, and good spectral and spatial overlap. To maximize the efficiency of OPA, we want both the spectral and spatial overlap to be as good as possible.

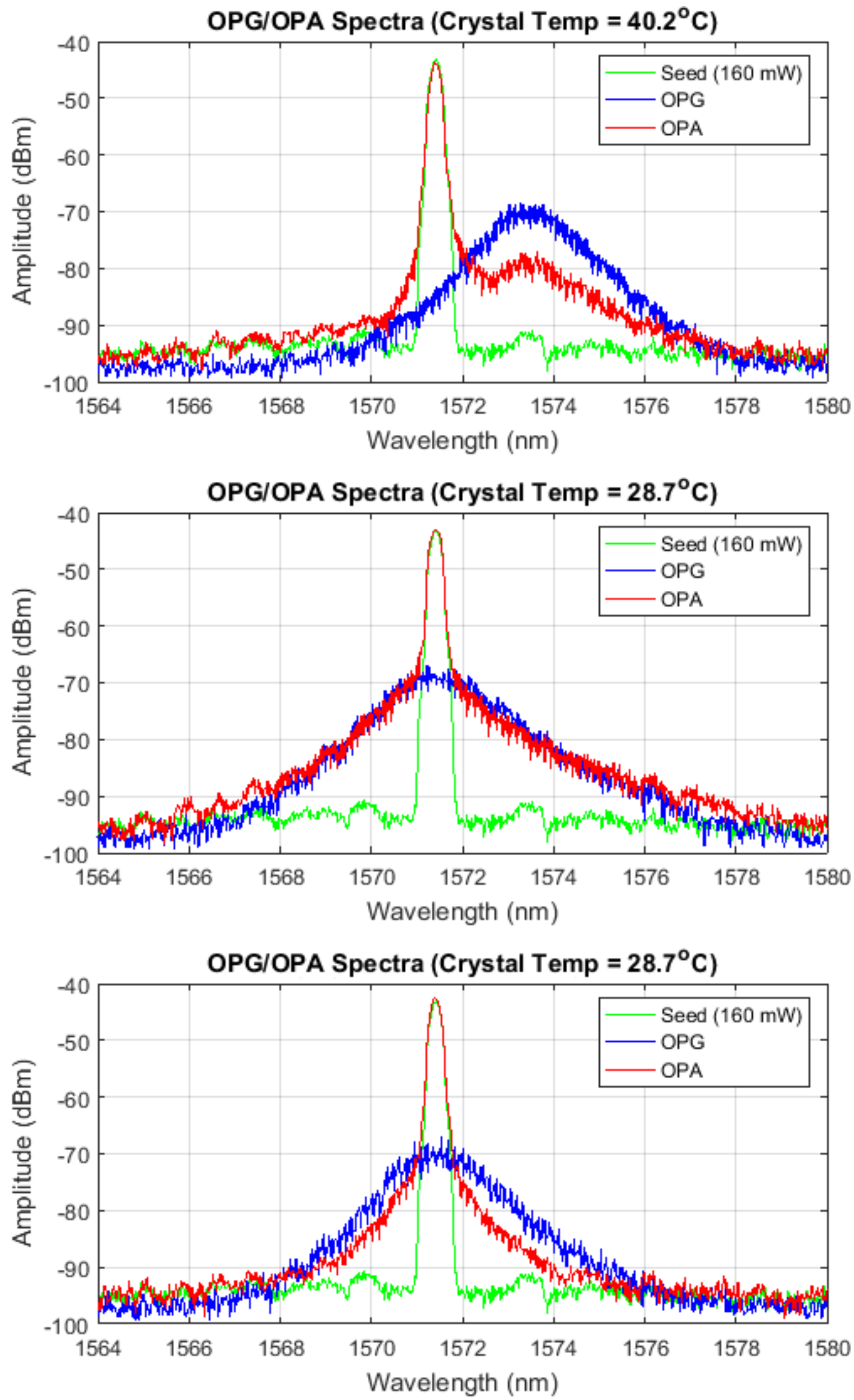


Figure 5.11 OPG/OPA output spectra with poor spectral overlap (top), poor spatial overlap (middle), good spectral and spatial overlap (bottom).

Spectral overlap is the overlap between the peak of the OPG output spectrum and the peak of the seed laser spectrum. Figure 5.11 (top) displays the case of poor spectral overlap in which the peak of the OPG spectrum does not line up with the peak of the seed laser spectrum. It indicates that a quasi-phase-matching condition for the crystal is not quite met. As discussed in Chapter 3, temperature tuning can change the poling period of the PPLN crystal and thus alter the QPM, resulting in different wavelengths of the OPG output spectrum peak as shown in the blue plot in Figure 5.9. Good spatial overlap is observed at the crystal temperature of 28.7 $^{\circ}$C as shown in Figure 5.11 (middle and bottom).

Spatial overlap is the overlap between the pump and seed beams within the crystal. The spatial overlap may be evaluated by comparing the features of the OPG and OPA output spectra. As shown in Figure 5.11 (middle), the sidebands of the OPA spectrum, which is nearly as broad as the OPG spectrum, indicates a poor spatial overlap. Ideally a good spatial overlap should provide OPA sidebands that are narrower than the OPG spectrum as demonstrated in Figure 5.11 (bottom). The broadening of the OPA sidebands in the case of poor spatial overlap is most likely the result of domination of the spontaneous emission over the stimulated emission when the seed is not lined up with the pump beam.

Next we take a look at the effect of the input pump energy on OPG and OPA output spectra in the case where both spectral and spatial overlap are good. As shown in Figure 5.12, the OPG spectrum FWHM (the width at -3 dBm down from the peak) gets broader while the OPA spectrum FWHM gets narrower as the input pump energy increases from 2.75 mJ (top) to 3.56 mJ (middle) and to 5.31 mJ (bottom). The different results are due to different types of emission, i.e. spontaneous in OPG and stimulated in OPA. Although

OPG and OPA linewidths change in an opposite way, their line amplitudes or intensities share a similarity in which they both increase as the pump energy increases. This makes sense because more incident photons should result in more emitted photons, at least before the crystal becomes saturated. Another observable feature is the broadening of OPA sidebands when the pump energy increases. This effect is most likely because the seed power is not relatively high enough to stimulate the gain at higher pump energies. Increasing seed power is expected to narrow down the sidebands again.

It should be noted that the measurements shown in Figure 5.12 cannot be used for comparison between the output intensities of OPG/OPA and the seed. It appears in these plots that the seed intensity is almost as high as the OPA intensity which is not the case. This is because the OSA captures and handles the OPG/OPA outputs, which are pulsed and the seed laser, which is CW, differently. For this reason, a separate output energy measurement using a different instrument is needed to make the comparison.

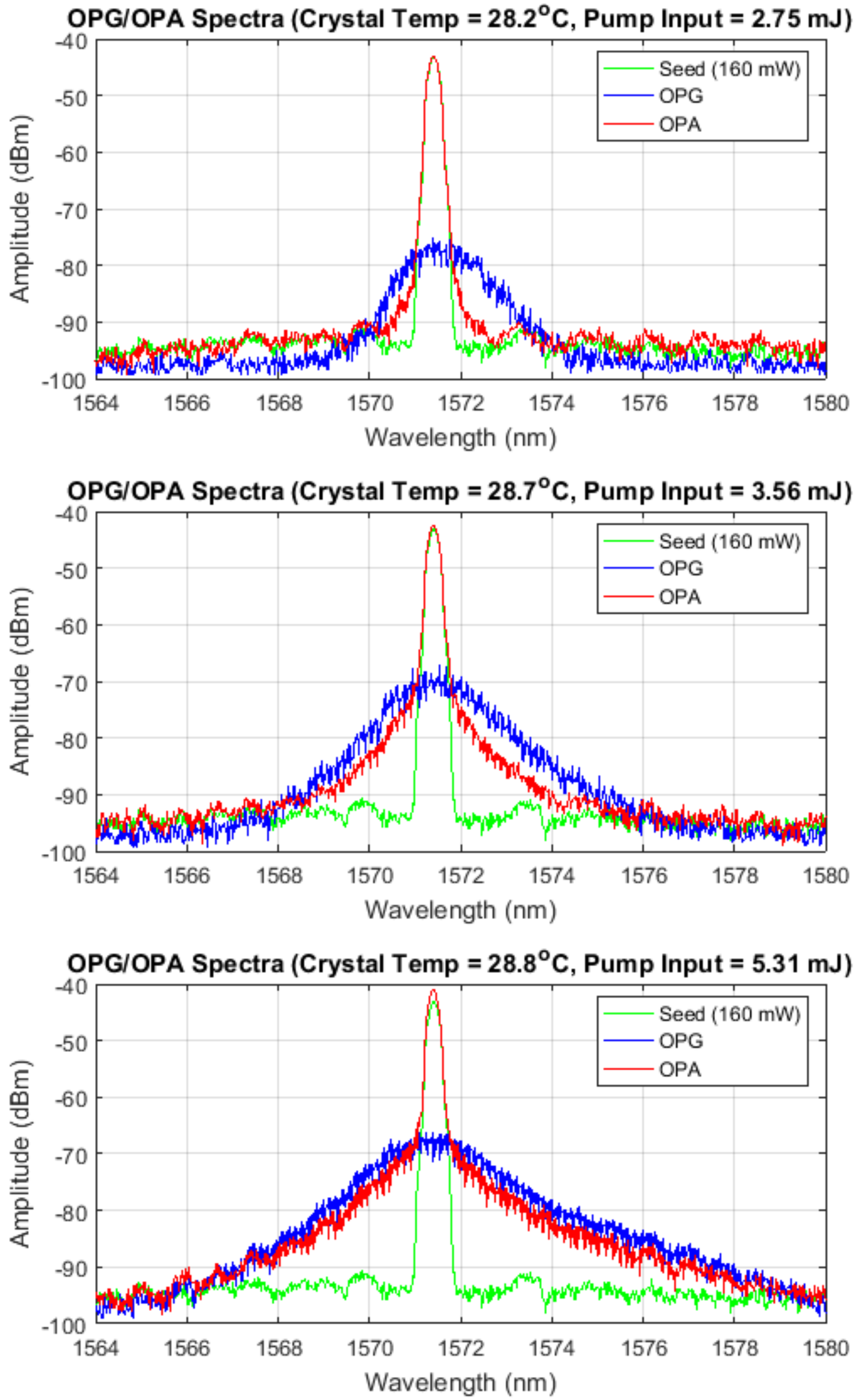
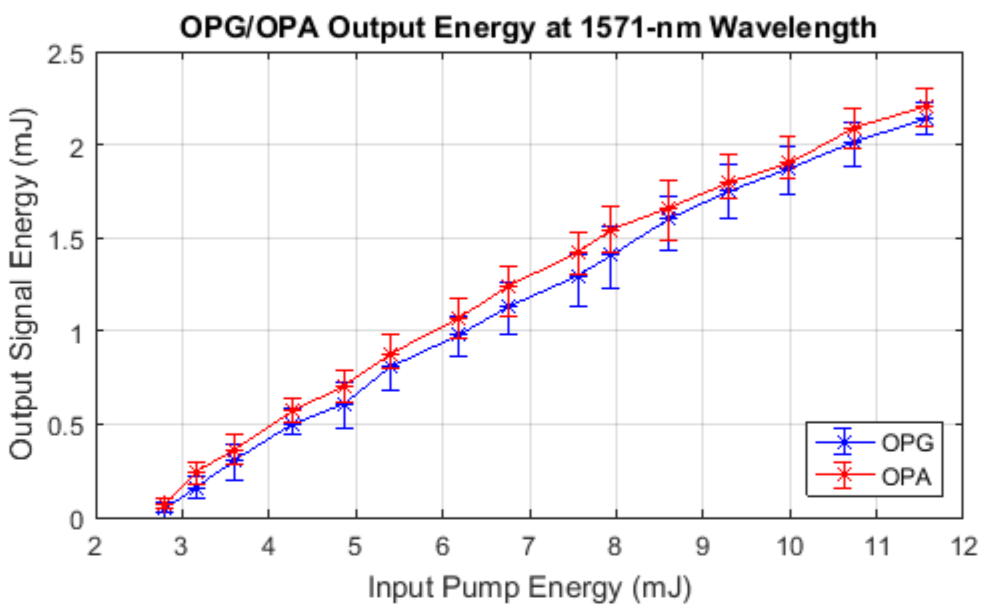


Figure 5.12 OPG/OPA output spectra with different input pump energies.

<u>Output Energy</u>

A Thorlabs energy meter model PM100D with ES111C power detector is used in the output energy measurements which give the results shown in Figure 5.13 below.



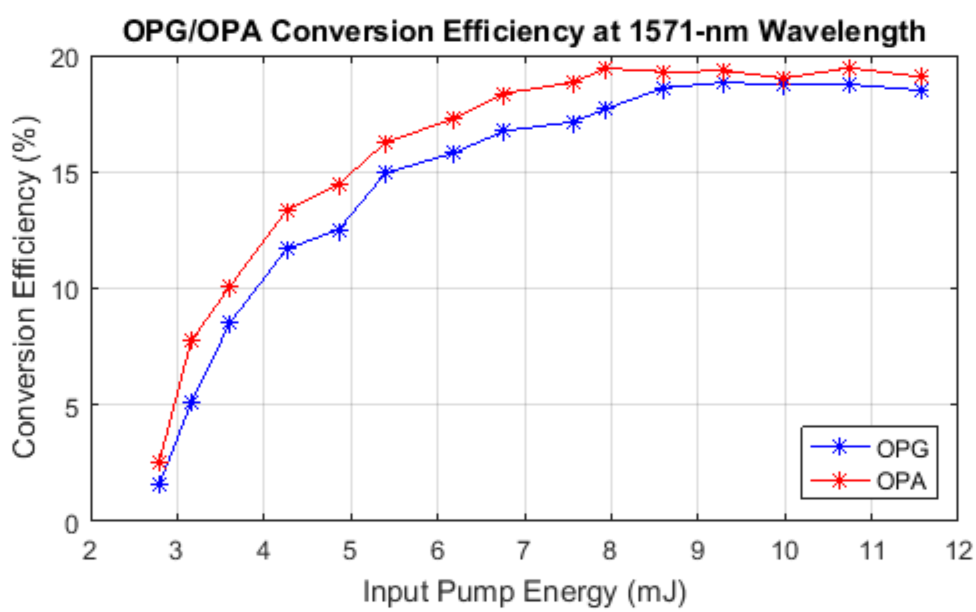


Figure 5.13 OPG/OPA output energy (top) and conversion efficiency (bottom). In the measurements, crystal temperature was 28.8 °C and the seed power was 160 mW.

The output signal energy data in Figure 5.13 (top) is the output at the crystal exit location in which output measurements at the detector location have been corrected by the transmissions of all the optic along the path in between shown in Figure 5.1. The conversion efficiency shown in Figure 5.13 (bottom) is defined as a percentage ratio between the output signal energy and the input pump energy. As expected, on average the output energy in OPA are higher than in OPG. The conversion efficiency reaches a maximum of above 18% and 19% for OPG and OPA respectively. At these rates, a 3-mJ output needed for the DIAL transmitter is possible with roughly 17- and 16-mJ input pump energy. Note that while the crystal damage threshold has not been confirmed by the manufacturer, it is estimated to be around 23 mJ at the wavelength and size of the pump beam. To prevent damaging the crystal and all other optics, all experimental measurements are limited at 12-mJ input pump energy.

SNLO Comparison

To estimate the values of d_{eff} which is a gain term and Δk which is a loss term in the system, the OPA output energy modeled in SNLO is used to compare with the OPA experimental data. Based on the pump-seed-crystal alignment shown in Figure 5.8 and the crystal refractive index contained in SNLO, the SNLO input values for OPA model is shown in Figure 5.14. By varying the input values of the pump energy, d_{eff}, and Δk, SNLO can calculate and predict the output signal energy of the system. The conversion efficiencies are calculated and plotted as shown in Figure 5.15 for comparison. The best-fit result to the experimental data has $d_{eff} \approx 13$ pm/V and $\Delta k \approx 300$ m^{-1} shown as the red curve in the bottom-right plot. In the comparison, the experimental data at low input pump energies of below 3 mJ and at high input pump energies of above 8 mJ were neglected. The reason for neglecting the low pump energies is that the detector cannot accurately detect extremely low output energies and thus produces large errors. The reason for neglecting the high pump energies is because SNLO model does not take the effect of crystal saturation at high pump energies into account.

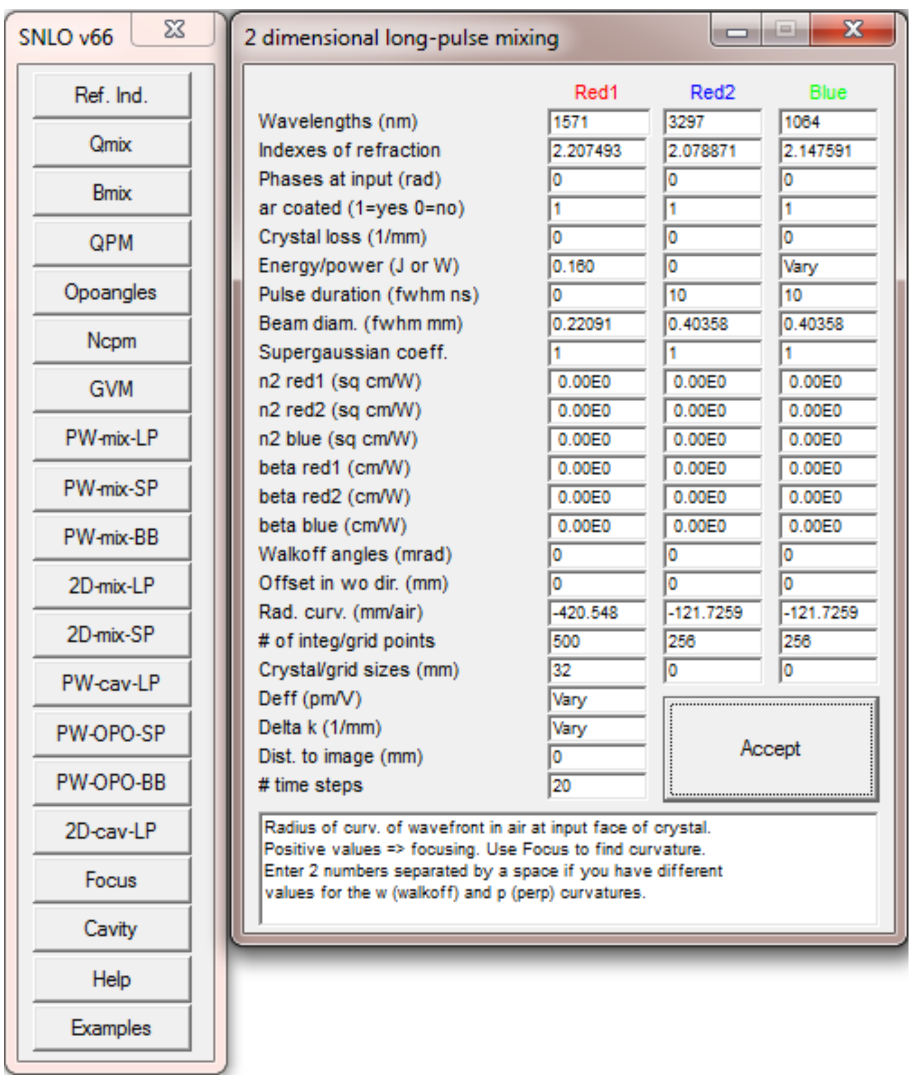


Figure 5.14 Input in SNLO model for OPA output energy.

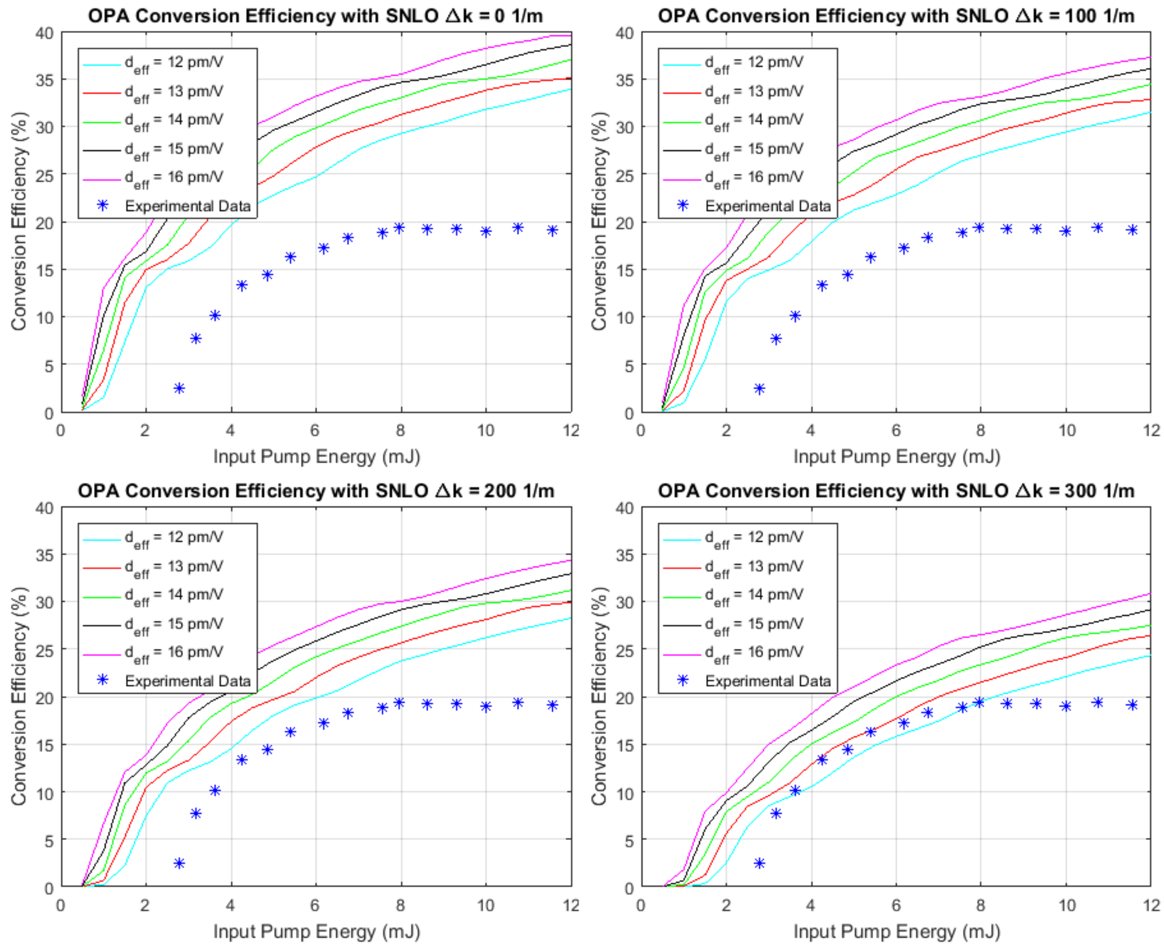


Figure 5.15 Conversion efficiency comparison between OPA and SNLO model with different values of d_{eff} and Δk.

An important point should be made about Δk. One might expect $\Delta k = 0$ when we maximize the output by tuning crystal temperature to adjust for quasi-phase-matching. In the ideal case such as in SNLO, Δk accounts only for phase mismatch; however, in the experiment, Δk is used to account for other losses in the system as well. Some of the losses could be the mismatch of the focusing between the pump and seed beams inside the crystal, the pump-seed overlap imperfection, and the different phase-fronts between the pump and seed beams as they enter the crystal. An estimated high value of Δk of about 300 m^{-1} is actually quite reasonable when taking all these other losses into account. Both d_{eff} and Δk estimates will be used as inputs in SNLO OPO model in the next chapter.

OPTICAL PARAMETRIC OSCILLATION

Instrument Design

The OPO schematic and image are shown in Figure 6.1 and Figure 6.2. In addition to the equipment used in the OPA, the OPO consists of an optical cavity and a locking system. The cavity is hemispherical and is designed to resonate at the seed wavelength of 1571 nm. The locking system uses a dither locking technique to adjust the cavity length that will maintain the cavity at resonance. The locking system uses a function generator (Stanford Research Systems DS345) and an acousto-optic modulator (AOM) (Isomet 1205C-2 modulator with 620C-80 driver) to modulate the seed laser. This modulated signal is then amplified and incident on the optical cavity. An InGaAs detector (New Focus 2011) is used to detect the cavity transmission output and convert the signal into an analog signal. A hold circuit is used to clean unwanted parts of this output signal before it is sent to the lock-in amplifier (Stanford Research Systems SR510). Together with the dither signal from the function generator, the lock-in amplifier can extract the error signal which is proportional to the change in cavity length needed to maintain the resonance. A ramp generator (Burleigh RC-44) is used to control a piezo-electric transducer (PZT) in accordance with the error signal sent from the log-in amplifier. Through contractions and expansions of the PZT in which the cavity mirror is mounted, the cavity length can be adjusted to maintain the resonance. More detail about dither locking is discussed in the next section.

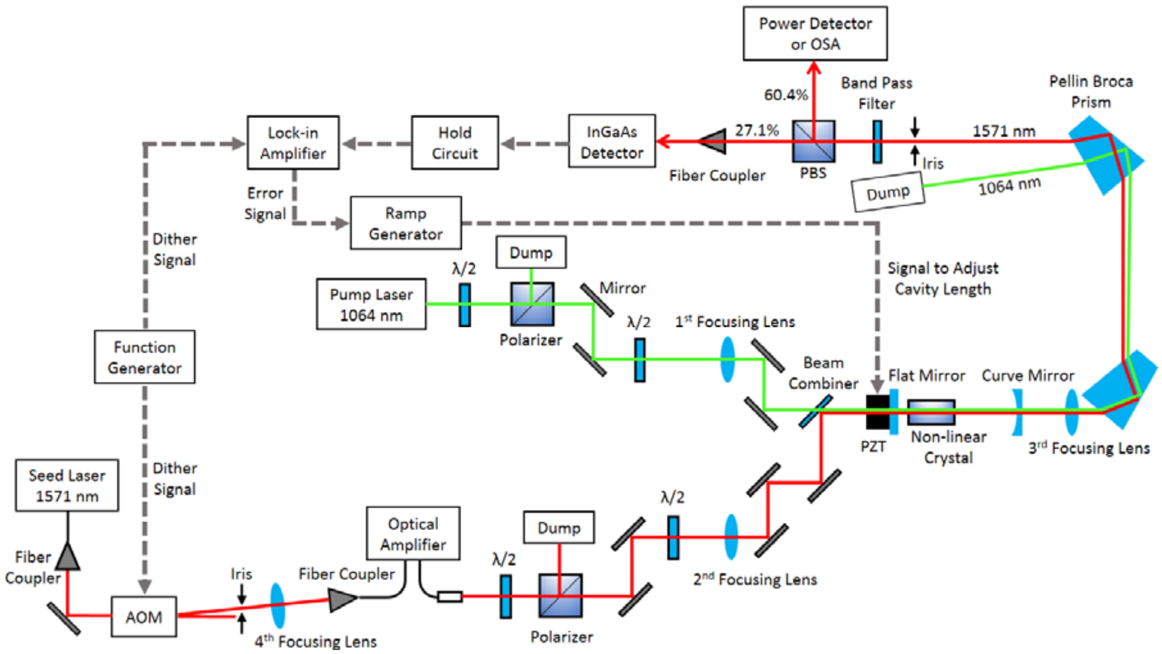


Figure 6.1 Schematic of the OPO with the locking system.

Figure 6.2 Image of the OPO with the locking system.

Dither Locking Theory

Dither locking theory involves a comparison between a dither frequency on the input laser and the frequency of the output laser transmitted through the optical cavity to determine whether the cavity is on- or off-resonance. In this research, the OPO cavity is designed to resonate only at the seed/signal wavelength. This type of OPO where only one wavelength is resonated whether it is the seed, pump, or idler wavelength is called a singly-resonant OPO.

Ideally the cavity can be aligned and maintain the resonant condition. However, in practice, the cavity length always experiences small changes due to temperature fluctuations and noises. These changes even with a tiny length can dramatically reduce the cavity transmission. To put this into perspective, first consider a transmission of an optical cavity which is usually described by the parameters shown in Figure 6.3. A cavity usually has many transmission modes, but only two are shown as the peaks in the figure. The free spectral range (FSR) is the spacing in wavelength or frequency between two successive modes such as peak-to-peak shown in the figure. Depending on whether one looks in the wavelength or frequency domain, the FSR can be represented as either $\Delta\lambda$ or $\Delta\nu$. In terms of frequency, the FSR is given by

$$\text{FSR} = \Delta\nu = \frac{c}{2nL} \tag{6.1}$$

and in terms of wavelength, FSR is given by

$$\text{FSR} = \Delta\lambda = \frac{\lambda^2}{2nL} \tag{6.2}$$

where c is the speed of light, n is the refractive index of a medium inside the cavity, L is the cavity length, and λ is the wavelength in vacuum. Similarly, the FWHM can be expressed either in terms of wavelength $\delta\lambda$ or frequency $\delta\nu$. The ratio between the FSR and FWHM is called finesse which can be expressed as

$$F = \frac{\Delta\lambda}{\delta\lambda} = \frac{\Delta\nu}{\delta\nu} \tag{6.3}$$

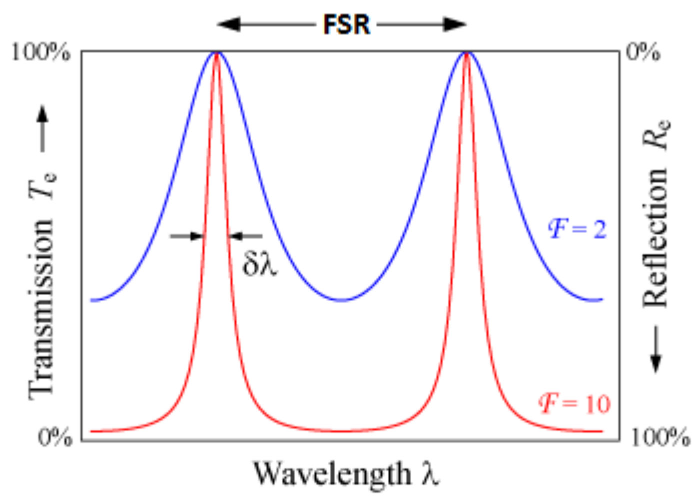


Figure 6.3 Optical cavity transmission.

Next we consider the change in cavity length ΔL needed to shift the resonant mode by a frequency $\Delta\nu$, which is given by [48]

$$\Delta L = \frac{pc}{2n(\nu + \Delta\nu)} - L \tag{6.4}$$

where $p = \text{int}\left(\frac{2nL}{\lambda}\right)$, L is the cavity length, $\nu = \frac{pc}{2nL}$ is the frequency of the longitudinal modes, n is the refractive index of a medium inside the cavity, and c is the speed of light.

To understand how a tiny change in cavity length can seriously affect the transmission, assume a cavity length of 50 cm with a refractive index of 1 and a resonant wavelength of 1571 nm. To determine how much the cavity length needs to change in order to move the resonant mode by half of a FSR, we set $\Delta\nu = \frac{1}{2}\frac{c}{2nL}$. Substituting this into equation (6.4) and doing some algebra, we get

$$\Delta L = \frac{-L}{2p+1} \tag{6.5}$$

which results in $\Delta L = 393$ nm. This shows that even a change in cavity length this small can cause the cavity transmission to go from 100% to near 0%. A change in cavity length of such small magnitude can definitely occur from minor temperature fluctuations or noises near the optical cavity.

To achieve the highest performance, the OPO cavity length needs to be locked at a configuration where the cavity always has maximum transmission. By "locked", it means one of the mirrors is made to translate along the optical axis in small increments in response to a feedback loop that determines which direction the mirror needs to move in order to maintain the maximum transmission. To accomplish this, a sinusoidal signal of frequency ω_d from a function generator is fed into an acousto-optic modulator (AOM) that modulates the input seed laser with a Doppler shift. This creates a dither signal that oscillates around the seed laser wavelength between $\lambda_0 - \Delta\lambda$ and $\lambda_0 + \Delta\lambda$ at a frequency ω_d called dither frequency as demonstrated in Figure 6.4. Here λ_0 is the wavelength of the seed laser which is 1571 nm in the case of this research and $\Delta\lambda$ is the maximum wavelength change due to the dithering. It should be emphasized that $\Delta\lambda$ here does not represent a FSR which is much larger than a FWHM as shown in Figure 6.3, but rather another quantity which is much smaller than the FWHM of the resonant peak of the cavity as shown in Figure 6.5 (left).

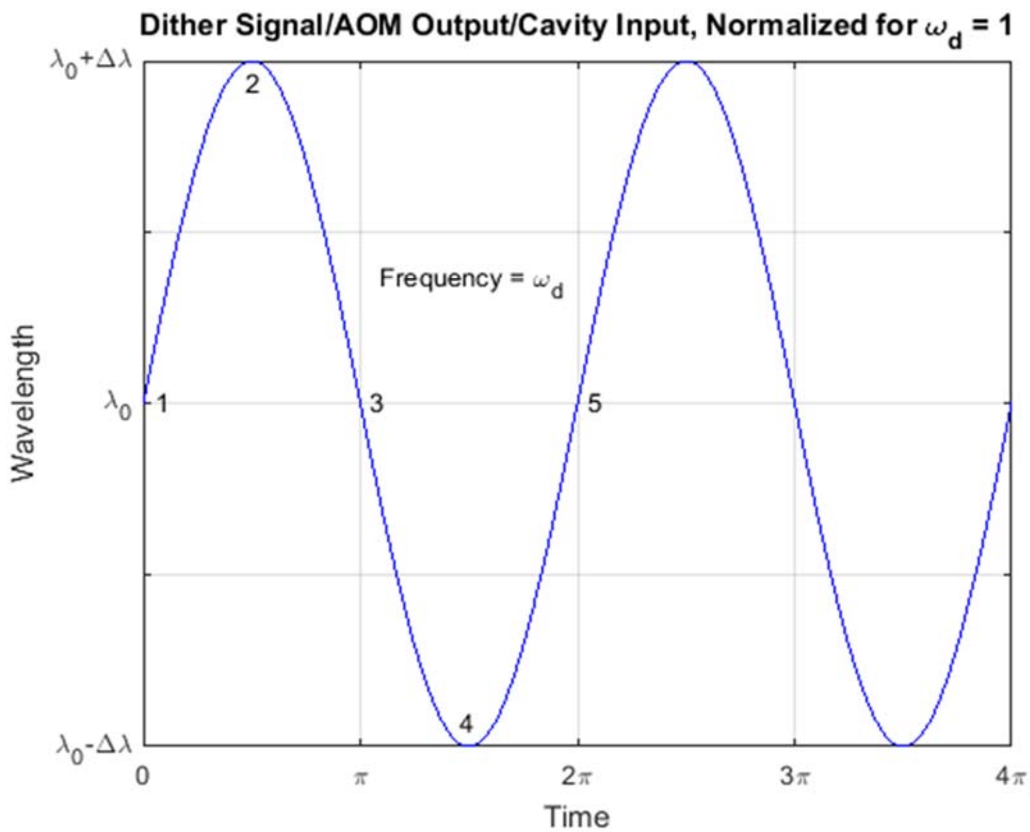


Figure 6.4 Dither signal as a function of time. This is the output of the AOM and the input of the optical cavity.

Next we consider the effect of dithering input on the cavity transmission output. The first case is when the cavity is on-resonant at λ_0. This is the situation when the peak of the cavity transmission wavelength λ falls exactly on λ_0 as shown in Figure 6.5 (left). By tracing number 1 to 5 from the dithering signal in Figure 6.4 onto Figure 6.5 (left) to figure out their cavity transmission amplitudes, we can map the cavity transmission output as a function of time shown in Figure 6.5 (right). It reveals that the cavity transmission output oscillates with a frequency $2\omega_d$ and an amplitude

$$A_{\text{on resonance}} = A(\lambda_0) - A(\lambda_0 \pm \Delta\lambda) \tag{6.6}$$

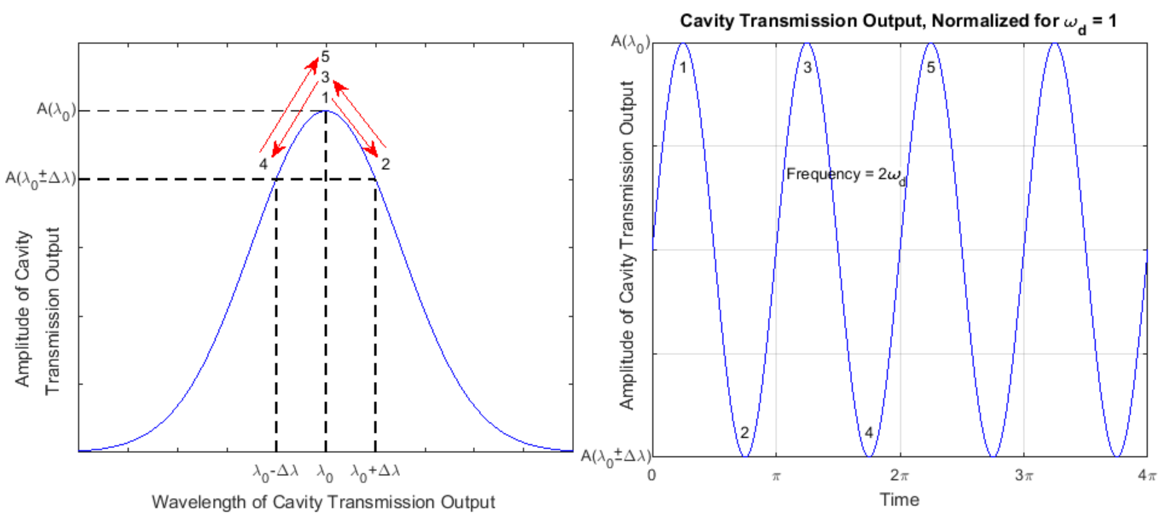


Figure 6.5 Cavity transmission output when the cavity is on-resonant at λ_0.

The second case is when the cavity is off-resonance with $\lambda > \lambda_0$ as shown in Figure 6.6 (left). By using the similar technique of tracing and mapping number 1 to 5, we get the cavity transmission output shown in Figure 6.6 (right) with a frequency ω_d and an amplitude

$$A_{\text{off resonance}} = A(\lambda_0 + \Delta\lambda) - A(\lambda_0 - \Delta\lambda) \tag{6.7}$$

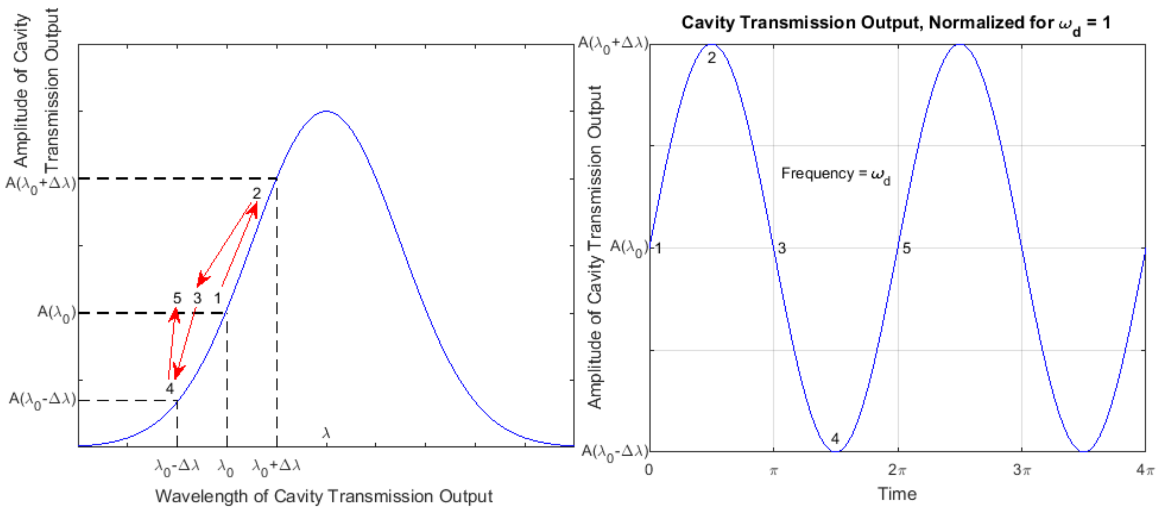


Figure 6.6 Cavity transmission output when the cavity is off-resonant. In the case where $\lambda > \lambda_0$, the cavity transmission is in phase with the dither signal.

The third case is when the cavity is off-resonance with $\lambda < \lambda_0$ as shown in 102 (left). Again from the similar technique, we get the cavity transmission output with the same frequency and amplitude as in the second case but with a 180° phase shift as shown in Figure 6.7 (right).

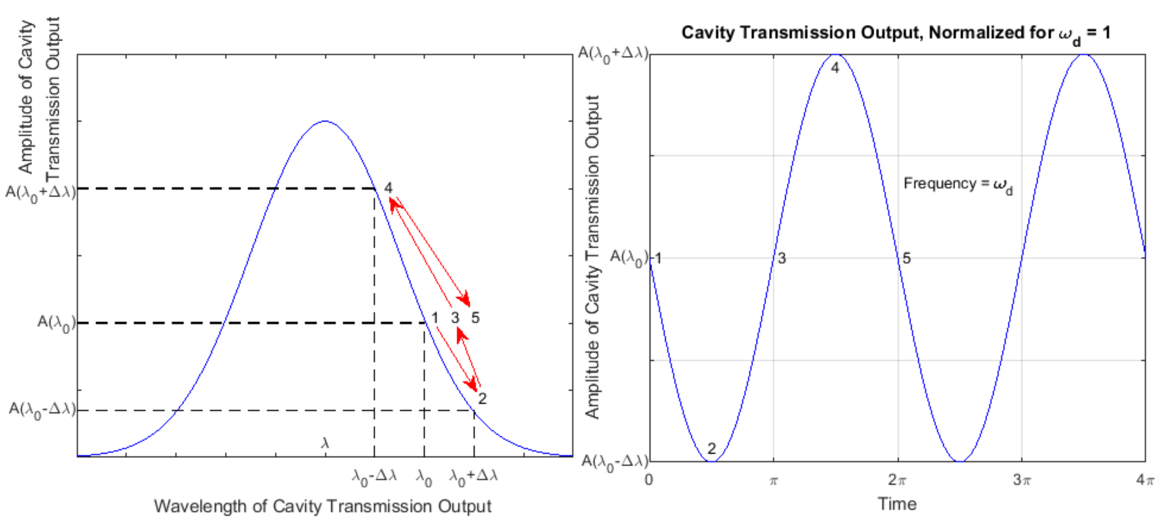


Figure 6.7 Cavity transmission output when the cavity is off-resonant. In the case where $\lambda < \lambda_0$, the cavity transmission is out of phase with the dither signal.

By examining the cavity transmission amplitudes in all three cases described by equation (6.6) and (6.7), it is evident that as λ approaches λ_0, the cavity transmission amplitude decreases and has the smallest value when $\lambda = \lambda_0$. Additionally, the cavity transmission frequency has the highest frequency of $2\omega_d$ when $\lambda = \lambda_0$ and decreases to ω_d as λ deviates from λ_0. Since the cavity transmission changes in such predictable way, we can theoretically plot what is called the error signal $\varepsilon(\lambda)$ which is essentially the wavelength derivative of the resonance feature of the cavity transmission. So far we have assumed a Gaussian shape for the resonance feature and therefore the error signal is the derivative of a Gaussian as shown in Figure 6.8.

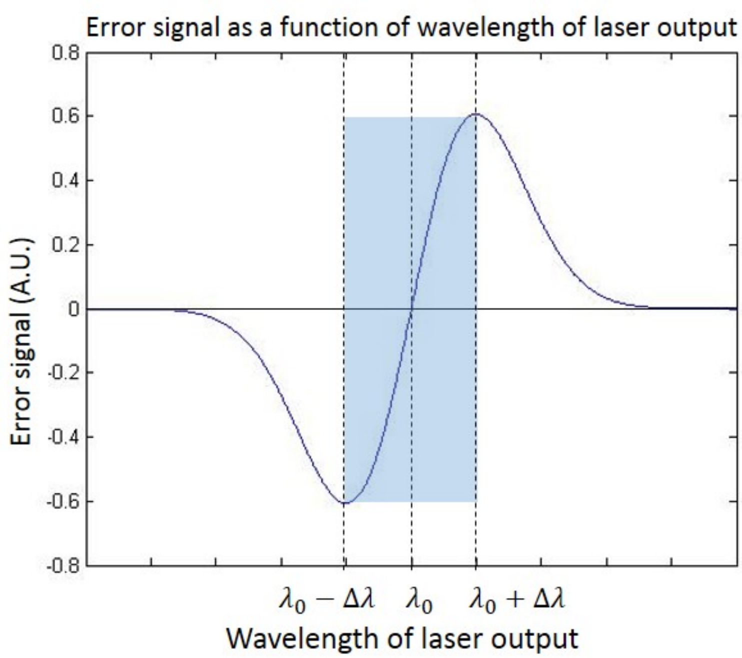


Figure 6.8 Error signal as a function of cavity transmission wavelength. The cavity is on-resonant when $\lambda = \lambda_0$ where the error signal is zero. To maintain resonance, the cavity transmission wavelength must stay within shaded region which indicates the range of the dithering wavelength. Plot courtesy by Briana Jones, Montana State University [49].

Figure 6.8 demonstrates the error signal as a function of cavity transmission wavelength. When on-resonance ($\lambda = \lambda_0$), the error signal is zero. When off-resonance in one direction ($\lambda > \lambda_0$), the error signal is positive and increases as $\Delta\lambda$ increases. When off-resonance in the other direction ($\lambda < \lambda_0$), the error signal is negative and decreases as $\Delta\lambda$ increases. When experimentally trying to lock the cavity, we need the cavity to stay within the shaded region on the plot otherwise the system would not know which way it needs to adjust to go back to resonance.

Methods and Setup

Optical Cavity

A linear cavity used in the OPO consists of two mirrors—one flat and one concave with 500-mm radius of curvature. The flat mirror has a measured reflectivity of 99% at the seed wavelength (1571 nm) and 24% at the pump wavelength (1064 nm). The curved mirror has measured reflectivity of 48% at the seed and 21% at the pump wavelength. The reflectivity at the idler wavelength (3297 nm) is estimated to be 30% for both mirrors. Although SNLO model predicts a better OPO performance with the other pairs of mirror selection, they must be custom built which requires extra time and budget that we cannot afford. Fortunately, the mentioned mirrors were already available in the lab and their performance as predicted by SNLO model is still promising.

To build a cavity that will resonate at the seed wavelength, each mirror must be placed where its radius of curvature (ROC) matches that of the seed laser. The ROCs of the seed and pump beams calculated based on the Gaussian fits in Figure 5.8 are plotted in blue and red, respectively, as shown in Figure 6.9. Based on the matching between the mirror and the seed ROCs, the flat mirror is located at 186 mm as indicated by the dashed black line and the curved mirror at 678 mm as indicated by the dashed blue line. The final OPO cavity with the crystal inside is demonstrated in Figure 6.10.

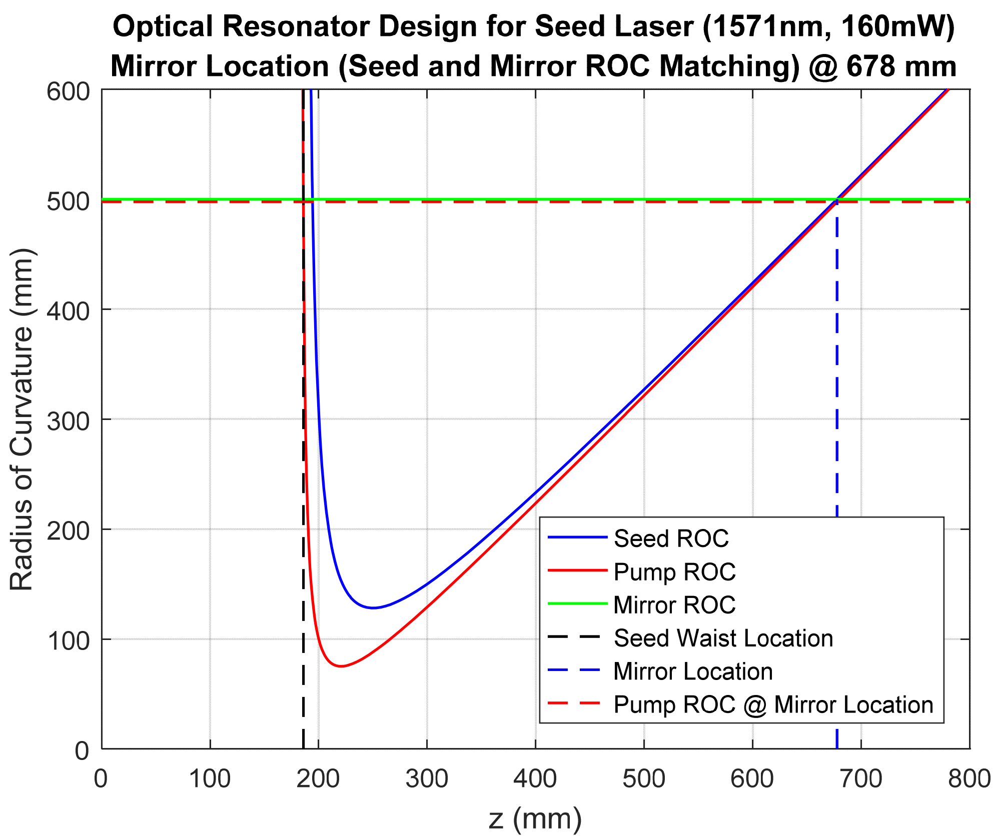


Figure 6.9 Optical cavity design for the location of the curved mirror. To resonate at the seed wavelength, each mirror must be placed at where its radius of curvature (ROC) matches that of the seed laser. From the measurements, the waist of the seed beam is located at 186 mm (dashed black line). This is where the flat mirror should be located since their ROC of infinity matches. The seed and pump beam ROCs calculated based on the Gaussian fits in Figure 5.8 are plotted in blue and red, respectively. The intercept between the curved mirror ROC (green line) and the seed ROC (solid blue line) is where the curved mirror should be located as indicated by the dashed blue line at 678 mm.

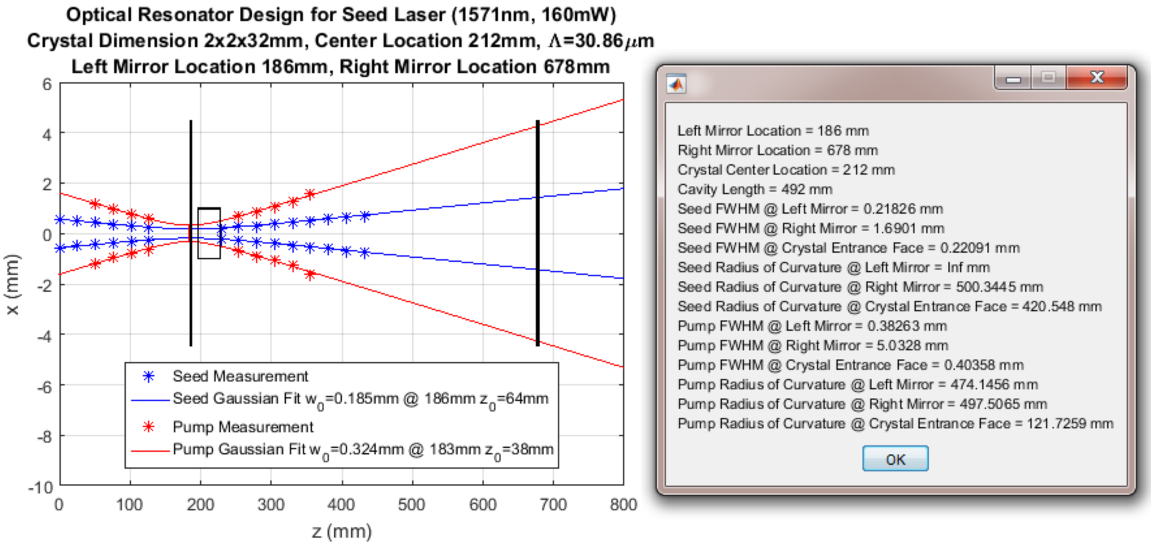


Figure 6.10 Schematic of the OPO cavity. The rectangle represents the crystal with its actual dimension. The left vertical black line represents the flat mirror. The right vertical black line represents the curved mirror. Note that the mirror locations are exact but the mirror diameters and thicknesses are not to scale.

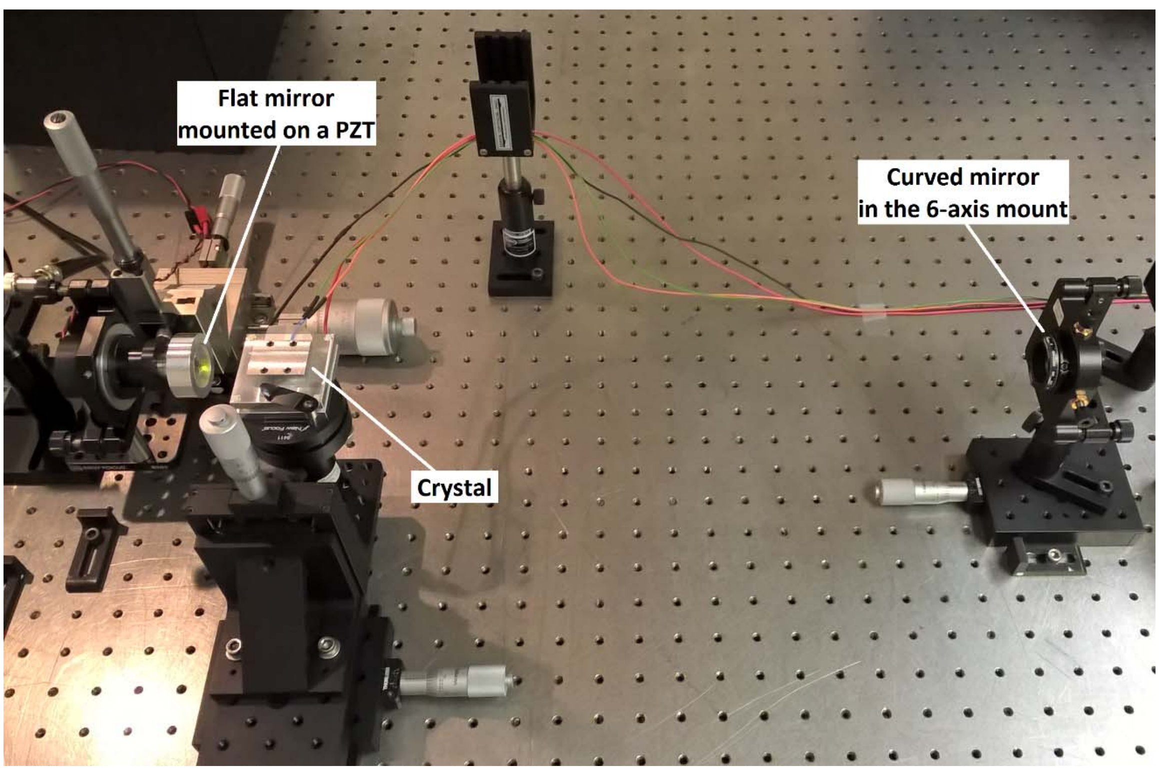


Figure 6.11 Image of the OPO cavity.

As shown in Figure 6.11, the flat mirror is mounted on a PZT which is used to mechanically adjust the cavity length according to the error signal when locking. Besides locking, the PZT can also be used for fine alignment of the cavity. By using a ramp generator to apply a saw-tooth-like voltage signal on the PZT, it will expand and contract at a length proportional to that applied voltage. This effectively scans the cavity length so that the cavity transmission such as one shown in Figure 6.3 can be observed. During the PZT scan with only the seed laser turned on, an InGaAs detector is used to measure the cavity transmission signal. By connecting the ramp generator and the detector to the oscilloscope, we can simultaneously observe both the applied voltage and the cavity transmission such as shown in Figure 6.12. To align the cavity, both mirror mounts which allow 6 degree of freedom including three translation axes, tip, tilt, and rotation, must be adjusted such that cavity transmission shown in blue is peaked-up.

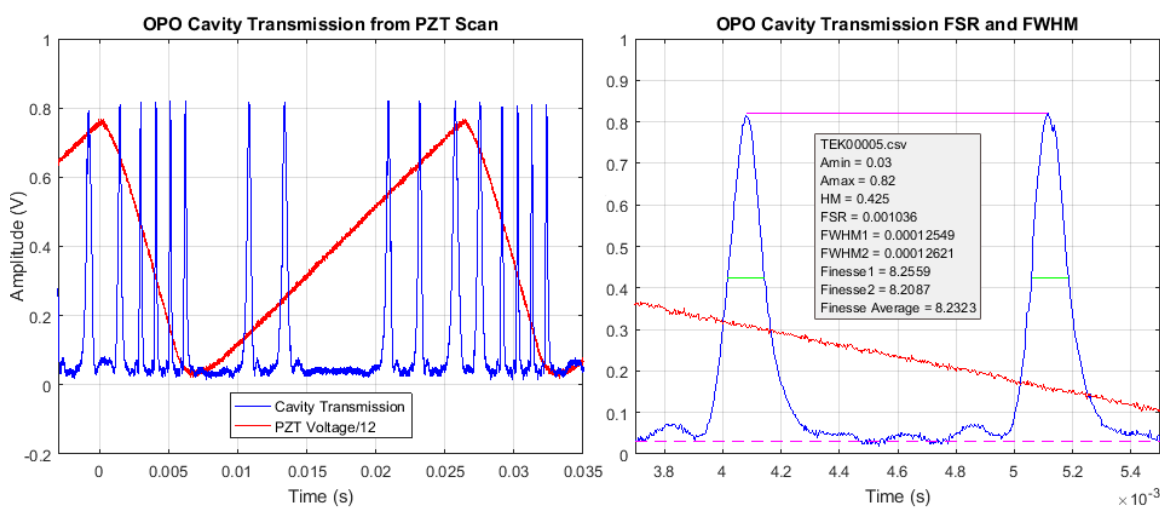


Figure 6.12 OPO cavity transmission from the PZT scan (left) with the zoom-in (right). In the right plot, the solid pink line represents the FSR. The green lines represent the FWHM of each peak. The amplitude of the green lines is determined as the half way between the dashed and solid pink lines. The measured values of the FSR and FWHM are used to calculate the finesse as shown in the box of the right plot.

Because of a narrow linewidth desired for the DIAL transmitter, we want the cavity to have the highest finesse possible. To maximize the finesse, the amplitude of the cavity transmission needs to be peaked-up such as shown in Figure 6.12. The cavity finesse can be calculated from the measured FSR and FWHM using equation (6.3). From the two selected peaks shown in Figure 6.12 (right), the finesse has an average value of 8.23. To gauge how well this value is, we compare it with the Airy finesse of a Fabry-Perot interferometer

$$F \approx \frac{\pi \left(R_1 R_2 \right)^{\frac{1}{4}}}{1 - \sqrt{R_1 R_2}} \tag{6.8}$$

where R_1 and R_2 are reflectivity of the cavity mirrors at the resonated wavelength. By substituting the mirror reflectivity values described earlier for R_1 and R_2, the cavity finesse is estimated to be 8.40. Since the measured value is very close to the estimate, a good cavity alignment is assured.

Dither Locking

As shown in Figure 6.1 diagram and as described in the dither locking theory section, the locking method in this experiment starts at a function generator that feeds the AOM with a sinusoidal signal of 5000-Hz dither frequency. The AOM then oscillates the input seed laser with a Doppler shift that creates the optical dither signal shown in Figure 6.4. This signal is then amplified and incident on the optical cavity. The cavity transmission output is split by a beam splitter with a small portion being sent to the InGaAs detector and large portion to the power detector/OSA. The analog signal of the cavity transmission produced by the InGaAs detector together with the analog dither signal from the function generator are used by the lock-in amplifier to determine and extract the error signal. However, one issue arises at this step of the locking process. While the locking goes smoothly for the CW seed laser, the high-intensity pulsed output can cause the InGaAs detector to oversaturate and thus creates a small valley in its analog output. This small valley can be mistakenly determined as a peak while locking since its output also has a frequency of $2\omega_d$. This makes the locking rather difficult. For this reason, a hold circuit designed to ignore high-intensity pulses is inserted in between the InGaAs detector and the lock-in amplifier.

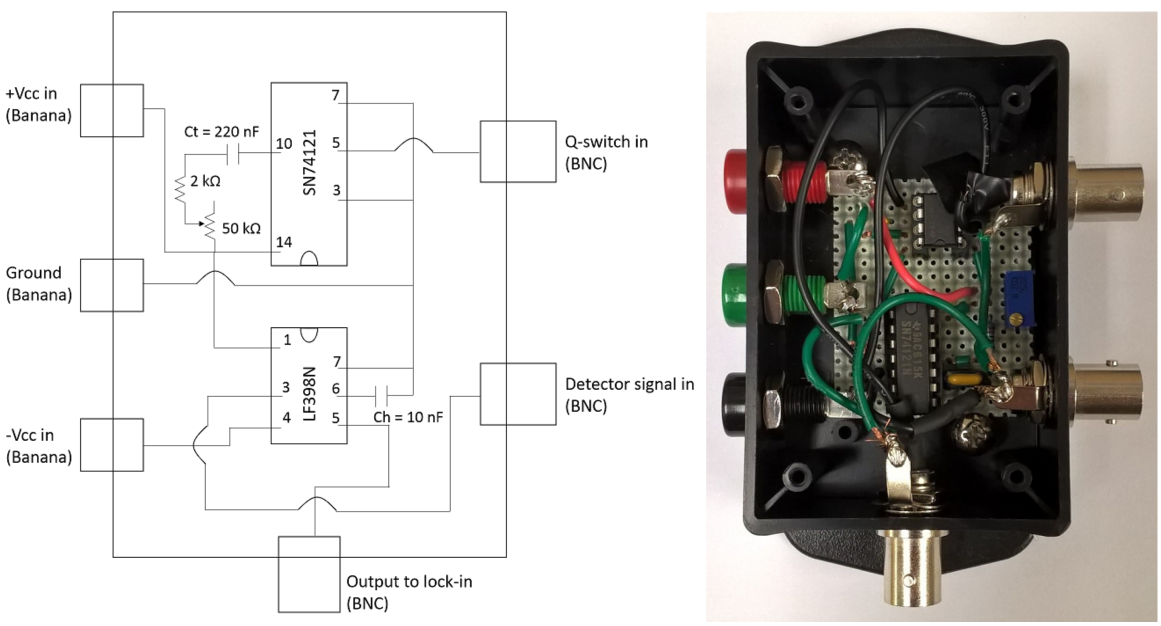


Figure 6.13 Hold circuit diagram (left) and image (right) designed to ignore the analog signals of high-intensity pulses. A Schmitt trigger (SN74121) and a sample and hold (LF398N) are used in conjunction to time the arrival of the large signal pulses and to set how long to ignore the pulses. A variable resistor is used to fine-tune the hold time. [49]

As shown in Figure 6.13, the hold circuit consists of a Schmitt trigger and a sample-and-hold chip. The Schmitt trigger produces a square wave that starts at the beginning of the rising edge of the high-intensity signal and ends at a time $t = 0.7CR$ where C and R are the capacitance and resistance. Given the values 2 kΩ fixed resistor and 50 kΩ variable resistor for R and 220 nF for C, t can range between 0.3 to 8 ms. This is the length of time that the circuit ignores the high-intensity signal from the detector. The time needs to be set such that it is long enough to completely block the valley and yet short enough to prevent the PZT from changing the cavity length too much. The sample-and-hold chip holds the values of the detector signal for the duration t of the square wave from the Schmitt trigger. This provides a clean cavity transmission signal which can be used by the lock-in amplifier to extract the error signal. The error signal which contains information about the change increment of the moving direction of the cavity, is then used by the ramp generator to control the PZT accordingly.

In the experiment, an oscilloscope is used to monitor the locking. To do this, a split signal from the InGaAs detector and an analog output signal from the ramp generator are sent to the oscilloscope input channels. The appearance of these signals on the oscilloscope screen allows us to monitor the locking in real time. This is important because oftentimes the locking can be unstable due to large temperature fluctuations and building noises. At a very unstable stage, the locking can totally go off the shaded region shown in Figure 6.8 causing the locking loss. When this occurs, the signal amplitude appeared on the oscilloscope screen will drop to zero. If this happens, we can try to relock the cavity by adjusting the ramp amplitude and bias to peak the signal amplitude back up.

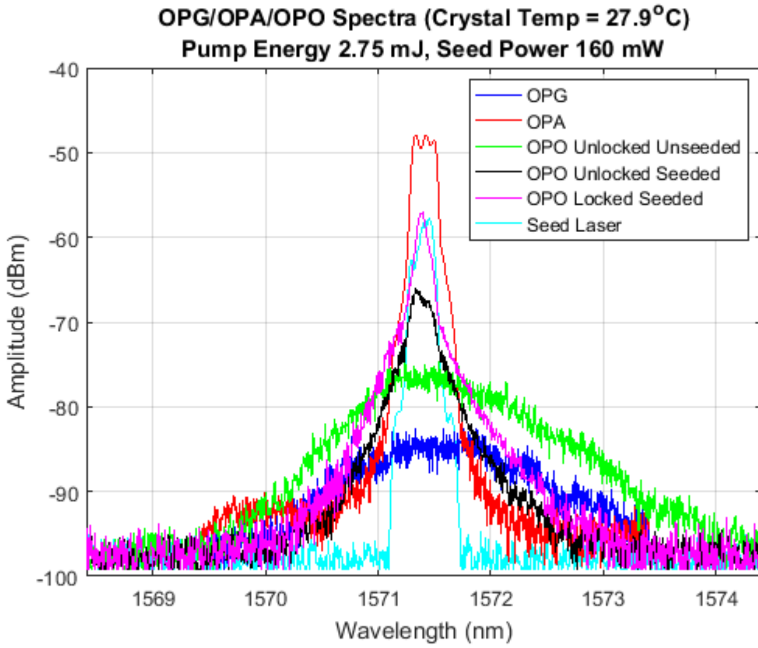
Results

OPO data are taken in three scenarios—unseeded unlocked, seeded unlocked, and seeded locked. The unseeded unlocked OPO is similar to the OPG but with the presence of an optical cavity. It therefore only has the pump laser turned on and thus relies on spontaneous emission. Note that without the seed laser turned on, the cavity locking is not possible. Hence there is no such case as an unseeded locked OPO because of the absence of the seed laser that is necessary for locking. Likewise, unseeded unlocked OPO is similar to OPA but again with the presence of an optical cavity. Both the pump and seed lasers are turned on while the locking system remains off in this case. Finally, the seeded locked OPO has both the pump and seed lasers and the locking system turned on.

Output Spectrum

Before looking at the spectrum measurement results, it should be noted that the OPG/OPA data are the same data taken in Chapter 5. It should also be noted that the spectrum measurements from the OSA cannot be used for intensity comparison between the OPG/OPA and the OPO, as well as the seed laser. Although the OPO is built upon the OPG/OPA setup, adding an optical cavity is always going to change the output beam shape and direction. Even at best re-aligning effort, the amount of light that is fiber coupled into the OSA still reduces significantly. Consequently, all of the OPO peaks appear smaller than the OPA peak, but this does not necessary mean the OPA output energy is greater than that of the OPOs. As far as the seed spectrum is concerned, the CW seed and the pulsed OPG/OPA/OPO outputs are handled differently by the OSA, so their measurements are again not comparable. Nevertheless, FWHM determined from the OSA measurements can be used for comparison among each other.

In the output spectrum measurements shown in Figure 6.14, the OSA is set at the highest resolution of 0.06 nm with a high sensitivity of -90 dBm. Data in each plot are averaged between 5 to 10 pulses or until the pattern of the plot does not appear to change much. As expected, the FWHM of the seeded-locked OPO is the narrowest among the five cases, followed by the seeded-unlocked OPO, OPA, unseeded-unlocked OPO, and OPG. The unseeded cases have broader spectra because they depend on spontaneous emission rather than stimulated emission like in the seeded cases. The optical cavity proved to narrow down the spectra as evidently shown when comparing between the unseeded-unlocked OPO and OPG or between the seeded-unlocked OPO and OPA. The OPO proved to be much more effective when the cavity is locked as the FWHM reduces by half when going from the seeded-unlocked to the seeded-locked OPO. Meanwhile all five cases demonstrate good spectral overlap between the pump and seed beams as all of their peaks are located at the seed laser wavelength of 1571.406 nm. Good spatial overlap between the pump and seed beams are also observed in all cases since the sidebands become narrower when going from the unseeded cases to the seeded cases. Both spectral and spatial overlaps imply a good phase matching for the crystal and a good beam alignment inside the optical cavity.

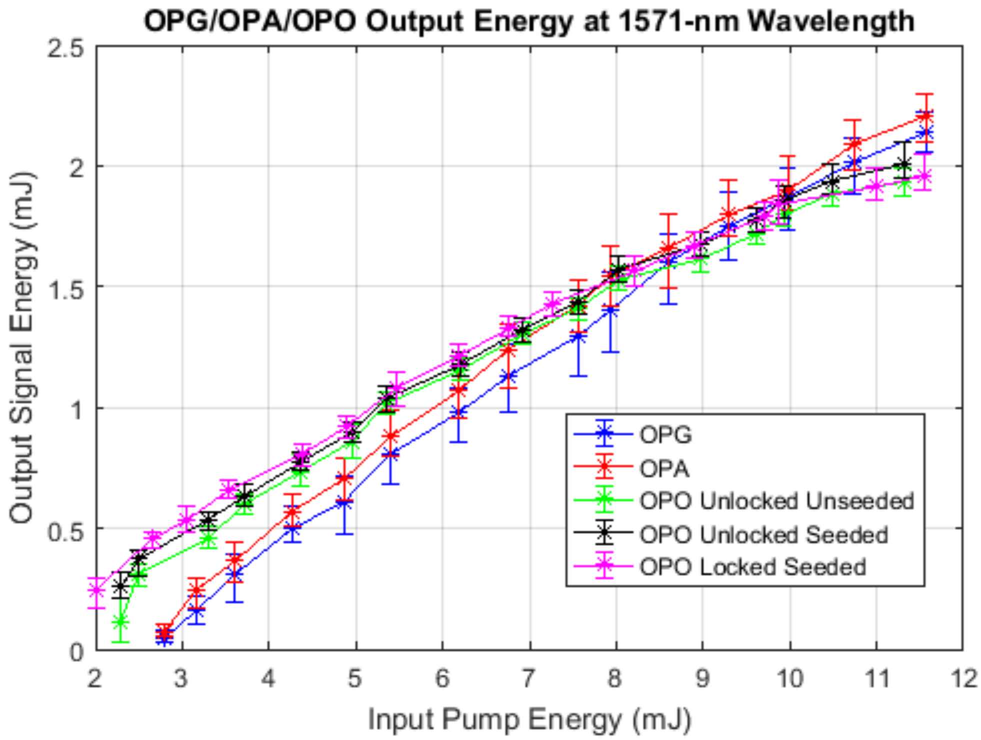


	FWHM (nm)
OPG	1.4000
OPA	0.2232
Unseeded-unlocked OPO	1.1000
Seeded-unlocked OPO	0.1898
Seeded-locked OPO	0.0941

Figure 6.14 OPG/OPA/OPO spectra (left) and the corresponding FWHM measurements (right). FWHMs are carefully measured at -3dBm down from the peaks.

<u>Output Energy</u>

Results of the output energy measurements using a Thorlabs energy meter model PM100D with ES111C power detector are shown in Figure 6.15. At below 8-mJ input pump energies, the output energy and conversion efficiency from the seeded-locked OPO is the highest among the five cases, followed by the seeded-unlocked OPO, unseeded-unlocked OPO, OPA, and OPG, which was expected. However, at above 8-mJ input pump energies, all three OPO output energies drop down and become lower than that of OPG and OPA. This is likely due to the broadening of the spectra at higher pump energies, which may be corrected with a higher seed power.



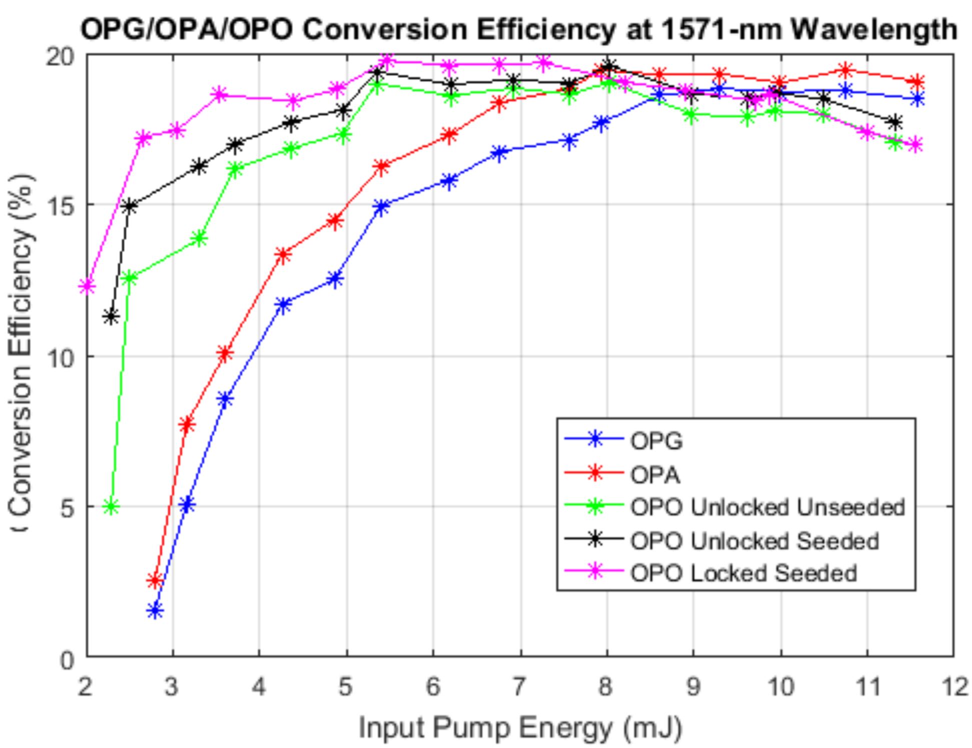
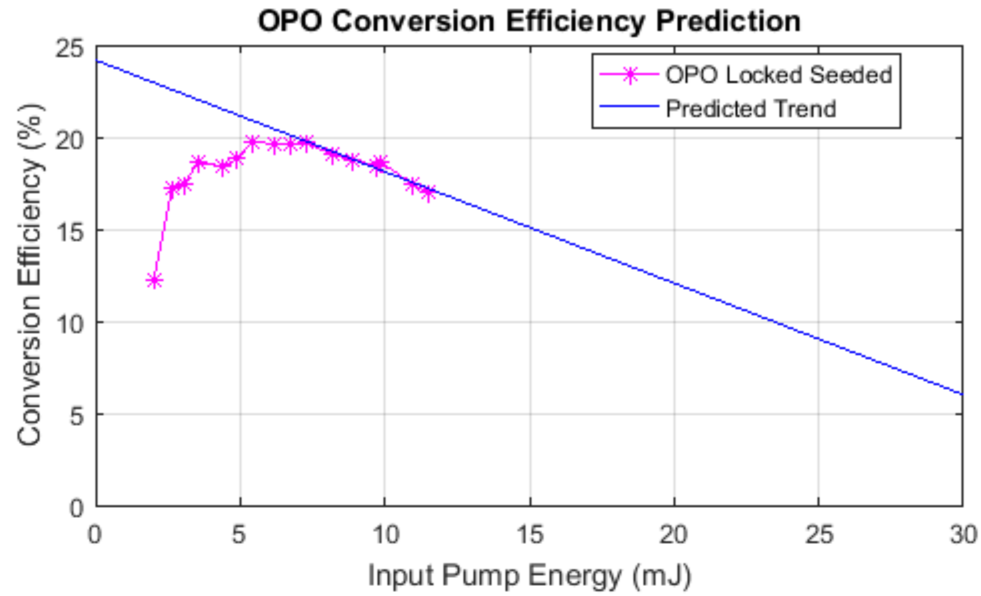


Figure 6.15 OPG/OPA/OPO output energy (top) and conversion efficiency (bottom). In the measurements, crystal temperature was 27.9 ˚C and the seed power was 160 mW.

In Figure 6.15 (bottom), the conversion efficiency of the seeded-locked OPO reaches a maximum near 20% and levels out then decreases, while the conversion efficiency of the OPA reaches near 19% and levels out. Higher input pump energies are necessary to determine how the OPA conversion efficiency will fall. Because of the narrow linewidth needed for the DIAL, we can focus solely on the seeded-locked OPO even though the OPA tends to outrun the OPO at high input pump energies as shown in Figure 6.15 (top). The same seeded-locked OPO conversion efficiency data is plotted in Figure 6.16 (top), but with a linear fit line to predict its falling trend. This efficiency trend is then used to predict the output energy trend shown in Figure 6.16 (bottom). Although the output energy trend does not reach a value of 3 mJ required for the DIAL, its close maximum value near 2.5 mJ at 20-mJ input pump energy suggests a possible success. Using a higher seed power can certainly improve the OPO efficiency. Additionally, further enhancement of the cavity including better mode-matching, better locking, and choice of mirrors i.e. different curvature and reflectivity could also assist in generating more output energy while keeping the spectrum narrow. We must however keep in mind not to risk damaging the crystal. Future work in optimization focusing on pump beam sizes within the crystal, damage threshold of the crystal, and the generated output energy at 1571 nm is needed to avoid damage the crystal.



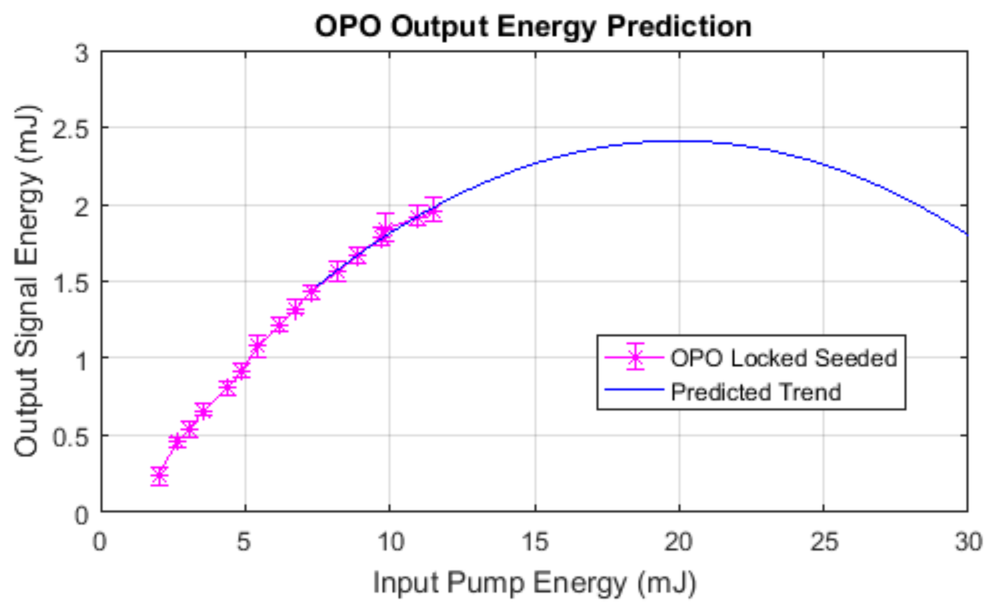


Figure 6.16 Predictions of conversion efficiency and output energy for seeded-locked OPO.

<u>SNLO Comparison</u>

The SNLO model for the OPO uses the measured values shown in Figure 6.10 as its inputs. The screenshot of the SNLO 2D-cav-LP inputs is shown in Figure 6.17. It should be noted that the SNLO assumes a symmetric cavity. Therefore, the inputs for the flat mirror are modified as if they are for another curved mirror that would make the cavity symmetric. The d_{eff} value of 13 pm/V acquired from the OPA model is used in the OPO model. Δk is set to zero for the case of perfect QPM. The values of the input pump energy vary from 1 to 16 mJ.

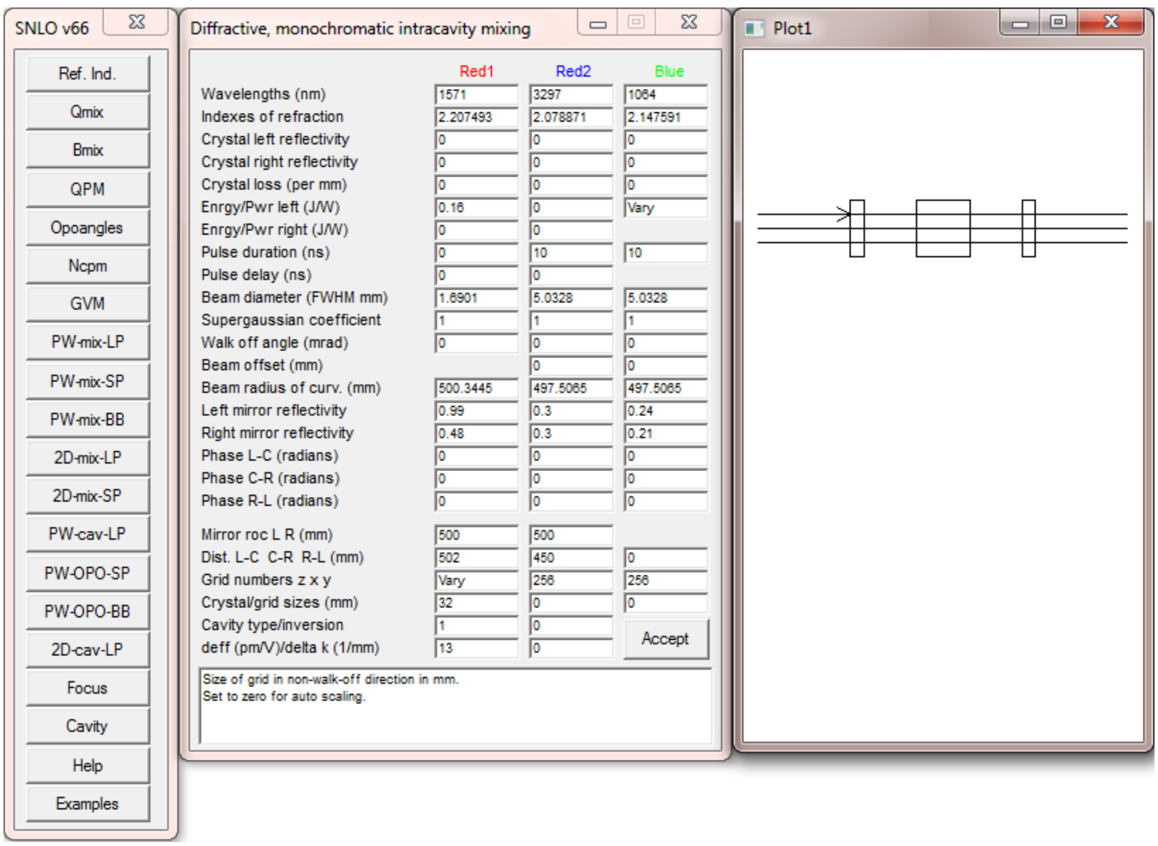


Figure 6.17 Input in SNLO model for OPO output energy.

Results from the SNLO model in comparison with the OPO measurements are shown in Figure 6.18. Although with a lot of noise, SNLO displays similar trend with the OPO. This confirms experimental success in terms of system alignment, phase matching, and perhaps locking. With a better choice of cavity mirrors, the OPO efficiency could potentially meet the required specs of the DIAL.

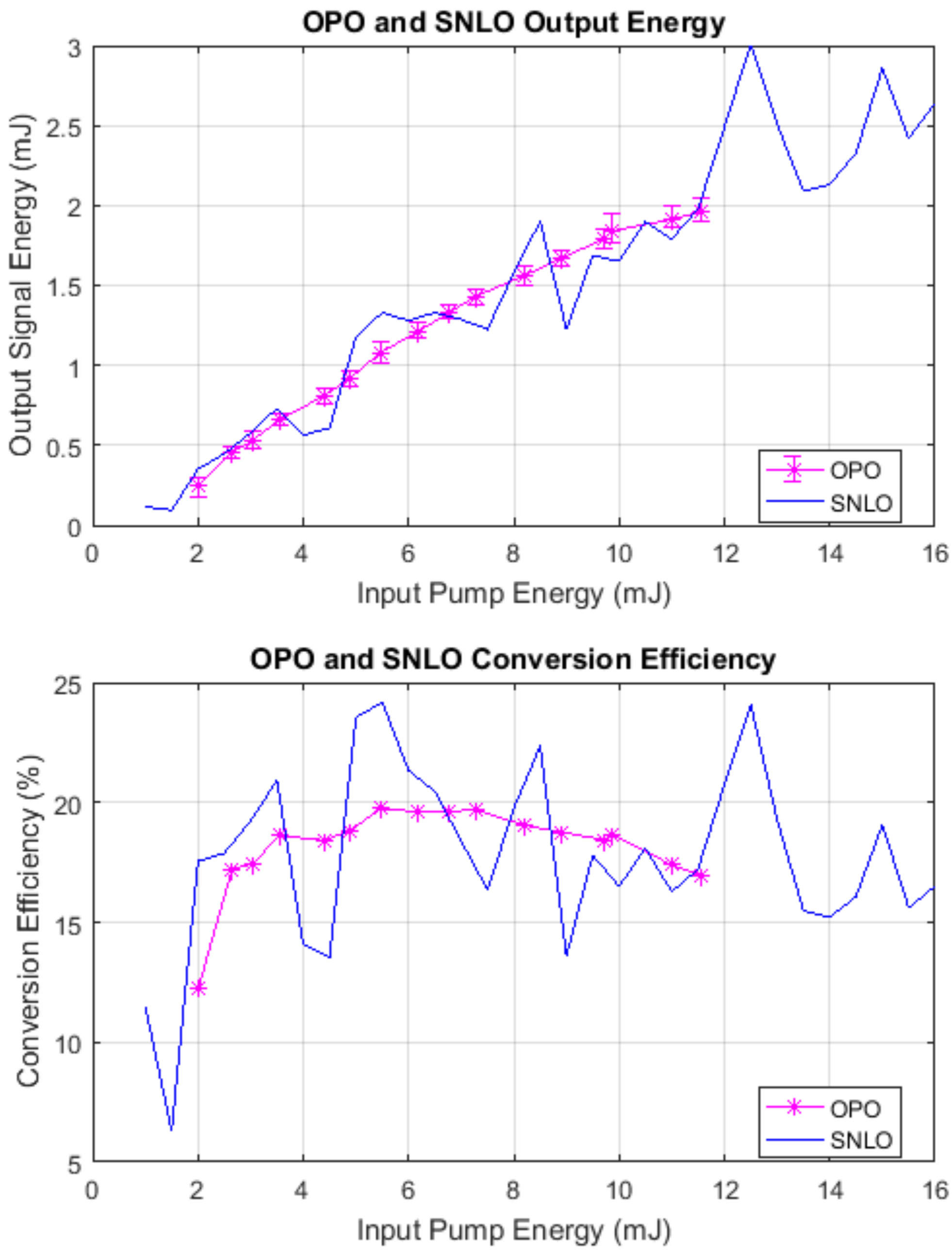


Figure 6.18 Output energy and conversion efficiency comparisons between OPO and SNLO model with $d_{eff} = 13$ pm/V and $\Delta k = 0$.

DISCUSSION OF INITIAL DIAL DESIGN

Following the promising results of the OPO in the last chapter, this chapter proposes a design for a DIAL system incorporating the OPO. In addition to the seed wavelength of 1571 nm presented in this dissertation, the DIAL design embraces another seed wavelength of 1645 nm for methane (CH_4) measurements. The potential use of an OPO near a wavelength of 1645 nm in a DIAL is presented in Briana Jones (2015) [49]. The complete initial design of the dual-seed-wavelength DIAL with the schematic is discussed in the last section of this chapter. In the design work, we start by modeling the DIAL performance based on the known parameters to estimate its errors.

DIAL Performance Model

A differential error analysis based on the DIAL equation yields an error as a function of range, $\Delta N_{mol}(r)/N_{mol}(r)$, given by [50]

$$\frac{\Delta N_{mol}(r)}{N_{mol}(r)} = \frac{1}{2N_{mol}(r)\left[\sigma_{mol}(\lambda_{on},r)-\sigma_{mol}(\lambda_{off},r)\right](r_2-r_1)}\left\{\begin{array}{l}\dfrac{\left[N_S(\lambda_{on},r_1)+N_{BG}\right]F+N_D}{N_S^2(\lambda_{on},r_1)}+\\ \dfrac{\left[N_S(\lambda_{off},r_1)+N_{BG}\right]F+N_D}{N_S^2(\lambda_{off},r_1)}+\\ \dfrac{\left[N_S(\lambda_{on},r_2)+N_{BG}\right]F+N_D}{N_S^2(\lambda_{on},r_2)}+\\ \dfrac{\left[N_S(\lambda_{off},r_2)+N_{BG}\right]F+N_D}{N_S^2(\lambda_{off},r_2)}\end{array}\right\}^{\frac{1}{2}}$$

(7.1)

where $N_{mol}(r)$ is the number density for the molecule of interest, $\sigma(\lambda, r)$ is the molecular cross section, λ_{on} and λ_{off} are the online and offline wavelengths, r_1 is the near range and r_2 is the farther range, $N_S(\lambda, r)$ is the number of photoelectrons seen by the DIAL detector, N_{BG} is the background noise, F is the detector excess noise factor, and N_D is the detector and circuit noise.

The number of photoelectrons seen by the DIAL detector can be calculated based on the lidar equation, equation (2.12). The background noise can be calculated from $N_{BG}(\lambda) = \frac{\pi\lambda s_{BG}(\lambda)}{4hc}\tau A\varepsilon_d(\lambda)\varepsilon_r(\lambda)BW_{filter}FOV^2$ where $s_{BG}(\lambda)$ is the background irradiance, BW_{filter} is the filter bandwidth, and FOV is the receiver field of view, h is the Planck's constant, c is the speed of light, τ is the pulse duration, A is the area of the receiving telescope, $\varepsilon_d(\lambda)$ is the detector efficiency, $\varepsilon_r(\lambda)$ is the receiver efficiency. The detector noise is calculated from $N_D = \frac{I_n\tau\sqrt{BW}}{qM}$ where I_n is the noise current spectral density, BW is the detection system bandwidth, q is the electron charge, and M is the multiplication gain of the detector. The noise current spectral density is found from $I_n = \sqrt{2qI_dMF + I_{na}^2 + \frac{V_{na}^2}{R_f^2} + \frac{4kT_K}{R_f} + \frac{(2\pi V_{na}C_dBW)^2}{3}}$ where I_d is the detector dark current, I_{na} and V_{na} are the preamplifier current and voltage noise spectral density, R_f is preamplifier feedback resistance, C_d is the detector and preamplifier equivalent capacitance. [50, 51]

The DIAL performance can be modeled using equation (7.1) to calculate the percent error expected based on a model atmosphere and instrument parameters. The instrument parameters and the atmospheric parameters used for the performance modeling are listed in Table 7.1 and 7.2 consecutively. For the 3-mJ pulse energy, 10-ns pulse duration, and 1-kHz pulse repetition rate, the eye-safe operation of the DIAL can be found from the American National Standards for Safe Use of Lasers (ANSI Z136.1-1993) [52]. Eye-safe operation is a requirement for autonomous DIAL operation, particularly in the horizontal direction. The maximum permissible exposure (MPE) for either the CO_2 or CH_4 operating wavelengths is 100 μJ/cm^2. To operate below the MPE, the 3-mJ pulse must be spread over an area of greater than 30 cm^2.

Table 7.1 Transmitter and receiver parameters used for the DIAL performance modeling.

Transmitter		**Receiver**	
CO_2 On-Line Wavelength	1571.4060 nm	Area	0.114 m^2
CO_2 Off-Line Wavelength	1571.2585 nm	Field of View	500 μrad
CH_4 On-Line Wavelength	1645.5518 nm	Filter Bandwidth	1 nm
CH_4 Off-Line Wavelength	1645.3724 nm	Receiver Efficiency	0.55
Pulse Energy	3 mJ	Detector Efficiency	0.8
Pulse Duration	10 ns	Excess Noise Factor	2
Pulse Repetition Rate	1 kHz	Detection System Bandwidth	3.5 MHz
		Dark Current	60 nA
		Multiplication Factor	10
		Current Noise Spectral Density	500 fA/$\sqrt{HZ}$
		Voltage Noise Spectral Density	3 nV/$\sqrt{HZ}$

Table 7.2 Atmospheric parameters used for the DIAL performance modeling.

Atmospheric Parameters			
Backscatter Coefficient	2x10^{-6} m^{-1}sr^{-1}	Background Irradiance	4x10^{-3} W/m^2 μm sr
Lidar Ratio	50 sr	Temperature	293 K
Extinction Coefficient	1x10^{-4} m^{-1}	Pressure	0.85 atm
CO_2 Absorption Cross Section	8.72x10^{-27} m^2	CO_2 Concentration	390 ppm
CH_4 Absorption Cross Section	1.60x10^{-24} m^2	CH_4 Concentration	1.8 ppm

The expected performance for CO_2 profiling is shown in Figure 7.1 as a plot of the percent error as a function of range. The black dashed (solid, dotted) line indicates an error of 1% (2%, 0.5%) and is shown for reference. The blue (red, green) lines indicate temporal averaging of 1 minute (5 minutes, 15 minutes) while the solid (dashed, dotted) line represents spatial averaging over 75 m (150 m, 300 m). The performance model indicates that for averaging times of 5 minutes (2.5 minutes of on-line returns and 2.5 minutes of off-line returns), a maximum range of greater than 4 km (5 km, 7 km) is achievable for spatial averaging of 75 m (150 m, 300 m) for an error of less than 1% (3.9 ppm). For vertical profiling, this indicates that profiles through the atmospheric boundary layer are achievable with useful temporal and spatial resolution.

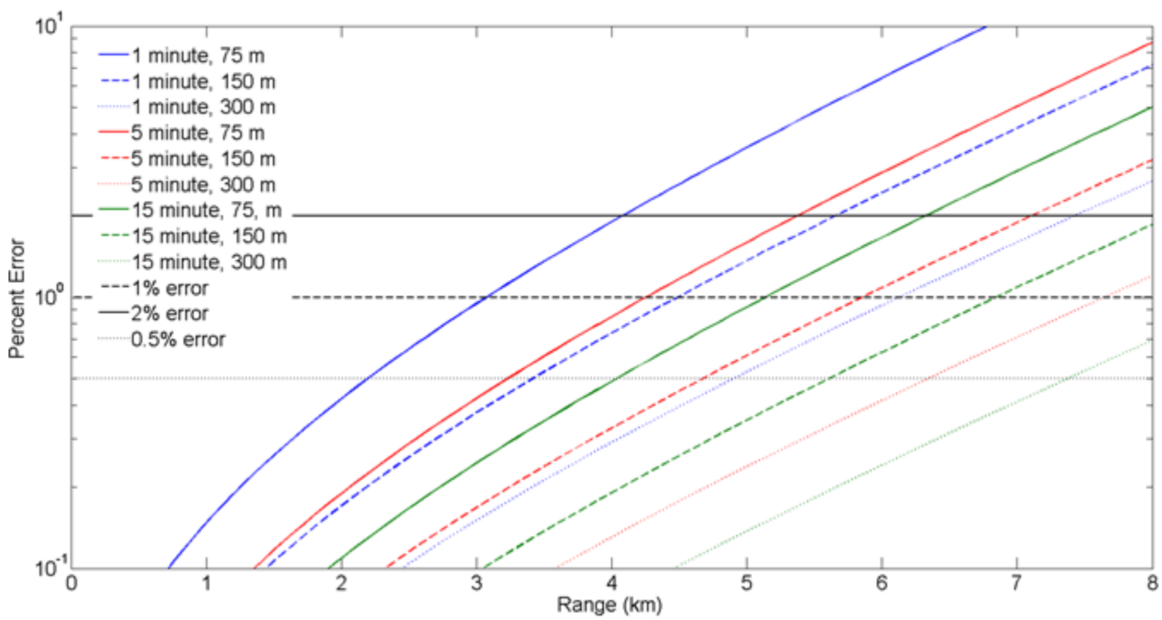


Figure 7.1 The percent error in the DIAL retrieval of CO_2 as a function of range for temporal averaging ranging from 1 minute to 15 minutes and spatial averaging ranging from 75 m to 300 m.

The expected performance for CH_4 profiling is shown in Figure 7.2 as a plot of the percent error as a function of range. The performance model indicates that for averaging times of 5 minutes (2.5 minutes of on-line returns and 2.5 minutes of off-line returns), a maximum range of greater than 2 km (4 km, 6 km) is achievable for spatial averaging of 75 m (150 m, 300 m) for an error of less than 2%.

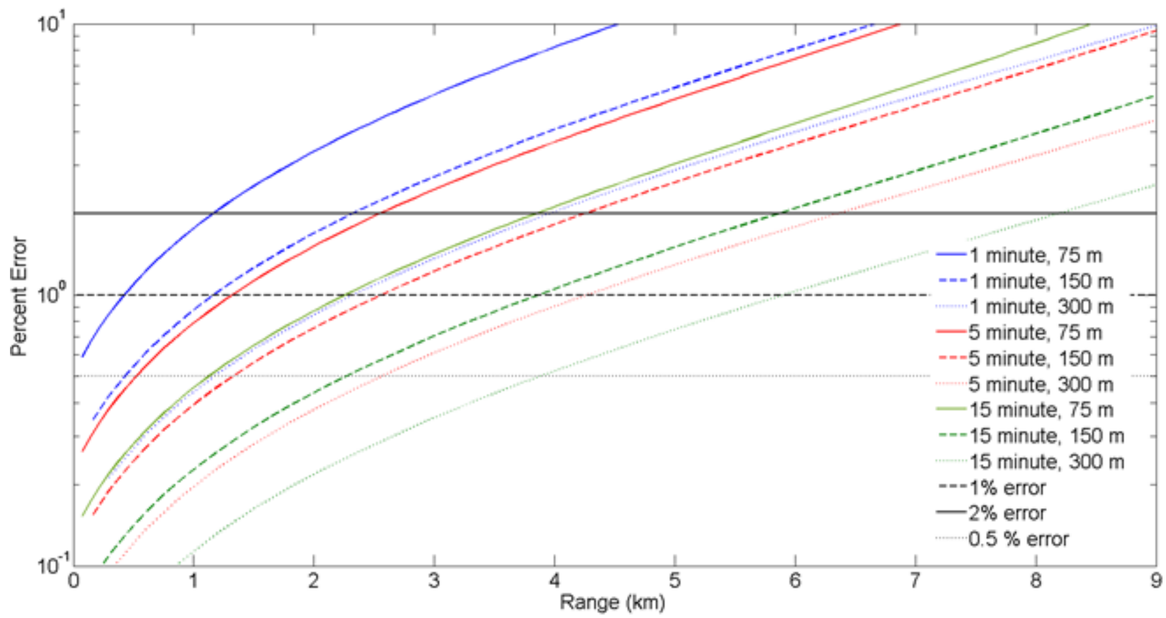


Figure 7.2 The percent error in the DIAL retrieval of CH_4 as a function of range for temporal averaging ranging from 1 minute to 15 minutes and spatial averaging ranging from 75 m to 300 m.

Initial DIAL Transmitter Design

The initial design of the DIAL laser transmitter includes the operations at the on-line and off-line wavelengths of both CO_2 and CH_4. Because no high-power pulsed-laser source is commercially available at these wavelengths, the laser transmitter design utilizes a singly-resonant OPO similar to that in Chapter 6, but with some modifications.

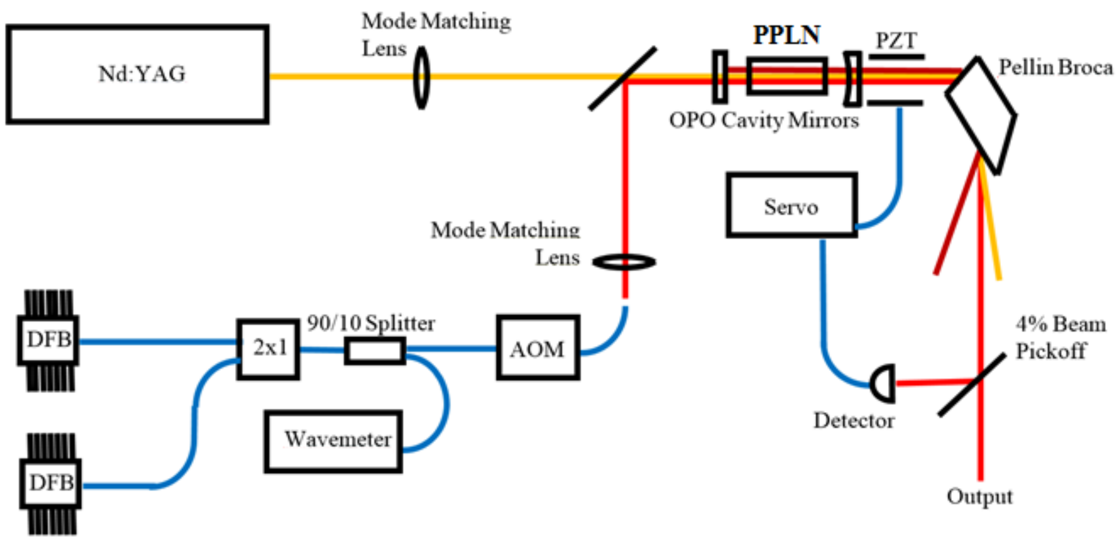


Figure 7.3 Schematic of the laser transmitter design based on a singly-resonant OPO.

A schematic of the proposed laser transmitter design is shown in Figure 7.3. In the design, a pulsed Nd:YAG laser operating at the fundamental 1064 nm wavelength is used to pump the singly-resonant OPO. Two distributed feedback (DFB) diode lasers with continuous wave (cw) output is used for injection seeding, one DFB laser operating at the on-line wavelength and the other DFB laser operating at the off-line wavelength. The DFB lasers is coupled to a 2×1 fiber optic switch to allow switching between the on-line and off-line wavelengths. After the 2×1 fiber optic switch, an inline 90/10 beam splitter is used to direct 10% of the DFB laser output to a wavemeter for seed laser locking. When the switch is set for the on-line DFB laser, the wavemeter is read and compared to a set wavelength, and a correction signal is sent to the diode laser controller (not shown in Figure 7.3) to adjust the current to maintain the set wavelength. When the switch is set for the off-line DFB laser, the wavemeter is read and compared to a second set wavelength and a correction signal will be sent to a second diode laser controller where the off-line laser current can be adjusted to maintain the set wavelength. The 90% port of the fiber splitter directs the laser light to an in-line acousto-optic modulator (AOM) that produces a small frequency oscillation used for locking the OPO cavity to the seed laser. After the AOM, the injection seed laser is combined with the pulsed pump laser and is incident on the OPO cavity with mirrors that provide a resonance at the signal wavelength while allowing a single pass for the pump and idler wavelengths. The nonlinear interaction then takes place in a PPLN crystal similar to the crystal used in this dissertation. As presented in the last two chapters and in Briana Jones (2015) [49], phase matching can be achieved for a signal wavelength of 1571 nm with a crystal of 30.86-μm poling period at 27.9 $^{\circ}$C temperature and near 1645 nm with a crystal of 31.447-μm poling period at 44 $^{\circ}$C temperature. After the OPO cavity, the pump, signal, and idler beams are separated using a dispersive optic such as a Pellin-Broca prism, a 4% pick-off optic will be used to direct a portion of the signal light to a detector to be used for locking while the remainder of the signal beam will provide the output for the DIAL laser transmitter.

The locking of the OPO cavity to the injection seed laser will be accomplished using the dither locking technique as described in Chapter 6 of this dissertation. Two important issues that will arise with this locking scheme include the switching between the on-line and off-line wavelengths and the generation of the pulsed signal. To handle the switching between the on-line and off-line wavelengths, the on-line wavelength will be locked to the absorption line center via the optical wavemeter. The off-line wavelength will then be locked, via the optical wavemeter, to a wavelength that is an integral number of cavity free spectral ranges away from the on-line wavelength so that the OPO cavity will be resonant at both wavelengths with the same OPO cavity length. To handle the pulse generation at the signal wavelength, the servo electronics will have a unity gain at a frequency of approximately 1 kHz due to the response time associated with adjusting the OPO cavity via the PZT. Since the pulse is on the order of 10 ns corresponding to a frequency of 100 MHz, the servo will not be able to provide any control signal associated with this pulse. In simple terms, the feedback loop cannot react fast enough to try to make a correction to the cavity length over the 10 ns pulse duration.

Initial DIAL Design

A schematic of the complete initial design of the CO_2/CH_4 DIAL is shown in Figure 7.4. In the design, the output from the singly-resonant OPO laser transmitter is first incident on a pickoff optic to allow the monitoring of the laser transmitter output. The beam is then expanded and is shaped using a matched axicon pair which produces an annular beam. This annular beam is then focused into the telescope using a lens to match the f/# of the telescope. After the lens, the beam passes through an elliptical mirror with a center hole bored through the mirror. This transmitted beam then fills up the inner part of the telescope and, because of the annular shape, the outgoing beam is not blocked by the secondary mirror of the telescope. This shared telescope design has two benefits. First, the shared telescope provides for a mechanically stable transceiver needed for long term autonomous operation. Second, the outgoing beam can be expanded over a larger area. Allowing the outgoing annular beam to have a 10.16 cm inner diameter and a 15.24 cm diameter allows a 3-mJ pulse to have an intensity of 29.7 $\mu J/cm^2$, a factor of greater than 3 below the 100 μJ/cm^2 needed for eye-safe operation. Using a 40.64-cm diameter Dobsonian telescope implies that the collection area of the telescope is 1140 cm^2 (area between diameters of 15.25 cm and 40.64 cm), the value used in the aforementioned performance model.

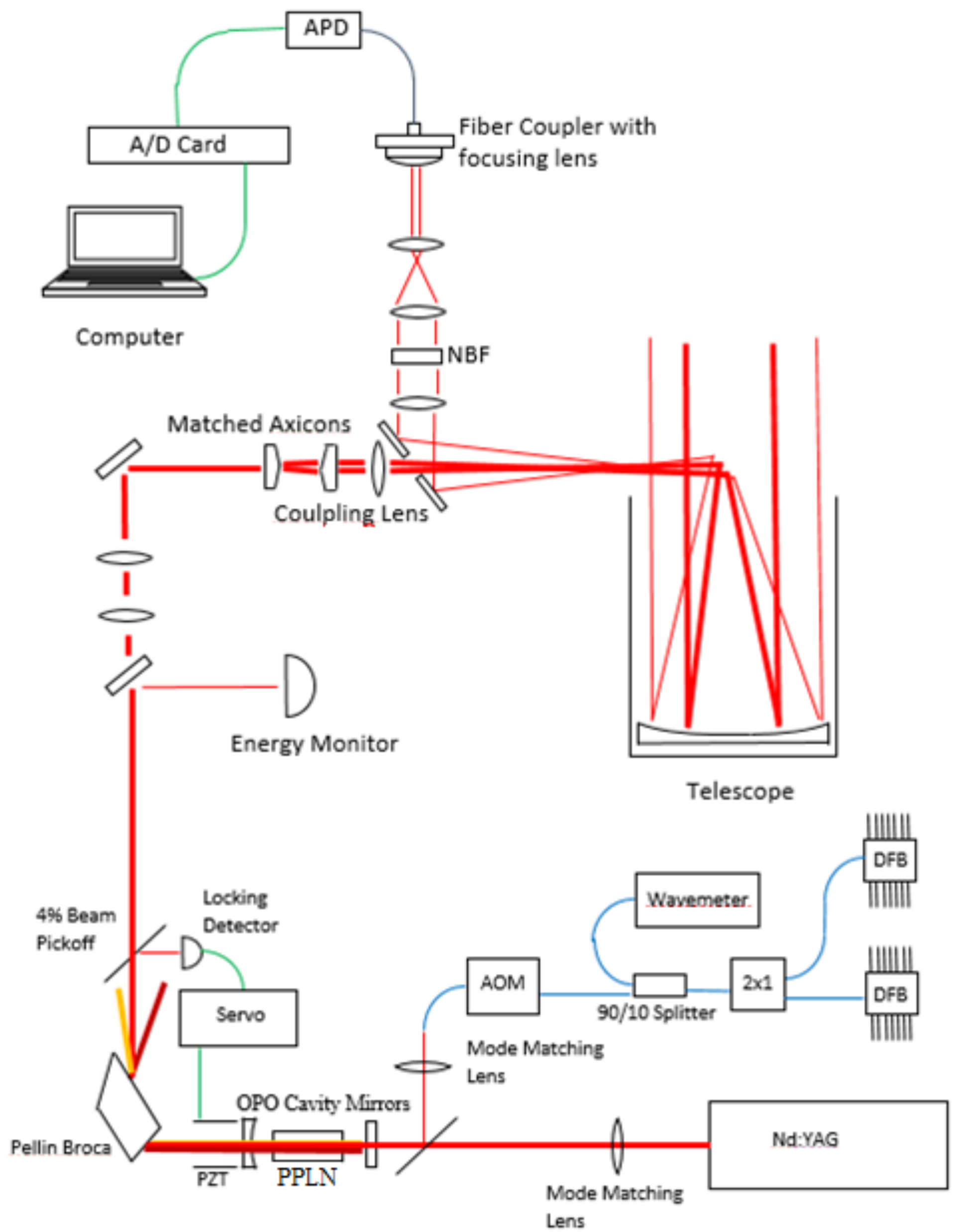


Figure 7.4 Schematic of the DIAL design.

Light scattered in the atmosphere is collected by the telescope and is sent back to the elliptic mirror. Only light scattered by the outer portion of the telescope mirror will be reflected from the elliptical mirror where the light is collimated, passes through a narrowband filter and is coupled into a multimode fiber that will deliver the collected light to an avalanche photodiode (APD). The output from the photodiode will be monitored using a high speed analog to digital (A/D) convertor which will be monitored via a computer.

A program will be developed in the Labview programming environment to operate the DIAL instrument. This program will allow the used to define the on-line and off-line wavelengths for locking and will collect and store an averaged return signal for the on-line and off-line data along with ancillary information including a time stamp and other relevant atmospheric parameters such as temperature and pressure. This program will allow data to be display to allow the user to assess the instrument operations easily. Data analysis software will then be developed using the Matlab programming environment.

CONCLUSIONS

To understand the climate change or global warming whether from anthropogenic or natural causes, we need to develop new tools and technologies for making accurate measurements of atmospheric components. Two technologies developed in this dissertation are the MPL which is a remote sensing for aerosol measurements and the OPO which can be utilized in a DIAL transmitter used for CO_2 and CH_4 measurements. The ability to accurately measure both aerosols and CO_2 is important for building a climate model which helps us to understand our climate system. According to IPCC's Fifth Assessment Report, aerosols currently remain at a low level of scientific understanding and still dominate the uncertainty in the total anthropogenic radiative forcing. As for CO_2, the full picture of the carbon cycle remains incomplete since many sources and sinks of CO_2 in nature as well as human contributions have neither been detected nor measured. Because of the spatial and temporal variations of atmospheric aerosols and greenhouse gases, small-scale, long-term measurements from various locations can complement a larger global climate monitoring effort. The instruments developed in this dissertation contributes to a small-scale remote sensing facility at Montana State University which has a goal of coordinating instruments and measurements for long-term observation studies of atmospheric aerosols and greenhouse gases from a rural continental site.

MPL

The MPL was successfully built including the DAQ software that incorporated scanning capability. The DAQ software embeds many new real-time plots which are useful for the system alignment. The software also embeds the mount control which allows a precise shooting direction and introduces a scanning capability to the system. In addition, the software code is better organized and is gradually built up from many smaller codes for easier tracking and understanding. The system efficiency is enhanced in terms of the increasing power of the transmitted beam and the new transmitter-receiver aligning technique which provides a good alignment in much less time. The system accuracy is improved by the utilization of the vertical-horizontal overlap matching technique. The system is more reliable since it was moved to a temperature controlled room which eliminates the alignment loss due to temperature fluctuations.

As presented in the MPL results section, good agreements between the total AODs retrieved from the MPL and that acquired from the AERONET's sun-sky photometer demonstrate a good MPL accuracy. However, we observed that the MPL lidar ratio are oftentimes much lower than the expected values. It is rather difficult for us at this point to gauge how accurate the MPL lidar ratio is since the lidar ratio acquired from the sun-sky photometer is sometimes not even realistic i.e. it exceeds 100 sr. Other than that issue, the MPL proved to provide a reliable total AOD which agree with AERONET's and correlate well with atmospheric conditions. In addition, it is capable of providing other useful information that the sun-sky photometer is incapable of, such as range resolved AOD, range resolved β_a, and a direct measurement of the aerosol boundary layer. These show that the MPL can be used in conjunction with the AERONET's sun-sky photometer for atmospheric aerosol study in the lower troposphere.

OPO

The OPO proved to have a good potential for use as a DIAL laser transmitter that requires an output energy of 3-mJ at 1571 nm with a repetition rate of 1 kHz and a linewidth on the order of 3 MHz (24.7 fm) in order for a DIAL to meet an error of less than 2%. Our results indicate that an OPO is the best choice for the DIAL transmitter due to its high output energy at 1571 nm and narrow linewidth. With dither locking and a seeding injection of 160-mW power, the OPO produces a maximum measured output energy of 2 mJ at an input pump energy of 11.5 mJ and predicts a possible maximum output energy of 2.5 mJ at an input pump energy of 20-mJ. The steady state of the OPO occurs between the input pump energies of 5.5-7.5 mJ when the conversion efficiency reaches 20%. The achievable narrowest linewidth from the OPO is 11.4 GHz (0.0941 nm) measured with a 0.06 nm resolution. Although the DIAL transmitter requirements are not quite met, the close results suggest a potential success of an OPO, especially when there is still room for improvements. To achieve a higher output energy and narrower linewidth, the OPO can be improved with a better choice of mirrors, better mode-matching, better locking, and higher-power seed laser. Without further narrowing the linewidth, the output energy can be increased with increased input pump energies. However, we must take precaution when tuning up the input pump energies as they shall not exceed the damage threshold of the crystal.

Based on good agreements between the SNLO model and OPO experimental data, SNLO is demonstrated to be a reliable source for predicting an OPO performance. However, before we can use SNLO to model an OPO, we must first experimentally acquire a d_{eff} value from an OPA. As an example, in this research, an estimated d_{eff} value of 13 pm/V with $\Delta k \approx 300$ m^{-1} is found when fitting the SNLO OPA model to OPA data. Using this value of d_{eff} and $\Delta k \approx 0$ as inputs, the SNLO OPO model yields the results that agree with the OPO experimental data.

Overall, the OPO results show promise for its use as a DIAL laser transmitter. With the mentioned OPO improvements, the DIAL requirements are very likely achievable. Optimization of an OPO cavity, input pump and seed intensities within the crystal, damage threshold of the crystal, and the generated output energy at 1571 nm as predicted by SNLO, is highly recommended for future work. This will help make an OPO to reach its full potential, prevent damaging the crystal, and identify necessary parts or equipment, which could save much time and cost.

REFERENCES CITED

1. Hinds, W.C., *Aerosol Technology Properties, Behavior, and Measurement of Airborne Particles*. Second ed. 1999, New York: John Wiley & Sons, Inc. 28.

2. Team, E.S.C. *Just 5 questions: Aerosols*. 2.16 [cited 2016 15 November]; Available from: http://climate.nasa.gov/news/215/just-5-questions-aerosols/.

3. Voiland, A. *Aerosols: Tiny Particles, Big Impact*. 2010 [cited 2016 11 November]; Available from: http://earthobservatory.nasa.gov/Features/Aerosols/.

4. Chin, M., et al., *Intercontinental transport of pollution and dust aerosols: implications for regional air quality.* Atmos. Chem. Phys., 2007. **7**(21): p. 5501-5517.

5. Edenhofer, O., R. Pichs-Madruga, Y. Sokona, E. Farahani, S. Kadner, K., A.A. Seyboth, I. Baum, S. Brunner, P. Eickemeier, B. Kriemann, J. Savolainen, S. Schlömer, C. von Stechow, T. Zwickel and J.C., and Minx, *IPCC, 2014: Climate Change 2014: Mitigation of Climate Change. Contribution of Working Group III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. 2014: Cambridge, United Kingdom and New York, NY, USA.

6. Robock, A., *Introduction: Mount Pinatubo as a Test of Climate Feedback Mechanisms*, in *Volcanism and the Earth's Atmosphere*. 2013, American Geophysical Union. p. 1-8.

7. RG Roosen, C.K., RJ Angione, *Worldwide Variations in Atmospheric Transmission: Baseline Results from Smithsonian Observations.* Bulletin of the American Meteorological Society 1973. **54**(4).

8. Holben, B.N., et al., *An emerging ground-based aerosol climatology: Aerosol optical depth from AERONET.* Journal of Geophysical Research: Atmospheres, 2001. **106**(D11): p. 12067-12097.

9. Flowers, E.C., R.A. McCormick, and K.R. Kurfis, *Atmospheric Turbidity over the United States, 1961–1966.* Journal of Applied Meteorology, 1969. **8**(6): p. 955-962.

10. Volz, F.E., *Some results of turbidity networks1.* Tellus, 1969. **21**(5): p. 625-630.

11. Holben, B. *AERONET Aerosol Robotic Network*. 2015 [cited 2016 17 November]; Available from: http://aeronet.gsfc.nasa.gov/new_web/system_descriptions_ instrument.html.

12. Agency, U.E.P. *Global Greenhouse Gas Emissions Data*. 2016 9 August 2016 [cited 2016 16 November]; Available from: https://www.epa.gov/ghgemissions/global-greenhouse-gas-emissions-data.

13. Voiland, A. *Why So Many Global Temperature Records?* 2015 21 January 2015 [cited 2016 16 November]; Available from: http://www.giss.nasa.gov/research/features/201501_gistemp/.

14. Laboratory, E.S.R. *Carbon Cycle Science*. 2016 [cited 2016 18 November]; Available from: http://www.esrl.noaa.gov/research/themes/carbon/.

15. Tans, P., *Global Greenhouse Gas Reference Network*, G.M.D. NOAA Earth System Research Laboratory, Editor. 2016.

16. U.S. Department of Commerce, N. *Global Greenhouse Gas Reference Network*. [cited 2016 November 18]; Available from: https://www.esrl.noaa.gov/gmd/ccgg/ggrn.php?projects%5B%5D=1&projects%5B%5D=2&projects%5B%5D=3&projects%5B%5D=4&projects%5B%5D=5&active=on.

17. Fernald, F.G., *Analysis of atmospheric lidar observations: some comments.* Applied Optics, 1984. **23**(5): p. 652-653.

18. Fernald, F.G., B.M. Herman, and J.A. Reagan, *Determination of Aerosol Height Distributions by Lidar.* Journal of Applied Meteorology, 1972. **11**(3): p. 482-489.

19. Vladimir Kovalev, W.E., *Elastic Lidar*. 2004, New Jersey: John Wiley & Sons, Inc. 615.

20. Holben, B. *Version 2 Direct Sun Algorithm*. [cited 2016 March 20]; Available from: https://aeronet.gsfc.nasa.gov/cgi-bin/type_one_station_opera_v2_new?site=Bozeman&nachal=2&level=1&place_code=10.

21. Holben, B. *Version 2 Inversion Products*. [cited 2017 March 20]; Available from: https://aeronet.gsfc.nasa.gov/cgi-bin/webtool_opera_v2_inv?stage=3&region=United_States_West&state=Montana&site=Bozeman&place_code=10&if_polarized=0.

22. Shaw, J. *Weather Station*. [cited 2016 November 28]; Available from: http://orsl.eps.montana.edu/weather/.

23. Wallace, J.M. and P.V. Hobbs, *Atmospheric Science: An Introductory Survey*. Second ed. 2006: Elsevier Academic Press.

24. Bohren, C.F. and D.R. Huffman, *Absorption and Scatterrting of Light by Small Particles*. 1983: John Wiley & Sons, Inc.

25. Zwillinger, D., *Standard Mathematical Tables and Formulae*. 31 ed. 2003, New York: CRC Press LLC.

26. Moen, D., *Two Channel Receiver Design and Implementation for a Ground Based Micro-Pulse Differential Absorption Lidar (DIAL) Instrument*, in *Electrical and Computer Engineering*. 2016, Montana State University: Bozeman, MT.

27. Stephens, G.L., *Remote Sensing of the Lower Atmosphere: An Introduction*. 1994, New York: Oxford University Press, Inc.

28. Goody, R.M. and Y.L. Yung, *Atmospheric Radiation: Theoretical Basis*. 1989, New York: Oxford University Press, Inc.

29. Nehrir, A.R., *Development of and eye-safe diode-laser-based micro-pulse differential absorption lidar (MP-DIAL) for atmospheric water-vapor and aerosol studies*. 2011, Montana State University: Montana

30. Rothman, L.S., et al., *The HITRAN molecular spectroscopic database: edition of 2000 including updates through 2001.* Journal of Quantitative Spectroscopy and Radiative Transfer. **82**(1–4): p. 5-44.

31. Phelan, R., et al., *Absorption line shift with temperature and pressure: impact on laser-diode-based H2O sensing at 1.393 µm.* Applied Optics, 2003. **42**(24): p. 4968-4974.

32. Lenoble, J., *Atmospheric Radiative Transfer (Studies in Geophysical Optics and Remote Sensing)*. 1993, India: Deepak Publication.

33. Thomas, G.E. and K. Stamnes, *Radiative Transfer in the Atmosphere and Ocean*. 1999, Cambridge, U.K.: Canbridge University Press.

34. Vidler, M. and J. Tennyson, *Accurate partition function and thermodynamic data for water.* The Journal of Chemical Physics, 2000. **113**(21): p. 9766-9771.

35. Johnson, W., K.S. Repasky, and J.L. Carlsten, *Micropulse differential absorption lidar for identification of carbon sequestration site leakage.* Applied Optics, 2013. **52**(13): p. 2994-3003.

36. Boyd, R.W., *Nonlinear Optics*. Second ed. 2003, San Diego, CA: Academic Press, Elsevier Science.

37. Jones, B.L., *Development of a Singly-Resonant Optical Parametric Oscillator for Carbon Cycle Science*, in *Electrical and Conputer Science Engineering*. 2015, Montana State University: Bozeman, MT.

38. Yariv, A., *Quantum Electronics*. Second ed. 1975, Canada: John Wiley & Sons, Inc.

39. Griffiths, D.J., *Introduction to Electrodynamics*. Third ed. 1999, New Jersey, USA: Pearson Prentice Hall.

40. Haggstrom, M. *Birefringence*. 2010 30 November 2016 [cited 2017 23 January]; Available from: https://en.wikipedia.org/wiki/Birefringence.

41. Midwinter, J.E. and J. Warner, *The effects of phase matching method and of uniaxial crystal symmetry on the polar distribution of second-order non-linear optical polarization.* British Journal of Applied Physics, 1965. **16**(8): p. 1135.

42. Sheng, Y., Koynov K., J. Dou, B. Ma, J. Li, D. Zhang. *Nonlinear Photonic Crystals*. [cited 2017 March 20]; Available from: http://www2.mpip-mainz.mpg.de/~koynov/koynov/research/nlo/nonlinear/nonlinear.html.

43. GlobalSpec, I. *Chapter 11: Gaussian Beam Imagery*. [cited 2017 March 20]; Available from: http://www.globalspec.com/reference/72509/203279/chapter-11-gaussian-beam-imagery.

44. Smith, A. [cited 2017 January, 25]; Available from: http://www.as-photonics.com/snlo.

45. Thorlabs, I. *Shearing Interferometers*. 2017 [cited 2017 January 31]; Available from: https://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=2970.

46. Repasky, K.S., et al., *Observational Studies of Atmospheric Aerosols over Bozeman, Montana, Using a Two-Color Lidar, a Water Vapor DIAL, a Solar Radiometer, and a Ground-Based Nephelometer over a 24-h Period.* Journal of Atmospheric and Oceanic Technology, 2011. **28**(3): p. 320-336.

47. Boyd, G.D. and D.A. Kleinman, *Parametric Interaction of Focused Gaussian Light Beams.* Journal of Applied Physics, 1968. **39**(8): p. 3597-3639.

48. Kuhn, K., *Laser Engineering*, ed. First. 1998: Prentice Hall.

49. Jones, B.L., *Development of a Singly-Resonant Optical Parametric Oscillator for Carbon Cycle Science*, in *Electrical and Computer Engineering*. 2015, Montana State University: Bozeman, MT.

50. Refaat, T.F., et al., *Performance evaluation of a 1.6-μm methane DIAL system from ground, aircraft and UAV platforms.* Optics Express, 2013. **21**(25): p. 30415-30432.

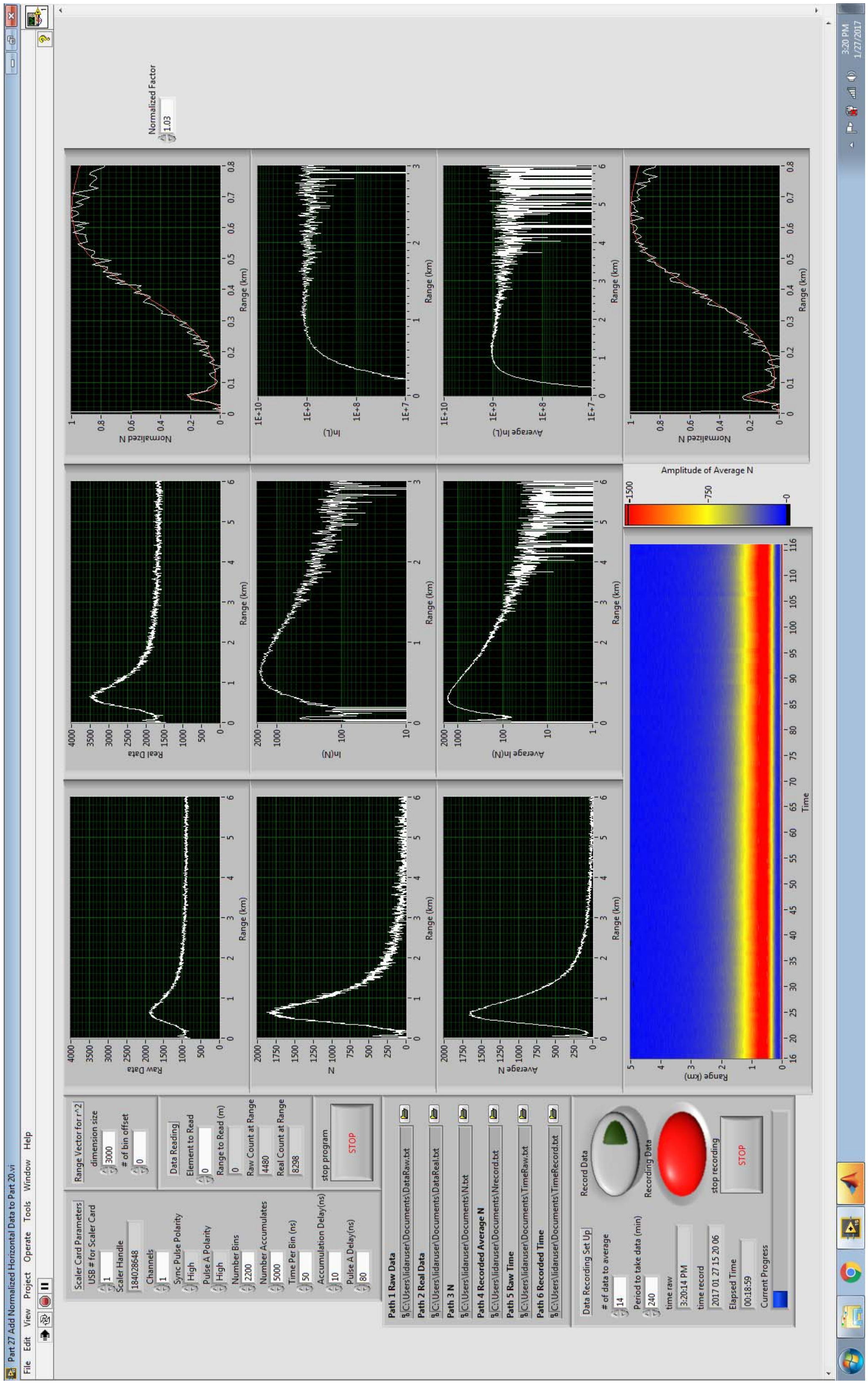
51. Ehret, G., et al., *Space-borne remote sensing of CO2, CH4, and N2O by integrated path differential absorption lidar: a sensitivity analysis.* Applied Physics B, 2008. **90**(3): p. 593-608.

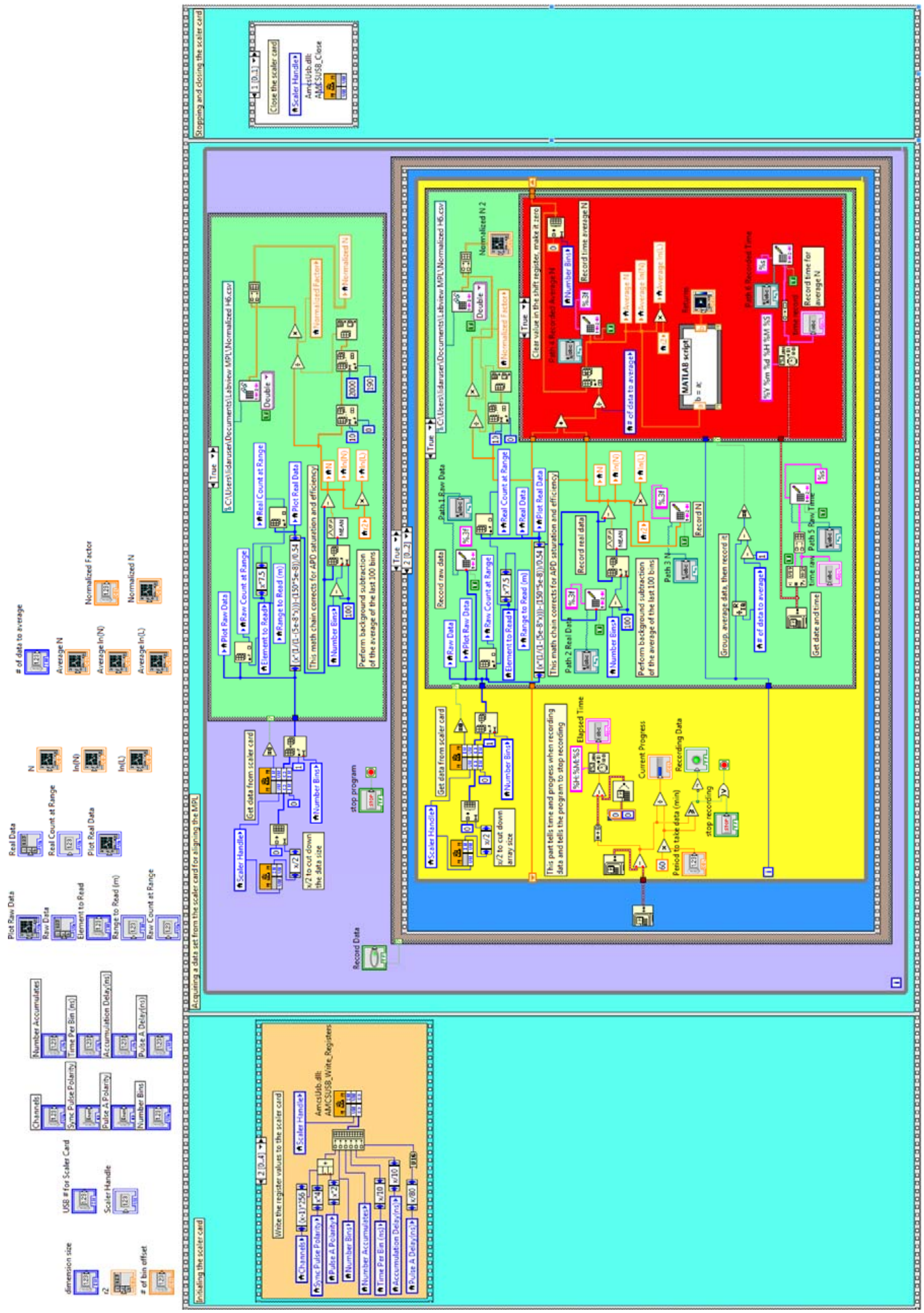
52. Institute, A.N.S., *American National Standard for Safe Use of Lasers, in: Z136.1-2007*. 2007, Laser Institute of America.

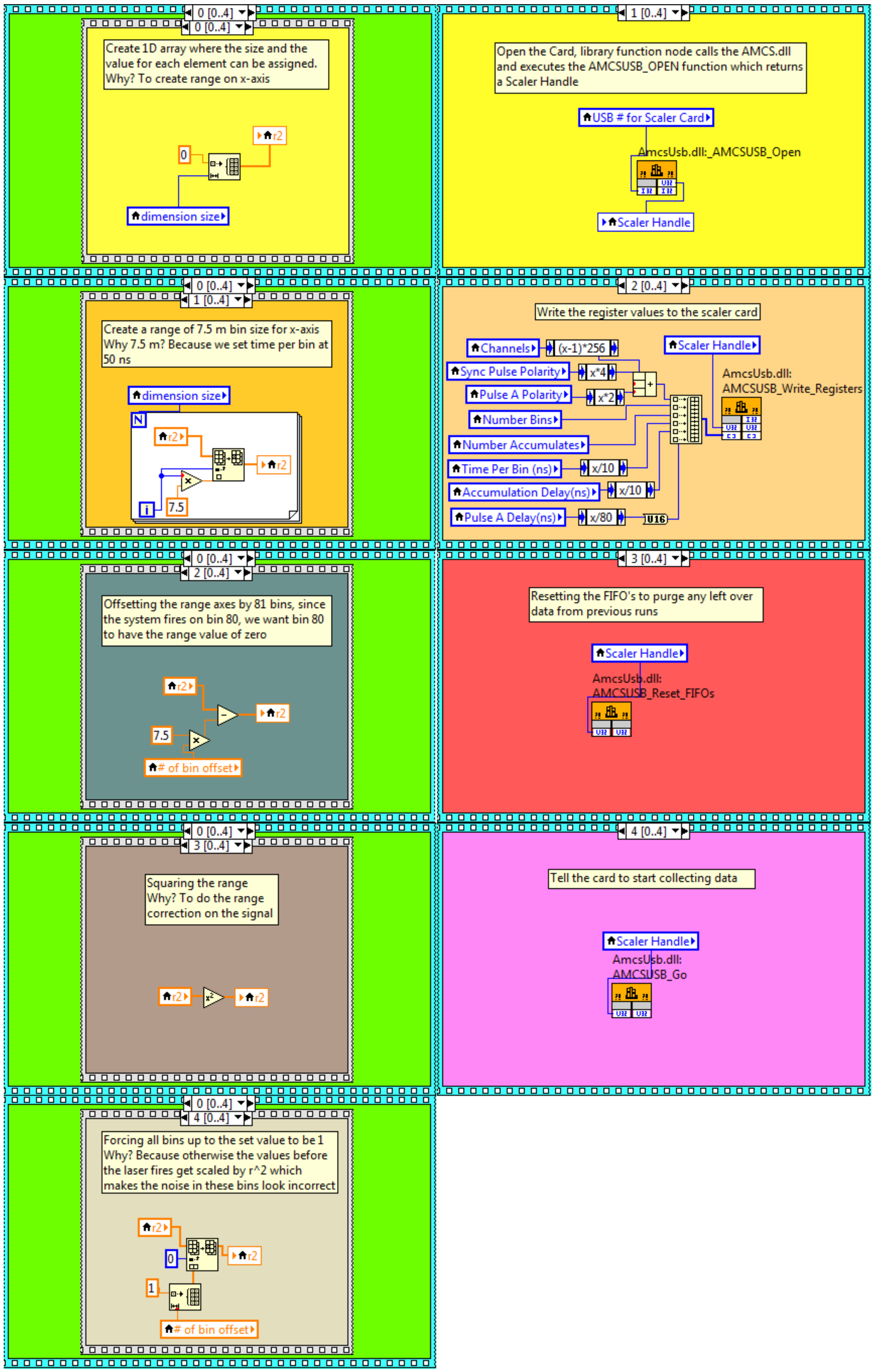
APPENDICES

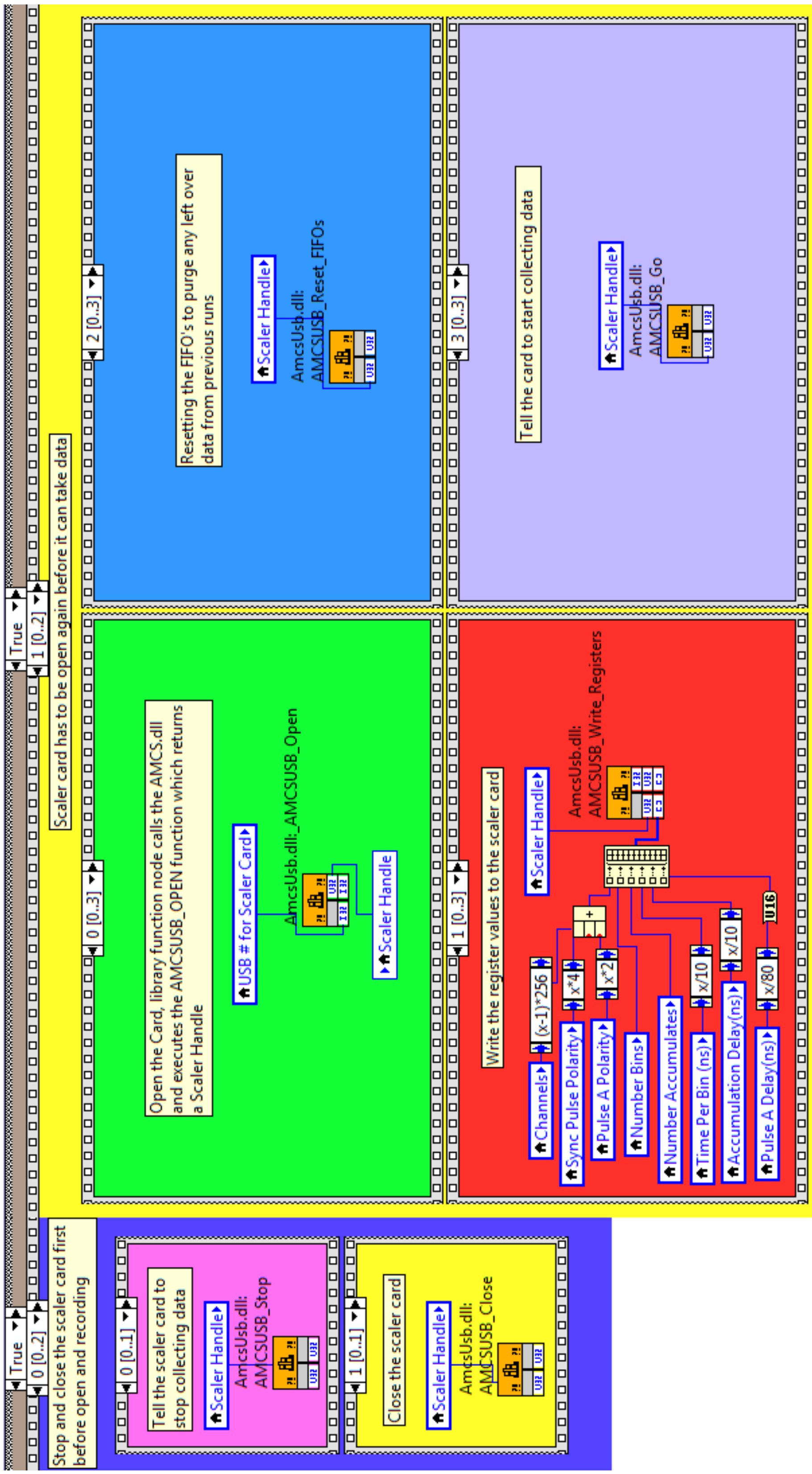
APPENDIX A

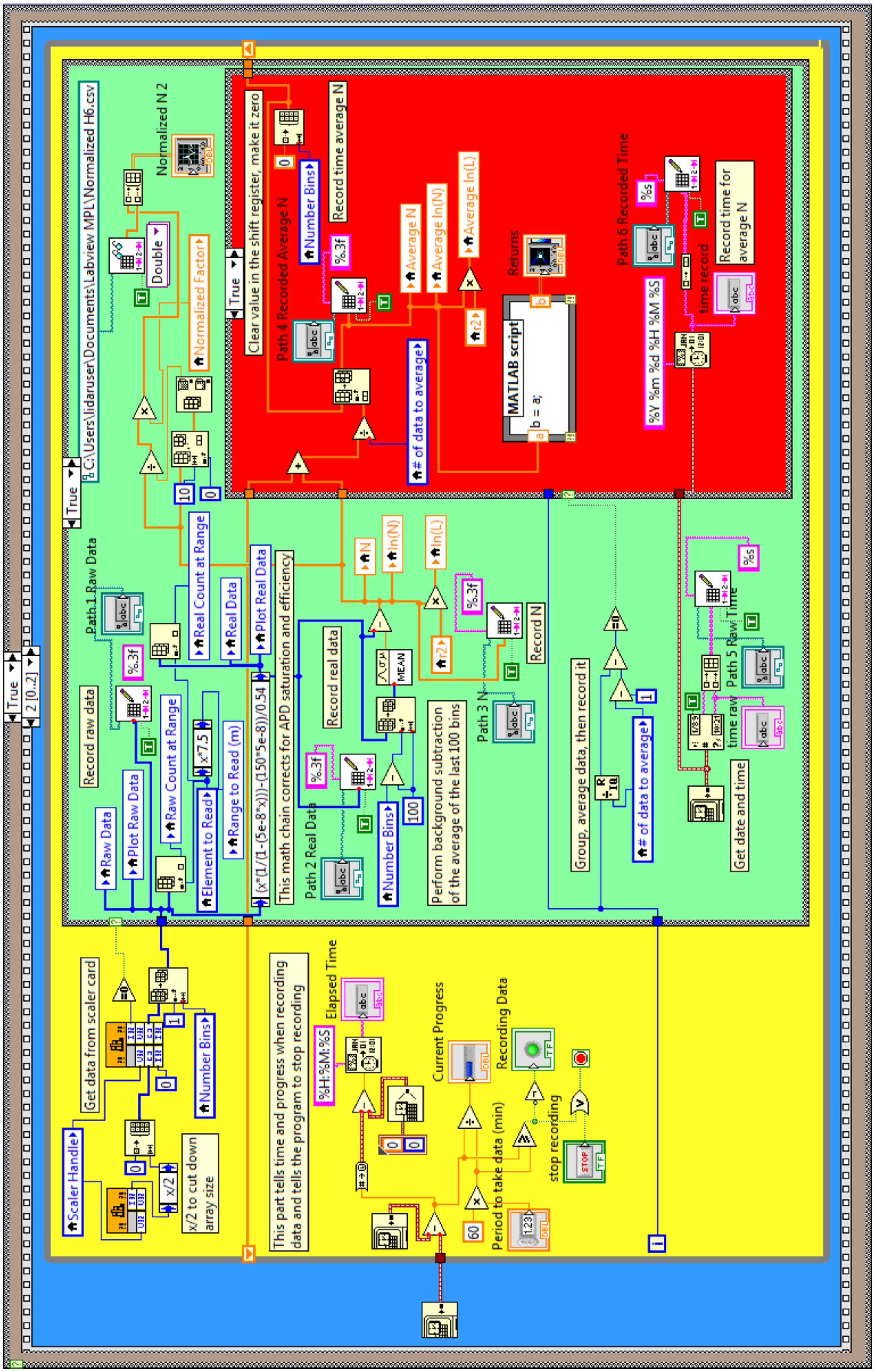
LABVIEW PROGRAM FOR MPL DATA ACQUISITION

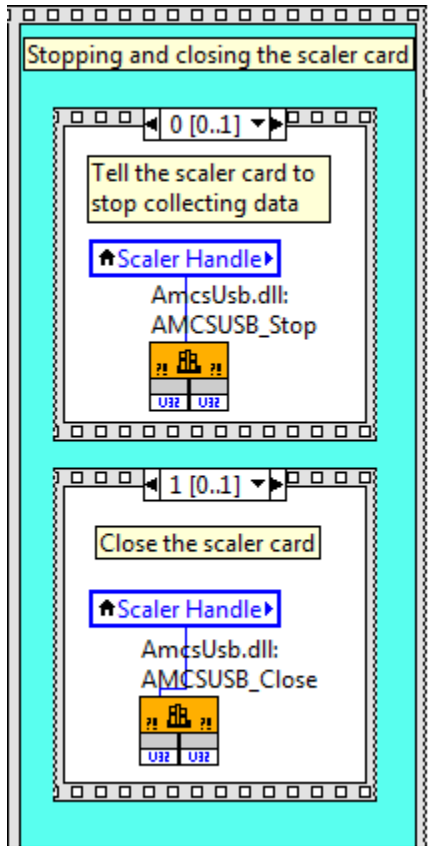

Part 27 Add Normalized Horizontal Data to Part 20.vi
File Edit View Project Operate Tools Window Help
Scaler Card Parameters
USB # for Scaler Card
Scaler Handle
184028648
Channels
Sync Pulse Polarity
High
Pulse A Polarity
High
Number Bins
2200
Number Accumulates
5000
Time Per Bin (ns)
50
Accumulation Delay(ns)
10
Pulse A Delay(ns)
80
Range Vector for r^2
dimension size
3000
# of bin offset
Data Reading
Element to Read
Range to Read (m)
Raw Count at Range
4480
Real Count at Range
8298
stop program
STOP
Path 1 Raw Data
C:\Users\lidaruser\Documents\DataRaw.txt
Path 2 Real Data
C:\Users\lidaruser\Documents\DataReal.txt
Path 3 N
C:\Users\lidaruser\Documents\N.txt
Path 4 Recorded Average N
C:\Users\lidaruser\Documents\Nrecord.txt
Path 5 Raw Time
C:\Users\lidaruser\Documents\TimeRaw.txt
Path 6 Recorded Time
C:\Users\lidaruser\Documents\TimeRecord.txt
Data Recording Set Up
# of data to average
14
Period to take data (min)
240
time raw
3:20:14 PM
time record
2017 01 27 15 20 06
Elapsed Time
00:18:59
Current Progress
Record Data
Recording Data
stop recording
Raw Data
Real Data
Normalized N
N
ln(N)
ln(L)
Average N
Average ln(N)
Average ln(L)
Range (km)
Time
Amplitude of Average N
Normalized Factor
1.03
3:20 PM
1/27/2017




0 [0..4]
0 [0..4]
Create 1D array where the size and the value for each element can be assigned. Why? To create range on x-axis
r2
0
dimension size
1 [0..4]
Open the Card, library function node calls the AMCS.dll and executes the AMCSUSB_OPEN function which returns a Scaler Handle
USB # for Scaler Card
AmcsUsb.dll:_AMCSUSB_Open
Scaler Handle
1 [0..4]
Create a range of 7.5 m bin size for x-axis Why 7.5 m? Because we set time per bin at 50 ns
dimension size
N
r2
r2
i
7.5
2 [0..4]
Write the register values to the scaler card
Channels
(x-1)*256
Scaler Handle
Sync Pulse Polarity
x*4
Pulse A Polarity
x*2
AmcsUsb.dll:
AMCSUSB_Write_Registers
Number Bins
Number Accumulates
Time Per Bin (ns)
x/10
Accumulation Delay(ns)
x/10
Pulse A Delay(ns)
x/80
U16
2 [0..4]
Offsetting the range axes by 81 bins, since the system fires on bin 80, we want bin 80 to have the range value of zero
r2
r2
7.5
# of bin offset
3 [0..4]
Resetting the FIFO's to purge any left over data from previous runs
Scaler Handle
AmcsUsb.dll:
AMCSUSB_Reset_FIFOs
3 [0..4]
Squaring the range Why? To do the range correction on the signal
r2
r2
4 [0..4]
Tell the card to start collecting data
Scaler Handle
AmcsUsb.dll:
AMCSUSB_Go
4 [0..4]
Forcing all bins up to the set value to be 1 Why? Because otherwise the values before the laser fires get scaled by r^2 which makes the noise in these bins look incorrect
r2
0
r2
1
# of bin offset

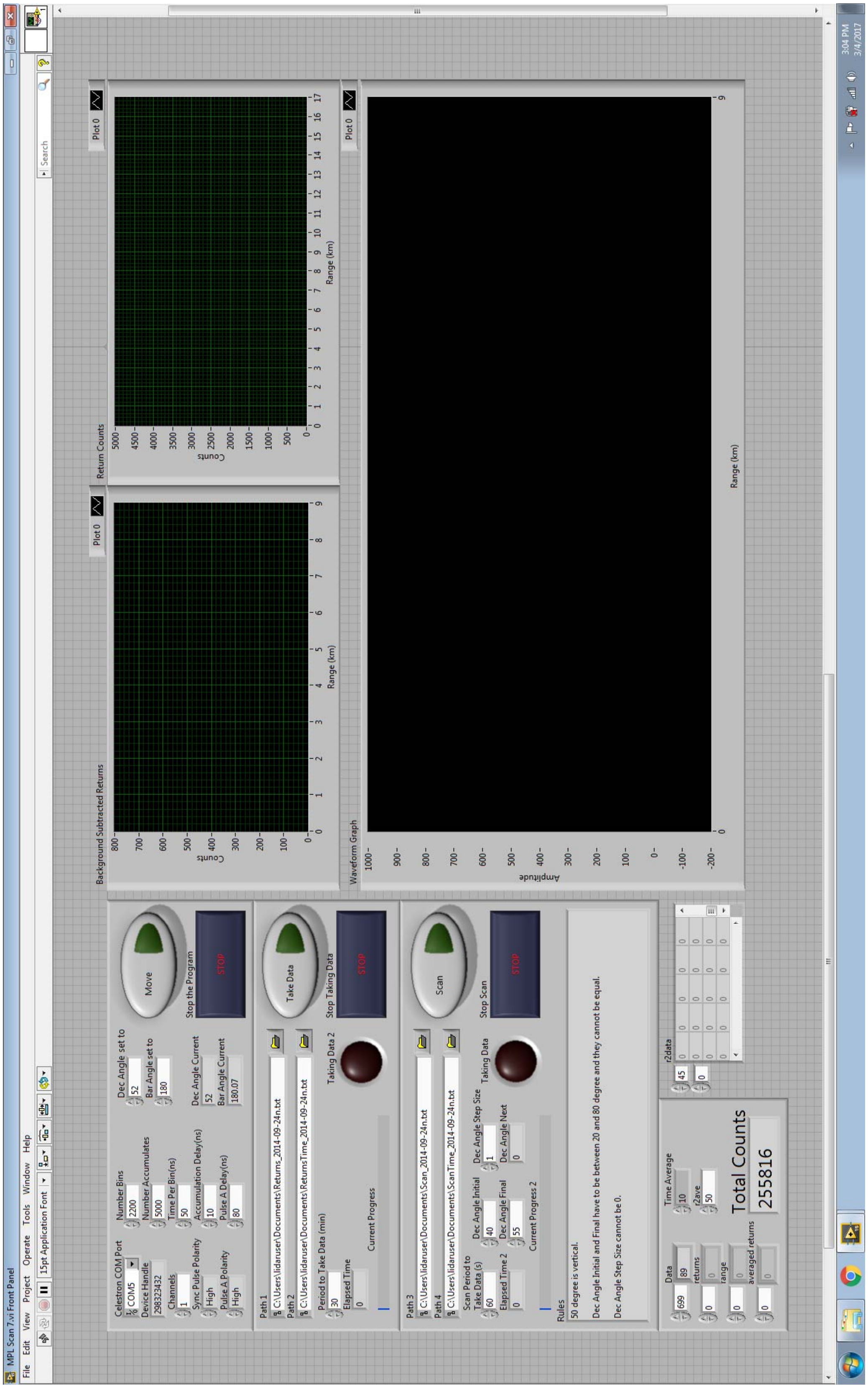

True
0 [0..2]
Stop and close the scaler card first before open and recording
0 [0..1]
Tell the scaler card to stop collecting data
Scaler Handle
AmcsUsb.dll: AMCSUSB_Stop
1 [0..1]
Close the scaler card
Scaler Handle
AmcsUsb.dll: AMCSUSB_Close
True
1 [0..2]
Scaler card has to be open again before it can take data
0 [0..3]
Open the Card, library function node calls the AMCS.dll and executes the AMCSUSB_OPEN function which returns a Scaler Handle
USB # for Scaler Card
AmcsUsb.dll:_AMCSUSB_Open
Scaler Handle
1 [0..3]
Write the register values to the scaler card
Scaler Handle
AmcsUsb.dll: AMCSUSB_Write_Registers
Channels
(x-1)*256
Sync Pulse Polarity
x*4
Pulse A Polarity
x*2
Number Bins
Number Accumulates
Time Per Bin (ns)
x/10
Accumulation Delay(ns)
x/10
Pulse A Delay(ns)
x/80
U16
2 [0..3]
Resetting the FIFO's to purge any left over data from previous runs
Scaler Handle
AmcsUsb.dll: AMCSUSB_Reset_FIFOs
3 [0..3]
Tell the card to start collecting data
Scaler Handle
AmcsUsb.dll: AMCSUSB_Go

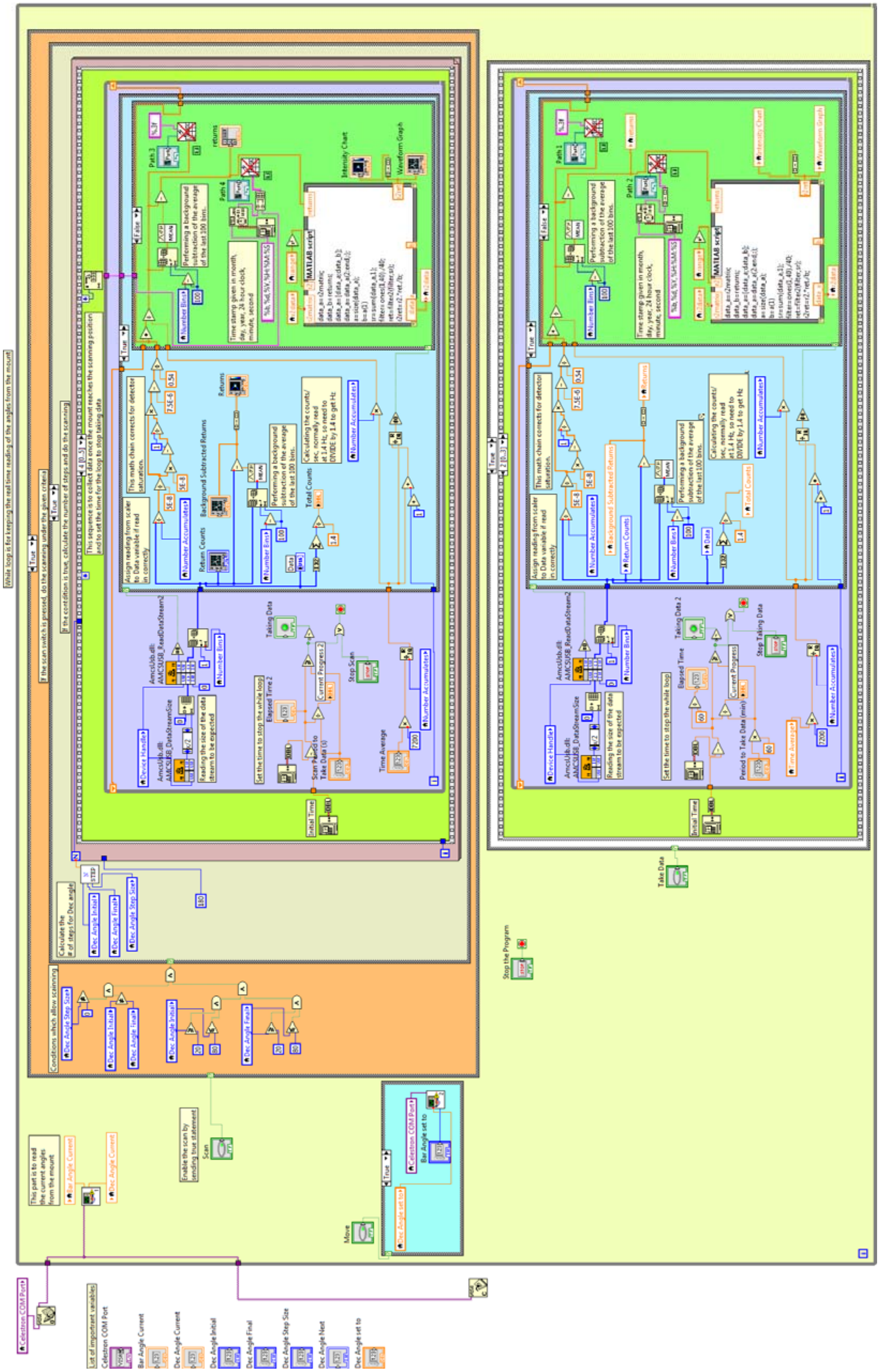

True
2 [0..2]
True
C:\Users\lidaruser\Documents\Labview MPL\Normalized H6.csv
Normalized N 2
Double
Normalized Factor
True
Clear value in the shift register, make it zero
Path 4 Recorded Average N
Number Bins
Record time average N
%.3f
Average N
Average ln(N)
Average ln(L)
r2
Returns
MATLAB script
b = a;
# of data to average
Path 6 Recorded Time
%s
Record time for average N
%Y %m %d %H %M %S
time record
Path.1 Raw Data
%.3f
Record raw data
Raw Data
Plot Raw Data
Raw Count at Range
Element to Read
Range to Read (m)
x*7.5
Real Count at Range
Real Data
Plot Real Data
(x*(1/(1-(5e-8*x)))-(150*5e-8))/0.54
This math chain corrects for APD saturation and efficiency
Path 2 Real Data
%.3f
Record real data
MEAN
Number Bins
100
Perform background subtraction of the average of the last 100 bins
Path 3 N
N
ln(N)
ln(L)
%.3f
r2
Record N
Group, average data, then record it
# of data to average
%s
Path 5 Raw Time
time raw
Get date and time
Scaler Handle
Get data from scaler card
Number Bins
x/2
x/2 to cut down array size
This part tells time and progress when recording data and tells the program to stop recording
%H:%M:%S
Elapsed Time
Current Progress
Recording Data
stop recording
Period to take data (min)
60

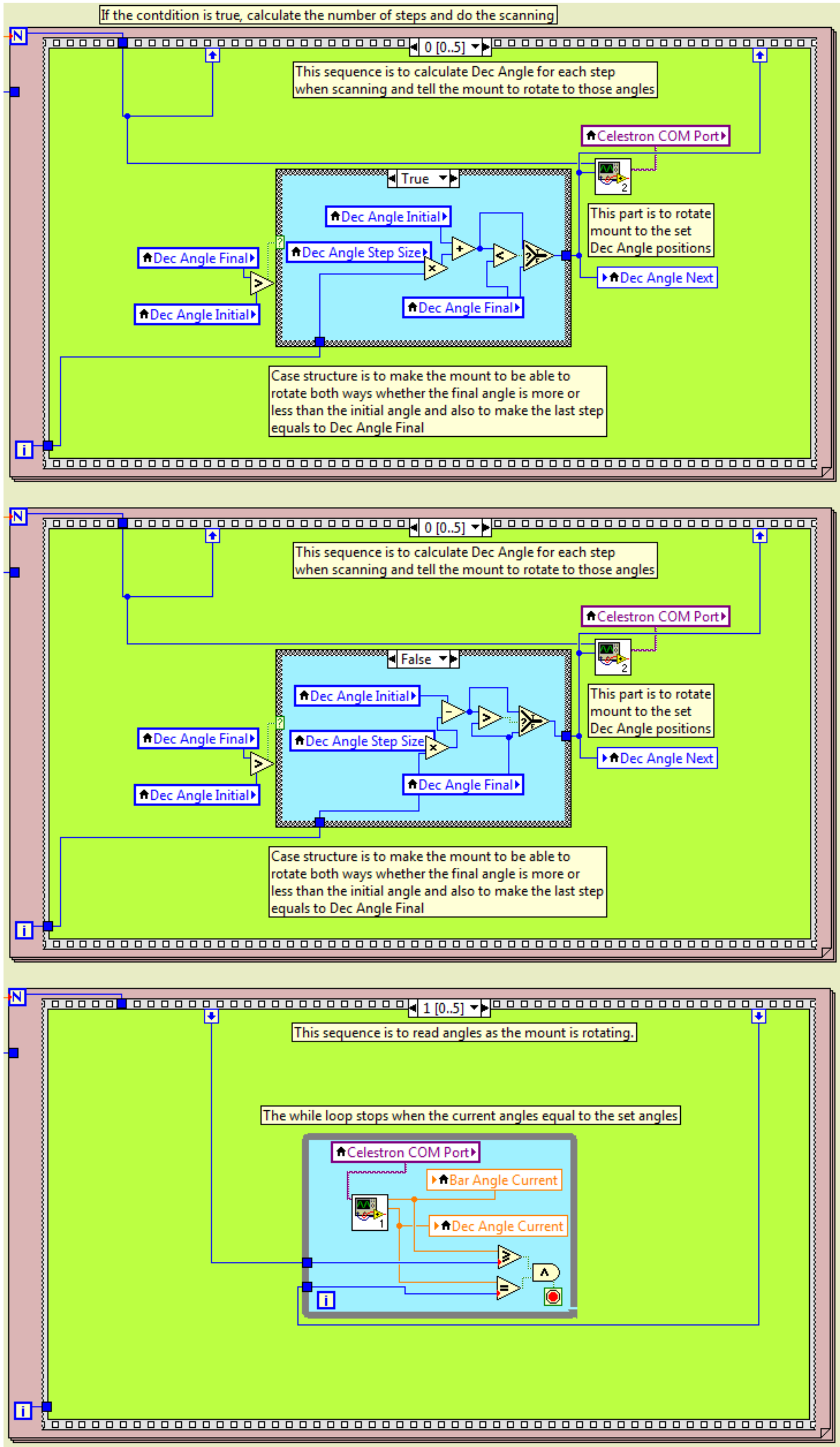

Stopping and closing the scaler card
0 [0..1]
Tell the scaler card to stop collecting data
Scaler Handle
AmcsUsb.dll: AMCSUSB_Stop
U32
U32
1 [0..1]
Close the scaler card
Scaler Handle
AmcsUsb.dll: AMCSUSB_Close
U32
U32

APPENDIX B

LABVIEW PROGRAAM FOR MPL SCAN


MPL Scan 7.vi Front Panel
File Edit View Project Operate Tools Window Help
15pt Application Font
Celestron COM Port
COM5
Device Handle
298323432
Channels
1
Sync Pulse Polarity
High
Pulse A Polarity
High
Number Bins
2200
Number Accumulates
5000
Time Per Bin(ns)
50
Accumulation Delay(ns)
10
Pulse A Delay(ns)
80
Dec Angle set to
52
Bar Angle set to
180
Dec Angle Current
52
Bar Angle Current
180.07
Move
Stop the Program
STOP
Path 1
C:\Users\lidaruser\Documents\Returns_2014-09-24n.txt
Path 2
C:\Users\lidaruser\Documents\ReturnsTime_2014-09-24n.txt
Period to Take Data (min)
30
Elapsed Time
0
Current Progress
Taking Data 2
Take Data
Stop Taking Data
STOP
Path 3
C:\Users\lidaruser\Documents\Scan_2014-09-24n.txt
Path 4
C:\Users\lidaruser\Documents\ScanTime_2014-09-24n.txt
Scan Period to Take Data (s)
60
Elapsed Time 2
0
Dec Angle Initial
40
Dec Angle Final
55
Current Progress 2
Dec Angle Step Size
1
Dec Angle Next
0
Taking Data
Scan
Stop Scan
STOP
Rules
50 degree is vertical.
Dec Angle Initial and Final have to be between 20 and 80 degree and they cannot be equal.
Dec Angle Step Size cannot be 0.
Data
699
89
returns
range
averaged returns
Time Average
10
r2ave
50
Total Counts
255816
r2data
45
Background Subtracted Returns
Counts
Range (km)
Plot 0
Return Counts
Waveform Graph
Amplitude
3:04 PM
3/4/2017




If the contdition is true, calculate the number of steps and do the scanning
0 [0..5]
This sequence is to calculate Dec Angle for each step when scanning and tell the mount to rotate to those angles
Celestron COM Port
True
Dec Angle Initial
Dec Angle Step Size
Dec Angle Final
Dec Angle Initial
Dec Angle Final
This part is to rotate mount to the set Dec Angle positions
Dec Angle Next
Case structure is to make the mount to be able to rotate both ways whether the final angle is more or less than the initial angle and also to make the last step equals to Dec Angle Final
0 [0..5]
This sequence is to calculate Dec Angle for each step when scanning and tell the mount to rotate to those angles
Celestron COM Port
False
Dec Angle Initial
Dec Angle Step Size
Dec Angle Final
Dec Angle Initial
Dec Angle Final
This part is to rotate mount to the set Dec Angle positions
Dec Angle Next
Case structure is to make the mount to be able to rotate both ways whether the final angle is more or less than the initial angle and also to make the last step equals to Dec Angle Final
1 [0..5]
This sequence is to read angles as the mount is rotating.
The while loop stops when the current angles equal to the set angles
Celestron COM Port
Bar Angle Current
Dec Angle Current

APPENDIX C

MATLAB CODE FOR TEMPERATURE AND PRESSURE MODELS

```
% Code for temperature and pressure models over Bozeman based on a group
of Sonde data

% Read data from these filenames i.e. 2013-11-26-19_27_12.tsv and get rid
of the first 39 lines header
Sonde1 = dlmread('2013-11-26-19_27_12.tsv','\t',39,0);
Sonde2 = dlmread('2013-11-26-15_43_35.tsv','\t',39,0);
Sonde3 = dlmread('2013-11-25-19_21_31.tsv','\t',39,0);
Sonde4 = dlmread('2013-11-25-00_00_37.tsv','\t',39,0);
Sonde5 = dlmread('2013-11-24-2_21_09.tsv','\t',39,0);
Sonde6 = dlmread('2013-11-24-19_37_23.tsv','\t',39,0);
Sonde7 = dlmread('2013-11-23-2_21_52.tsv','\t',39,0);
Sonde8 = dlmread('2013-11-23-21_48_24.tsv','\t',39,0);

% Sonde altitude in Meter (from column 7 of Sonde data)
z_Sonde1 = Sonde1(1:end,7);
z_Sonde2 = Sonde2(1:end,7);
z_Sonde3 = Sonde3(1:end,7);
z_Sonde4 = Sonde4(1:end,7);
z_Sonde5 = Sonde5(1:end,7);
z_Sonde6 = Sonde6(1:end,7);
z_Sonde7 = Sonde7(1:end,7);
z_Sonde8 = Sonde8(1:end,7);

% Shift Sonde altiudes to height from Bozeman (elevation 1524 m)
h_Sonde1 = z_Sonde1 - 1524;
h_Sonde2 = z_Sonde2 - 1524;
h_Sonde3 = z_Sonde3 - 1524;
h_Sonde4 = z_Sonde4 - 1524;
h_Sonde5 = z_Sonde5 - 1524;
h_Sonde6 = z_Sonde6 - 1524;
h_Sonde7 = z_Sonde7 - 1524;
h_Sonde8 = z_Sonde8 - 1524;

% Sonde temperature in Kelvin (from column 3 of Sonde data)
T_Sonde1 = Sonde1(1:end,3);
T_Sonde2 = Sonde2(1:end,3);
T_Sonde3 = Sonde3(1:end,3);
T_Sonde4 = Sonde4(1:end,3);
T_Sonde5 = Sonde5(1:end,3);
T_Sonde6 = Sonde6(1:end,3);
T_Sonde7 = Sonde7(1:end,3);
T_Sonde8 = Sonde8(1:end,3);

% Sonde pressure in Pascal (from column 8 of Sonde data)
P_Sonde1 = 100*Sonde1(1:end,8);
P_Sonde2 = 100*Sonde2(1:end,8);
P_Sonde3 = 100*Sonde3(1:end,8);
P_Sonde4 = 100*Sonde4(1:end,8);
P_Sonde5 = 100*Sonde5(1:end,8);
P_Sonde6 = 100*Sonde6(1:end,8);
P_Sonde7 = 100*Sonde7(1:end,8);
P_Sonde8 = 100*Sonde8(1:end,8);
```

```
% Constants for temperature ans pressure model calculation
g = 9.81; % m/s^2
Cp = 1004; % J/kg K
R = 287.053; % J/kg K

% Height vector in meter for temperature ans pressure model calculation
r = 7.5:7.5:7.5*3000;
% Heights where atmospheric temperature falls at lapse rate
ra = 7.5*round(1900/7.5);
rb = 7.5*round(10200/7.5);

% Assign a value for lapse rate
LapseRate = 0.0077; % K/m

% Assume surface temperature and pressure values for model calculations
Ts = Sonde4(1,3); % Surface temperature in Kelvin
Ps = P_Sonde4(1); % Surface pressure in Pascal

% Temperature Model Calculation
T1 = Ts*ones(1,find(r==ra)); % Make T=Ts for r<ra
T2 = Ts-(LapseRate*r(1,find(r==ra)+1:end)); % Make T = Ts-LapseRate*r for
r>ra
T2 = T2(1,:) + Ts(1,:) - T2(1,1); % Shift to make the continuity with T1
T = [T1(1,:), T2(1,:)]; % concatenate the matrices to get temperature
model
T(1,find(r==rb):end) = T(1,r==rb); % Make the final part of T (at r>rb)
constant

% Pressure Model Calculation for Bozeman condition
index = 1200;
P = Ps*(Ts./(Ts-LapseRate*r)).^(-0.0326/LapseRate); % Calculate pressure
in Pa
P(index:end)           =            P(index)*exp(-0.0353.*(r(index:end)-
index*7.5)./T(index));

% Pressure Model Calculation for average condition
P2   =   Ps*(Ts./(Ts-LapseRate*r)).^(-0.034164/LapseRate);   %   Calculate
pressure in Pa
P2(1466:end) = P2(1466)*exp(-0.034164.*(r(1466:end)-10995)./T(1466));

% Temperature plots
figure(1)
plot(T_Sonde1,h_Sonde1/1000,'g')
hold on;
plot(T,r/1000,'b')
hold on;
plot(T_Sonde1,h_Sonde1/1000,'g',...
     T_Sonde2,h_Sonde2/1000,'g',...
     T_Sonde3,h_Sonde3/1000,'g',...
     T_Sonde4,h_Sonde4/1000,'g',...
     T_Sonde5,h_Sonde5/1000,'g',...
     T_Sonde6,h_Sonde6/1000,'g',...
     T_Sonde7,h_Sonde7/1000,'g',...
```

```
     T_Sonde8,h_Sonde8/1000,'g');
hold on;
plot(T,r/1000,'b');
legend('Radiosonde Data','Model for Bozeman');
axis([200 290 0 20]);
xlabel('Temperature (K)');
ylabel('Height from Bozeman (km)');
title('Temperature Profile');
text(240,7.2,'Slope = -7.7 K/km','HorizontalAlignment','left')
grid on;

% Pressure plots
figure(2)
plot(P_Sonde1,h_Sonde1/1000,'g')
hold on;
plot(P2,r/1000,'r')
hold on;
plot(P,r/1000,'b')
hold on;
plot(P_Sonde1/1000,h_Sonde1/1000,'g',...
     P_Sonde2/1000,h_Sonde2/1000,'g',...
     P_Sonde3/1000,h_Sonde3/1000,'g',...
     P_Sonde4/1000,h_Sonde4/1000,'g',...
     P_Sonde5/1000,h_Sonde5/1000,'g',...
     P_Sonde6/1000,h_Sonde6/1000,'g',...
     P_Sonde7/1000,h_Sonde7/1000,'g',...
     P_Sonde8/1000,h_Sonde8/1000,'g','linewidth',1);
hold on;
plot(P2/1000,r/1000,'r');
hold on;
plot(P/1000,r/1000,'b');
legend('Radiosonde  Data','Model  for  Average  Condition  ','Model  for
Bozeman');
axis([0 90 0 20]);
xlabel('Pressure (kPa)');
ylabel('Height from Bozeman (km)');
title('Pressure Profile');
grid on;
```

APPENDIX D

MATLAB CODE FOR MPL OVERLAP RETREIVAL

```
% Code for overlap function retrieval

% Run the following codes first
run Step1_LoadHorizontal % To load data from the directory
run Step2_TimeAverage % To time average the data
run Step3_LinearlnLSegment % To set a range where ln(L) is linear

%% Calculations and plots
% Plot N
figure(101)
plot(r/1000,N,'b')
hold on;
line([ra/1000 ra/1000],ylim,'Color','g');
hold on;
line([rb/1000 rb/1000],ylim,'Color','g');
axis([0 10 0 10000]);
xlabel('Range (km)');
ylabel('N');
title(['Time average  ' filenameH{1} '  ' Time{1} ' - ' Time{end}]);
grid on;

% Calculate L
L = N.*r.^2;
L(L<1) = 1; % The value of L cannot be negative, otherwise Ln(L) is
invalid
lnL = log(L);

% Linear fit to ln(L)
yfit = FunctionLinearFit(r,ra,rb,lnL); % Call a function to perform a
linear fit

% Plot L and its linear fit on a log scale
figure(102)
semilogy(r/1000,L,'b');
hold on;
semilogy(r/1000,exp(yfit),'r');
hold on;
line([ra/1000 ra/1000],ylim,'Color','g');
hold on;
line([rb/1000 rb/1000],ylim,'Color','g');
axis([0 10 6*10^8 10^10]);
xlabel('Range (km)');
ylabel(texlabel('L'));
title(['Time average  ' filenameH{1} '  ' Time{1} ' - ' Time{end}]);
grid on;
dim = [.353 .46 .2 .2];
str = 'y = -0.000122x + 22.171454'; % These slope and intercept are
acquired from the linear fit line
annotation('textbox',dim,'String',str,'FitBoxToText','on');

% Plot L and its linear fit on a linear scale
figure(103)
plot(r/1000,lnL,'b');
```

```
hold on;
plot(r/1000,yfit,'r');
hold on;
line([ra/1000 ra/1000],ylim,'Color','g');
hold on;
line([rb/1000 rb/1000],ylim,'Color','g');
axis([0 10 log(6*10^8) log(10^10)]);
xlabel('Range (km)');
ylabel(texlabel('ln(L)'));
title(['Time average  ' filenameH{1} '  ' Time{1} ' - ' Time{end}]);
grid on;
dim = [.353 .46 .2 .2];
str = 'y = -0.000122x + 22.171454'; % These slope and intercept are
acquired from the linear fit line
annotation('textbox',dim,'String',str,'FitBoxToText','on');

% Find the ratio between ln(L) and the linear fit line to take a look at
the error
Ratio = lnL./yfit;

% Plot the ratio
figure(104)
plot(r/1000,Ratio,'b');
hold on;
plot(r/1000,1,'r');
hold on;
line([ra/1000 ra/1000],ylim,'Color','g');
hold on;
line([rb/1000 rb/1000],ylim,'Color','g');
axis([0 10 0.98 1.02]);
xlabel('Range (km)');
ylabel('Ratio');
title(['Time average  ' filenameH{1} '  ' Time{1} ' - ' Time{end}]);
grid on;

% Calculate the overlap function
O = FunctionOverlap(r,ra,rb,lnL); % Call a function to perform an ovelap
calculation

% Plot O to see what it looks like first
figure(105)
plot(r/1000,O,'b');
axis([0 10 0 1.2]);
xlabel('Range (km)');
ylabel('O');
title(['Time average  ' filenameH{1} '  ' Time{1} ' - ' Time{end}]);
grid on;

% Modify O by setting O=1 in the full overlap region
Omodify = FunctionOverlapModify(r,ra,rb,rc,lnL); % Call a function to
modify the overlap function
index = round(rc/7.5);
```

```
% Plot O with the modified O
figure(106)
plot(r(1:index)/1000,O(1:index),'b','linewidth',3);
hold on;
plot(r(index:end)/1000,O(index:end),'b','linewidth',1);
hold on;
plot(r(1:index)/1000,Omodify(1:index),'r','linewidth',1);
hold on;
plot(r(index:end)/1000,Omodify(index:end),'r','linewidth',1);
axis([0 10 0 1.2]);
xlabel('Range (km)');
ylabel('O');
title(['Time average  ' filenameH{1} '   ' Time{1} ' - ' Time{end}]);
grid on;

% Apply the overlap correction to N and L
LO = L./O;
NO = N./O;

% Plot N and L with overlap correction
figure(107)
h                                                                          =
semilogy(r/1000,N,'g',r/1000,NO,'k',r/1000,L/1000000,'b',r/1000,LO/(100
0000),'r');
h(1).LineWidth = 2;
h(3).LineWidth = 2;
axis([0 10 1 10^5]);
legend('N','N/O',texlabel('L (km^2)'),'L/O (km^2)','Location','best');
xlabel('Range (km)');
ylabel('N, L');
title(['Time average  ' filenameH{1} '   ' Time{1} ' - ' Time{end}]);
grid on;

% Apply the modifyed overlap correction to N and L
LOmodify = L./Omodify;
NOmodify = N./Omodify;

% Plot N and L with modified overlap correction
figure(108)
h                                                                          =
semilogy(r/1000,N,'g',r/1000,NOmodify,'k',r/1000,L/1000000,'b',r/1000,L
Omodify/(1000000),'r');
h(1).LineWidth = 2;
h(3).LineWidth = 2;
axis([0 10 1 10^5]);
legend('N','N/O',texlabel('L (km^2)'),'L/O (km^2)','Location','best');
xlabel('Range (km)');
ylabel('N, L');
title(['Time average  ' filenameH{1} '   ' Time{1} ' - ' Time{end}]);
grid on;

%% Choice to save O to the directory (folder "Lidar")
```

```
choice    =    menu(['Save    Omodify    as    O'    filenameH{1}(1,2:end)
'.mat'],'Yes','No');
if choice==1
    save(fullfile(savdirO,['O' filenameH{1}(1,2:end)]),'Omodify');
else
end

clear prompt; clear dlg_title; clear num_lines; clear def; clear answer4;
clear answer5; clear i;
```

```
% This is Step1_LoadHorizontal code

% This code is to load horizontal data from the directory specified in
"Path"

run Path % Run the code named "Path" first
ref = 1;

%% Input
prompt = {'Enter Lidar horizontal file name'};
dlg_title = 'File Name';
num_lines = 1;
def = {'H6'};
answer1 = inputdlg(prompt,dlg_title,num_lines,def);

%% Assign file name and download data
filenameH = answer1(1);
N = dlmread([filenameH{1} '.txt'],'\t');
Time = importdata(['Time' filenameH{1} '.txt'],' ');
Time = cellstr(datestr(Time));
Timenum = datenum(Time);

clear prompt; clear dlg_title; clear num_lines; clear def; clear answer1
```

```
% This is Step2_TimeAverage code

% This code is to time average a set or several sets of both horizontal
and vertical data

if ref==1
    % Time window to average
    prompt = {'Enter start time','Enter end time'};
    dlg_title = 'Time Average';
    num_lines = 1;
    def = {Time{1},Time{end}};
    answer3 = inputdlg(prompt,dlg_title,num_lines,def);

    % Get start and end time
    Start = answer3{1};
    End = answer3{2};

    % Convert start and end time to numeric
    Startnum = datenum(Start);
    Endnum = datenum(End);

    % Find the index where Time is closest to Start and End
    indexStart       =       find(abs(Timenum-Startnum)==min(abs(Timenum-
Startnum)));
    indexEnd = find(abs(Timenum-Endnum)==min(abs(Timenum-Endnum)));

    % Take only data within the time window
    N = N(indexStart:indexEnd,:);
    Time = Time(indexStart:indexEnd,:);
    Timenum = datenum(Time);

    % Average the data
    N = mean(N);

    clear prompt; clear dlg_title; clear num_lines; clear def; clear
answer3
    clear  Start;  clear  End;  clear  Startnum;  clear  Endnum;  clear
indexStart; clear indexEnd;

elseif ref==2
    Nmean = cell(size(N,1),size(N,2));
    for i=1:size(N,1)
        Nmean{i,1} = mean(N{i,1});
    end
    N = Nmean;
    clear Nmean, clear i;

else
    % Average NH
    Nmean = cell(size(NH,1),size(NH,2));
    for i=1:size(NH,1)
        Nmean{i,1} = mean(NH{i,1});
```

```
    end
    NH = Nmean;
    clear Nmean, clear i;
    % Average NV
    Nmean = cell(size(NV,1),size(NV,2));
    for i=1:size(NV,1)
        Nmean{i,1} = mean(NV{i,1});
    end
    NV = Nmean;
    clear Nmean, clear i;
end
```

```
% This is Step3_LinearlnLSegment code

% This code is to set a range where ln(L) is linear

%% Input
prompt = {'Segment for linear ln(L) (m)','Full overlap begin (m)'};
dlg_title = 'Linear ln(L)';
num_lines = 1;
def = {'[1700 8000]','1700'};
answer4 = inputdlg(prompt,dlg_title,num_lines,def);

limit = str2num(answer4{1});
ra = limit(1);
rb = limit(2);
rc = str2double(answer4{2});

clear prompt; clear dlg_title; clear num_lines; clear def; clear limit;
```

```
% This is Path code

% This code is to add directories where data is saved and results to be
saved

% Input box asking to clear old information
prompt  =  {'Close  All  Figures?','Clear  Command  Window?','Clear  Work
Space?'};
dlg_title = 'Input';
num_lines = 1;
def = {'yes','yes','yes'};
answer = inputdlg(prompt,dlg_title,num_lines,def);
answerL = strcmp(answer,'yes');

if answerL(1)==1;
    close all;
else
end
if answerL(2)==1;
    clc;
else
end
if answerL(3)==1;
    clear;
else
end

addpath('C:\Users\Quantum\Documents\Research\MPL\Data\2016 Lidar');
addpath('C:\Users\Quantum\Documents\Research\MPL\Data\AERONET');
addpath('C:\Users\Quantum\Documents\Research\MPL\Data\ECE');
addpath('C:\Users\Quantum\Documents\Research\MPL\Data\Results 2015 V8 10
min');

savdirO = 'C:\Users\Quantum\Documents\Research\MPL\Data\2015 Lidar';
savdir = 'C:\Users\Quantum\Documents\Research\MPL\Data\Results';

%% Create range vector (m)
r = 7.5:7.5:7.5*2200;
```

```
% These are function codes

% Function for linear fit
function yfit = LinearFit(x,xa,xb,y) % Linear fit curve y(x) for segment
between xa and xb
    index = (x >= xa) & (x <= xb); % Get the index of the line segment
    p = polyfit(x(index),y(index),1); % Polynomial coefficient=1 for a
straight line
    yfit = p(2)+x.*p(1); % Compute the best-fit line
end

% Function for overlap function calculation
function O = Overlap(x,xa,xb,y) % Linear fit curve y(x) for segment
between xa and xb
    index = (x >= xa) & (x <= xb); % Get the index of the line segment
    p = polyfit(x(index),y(index),1); % Polynomial coefficient=1 for a
straight line
    yfit = p(2)+x.*p(1); % Compute the best-fit line
    O = exp(y-yfit); % Calculate O, can also use O=L./exp(p(2)+p(1)*r)
end

% Function for modified overlap function calculation
function O = ModifyOverlap(x,xa,xb,xc,y) % Full overlap begin at xc
    index = (x >= xa) & (x <= xb); % Get the index of the line segment
    p = polyfit(x(index),y(index),1); % Polynomial coefficient=1 for a
straight line
    yfit = p(2)+x.*p(1); % Compute the best-fit line
    O = exp(y-yfit); % Calculate O, can also use O=L./exp(p(2)+p(1)*r)

    c = find(x==7.5*round(xc/7.5)); % Find the cell to begin dropping O
    O = O(1:c(size(c,2))); % Cut O up to where r = xc
    Oextend = ones(1, 2200-c(size(c,2))); % Create a vector 1 where O is
dropped
    O = [O,Oextend]; % Replace O = 1 to O
end
```

APPENDIX E

MATLAB CODE FOR AEROSOL INVERSION

```
% Code for aerosol inversion contains 12 steps written in smaller
separated codes

% This is Step1_LoadSync code

% This code is to load data from lidar, weather station, and AERONET

run Path

%% Time zone correction for AERONET
choice = menu('Choose time zone for Bozeman','GMT-6','GMT-7');
if choice==1
    TZD = '6';
else
    TZD = '7';
end
%%
prompt = {'Vertical data file name'};
dlg_title = 'File Name';
num_lines = 1;
def = {'V29'};
answer1 = inputdlg(prompt,dlg_title,num_lines,def);

bar = waitbar(0,'Loading Lidar data, please wait');

% Import Lidar data
FilenameLidar = answer1{1};
waitbar(0.05,bar,'Loading Lidar data, please wait');
DataLidar = dlmread([FilenameLidar '.txt'],'\t');
waitbar(0.1,bar,'Loading Lidar data, please wait');
TimeLidar = dlmread(['Time' FilenameLidar '.txt'],' ');
waitbar(0.15,bar,'Loading Lidar data, please wait');
TimeLidar = cellstr(datestr(TimeLidar)); % Change to string
TimenumLidar = datenum(TimeLidar);

waitbar(0.2,bar,'Loading ECE data, please wait');

% Get # of days (how long) the Lidar data was taken, then use that to
% create the needed file names for ECE weather station and AERONET
day = datenum(TimeLidar{end}(1:10))-datenum(TimeLidar{1}(1:10))+1;
for i = 1:day
    date{i,1} = datestr(datenum(TimeLidar{1}(1:11))+i-1,'mm/dd/yyyy');
    FilenameECE{i,1} = ['ECE_' date{i}(7:10) date{i}(1:2) date{i}(4:5)];
    FilenameAOD{i,1} = ['AOD_' date{i}(7:10) date{i}(1:2) date{i}(4:5)];
    FilenameSSA{i,1} = ['SSA_' date{i}(7:10) date{i}(1:2)];
    FilenamePhaseFunction{i,1}    =    ['PhaseFunction_'    date{i}(7:10)
date{i}(1:2)];
end
waitbar(0.25,bar,'Loading ECE data, please wait');

% Import ECE data
for i = 1:size(FilenameECE,1)
```

```
    ECE{i,1} = xlsread([FilenameECE{i} '.csv']);
    ECE{i,1}(289:end,:) = []; % For full day use 289:end
    DataECE{i,1} = [ECE{i,1}(:,4),ECE{i,1}(:,6)];
    TimenumECE{i,1} = datenum(date{i})+datenum(ECE{i}(:,1));
end
waitbar(0.3,bar,'Loading ECE data, please wait');
FilenameECE = cell2mat(FilenameECE);
DataECE = cell2mat(DataECE);
DataECE(:,1) = 100*DataECE(:,1); % Change pressure unit
DataECE(:,2) = 273+DataECE(:,2); % Change temperature unit
TimenumECE = cell2mat(TimenumECE);
TimeECE = cellstr(datestr(TimenumECE));

waitbar(0.4,bar,'Loading AERONET data, please wait');

%% Import AERONET AOD
for i = 1:size(FilenameAOD,1)
    AOD{i,1} = importdata([FilenameAOD{i} '.xlsx']);
end
waitbar(0.45,bar,'Loading AERONET data, please wait');
for i = 1:size(AOD,1)
    for j = 1:size(AOD{i,1}.textdata)-5
        TimeAOD{i,1}{j,1}     =     [AOD{i,1}.textdata{j+5,1}     '     '
datestr(AOD{i,1}.data(j,1),'HH:MM:SS')];
    end
    TimenumAOD{i,1} = datenum(TimeAOD{i,1},'dd/mm/yyyy HH:MM:SS')- ...
        datenum(['00/01/0000 0' TZD ':00:00'],'dd/mm/yyyy HH:MM:SS'); %
Put date and time together,
                                                                   %convert
to numeric and subtract time zone difference
    AOD500{i,1} = AOD{i,1}.data(:,12);
    AOD675{i,1} = AOD{i,1}.data(:,6); %#ok<*SAGROW>
end
waitbar(0.55,bar,'Loading AERONET data, please wait');

FilenameAOD = cell2mat(FilenameAOD);
AOD500 = cell2mat(AOD500);
AOD675 = cell2mat(AOD675);
AOD532 = (532-500)/(675-500)*(AOD675-AOD500)+AOD500;
TimenumAOD = cell2mat(TimenumAOD);
TimeAOD = cellstr(datestr(TimenumAOD));
waitbar(0.6,bar,'Loading AERONET data, please wait');

% Import AERONET SSA
SSA = importdata([FilenameSSA{1} '.xlsx']);
for j = 1:size(SSA.textdata)-4
    TimeSSA{j,1}        =        [SSA.textdata{j+4,1}        '        '
datestr(SSA.data(j,1),'HH:MM:SS')];
end
waitbar(0.7,bar,'Loading AERONET data, please wait');

TimenumSSA = datenum(TimeSSA,'mm/dd/yyyy HH:MM:SS')- ...
```

```
    datenum(['00/01/0000 0' TZD ':00:00'],'dd/mm/yyyy HH:MM:SS'); % Put
date and time together,
    %convert to numeric and subtract time zone difference
TimeSSA = cellstr(datestr(TimenumSSA));
FilenameSSA = cell2mat(FilenameSSA);
SSA440 = SSA.data(:,3);
SSA675 = SSA.data(:,4);
SSA532 = (532-440)/(675-440)*(SSA675-SSA440)+SSA440;
waitbar(0.8,bar,'Loading AERONET data, please wait');

% Import AERONET phase function
PhaseFunction = importdata([FilenamePhaseFunction{1} '.xlsx']);
for j = 1:size(PhaseFunction.textdata)-4
    TimePhaseFunction{j,1}    =    [PhaseFunction.textdata{j+4,1}    '    '
datestr(PhaseFunction.data(j,1),'HH:MM:SS')];
end
waitbar(0.9,bar,'Loading AERONET data, please wait');

TimenumPhaseFunction = datenum(TimePhaseFunction,'mm/dd/yyyy HH:MM:SS')-
...
    datenum(['00/01/0000 0' TZD ':00:00'],'dd/mm/yyyy HH:MM:SS'); % Put
date and time together,
    %convert to numeric and subtract time zone difference
waitbar(1,bar,'Loading AERONET data, please wait');
TimePhaseFunction = cellstr(datestr(TimenumPhaseFunction));
FilenamePhaseFunction = cell2mat(FilenamePhaseFunction);
PhaseFunction440 = PhaseFunction.data(:,85);
PhaseFunction675 = PhaseFunction.data(:,168);
PhaseFunction532        =         (532-440)/(675-440)*(PhaseFunction675-
PhaseFunction440)+PhaseFunction440;

close(bar);
%%
clear prompt; clear dlg_title; clear num_lines; clear def; clear answer1;
clear bar; clear choice; clear date; clear day; clear ECE; clear i; clear
j;
clear AOD; clear PhaseFunction; clear SSA;
```

```
% This is Step2_TimeSync code

% This code is to synchronize the time for lidar, weather station, and AERONET data

run Step1_LoadSync

%% Create time sync
prompt = {'Enter sync step (HH:MM:SS)','Enter sync start time','Enter sync end time'};
dlg_title = 'Sync';
num_lines = 1;
def = {'00:05:00',TimeLidar{1},TimeLidar{end}};
answer3 = inputdlg(prompt,dlg_title,num_lines,def);

step = answer3{1};
stepnum = datenum(['01/00/0000 ' step]);
Start = answer3{2};
End = answer3{3};
periodnum = datenum(End)-datenum(Start);
period = datestr(periodnum, 'dd HH:MM:SS');
TimeSync = (1:1:1+round(periodnum/stepnum)) - 1;
TimenumSync = stepnum*TimeSync' + datenum(Start);
TimeSync = cellstr(datestr(TimenumSync));

f = figure;
movegui(f,'northwest');
plot(TimenumECE,0.4+zeros(1,size(TimenumECE,1)),'b*',...
    TimenumAOD,0.3+zeros(1,size(TimeAOD,1)),'r*',...
    TimenumSSA,0.2+zeros(1,size(TimeSSA,1)),'k*',...
    TimenumLidar,0.1+zeros(1,size(TimeLidar,1)),'g*',...
    TimenumSync,zeros(1,size(TimeSync,1)),'m*');
xlim([min([TimenumECE;    TimenumAOD;    TimenumLidar;    TimenumSync])
max([TimenumECE; TimenumAOD; TimenumLidar; TimenumSync])]);
ylim([-0.4 1.4]);
datetick('x','HHPM','keeplimits');
legend('Weather       Station','AERONET       AOD','AERONET       SSA,
PFN','Lidar','Sync');
xlabel('Time (MDT)');
ylabel('Available Data Point');
title({['Weather Station ' TimeECE{1} ' - ' TimeECE{end}];...
       ['           AERONET ' TimeAOD{1} ' - ' TimeAOD{end}];...
       ['                  Lidar ' TimeLidar{1} ' - ' TimeLidar{end}];...
       ['                    Sync ' TimeSync{1} ' - ' TimeSync{end}]});
grid on;

%% Syncronize Lidar data by interpolation
DataLidarinterp = zeros(size(TimeSync,1),size(r,2));
for i = 1:size(DataLidar,2)
    DataLidarinterp(:,i)                                                 =
interp1(TimenumLidar,DataLidar(:,i),TimenumSync);
end
```

```
%% Syncronize ECE data by interpolation
DataECEinterp = zeros(size(TimeSync,1),size(r,2));
for i = 1:size(DataECE,2)
    DataECEinterp(:,i) = interp1(TimenumECE,DataECE(:,i),TimenumSync);
end

f = figure;
movegui(f,'north');
plot(TimenumECE,DataECE(:,2),'b','LineWidth',2);
hold on;
plot(TimenumSync,DataECEinterp(:,2),'mo');
xlim([TimenumECE(1) TimenumECE(end)]);
ylim('auto');
datetick('x','HHPM','keeplimits');
legend('Real Data','Sync Data','Location','Best');
xlabel('Time (MDT)');
ylabel('T_s (K)');
title({['Weather Station ' TimeECE{1} ' - ' TimeECE{end}];...
       ['           AERONET ' TimeAOD{1} ' - ' TimeAOD{end}];...
       ['                  Lidar ' TimeLidar{1} ' - ' TimeLidar{end}];...
       ['                    Sync ' TimeSync{1} ' - ' TimeSync{end}]});
grid on;

f = figure;
movegui(f,'northeast');
plot(TimenumECE,DataECE(:,1)/1000,'b','LineWidth',2);
hold on;
plot(TimenumSync,DataECEinterp(:,1)/1000,'mo');
xlim([TimenumECE(1) TimenumECE(end)]);
ylim('auto');
datetick('x','HHPM','keeplimits');
legend('Real Data','Sync Data','Location','Best');
xlabel('Time (MDT)');
ylabel('P_s (kPa)');
title({['Weather Station ' TimeECE{1} ' - ' TimeECE{end}];...
       ['           AERONET ' TimeAOD{1} ' - ' TimeAOD{end}];...
       ['                  Lidar ' TimeLidar{1} ' - ' TimeLidar{end}];...
       ['                    Sync ' TimeSync{1} ' - ' TimeSync{end}]});
grid on;

%% Syncronize AERONET data by interpolation
AOD532interp = interp1(TimenumAOD,AOD532,TimenumSync);
SSA532interp = interp1(TimenumSSA,SSA532,TimenumSync);
PhaseFunction532interp                                                        =
interp1(TimenumPhaseFunction,PhaseFunction532,TimenumSync);

f = figure;
movegui(f,'southwest');
plot(TimenumAOD,AOD500,'b*-',TimenumAOD,AOD532,'g*-
',TimenumAOD,AOD675,'r*-');
hold on;
plot(TimenumSync,AOD532interp,'mo');
xlim([TimenumECE(1) TimenumECE(end)]);
```

```
ylim('auto');
datetick('x','HHPM','keeplimits');
legend('500 nm','532 nm','675 nm','Sync Data','Location','Best');
xlabel('Time (MDT)');
ylabel('AERONET AOD');
title({['Weather Station ' TimeECE{1} ' - ' TimeECE{end}];...
       ['           AERONET ' TimeAOD{1} ' - ' TimeAOD{end}];...
       ['                  Lidar ' TimeLidar{1} ' - ' TimeLidar{end}];...
       ['                    Sync ' TimeSync{1} ' - ' TimeSync{end}]});
grid on;

f = figure;
movegui(f,'south');
plot(TimenumSSA,SSA440,'b*-',TimenumSSA,SSA532,'g*-
',TimenumSSA,SSA675,'r*-');
hold on;
plot(TimenumSync,SSA532interp,'mo');
xlim([TimenumECE(1) TimenumECE(end)]);
ylim('auto');
datetick('x','HHPM','keeplimits');
legend('440 nm','532 nm','675 nm','Sync Data','Location','Best');
xlabel('Time (MDT)');
ylabel('AERONET SSA');
title({['Weather Station ' TimeECE{1} ' - ' TimeECE{end}];...
       ['           AERONET ' TimeAOD{1} ' - ' TimeAOD{end}];...
       ['                  Lidar ' TimeLidar{1} ' - ' TimeLidar{end}];...
       ['                    Sync ' TimeSync{1} ' - ' TimeSync{end}]});
grid on;

f = figure;
movegui(f,'southeast');
plot(TimenumPhaseFunction,PhaseFunction440,'b*-
',TimenumPhaseFunction,PhaseFunction532,'g*-
',TimenumPhaseFunction,PhaseFunction675,'r*-');
hold on;
plot(TimenumSync,PhaseFunction532interp,'mo');
xlim([TimenumECE(1) TimenumECE(end)]);
ylim('auto');
datetick('x','HHPM','keeplimits');
legend('440 nm','532 nm','675 nm','Sync Data','Location','Best');
xlabel('Time (MDT)');
ylabel('AERONET 180^o Phase Function');
title({['Weather Station ' TimeECE{1} ' - ' TimeECE{end}];...
       ['           AERONET ' TimeAOD{1} ' - ' TimeAOD{end}];...
       ['                  Lidar ' TimeLidar{1} ' - ' TimeLidar{end}];...
       ['                    Sync ' TimeSync{1} ' - ' TimeSync{end}]});
grid on;

%% Assign variable names
% msgbox('Data are syncronized by interpolation approach');
N = DataLidarinterp;
Ps = DataECEinterp(:,1);
Ts = DataECEinterp(:,2);
AOD = AOD532interp;
```

```
SSA = SSA532interp;
PhaseFunction = PhaseFunction532interp;
Time = TimeSync;
Timenum = TimenumSync;
TimeSSA532 = TimeSSA;
TimenumSSA532 = TimenumSSA;

N(isnan(N)) = 0; % Find and set NaN in N to zero
index = find(N(:,1)==0);
N(index,:) = [];
Ps(index,:) = [];
Ts(index,:) = [];
AOD(index,:) = [];
SSA(index,:) = [];
PhaseFunction(index,:) = [];
Time(index,:) = [];
Timenum(index,:) = [];

%% Take a first look at AERONET Sa
SaAERONETsync = 4*pi./(PhaseFunction.*SSA);
SaAERONET = 4*pi./(PhaseFunction532.*SSA532);

f = figure;
movegui(f,'center');
plot(Timenum,SaAERONETsync,'go-',TimenumSSA532,SaAERONET,'r*');
xlim([min(Timenum) max(Timenum)]);
ylim([0 200]);
datetick('x','HHMM','keeplimits');
xlabel('Time (MDT)');
ylabel('S_a (sr)');
title({['Weather Station ' TimeECE{1} ' - ' TimeECE{end}];...
       ['           AERONET ' TimeAOD{1} ' - ' TimeAOD{end}];...
       ['                  Lidar ' TimeLidar{1} ' - ' TimeLidar{end}];...
       ['                     Sync ' TimeSync{1} ' - ' TimeSync{end}]});
grid on;
legend('S_a AERONET Sync','S_a AERONET','Location','Best');

%% Clear unnecessary data
clear prompt; clear dlg_title; clear num_lines; clear def; clear answer3
clear DataECE; clear DataECEinterp; clear DataLidar; clear
DataLidarinterp;
clear AOD500; clear AOD675; clear AOD532interp;
clear SSA440; clear SSA675; clear SSA532interp;
clear PhaseFunction440; clear PhaseFunction675; clear
PhaseFunction532interp;
clear TimeECE; clear TimenumECE; clear TimeLidar; clear TimenumLidar;
clear TimeSync; clear TimenumSync;
clear TimeAOD; clear TimenumAOD; clear TimePhaseFunction; clear
TimenumPhaseFunction; clear TimeSSA; clear TimenumSSA;
clear FilenameAOD; clear FilenameECE; clear FilenameSSA; clear
FilenamePhaseFunction;
clear Start; clear End; clear i; clear periodnum; clear stepnum; clear
index;
```

```
% This is Step3_ApplyOverlap code

% This code is to select, download, and apply O to the time sync L

run Step2_TimeSync

%% Download O
prompt = {'Enter overlap file name'};
dlg_title = 'File Name';
num_lines = 1;
def = {'O54'};
answer4 = inputdlg(prompt,dlg_title,num_lines,def);

FilenameO = answer4{1};
O = importdata([FilenameO '.mat']);
clear prompt; clear dlg_title; clear num_lines; clear def; clear answer4

%% Apply O to L
L = zeros(size(N,1),size(N,2));
LO = zeros(size(N,1),size(N,2));
for i = 1:size(Time,1)
    L(i,:) = N(i,:).*r.^2;
    LO(i,:) = L(i,:)./O;
end
```

```
% This is Step4_NoiseReduction code

% This code is to reduce noise in L and modify L near ground

run Step3_ApplyOverlap

%% Noise reduction

order = 3;
fcutoff = 0.1;
[B,A] = butter(order,fcutoff);
shift = 7;

LO_NR = zeros(size(LO,1),size(LO,2));
LO_NRshift = zeros(size(LO,1),size(LO,2));

for i=1:size(Time,1)
LO_NR(i,:) = filter(B,A,LO(i,:));
LO_NRshift(i,1:size(LO,2)-shift+1) = LO_NR(i,shift:end);
end
for i=1:size(Time,1)
    LO_NRshift(i,1:70) = 5*10^6*(70-(1:70)) + LO_NRshift(i,70);
end

% Modify L
rmodify1 = 1000; % Range to modify from ground
slope1 = -1.5*10^5;
LO_modify1 = LO_NRshift;
for i = 1:size(Time,1)
    LO_modify1(i,1:round(rmodify1/7.5))                              =
slope1*r(1:round(rmodify1/7.5))  +  LO_NRshift(i,round(rmodify1/7.5))  -
slope1*7.5*round(rmodify1/7.5);
end

rmodify2 = 4000; % Range to modify from ground
slope2  =  ((0.4/(7.5*round(rmodify2/7.5)))*r(1:round(rmodify2/7.5))  +
0.6);
LO_modify2 = LO_NRshift;
for i = 1:size(Time,1)
    LO_modify2(i,1:round(rmodify2/7.5))                              =
slope2.*LO_NRshift(i,1:round(rmodify2/7.5));
end

figure
plot(r(1:round(rmodify2/7.5)),slope2)
xlim('auto');
ylim('auto');
xlabel('Range (km)');
ylabel('Multiply Factor');
title('Multiply Function for L/O Modify 2');
grid on;
```

```
for i = 1:size(Time,1)
    figure(101)
    h                                                                           =
plot(r/1000,L(i,:),'r',r/1000,LO(i,:),'g',r/1000,LO_NRshift(i,:),'b',r/
1000,LO_modify1(i,:),'k',r/1000,LO_modify2(i,:),'m');
    h(1).LineWidth = 2;
    h(3).LineWidth = 2;
    xlim([0 8]);
    ylim([0 3*10^9]);
    xlabel('Range (km)');
    ylabel('L');
    title([FilenameLidar ' @ ' Time{i}]);
    grid on;
    legend('L','L/O','L/O  Noise  Reduction','L/O  Modify  1','L/O  Modify
2','Location','Northeast');
    pause(0.01);
end

clear L;
L = LO_NRshift;
L(L<1) = 1;

clear DataLidar_NR; clear DataLidar_NRshift; clear shift;
clear order; clear fcutoff; clear A; clear B; clear i; clear h; clear
rmodify;
clear  LO;  clear  LO_NR;  clear  LO_NRshift;  clear  LO_modify1;  clear
LO_modify2;
```

```
% This is Step5_T_P_Betam_Sigmam_Tm code

% This code is to calculate and plot T, P, beta_m, sigma_m, and T_m

run Step4_NoiseReduction

%% Input
prompt = {'Lapse rate (K/km)','Segment for linear slope T (km)'};
dlg_title = 'T,P,beta,sigma,Tm';
num_lines = 1;
def = {'7.7','[1.9 10.2]'};
answer5 = inputdlg(prompt,dlg_title,num_lines,def);

LapseRate = str2num(answer5{1}); % Change unit to K/m
limit = str2num(answer5{2});
ra = limit(1)*1000; % Change unit from km to m
rb = limit(2)*1000;
ra = 7.5*round(ra/7.5 - 1);
rb = 7.5*round(rb/7.5);
clear prompt; clear dlg_title; clear num_lines; clear def; clear answer5
clear limit;

%% Calculation
bar = waitbar(0,'Processing data, please wait');

% Fixed Constants
Lamda = 532; % Wavelength in nm
g = 9.81; % m/s^2
Cp = 1004; % J/kg K
R = 287.053; % J/kg K
Sm = 8*pi/3; % Molecular Lidar ratio
index = 1200;

for i = 1:size(Time,1)
    % Temperature
    T1(i,:) = Ts(i,1)*ones(1,find(r==ra)); %#ok<*SAGROW> % Make T=Ts for
r<ra
    T2(i,:) = Ts(i,1)-(LapseRate/1000)*r(1,find(r==ra)+1:end); % Follow
T = Ts-LapseRate*r for r>=ra
    T2(i,:) = T2(i,:) + Ts(i,1) - T2(i,1); % Shift to make the continuity
with T1
    T(i,:) = [T1(i,:), T2(i,:)]; % Get temperature model
    T(i,find(r==rb):end) = T(i,r==rb); % Make the final part of T constant

    % Pressure
    P(i,:)     =     Ps(i,:)*(Ts(i,1)./(Ts(i,1)-(LapseRate/1000)*r)).^(-
0.0326/(LapseRate/1000)); % Pressure in Pa
    P(i,index:end)       =       P(i,index)*exp(-0.0353.*(r(index:end)-
index*7.5)./T(i,index));

    %   P(i,:)   =   Ps(i,:)*(Ts(i,:)./(Ts(i,:)-(LapseRate/1000)*r)).^(-
0.034164/(LapseRate/1000)); % Pressure in Pa
```

```
    P(i,:) = P(i,:)/1000; % Pressure in kPa

    % Molecular Back Scattering
    beta_m(i,:) = 374280*P(i,:)./(T(i,:)*Lamda^4); % 1/m Sr
    % Molecular Extinction
    sigma_m(i,:) = Sm*beta_m(i,:); % 1/m Sr

    % Integrate beta_m from 0 to r
    beta_m_area(i,:) = beta_m(i,:)*7.5;
    for j = 1:size(r,2)
        integrate_beta_m(i,j) = sum(beta_m_area(i,1:j));
    end

    % Integration sigma_m from 0 to r
    sigma_m_area(i,:) = sigma_m(i,:)*7.5;
    for j = 1:size(r,2)
        integrate_sigma_m(i,j) = sum(sigma_m_area(i,1:j));
    end
    waitbar(i/size(Time,1),bar,'Processing data, please wait');
end

clear T1; clear T2;

% Molecular Transmission
Tm = exp(-integrate_sigma_m);

close(bar);
```

```
% This is Step6_C code

% This code is to calculate the calibration constant C

run Step5_T_P_Betam_Sigmam_Tm

%% Input
prompt = {'Segment for C averaging (m)'};
dlg_title = 'C';
num_lines = 1;
def = {'[4000 5000]'};
answer6 = inputdlg(prompt,dlg_title,num_lines,def);

limit = str2num(answer6{1});
rc = limit(1);
rd = limit(2);
rc = 7.5*round(rc/7.5);
rd = 7.5*round(rd/7.5);
clear prompt; clear dlg_title; clear num_lines; clear def; clear answer6
clear limit;

%% Calculation
bar = waitbar(0,'Processing data, please wait');

% Transmitted photons
Planck = 6.63*10^-34; % Planck constant unit m^2 kg/s
c = 3*10^8; % Speed of light
E_photon = Planck*c/(Lamda*10^-9); % Energy of a photon in J
E_pulse = 3*10^-6; % Average energy of a laser pulse in J
NumberAccumulation = 5000; % # of pulses being added up and count
Nt = E_pulse*NumberAccumulation/E_photon; % # of transmitted photons

C = zeros(size(L,1),size(L,2));
for i = 1:size(Time,1)
    % Calibration Constant
    C(i,:)   =   L(i,:)./(Nt*beta_m(i,:).*Tm(i,:).^2*exp(-2*AOD(i)));   %
Sr*m^3
    waitbar(i/size(Time,1),bar,'Processing data, please wait');
end

close(bar);

%% Plot C
for i = 1:size(Time,1)
    figure(107)
    plot(C(i,:),r/1000,'b');
    xlim([0 0.1]);
    ylim([0 6]);
    % line(xlim,[rc/1000 rc/1000],'Color','g');
    % line(xlim,[rd/1000 rd/1000],'Color','g');
    xlabel('C_\lambda (Sr*m^3)');
    ylabel('Height from Bozeman (km)');
```

```
    title({[FilenameLidar ' SyncStep=' step ' @ ' Time{i}];
        ['Ts=' num2str(Ts(i),'%.1f') ' K  Ps=' num2str(Ps(i)/1000,'%.1f')
' kPa  LapseRate=' num2str(LapseRate) ' K/km'];
        ['C_\lambda    Average    Range=['    num2str(rc,'%.0f')    '    '
num2str(rd,'%.0f') '] m']});
    grid on;
    pause(0.01);
end

%% C average
Cavg = zeros(size(Time,1),1);
for i = 1:size(Time,1)
    Cavg(i,:) = mean(C(i,find(r==rc):find(r==rd)));
end

figure
plot(Timenum,Cavg,'b');
xlim([min(Timenum) max(Timenum)]);
ylim([0 0.06]);
datetick('x','HHMM','keeplimits');
ylabel('C_\lambda Average');
title({[FilenameLidar ' SyncStep=' step ' ' Time{1} ' - ' Time{end}];
    ['LapseRate=' num2str(LapseRate) ' K/km  C_\lambda Average Rangee=['
num2str(rc,'%.0f') ' ' num2str(rd,'%.0f') '] m']});
grid on;

clear bar; clear i;
```

```
% This is Step7_LidarInversion1 code

% This code is to perform invesion by scanning Sa with step=1

bar = waitbar(0,'Performing inversion, Sa scanning step = 1');

% Location to cut AOD where beta_a begins to be zero
re = rd;
re = 7.5*round(re/7.5);
index = find(r==re);

% Create 3D or 2D matrix for each variable
argument = zeros(size(Time,1),size(Sa,2),size(r,2));
argument_area = zeros(size(Time,1),size(Sa,2),size(r,2));
integrate_argument = zeros(size(Time,1),size(Sa,2),size(r,2));
beta_total = zeros(size(Time,1),size(Sa,2),size(r,2));
beta_a = zeros(size(Time,1),size(Sa,2),size(r,2));
beta_a_area = zeros(size(Time,1),size(Sa,2),size(r,2));
integrate_beta_a = zeros(size(Time,1),size(Sa,2),size(r,2));
AOD_Lidar = zeros(size(Time,1),size(Sa,2),size(r,2));
AOD_Lidar_max = zeros(size(Time,1),size(Sa,2));
PE = zeros(size(Time,1),size(Sa,2));

% Convert these variables from 2D to 3D matrix
L_Temp = zeros(size(Time,1),size(Sa,2),size(r,2));
integrate_beta_m_Temp = zeros(size(Time,1),size(Sa,2),size(r,2));
Tm_Temp = zeros(size(Time,1),size(Sa,2),size(r,2));
beta_m_Temp = zeros(size(Time,1),size(Sa,2),size(r,2));
for j = 1:size(Sa,2)
    L_Temp(:,j,:) = L;
    integrate_beta_m_Temp(:,j,:) = integrate_beta_m;
    Tm_Temp(:,j,:) = Tm;
    beta_m_Temp(:,j,:) = beta_m;
    waitbar(0.1*j/(size(Sa,2)),bar,'Performing  inversion,  Sa  scanning
step = 1');
end
clear L; clear integrate_beta_m; clear Tm; clear beta_m; clear j;
L = L_Temp;
integrate_beta_m = integrate_beta_m_Temp;
Tm = Tm_Temp;
beta_m = beta_m_Temp;
clear  L_Temp;  clear  integrate_beta_m_Temp;  clear  Tm_Temp;  clear
beta_m_Temp;

%% Lidar inversion
for i = 1:size(Time,1)
    for j = 1:size(Sa,2)
        % Beta_total the whole range
        argument(i,j,:)                =                L(i,j,:).*exp(-
2*Sa(i,j)*integrate_beta_m(i,j,:))./(Tm(i,j,:).^2);  %  Argument  inside
the integration denominator
        argument_area(i,j,:) = argument(i,j,:)*7.5;
        for k = 1:size(r,2)
```

```
            integrate_argument(i,j,k)  =  sum(argument_area(i,j,1:k));  %
Integration denominator
        end
        beta_total(i,j,:)            =            (L(i,j,:).*exp(2*(Sm-
Sa(i,j))*integrate_beta_m(i,j,:)))./((Nt*Cavg(i))-
(2*Sa(i,j)*integrate_argument(i,j,:)));

        % Aerosol Back Scatter
        beta_a(i,j,:) = beta_total(i,j,:) - beta_m(i,j,:);
        % beta_a(beta_a<0) = 0;

        % Integrate beta_a
        beta_a_area(i,j,:) = beta_a(i,j,:)*7.5;
        for k = 1:size(r,2)
            integrate_beta_a(i,j,k) = sum(beta_a_area(i,j,1:k));
        end

        % AOD Lidar
        AOD_Lidar(i,j,:) = Sa(i,j)*integrate_beta_a(i,j,:);
        AOD_Lidar_max(i,j) = max(AOD_Lidar(i,j,1:index));

        % Percent difference between AOD and AOD_Lidar
        PE(i,j) = 100*abs(AOD(i) - AOD_Lidar_max(i,j))/AOD(i);

        waitbar(0.1+0.9*(size(Sa,2)*(i-1)                              +
j)/(size(Sa,2)*size(Time,1)),bar,'Performing inversion, Sa scanning step
= 1');
    end
end
clear i; clear j; clear k; clear index;

%% Find Sa that yields minimum percent error in AOD
PEmin = min(PE,[],2);
index = zeros(size(Time,1),1);
for i = 1:size(Time,1)
    index(i) = find(PE(i,:)==PEmin(i),1);
end
index(index==1) = 2;
SaLidar = zeros(size(Time,1),1);
for i = 1:size(Time,1)
    SaLidar(i,:) = Sa(i,index(i));
end
clear i

%% Convert these variables back from 3D to 2D matrix
L_Temp = zeros(size(Time,1),size(r,2));
integrate_beta_m_Temp = zeros(size(Time,1),size(r,2));
Tm_Temp = zeros(size(Time,1),size(r,2));
beta_m_Temp = zeros(size(Time,1),size(r,2));
for i = 1:size(Time,1)
    L_Temp(i,:) = L(i,1,:);
    integrate_beta_m_Temp(i,:) = integrate_beta_m(i,1,:);
    Tm_Temp(i,:) = Tm(i,1,:);
```

```
    beta_m_Temp(i,:) = beta_m(i,1,:);
end
clear L; clear integrate_beta_m; clear Tm; clear beta_m; clear i;
L = L_Temp;
integrate_beta_m = integrate_beta_m_Temp;
Tm = Tm_Temp;
beta_m = beta_m_Temp;
clear  L_Temp;  clear  integrate_beta_m_Temp;  clear  Tm_Temp;  clear
beta_m_Temp;

close(bar); clear bar;
```

```
% This is Step7_LidarInversion2 code

% This code is to perform invesion by scanning Sa with step=0.1

bar = waitbar(0,'Performing inversion, Sa scanning step = 0.1');

% Location to cut AOD where beta_a begins to be zero
re = rd;
re = 7.5*round(re/7.5);
index = find(r==re);

% Create 3D or 2D matrix for each variable
argument = zeros(size(Time,1),size(Sa,2),size(r,2));
argument_area = zeros(size(Time,1),size(Sa,2),size(r,2));
integrate_argument = zeros(size(Time,1),size(Sa,2),size(r,2));
beta_total = zeros(size(Time,1),size(Sa,2),size(r,2));
beta_a = zeros(size(Time,1),size(Sa,2),size(r,2));
beta_a_area = zeros(size(Time,1),size(Sa,2),size(r,2));
integrate_beta_a = zeros(size(Time,1),size(Sa,2),size(r,2));
AOD_Lidar = zeros(size(Time,1),size(Sa,2),size(r,2));
AOD_Lidar_max = zeros(size(Time,1),size(Sa,2));
PE = zeros(size(Time,1),size(Sa,2));

% Convert these variables from 2D to 3D matrix
L_Temp = zeros(size(Time,1),size(Sa,2),size(r,2));
integrate_beta_m_Temp = zeros(size(Time,1),size(Sa,2),size(r,2));
Tm_Temp = zeros(size(Time,1),size(Sa,2),size(r,2));
beta_m_Temp = zeros(size(Time,1),size(Sa,2),size(r,2));
for j = 1:size(Sa,2)
    L_Temp(:,j,:) = L;
    integrate_beta_m_Temp(:,j,:) = integrate_beta_m;
    Tm_Temp(:,j,:) = Tm;
    beta_m_Temp(:,j,:) = beta_m;
    waitbar(0.1*j/(size(Sa,2)),bar,'Performing inversion, Sa scanning
step = 0.1');
end
clear L; clear integrate_beta_m; clear Tm; clear beta_m; clear j;
L = L_Temp;
integrate_beta_m = integrate_beta_m_Temp;
Tm = Tm_Temp;
beta_m = beta_m_Temp;
clear L_Temp; clear integrate_beta_m_Temp; clear Tm_Temp; clear
beta_m_Temp;

%% Lidar inversion
for i = 1:size(Time,1)
    for j = 1:size(Sa,2)
        % Beta_total the whole range
        argument(i,j,:) = L(i,j,:).*exp(-
2*Sa(i,j)*integrate_beta_m(i,j,:))./(Tm(i,j,:).^2); % Argument inside
the integration denominator
        argument_area(i,j,:) = argument(i,j,:)*7.5;
        for k = 1:size(r,2)
```

```
            integrate_argument(i,j,k)  =  sum(argument_area(i,j,1:k));  %
Integration denominator
        end
        beta_total(i,j,:)            =             (L(i,j,:).*exp(2*(Sm-
Sa(i,j))*integrate_beta_m(i,j,:)))./((Nt*Cavg(i))-
(2*Sa(i,j)*integrate_argument(i,j,:)));

        % Aerosol Back Scatter
        beta_a(i,j,:) = beta_total(i,j,:) - beta_m(i,j,:);
        beta_a(beta_a<0) = 0;

        % Integrate beta_a
        beta_a_area(i,j,:) = beta_a(i,j,:)*7.5;
        for k = 1:size(r,2)
            integrate_beta_a(i,j,k) = sum(beta_a_area(i,j,1:k));
        end

        % AOD Lidar
        AOD_Lidar(i,j,:) = Sa(i,j)*integrate_beta_a(i,j,:);
        AOD_Lidar_max(i,j) = max(AOD_Lidar(i,j,1:index));

        % Percent difference between AOD and AOD_Lidar
        PE(i,j) = 100*abs(AOD(i) - AOD_Lidar_max(i,j))/AOD(i);

        waitbar(0.1+0.9*(size(Sa,2)*(i-1)                              +
j)/(size(Sa,2)*size(Time,1)),bar,'Performing inversion, Sa scanning step
= 0.1');
    end
end
clear i; clear j; clear k; clear index;

%% Find Sa that yields minimum percent error in AOD
PEmin = min(PE,[],2);
index = zeros(size(Time,1),1);
for i = 1:size(Time,1)
    index(i) = find(PE(i,:)==PEmin(i),1);
end
index(index==1) = 2;
SaLidar = zeros(size(Time,1),1);
for i = 1:size(Time,1)
    SaLidar(i,:) = Sa(i,index(i));
end
clear i

%% Convert these variables back from 3D to 2D matrix
L_Temp = zeros(size(Time,1),size(r,2));
integrate_beta_m_Temp = zeros(size(Time,1),size(r,2));
Tm_Temp = zeros(size(Time,1),size(r,2));
beta_m_Temp = zeros(size(Time,1),size(r,2));
for i = 1:size(Time,1)
    L_Temp(i,:) = L(i,1,:);
    integrate_beta_m_Temp(i,:) = integrate_beta_m(i,1,:);
    Tm_Temp(i,:) = Tm(i,1,:);
```

```
    beta_m_Temp(i,:) = beta_m(i,1,:);
end
clear L; clear integrate_beta_m; clear Tm; clear beta_m; clear i;
L = L_Temp;
integrate_beta_m = integrate_beta_m_Temp;
Tm = Tm_Temp;
beta_m = beta_m_Temp;
clear  L_Temp;  clear  integrate_beta_m_Temp;  clear  Tm_Temp;  clear
beta_m_Temp;

close(bar); clear bar;
```

```
% This is Step7_LidarInversion3 code

% This code is to perform invesion by scanning Sa with step=0.01

bar = waitbar(0,'Performing inversion, Sa scanning step = 0.01');

% Location to cut AOD where beta_a begins to be zero
re = rd;
re = 7.5*round(re/7.5);
index = find(r==re);

% Create 3D or 2D matrix for each variable
argument = zeros(size(Time,1),size(Sa,2),size(r,2));
argument_area = zeros(size(Time,1),size(Sa,2),size(r,2));
integrate_argument = zeros(size(Time,1),size(Sa,2),size(r,2));
beta_total = zeros(size(Time,1),size(Sa,2),size(r,2));
beta_a = zeros(size(Time,1),size(Sa,2),size(r,2));
beta_a_area = zeros(size(Time,1),size(Sa,2),size(r,2));
integrate_beta_a = zeros(size(Time,1),size(Sa,2),size(r,2));
AOD_Lidar = zeros(size(Time,1),size(Sa,2),size(r,2));
AOD_Lidar_max = zeros(size(Time,1),size(Sa,2));
PE = zeros(size(Time,1),size(Sa,2));

% Convert these variables from 2D to 3D matrix
L_Temp = zeros(size(Time,1),size(Sa,2),size(r,2));
integrate_beta_m_Temp = zeros(size(Time,1),size(Sa,2),size(r,2));
Tm_Temp = zeros(size(Time,1),size(Sa,2),size(r,2));
beta_m_Temp = zeros(size(Time,1),size(Sa,2),size(r,2));
for j = 1:size(Sa,2)
    L_Temp(:,j,:) = L;
    integrate_beta_m_Temp(:,j,:) = integrate_beta_m;
    Tm_Temp(:,j,:) = Tm;
    beta_m_Temp(:,j,:) = beta_m;
    waitbar(0.1*j/(size(Sa,2)),bar,'Performing inversion, Sa scanning step = 0.01');
end
clear L; clear integrate_beta_m; clear Tm; clear beta_m; clear j;
L = L_Temp;
integrate_beta_m = integrate_beta_m_Temp;
Tm = Tm_Temp;
beta_m = beta_m_Temp;
clear L_Temp; clear integrate_beta_m_Temp; clear Tm_Temp; clear beta_m_Temp;

%% Lidar inversion
for i = 1:size(Time,1)
    for j = 1:size(Sa,2)
        % Beta_total the whole range
        argument(i,j,:) = L(i,j,:).*exp(-2*Sa(i,j)*integrate_beta_m(i,j,:))./(Tm(i,j,:).^2); % Argument inside the integration denominator
        argument_area(i,j,:) = argument(i,j,:)*7.5;
        for k = 1:size(r,2)
```

```
            integrate_argument(i,j,k) = sum(argument_area(i,j,1:k)); %
Integration denominator
        end
        beta_total(i,j,:)          =          (L(i,j,:).*exp(2*(Sm-
Sa(i,j))*integrate_beta_m(i,j,:)))./((Nt*Cavg(i))-
(2*Sa(i,j)*integrate_argument(i,j,:)));

        % Aerosol Back Scatter
        beta_a(i,j,:) = beta_total(i,j,:) - beta_m(i,j,:);
        beta_a(beta_a<0) = 0;

        % Integrate beta_a
        beta_a_area(i,j,:) = beta_a(i,j,:)*7.5;
        for k = 1:size(r,2)
            integrate_beta_a(i,j,k) = sum(beta_a_area(i,j,1:k));
        end

        % AOD Lidar
        AOD_Lidar(i,j,:) = Sa(i,j)*integrate_beta_a(i,j,:);
        AOD_Lidar_max(i,j) = max(AOD_Lidar(i,j,1:index));

        % Percent difference between AOD and AOD_Lidar
        PE(i,j) = 100*abs(AOD(i) - AOD_Lidar_max(i,j))/AOD(i);

        waitbar(0.1+0.9*(size(Sa,2)*(i-1)                          +
j)/(size(Sa,2)*size(Time,1)),bar,'Performing inversion, Sa scanning step
= 0.01');
    end
end
clear i; clear j; clear k; clear index;

%% Find Sa that yields minimum percent error in AOD
PEmin = min(PE,[],2);
index = zeros(size(Time,1),1);
for i = 1:size(Time,1)
    index(i) = find(PE(i,:)==PEmin(i),1);
end
index(index==1) = 2;
SaLidar = zeros(size(Time,1),1);
for i = 1:size(Time,1)
    SaLidar(i,:) = Sa(i,index(i));
end
clear i

%% Convert these variables back from 3D to 2D matrix
L_Temp = zeros(size(Time,1),size(r,2));
integrate_beta_m_Temp = zeros(size(Time,1),size(r,2));
Tm_Temp = zeros(size(Time,1),size(r,2));
beta_m_Temp = zeros(size(Time,1),size(r,2));
for i = 1:size(Time,1)
    L_Temp(i,:) = L(i,1,:);
    integrate_beta_m_Temp(i,:) = integrate_beta_m(i,1,:);
    Tm_Temp(i,:) = Tm(i,1,:);
```

```
    beta_m_Temp(i,:) = beta_m(i,1,:);
end
clear L; clear integrate_beta_m; clear Tm; clear beta_m; clear i;
L = L_Temp;
integrate_beta_m = integrate_beta_m_Temp;
Tm = Tm_Temp;
beta_m = beta_m_Temp;
clear  L_Temp;  clear  integrate_beta_m_Temp;  clear  Tm_Temp;  clear
beta_m_Temp;

close(bar); clear bar;
```

```
% This is Step8_PercentErrorAOD code

% This code is to choose Sa scan step and calculate percent error in AOD

run Step6_C

%% Input
choice = menu('Choose Sa scanning step','1','0.1','0.01');

%%
Sa = zeros(size(Time,1),100);
for i = 1:size(Time,1)
    Sa(i,:) = 1:1:100;
end

if choice==1 % step=1
    run Step7_LidarInversion1
    PEmin1 = PEmin;
    SaLidar1 = SaLidar;

    for i = 1:size(Time,1)
        figure(108)
        semilogy(Sa(i,:),PE(i,:),'b');
        xlim([1 100]);
        ylim([0.001 100]);
        xlabel(texlabel('S_a (Sr)'));
        ylabel('Percent Error in AOD');
        title({[FilenameLidar ' SyncStep=' step ' @ ' Time{i}];
            ['Ts='      num2str(Ts(i),'%.1f')      '      K            Ps='
num2str(Ps(i)/1000,'%.1f')  '  kPa    LapseRate='  num2str(LapseRate)   '
K/km'];
            ['C_\lambda   Average   Range=['   num2str(rc,'%.0f')   '   '
num2str(rd,'%.0f') '] m  AOD=' num2str(AOD(i),'%.3f')]});
        grid on;
        pause(0.1);
    end
    figure
    plot(Timenum,PEmin1,'b');
    xlim([min(Timenum) max(Timenum)]);
    ylim([0 5]);
    datetick('x','HHMM','keeplimits');
    ylabel('Percent Error in AOD');
    title({[FilenameLidar ' SyncStep=' step ' on ' Time{1}(1:11)];
        ['LapseRate='  num2str(LapseRate)  '  K/km    C_\lambda  Average
Range=[' num2str(rc,'%.0f') ' ' num2str(rd,'%.0f') '] m']});
    grid on;
    legend('S_a scan step = 1');
    figure
    plot(Timenum,SaLidar1,'b');
    xlim([min(Timenum) max(Timenum)]);
    ylim([0 100]);
    datetick('x','HHMM','keeplimits');
    ylabel('S_a');
```

```
    title({[FilenameLidar ' SyncStep=' step ' on ' Time{1}(1:11)];
        ['LapseRate='  num2str(LapseRate)  '  K/km   C_\lambda  Average
Range=[' num2str(rc,'%.0f') ' ' num2str(rd,'%.0f') '] m']});
    grid on;
    legend('S_a scan step = 1');

elseif choice==2 % step=0.1
    run Step7_LidarInversion1
    PEmin1 = PEmin;
    SaLidar1 = SaLidar;

    clear argument; clear argument_area; clear integrate_argument; clear
beta_total; clear beta_a; clear beta_a_window; clear beta_a_area;
    clear integrate beta_a; clear AOD_lidar; clear AOD_Lidar_max; clear
clear PE; clear PEmin;  clear Sa_Lidar;
    SaTemp1 = zeros(size(Time,1),21);
    SaTemp2 = zeros(size(Time,1),118);
    for i = 1:size(Time,1)
        SaTemp1(i,:) = Sa(i,index(i)-1):0.1:Sa(i,index(i)+1);
        SaTemp2(i,:)      =      [Sa(i,1:index(i)-2)      SaTemp1(i,:)
Sa(i,index(i)+2:end)];
    end
    clear i; clear Sa;
    Sa = SaTemp2;
    clear SaTemp1; clear SaTemp2;
    run Step7_LidarInversion2
    PEmin2 = PEmin;
    SaLidar2 = SaLidar;

    for i = 1:size(Time,1)
        figure(108)
        semilogy(Sa(i,:),PE(i,:),'b');
        xlim([1 100]);
        ylim([0.1 100]);
        xlabel(texlabel('S_a (Sr)'));
        ylabel('Percent Error in AOD');
        title({[FilenameLidar ' SyncStep=' step ' @ ' Time{i}];
            ['Ts='     num2str(Ts(i),'%.1f')     '     K          Ps='
num2str(Ps(i)/1000,'%.1f')  '  kPa    LapseRate='  num2str(LapseRate)  '
K/km'];
            ['C_\lambda   Average   Range=['   num2str(rc,'%.0f')   '   '
num2str(rd,'%.0f') '] m  AOD=' num2str(AOD(i),'%.3f')]});
        grid on;
        pause(0.1);
    end
    figure
    plot(Timenum,PEmin1,'b',Timenum,PEmin2,'r');
    xlim([min(Timenum) max(Timenum)]);
    ylim([0 5]);
    datetick('x','HHMM','keeplimits');
    ylabel('Percent Error in AOD');
    title({[FilenameLidar ' SyncStep=' step ' on ' Time{1}(1:11)];
        ['LapseRate='  num2str(LapseRate)  '  K/km   C_\lambda  Average
Range=[' num2str(rc,'%.0f') ' ' num2str(rd,'%.0f') '] m']});
```

```
    grid on;
    legend('S_a scan step = 1','S_a scan step = 0.1');
    figure
    plot(Timenum,SaLidar1,'b',Timenum,SaLidar2,'r');
    xlim([min(Timenum) max(Timenum)]);
    ylim([0 100]);
    datetick('x','HHMM','keeplimits');
    ylabel('S_a');
    title({[FilenameLidar ' SyncStep=' step ' on ' Time{1}(1:11)];
        ['LapseRate='  num2str(LapseRate)  '  K/km    C_\lambda  Average
Range=[' num2str(rc,'%.0f') ' ' num2str(rd,'%.0f') '] m']]});
    grid on;
    legend('Sa scan step = 1','Sa scan step = 0.1');

else % step=0.01
    run Step7_LidarInversion1
    PEmin1 = PEmin;
    SaLidar1 = SaLidar;

    clear argument; clear argument_area; clear integrate_argument; clear
beta_total; clear beta_a; clear beta_a_window; clear beta_a_area;
    clear integrate beta_a; clear AOD_lidar; clear AOD_Lidar_max; clear
clear PE; clear PEmin;  clear Sa_Lidar;
    SaTemp1 = zeros(size(Time,1),21);
    SaTemp2 = zeros(size(Time,1),118);
    for i = 1:size(Time,1)
        SaTemp1(i,:) = Sa(i,index(i)-1):0.1:Sa(i,index(i)+1);
        SaTemp2(i,:)       =       [Sa(i,1:index(i)-2)       SaTemp1(i,:)
Sa(i,index(i)+2:end)];
    end
    clear i; clear Sa;
    Sa = SaTemp2;
    clear SaTemp1; clear SaTemp2;
    run Step7_LidarInversion2
    PEmin2 = PEmin;
    SaLidar2 = SaLidar;

    clear argument; clear argument_area; clear integrate_argument; clear
beta_total; clear beta_a; clear beta_a_window; clear beta_a_area;
    clear integrate beta_a; clear AOD_lidar; clear AOD_Lidar_max; clear
clear PE; clear PEmin;  clear Sa_Lidar;
    SaTemp1 = zeros(size(Time,1),21);
    SaTemp2 = zeros(size(Time,1),136);
    for i = 1:size(Time,1)
        SaTemp1(i,:) = Sa(i,index(i)-1):0.01:Sa(i,index(i)+1);
        SaTemp2(i,:)       =       [Sa(i,1:index(i)-2)       SaTemp1(i,:)
Sa(i,index(i)+2:end)];
    end
    clear i; clear Sa;
    Sa = SaTemp2;
    clear SaTemp1; clear SaTemp2;
    run Step7_LidarInversion3
    PEmin3 = PEmin;
    SaLidar3 = SaLidar;
```

```
    for i = 1:size(Time,1)
        figure(108)
        semilogy(Sa(i,:),PE(i,:),'b');
        xlim([1 100]);
        ylim([0.001 100]);
        xlabel(texlabel('S_a (Sr)'));
        ylabel('Percent Error in AOD');
        title({[FilenameLidar ' SyncStep=' step ' @ ' Time{i}];
            ['Ts='      num2str(Ts(i),'%.1f')      '      K           Ps='
num2str(Ps(i)/1000,'%.1f')  '  kPa    LapseRate='  num2str(LapseRate)  '
K/km'];
            ['C_\lambda   Average   Range=['   num2str(rc,'%.0f')   '   '
num2str(rd,'%.0f') '] m  AOD=' num2str(AOD(i),'%.3f')]});
        grid on;
        pause(0.1);
    end
    figure
    plot(Timenum,PEmin1,'b',Timenum,PEmin2,'r',Timenum,PEmin3,'g');
    xlim([min(Timenum) max(Timenum)]);
    ylim([0 5]);
    datetick('x','HHMM','keeplimits');
    ylabel('Percent Error in AOD');
    title({[FilenameLidar ' SyncStep=' step ' on ' Time{1}(1:11)];
        ['LapseRate='  num2str(LapseRate)  '  K/km    C_\lambda  Average
Range=[' num2str(rc,'%.0f') ' ' num2str(rd,'%.0f') '] m']});
    grid on;
    legend('S_a scan step = 1','S_a scan step = 0.1','S_a scan step =
0.01');
    figure

plot(Timenum,SaLidar1,'b',Timenum,SaLidar2,'r',Timenum,SaLidar3,'g');
    xlim([min(Timenum) max(Timenum)]);
    ylim([0 100]);
    datetick('x','HHMM','keeplimits');
    ylabel('S_a');
    title({[FilenameLidar ' SyncStep=' step ' on ' Time{1}(1:11)];
        ['LapseRate='  num2str(LapseRate)  '  K/km    C_\lambda  Average
Range=[' num2str(rc,'%.0f') ' ' num2str(rd,'%.0f') '] m']});
    grid on;
    legend('Sa scan step = 1','Sa scan step = 0.1','Sa scan step = 0.01');
end

clear i;
```

```
% This is Step9_Betatotal_Betam_Betaa code

% This code is to calculate beta_total, beta_m, and beta_a from the Sa
that provides minimum error

run Step8_PercentErrorAOD

%% Time evolve plots
% beta_total, beta_m
beta_total_Temp = zeros(size(Time,1),size(r,2));
beta_a_Temp = zeros(size(Time,1),size(r,2));
integrate_beta_a_Temp = zeros(size(Time,1),size(r,2));

for i = 1:size(Time,1)
    beta_total_Temp(i,:) = beta_total(i,index(i),:);
    beta_a_Temp(i,:) = beta_a(i,index(i),:);
    integrate_beta_a_Temp(i,:) = integrate_beta_a(i,index(i),:);
end
clear beta_total; clear beta_a; clear integrate_beta_a;
beta_total = beta_total_Temp;
beta_a = beta_a_Temp;
integrate_beta_a = integrate_beta_a_Temp;
clear beta_total_Temp; clear beta_a_Temp; clear integrate_beta_a_Temp;

for i = 1:size(Time,1)
    figure(109)
    plot(beta_total(i,:),r/1000,'b',beta_m(i,:),r/1000,'r');
    legend(texlabel('beta_total'),texlabel('beta_m'));
    xlim([0*10^-5 1*10^-5]);
    ylim([0 6]);
    line(xlim,[rc/1000 rc/1000],'Color','g');
    line(xlim,[rd/1000 rd/1000],'Color','g');
    xlabel(texlabel('beta_total and beta_m (1/m Sr)'));
    ylabel('Height from Bozeman (km)');
    title({[FilenameLidar ' SyncStep=' step ' @ ' Time{i}];
        ['Ts=' num2str(Ts(i),'%.1f') ' K  Ps=' num2str(Ps(i)/1000,'%.1f')
' kPa  LapseRate=' num2str(LapseRate) ' K/km'];
        ['C_\lambda    Average    Range=['    num2str(rc,'%.0f')    '    '
num2str(rd,'%.0f')   ']   m     AOD='   num2str(AOD(i),'%.3f')   '     S_a='
num2str(SaLidar(i),'%.2f')]});
    grid on;
    pause(0.1);
end

% beta_a
for i = 1:size(Time,1)
    figure(110)
    plot(beta_a(i,:),r/1000,'b');
    xlim([0*10^-5 1*10^-5]);
    ylim([0 6]);
    line(xlim,[rc/1000 rc/1000],'Color','g');
    line(xlim,[rd/1000 rd/1000],'Color','g');
    xlabel(texlabel('beta_a (1/m Sr)'));
```

```
    ylabel('Height from Bozeman (km)');
    title({[FilenameLidar ' SyncStep=' step ' @ ' Time{i}];
        ['Ts=' num2str(Ts(i),'%.1f') ' K  Ps=' num2str(Ps(i)/1000,'%.1f')
' kPa  LapseRate=' num2str(LapseRate) ' K/km'];
        ['C_\lambda    Average    Range=['    num2str(rc,'%.0f')    '    '
num2str(rd,'%.0f')   ']   m     AOD='   num2str(AOD(i),'%.3f')   '     S_a='
num2str(SaLidar(i),'%.2f')]});
    grid on;
    pause(0.1);
end

% Integrate_beta_a
for i = 1:size(Time,1)
    figure(111)
    h1 = plot(integrate_beta_a(i,:),r/1000,'b');
    h2      =      fill([integrate_beta_a(i,:)      integrate_beta_a(i,1)
integrate_beta_a(i,1)], [r/1000 r(end)/1000 r(1)/1000], 'b');
    xlim([0 0.04]);
    ylim([0 6]);
    line(xlim,[rc/1000 rc/1000],'Color','g');
    line(xlim,[rd/1000 rd/1000],'Color','g');
    xlabel(texlabel('integrate beta_a (1/m Sr)'));
    ylabel('Height from Bozeman (km)');
    title({[FilenameLidar ' SyncStep=' step ' @ ' Time{i}];
        ['Ts=' num2str(Ts(i),'%.1f') ' K  Ps=' num2str(Ps(i)/1000,'%.1f')
' kPa  LapseRate=' num2str(LapseRate) ' K/km'];
        ['C_\lambda    Average    Range=['    num2str(rc,'%.0f')    '    '
num2str(rd,'%.0f')   ']   m     AOD='   num2str(AOD(i),'%.3f')   '     S_a='
num2str(SaLidar(i),'%.2f')]});
    grid on;
    pause(0.1);
    if i==size(Time,1)
    else
        delete(h1);
        delete(h2);
    end
end

%% Plots
f = figure;
movegui(f,'southwest');
imagesc(Timenum, r/1000, beta_total');
datetick('x','HHMM','keeplimits');
set(gca,'YDir','normal');
set(gca, 'YLim',[0 16], 'CLim',[0 5*10^-6]);
colorbar;
colormap(jet);
xlabel({'Time (MDT)'; texlabel('beta_t_o_t_a_l (1/m Sr)')});
ylabel('Height from Bozeman (km)');
title({[FilenameLidar ' SyncStep=' step ' on ' Time{1}(1:11)];
    ['LapseRate=' num2str(LapseRate)  ' K/km  C_\lambda Average Range=['
num2str(rc,'%.0f') ' ' num2str(rd,'%.0f') '] m']});
grid on;
```

```
f = figure;
movegui(f,'south');
imagesc(Timenum, r/1000, beta_m');
datetick('x','HHMM','keeplimits');
set(gca,'YDir','normal');
set(gca, 'YLim',[0 16], 'CLim',[0 2*10^-6]);
colorbar;
colormap(jet);
xlabel({'Time (MDT)'; texlabel('beta_m (1/m Sr)')});
ylabel('Height from Bozeman (km)');
title({[FilenameLidar ' SyncStep=' step ' on ' Time{1}(1:11)];
    ['LapseRate=' num2str(LapseRate) ' K/km  C_\lambda Average Range=['
num2str(rc,'%.0f') ' ' num2str(rd,'%.0f') '] m']});
grid on;

f = figure;
movegui(f,'southeast');
imagesc(Timenum, r/1000, beta_a');
datetick('x','HHMM','keeplimits');
set(gca,'YDir','normal');
set(gca, 'YLim',[0 16], 'CLim',[0 2*10^-6]);
colorbar;
colormap(jet);
xlabel({'Time (MDT)'; texlabel('beta_a (1/m Sr)')});
ylabel('Height from Bozeman (km)');
title({[FilenameLidar ' SyncStep=' step ' on ' Time{1}(1:11)];
    ['LapseRate=' num2str(LapseRate) ' K/km  C_\lambda Average Range=['
num2str(rc,'%.0f') ' ' num2str(rd,'%.0f') '] m']});
grid on;

clear i; clear h1; clear h2;
```

```
% This is Step10_AOD_Sa code

% This code is to calculate lidar AOD, plot AOD and Sa both from lidar
and AERONET

run Step9_Betatotal_Betam_Betaa

%% Calculate Sa from AERONET SSA and phase function
clear Sa;
% SaAERONETsync = 4*pi./(PhaseFunction.*SSA);
% SaAERONET = 4*pi./(PhaseFunction532.*SSA532);

f = figure;
movegui(f,'northwest');
plot(Timenum,SaAERONETsync,'go-
',TimenumSSA532,SaAERONET,'ro',Timenum,SaLidar,'b*-');
xlim([min(Timenum) max(Timenum)]);
ylim([0 100]);
datetick('x','HHMM','keeplimits');
xlabel('Time (MDT)');
ylabel('S_a (sr)');
title({[FilenameLidar ' SyncStep=' step ' on ' Time{1}(1:11)];
    ['LapseRate=' num2str(LapseRate) ' K/km  C_\lambda Average Range=['
num2str(rc,'%.0f') ' ' num2str(rd,'%.0f') '] m']});
grid on;
legend('S_a    AERONET    Sync','S_a    AERONET    Raw    Data','S_a
Lidar','Location','Best');

%% Get AOD_Lidar from SaLidar that produce minimum error in AOD
AOD_Lidartemp = zeros(size(AOD_Lidar,1),size(AOD_Lidar,3));
for i = 1:size(Time,1)
    AOD_Lidartemp(i,:) = AOD_Lidar(i,index(i),:);
end
clear AOD_Lidar;
AOD_Lidar = AOD_Lidartemp;
clear AOD_Lidartemp;

for i = 1:size(Time,1)
        figure(112)
        h1 = plot(AOD_Lidar(i,:),r/1000,'b');
        h2 = fill([AOD_Lidar(i,:) AOD_Lidar(i,1) AOD_Lidar(i,1)], [r/1000
r(end)/1000 r(1)/1000], 'b');
        line([AOD(i) AOD(i)],[0 20],'Color','r','LineWidth',1);
        xlim([0 0.3]);
        ylim([0 6]);
        line(xlim,[rc/1000 rc/1000],'Color','g');
        line(xlim,[rd/1000 rd/1000],'Color','g');
        xlabel(texlabel('AOD_L_i_d_a_r'));
        ylabel('Height from Bozeman (km)');
        title({[FilenameLidar ' SyncStep=' step ' @ ' Time{i}];
            ['Ts='     num2str(Ts(i),'%.1f')     '     K          Ps='
num2str(Ps(i)/1000,'%.1f')  '  kPa    LapseRate='  num2str(LapseRate)  '
K/km'];
```

```
            ['C_\lambda   Average   Range=['   num2str(rc,'%.0f')   '   '
num2str(rd,'%.0f')  ']  m     AOD='  num2str(AOD(i),'%.3f')  '     S_a='
num2str(SaLidar(i),'%.2f')]});
        grid on;
        pause(0.1);
end

f = figure;
movegui(f,'northeast');
plot(Timenum,AOD,'ro',Timenum,AOD_Lidar(:,find(r==re)),'b*');
%#ok<*FNDSB>
xlim([min(Timenum) max(Timenum)]);
ylim([0 0.5]);
datetick('x','HHMM','keeplimits');
ylabel('AOD');
title({[FilenameLidar ' SyncStep=' step ' on ' Time{1}(1:11)];
    ['LapseRate=' num2str(LapseRate) ' K/km  C_\lambda Average Range=['
num2str(rc,'%.0f') ' ' num2str(rd,'%.0f') '] m']});
grid on;
legend('AERONET','Lidar');
```

```
% This is Step11_SaveResults code

% This code is to save the inversion results in the assigned directory

run Step10_AOD_Sa

%%
if choice==1 % step=1
    clear PEmin; PEmin = PEmin1; clear PEmin1;
    clear SaLidar; SaLidar = SaLidar1; clear SaLidar1;
elseif choice==2 % step=0.1
    clear PEmin; PEmin = PEmin2; clear PEmin1; clear PEmin2;
    clear SaLidar; SaLidar = SaLidar2; clear SaLidar1; clear SaLidar2;
else % step=0.01
    clear PEmin; PEmin = PEmin3; clear PEmin1; clear PEmin2; clear PEmin3;
    clear SaLidar; SaLidar = SaLidar3; clear SaLidar1; clear SaLidar2;
clear SaLidar3;
end

%% Save results
choice = menu('Save results to folder "Results" ?','Yes','No');
if choice==1
bar = waitbar(0,'Saving data, please wait');
    save(fullfile(savdir,[FilenameLidar ' Time']),'Time');
    save(fullfile(savdir,[FilenameLidar ' Timenum']),'Timenum');
waitbar(0.05,bar,'Saving data, please wait');
    save(fullfile(savdir,[FilenameLidar ' TimeSSA532']),'TimeSSA532');
    save(fullfile(savdir,[FilenameLidar                                '
TimenumSSA532']),'TimenumSSA532');
waitbar(0.1,bar,'Saving data, please wait');
    save(fullfile(savdir,[FilenameLidar ' r']),'r');
    save(fullfile(savdir,[FilenameLidar ' ra']),'ra');
    save(fullfile(savdir,[FilenameLidar ' rb']),'rb');
    save(fullfile(savdir,[FilenameLidar ' rc']),'rc');
    save(fullfile(savdir,[FilenameLidar ' rd']),'rd');
    save(fullfile(savdir,[FilenameLidar ' re']),'re');
waitbar(0.15,bar,'Saving data, please wait');
    save(fullfile(savdir,[FilenameLidar ' N']),'N');
waitbar(0.2,bar,'Saving data, please wait');
    save(fullfile(savdir,[FilenameLidar ' L']),'L');
waitbar(0.25,bar,'Saving data, please wait');
    save(fullfile(savdir,[FilenameLidar ' Ts']),'Ts');
waitbar(0.3,bar,'Saving data, please wait');
    save(fullfile(savdir,[FilenameLidar ' Ps']),'Ps');
waitbar(0.35,bar,'Saving data, please wait');
    save(fullfile(savdir,[FilenameLidar ' T']),'T');
waitbar(0.4,bar,'Saving data, please wait');
    save(fullfile(savdir,[FilenameLidar ' P']),'P');
waitbar(0.45,bar,'Saving data, please wait');
    save(fullfile(savdir,[FilenameLidar ' Tm']),'Tm');
waitbar(0.5,bar,'Saving data, please wait');
    save(fullfile(savdir,[FilenameLidar ' C']),'C');
waitbar(0.55,bar,'Saving data, please wait');
```

```
    save(fullfile(savdir,[FilenameLidar ' Cavg']),'Cavg');
    save(fullfile(savdir,[FilenameLidar ' PE']),'PE');
    save(fullfile(savdir,[FilenameLidar ' PEmin']),'PEmin');
waitbar(0.6,bar,'Saving data, please wait');
    save(fullfile(savdir,[FilenameLidar ' beta_m']),'beta_m');
waitbar(0.65,bar,'Saving data, please wait');
    save(fullfile(savdir,[FilenameLidar ' beta_a']),'beta_a');
waitbar(0.7,bar,'Saving data, please wait');
    save(fullfile(savdir,[FilenameLidar ' beta_total']),'beta_total');
waitbar(0.75,bar,'Saving data, please wait');
    save(fullfile(savdir,[FilenameLidar                                '
PhaseFunction']),'PhaseFunction');
    save(fullfile(savdir,[FilenameLidar                                '
PhaseFunction532']),'PhaseFunction532');
    save(fullfile(savdir,[FilenameLidar ' SSA']),'SSA');
    save(fullfile(savdir,[FilenameLidar ' SSA532']),'SSA532');
    save(fullfile(savdir,[FilenameLidar ' SaAERONET']),'SaAERONET');
    save(fullfile(savdir,[FilenameLidar                                '
SaAERONETsync']),'SaAERONETsync');
    save(fullfile(savdir,[FilenameLidar ' SaLidar']),'SaLidar');
waitbar(0.8,bar,'Saving data, please wait');
    save(fullfile(savdir,[FilenameLidar ' LapseRate']),'LapseRate');
    save(fullfile(savdir,[FilenameLidar ' step']),'step');
waitbar(0.85,bar,'Saving data, please wait');
    save(fullfile(savdir,[FilenameLidar ' AOD']),'AOD');
waitbar(0.9,bar,'Saving data, please wait');
    save(fullfile(savdir,[FilenameLidar ' AOD_Lidar']),'AOD_Lidar');
waitbar(0.95,bar,'Saving data, please wait');
    save(fullfile(savdir,[FilenameLidar                                '
AOD_Lidar_max']),'AOD_Lidar_max');
waitbar(1,bar,'Saving data, please wait');
    save(fullfile(savdir,FilenameO),'O');
close(bar);
else
end
```

```
% This is Step12_LoadResultsAndPlot code

% This code is to load the saved inversion results, reduce noise as fit,
then plot

% Noise Reduction
run Path

%%
prompt = {'Enter Filename'};
dlg_title = 'Input';
num_lines = 1;
def = {'V8'};
answer = inputdlg(prompt,dlg_title,num_lines,def);

% Download aerosol inversion results
FilenameLidar = def{1,1};
load('O54');
load([answer{1} ' AOD']);
load([answer{1} ' AOD_Lidar']);
load([answer{1} ' AOD_Lidar_max']);
load([answer{1} ' beta_a']);
load([answer{1} ' beta_m']);
load([answer{1} ' beta_total']);
load([answer{1} ' C']);
load([answer{1} ' Cavg']);
load([answer{1} ' L']);
load([answer{1} ' LapseRate']);
load([answer{1} ' N']);
load([answer{1} ' P']);
load([answer{1} ' PE']);
load([answer{1} ' PEmin']);
load([answer{1} ' PhaseFunction']);
load([answer{1} ' PhaseFunction532']);
load([answer{1} ' Ps']);
load([answer{1} ' ra']);
load([answer{1} ' rb']);
load([answer{1} ' rc']);
load([answer{1} ' rd']);
load([answer{1} ' re']);
load([answer{1} ' SaAERONET']);
load([answer{1} ' SaAERONETsync']);
load([answer{1} ' SaLidar']);
load([answer{1} ' SSA']);
load([answer{1} ' SSA532']);
load([answer{1} ' step']);
load([answer{1} ' T']);
load([answer{1} ' Time']);
load([answer{1} ' Timenum']);
load([answer{1} ' TimenumSSA532']);
load([answer{1} ' TimeSSA532']);
load([answer{1} ' Tm']);
load([answer{1} ' Ts']);
```

```
% clear answer; clear def; clear dlg_title; clear num_lines; clear prompt;

%% Plot L
L_NR = medfilt2(L,[5 5]);

f = figure;
movegui(f,'north');
imagesc(Timenum, r/1000, L');
datetick('x','HHMM','keeplimits');
set(gca,'YDir','normal');
set(gca, 'YLim',[0 10], 'CLim',[0 1.5*10^9]);
colorbar;
colormap(jet);
xlabel({'Time (MDT)'; texlabel('L/O')});
ylabel('Height from Bozeman (km)');
title({[FilenameLidar ' SyncStep=' step ' on ' Time{1}(1:11)];
    ['LapseRate=' num2str(LapseRate) ' K/km  C_\lambda Average Range=['
num2str(rc,'%.0f') ' ' num2str(rd,'%.0f') '] m']});
grid on;

f = figure;
movegui(f,'south');
imagesc(Timenum, r/1000, L_NR');
datetick('x','HHMM','keeplimits');
set(gca,'YDir','normal');
set(gca, 'YLim',[0 10], 'CLim',[0 1.5*10^9]);
colorbar;
colormap(jet);
xlabel({'Time (MDT)'; texlabel('L/O')});
ylabel('Height from Bozeman (km)');
title({[FilenameLidar ' SyncStep=' step ' on ' Time{1}(1:11)];
    ['LapseRate=' num2str(LapseRate) ' K/km  C_\lambda Average Range=['
num2str(rc,'%.0f') ' ' num2str(rd,'%.0f') '] m']});
grid on;

%% Plot beta

f = figure;
movegui(f,'center');
imagesc(Timenum, r/1000, beta_m');
datetick('x','HHMM','keeplimits');
set(gca,'YDir','normal');
set(gca, 'YLim',[0 10], 'CLim',[0 2*10^-6]);
colorbar;
colormap(jet);
xlabel({'Time (MDT)'; texlabel('beta_m (1/m Sr)')});
ylabel('Height from Bozeman (km)');
title({[FilenameLidar ' SyncStep=' step ' on ' Time{1}(1:11)];
    ['LapseRate=' num2str(LapseRate) ' K/km  C_\lambda Average Range=['
num2str(rc,'%.0f') ' ' num2str(rd,'%.0f') '] m']});
grid on;
```

```
beta_total_NR = medfilt2(beta_total,[5 5]);
beta_a_NR = medfilt2(beta_a,[5 5]);

f = figure;
movegui(f,'northwest');
imagesc(Timenum, r/1000, beta_total');
datetick('x','HHMM','keeplimits');
set(gca,'YDir','normal');
set(gca, 'YLim',[0 10], 'CLim',[0 4*10^-6]);
colorbar;
colormap(jet);
xlabel({'Time (MDT)'; texlabel('beta_t_o_t_a_l (1/m Sr)')});
ylabel('Height from Bozeman (km)');
title({[FilenameLidar ' SyncStep=' step ' on ' Time{1}(1:11)];
    ['LapseRate=' num2str(LapseRate) ' K/km  C_\lambda Average Range=['
num2str(rc,'%.0f') ' ' num2str(rd,'%.0f') '] m']});
grid on;

f = figure;
movegui(f,'southwest');
imagesc(Timenum, r/1000, beta_total_NR');
datetick('x','HHMM','keeplimits');
set(gca,'YDir','normal');
set(gca, 'YLim',[0 10], 'CLim',[0 4*10^-6]);
colorbar;
colormap(jet);
xlabel({'Time (MDT)'; texlabel('beta_t_o_t_a_l (1/m Sr)')});
ylabel('Height from Bozeman (km)');
title({[FilenameLidar ' SyncStep=' step ' on ' Time{1}(1:11)];
    ['LapseRate=' num2str(LapseRate) ' K/km  C_\lambda Average Range=['
num2str(rc,'%.0f') ' ' num2str(rd,'%.0f') '] m']});
grid on;

f = figure;
movegui(f,'northeast');
imagesc(Timenum, r/1000, beta_a');
datetick('x','HHMM','keeplimits');
set(gca,'YDir','normal');
set(gca, 'YLim',[0 10], 'CLim',[0 4*10^-6]);
colorbar;
colormap(jet);
xlabel({'Time (MDT)'; texlabel('beta_a (1/m Sr)')});
ylabel('Height from Bozeman (km)');
title({[FilenameLidar ' SyncStep=' step ' on ' Time{1}(1:11)];
    ['LapseRate=' num2str(LapseRate) ' K/km  C_\lambda Average Range=['
num2str(rc,'%.0f') ' ' num2str(rd,'%.0f') '] m']});
grid on;

f = figure;
movegui(f,'southeast');
imagesc(Timenum, r/1000, beta_a_NR');
datetick('x','HHMM','keeplimits');
set(gca,'YDir','normal');
```

```
set(gca, 'YLim',[0 10], 'CLim',[0 4*10^-6]);
colorbar;
colormap(jet);
xlabel({'Time (MDT)'; texlabel('beta_a (1/m Sr)')});
ylabel('Height from Bozeman (km)');
title({[FilenameLidar ' SyncStep=' step ' on ' Time{1}(1:11)];
    ['LapseRate=' num2str(LapseRate) ' K/km  C_\lambda Average Range=['
num2str(rc,'%.0f') ' ' num2str(rd,'%.0f') '] m']});
grid on;

%% Plot AOD

f = figure;
movegui(f,'northeast');
plot(Timenum,AOD,'ro',Timenum,AOD_Lidar(:,find(r==re)),'b*');
%#ok<*FNDSB>
xlim([min(Timenum) max(Timenum)]);
ylim([0 0.5]);
datetick('x','HHMM','keeplimits');
ylabel('AOD');
title({[FilenameLidar ' SyncStep=' step ' on ' Time{1}(1:11)];
    ['LapseRate=' num2str(LapseRate) ' K/km  C_\lambda Average Range=['
num2str(rc,'%.0f') ' ' num2str(rd,'%.0f') '] m']});
grid on;
legend('AERONET','Lidar');

%% Plot Sa
f = figure;
movegui(f,'northwest');
plot(Timenum,SaLidar,'b*-',Timenum,SaAERONETsync,'go-
',TimenumSSA532,SaAERONET,'ro');
xlim([min(Timenum) max(Timenum)]);
ylim([0 100]);
datetick('x','HHMM','keeplimits');
xlabel('Time (MDT)');
ylabel('S_a (sr)');
title({[FilenameLidar ' SyncStep=' step ' on ' Time{1}(1:11)];
    ['LapseRate=' num2str(LapseRate) ' K/km  C_\lambda Average Range=['
num2str(rc,'%.0f') ' ' num2str(rd,'%.0f') '] m']});
grid on;
legend('S_a       Lidar','S_a       AERONET','S_a       AERONET       Raw
Data','Location','Best');

%% Plot Cavg
order = 2;
fcutoff = 0.1;
[B,A] = butter(order,fcutoff);
shift = 3;

Cavg_NR = filter(B,A,Cavg);
Cavg_NR1         =          [Cavg_NR(shift+1:end);          mean(Cavg(end-
shift:end))*ones(shift,1)];
Cavg_NR1(1:8) = mean(Cavg(1:8));
```

```
figure
plot(Timenum,Cavg,'b',Timenum(1:end-
shift),Cavg_NR(shift+1:end),'g',Timenum,Cavg_NR1,'r');
legend(texlabel('C_a_v_g'),[texlabel('C_a_v_g  NR(')  num2str(order)  ','
num2str(fcutoff) ')']);
xlim([min(Timenum) max(Timenum)]);
ylim([0 0.04]);
datetick('x','HHMM','keeplimits');
xlabel('Time (MDT)');
ylabel('C_a_v_g');
title({[FilenameLidar ' SyncStep=' step ' on ' Time{1}(1:11)];
    ['LapseRate=' num2str(LapseRate) ' K/km  C_\lambda Average Range=['
num2str(rc,'%.0f') ' ' num2str(rd,'%.0f') '] m']});
grid on;
```

APPENDIX F

MATLAB CODE FOR OPA NON-DEPLETED AND DEPLETED PUMP

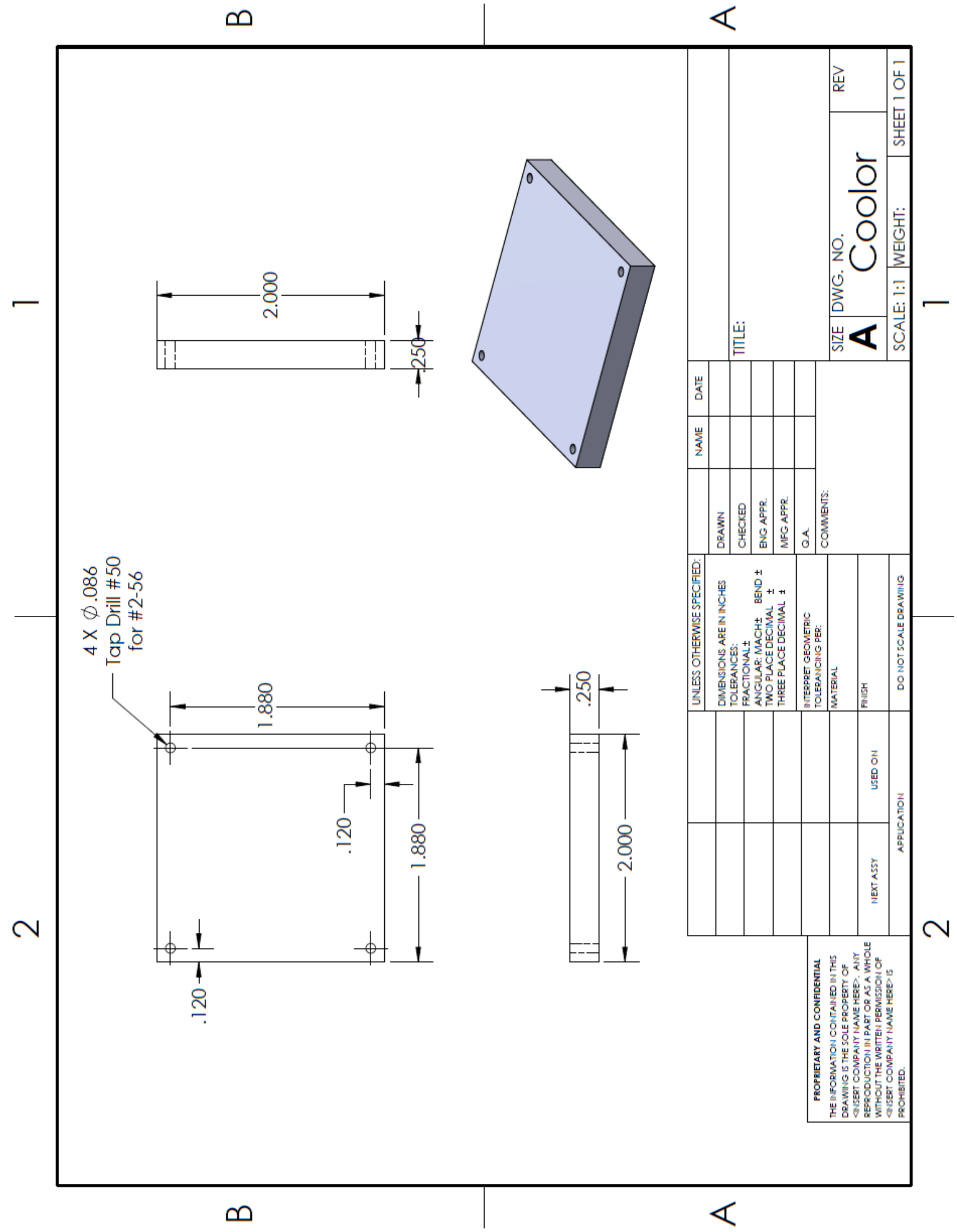
```
% Code for OPA depleted and non-depleted pump

tic;
format long

prompt = {'Number of steps in the numerical integration',...
    'Input pump energy (mJ)',...
    'Input signal power (mW)',...
    'Input idler power (mW)',...
    'Length of crystal (mm)',...
    'Crystal index of refraction (assume constant for each
wavelength)',...
    'd_e_f_f (pm/V)',...
    'Wavelength of signal (nm)',...
    'Wavelength of idler (nm)',...
    'Wavelength of pump (nm)',...
    'Radius of signal beam (mm)',...
    'Radius of pump beam (mm)'};
dlg_title = 'Input';
num_lines = 1;
def                                                                  =
{'100000','0:0.1:20','160','0','32','2.2','14','1571','3297','1064','0.
7','0.7'};
answer1 = inputdlg(prompt,dlg_title,num_lines,def);

% Parameters
numsteps = str2num(answer1{1}); %#ok<*ST2NM> %number of steps in the
numerical integration

% Creates vectors of zeros to place data in
As = zeros(numsteps,1);
Aistar = zeros(numsteps,1);
Ap = zeros(numsteps,1);
time = zeros(numsteps,1);

Epump = str2num(answer1{2}); % Input pump energy
EsDep = zeros(1,size(Epump,2)); % Output signal energy
PsDep = zeros(1,size(Epump,2)); % Output signal power
EiDep = zeros(1,size(Epump,2)); % Output idler energy
PiDep = zeros(1,size(Epump,2)); % Output idler power
EpDep = zeros(1,size(Epump,2)); % Output pump energy
PpDep = zeros(1,size(Epump,2)); % Output pump power

Ps0 = str2num(answer1{3})*10^-3; % Input signal power (W)
Pi0 = str2num(answer1{4})*10^-3; % Input idler power (W)

L = str2num(answer1{5})*10^-3; %length of crystal in m

% Constants
u0 = 1.26e-6; % Vacuum permeability (J s^2/m C^2)
e0 = 8.85e-12; % Vacuum permittivity (C^2/J m)
c = 3.0e8; % Speed of light (m/s)
```

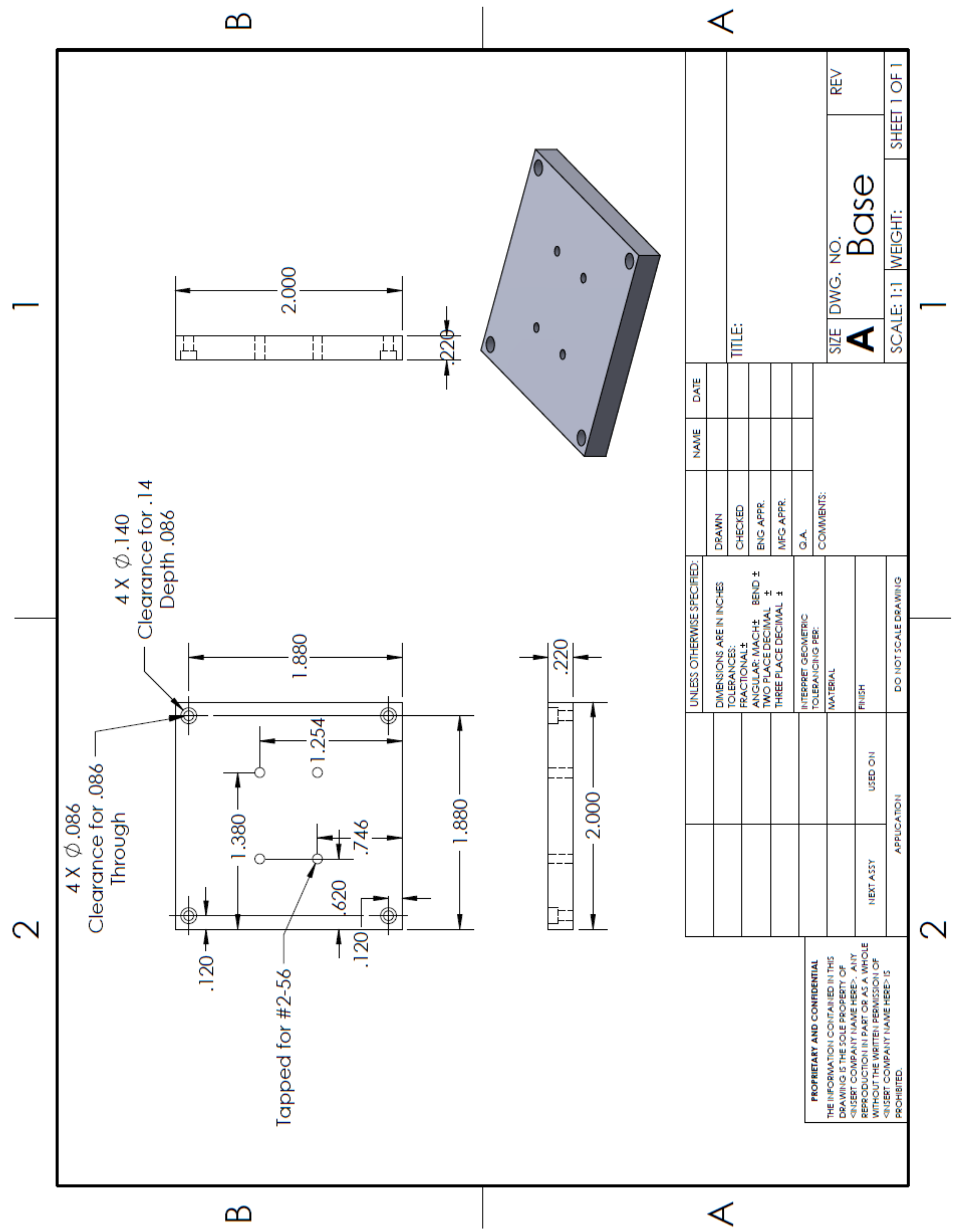
```
n = str2num(answer1{6}); % Index of refraction, assumed constant for each
wavelength
deff = (10^-12)*e0*str2num(answer1{7}); % d effective (C^3/J^2), to
convert to m/V, divide by e0

lambdas = str2num(answer1{8})*10^-9; % Wavelength of signal (m)
lambdai = str2num(answer1{9})*10^-9; % Wavelength of idler (m)
lambdap = str2num(answer1{10})*10^-9; % Wavelength of pump (m)

ws = (2*pi*c)/lambdas; % Angular frequency for signal (1/s)
wi = (2*pi*c)/lambdai; % Angular frequency for idler (1/s)
wp = (2*pi*c)/lambdap; % Angular frequency for pump (1/s)

Abs = (pi)*(str2num(answer1{11})*10^-3)^2; % Beam diameter with .1 cm
beam diamater (m^2)
Abp = (pi)*(str2num(answer1{12})*10^-3)^2;

As(1) = sqrt(((2*Ps0)/(Abs*ws))*(sqrt(u0/e0))); % Starting amplitude for
signal (J s^.5/m C)
Aistar(1) = sqrt(((2*Pi0)/(Abs*wi))*(sqrt(u0/e0))); % Starting amplitude
for idler (J s^.5/m C)

k = deff*(sqrt((u0*ws*wi*wp)/(e0*n^3))); % Gain term kappa (C/J s^.5)
pulsewidth = 10e-9; % Width of laser pulse (s)

bar = waitbar(0,'Performing integration, please wait');

% Depleted pump calculation

% This for loop calculates the output energy as a function of input pump
energy
for E = 1:size(Epump,2)

Ppeak = 0.94*(Epump(E)/1000)/pulsewidth; % Convert from energy to power
(W)
Ap(1) = sqrt(((2*Ppeak(1))/(Abs*wp))*(sqrt(u0/e0))); % Initial value of
Ap

    % This for loop numerically integrates the coupled wave equations
through the crystal. The output energy can be determined from this
    for z = 1:numsteps-1
        dt         = L/(numsteps*c); % d breaks up the crystal for the
integration
        time(1)  = dt;
        time(z)  = z*dt;

        dAs      = (-1i*k*c*Ap(z)*Aistar(z)*dt);
        dAistar = 1i*k*c*conj(Ap(z))*As(z)*dt;
        dAp      = -1i*k*c*As(z)*conj(Aistar(z))*dt ;

        As(z+1)      = As(z) + dAs;
```

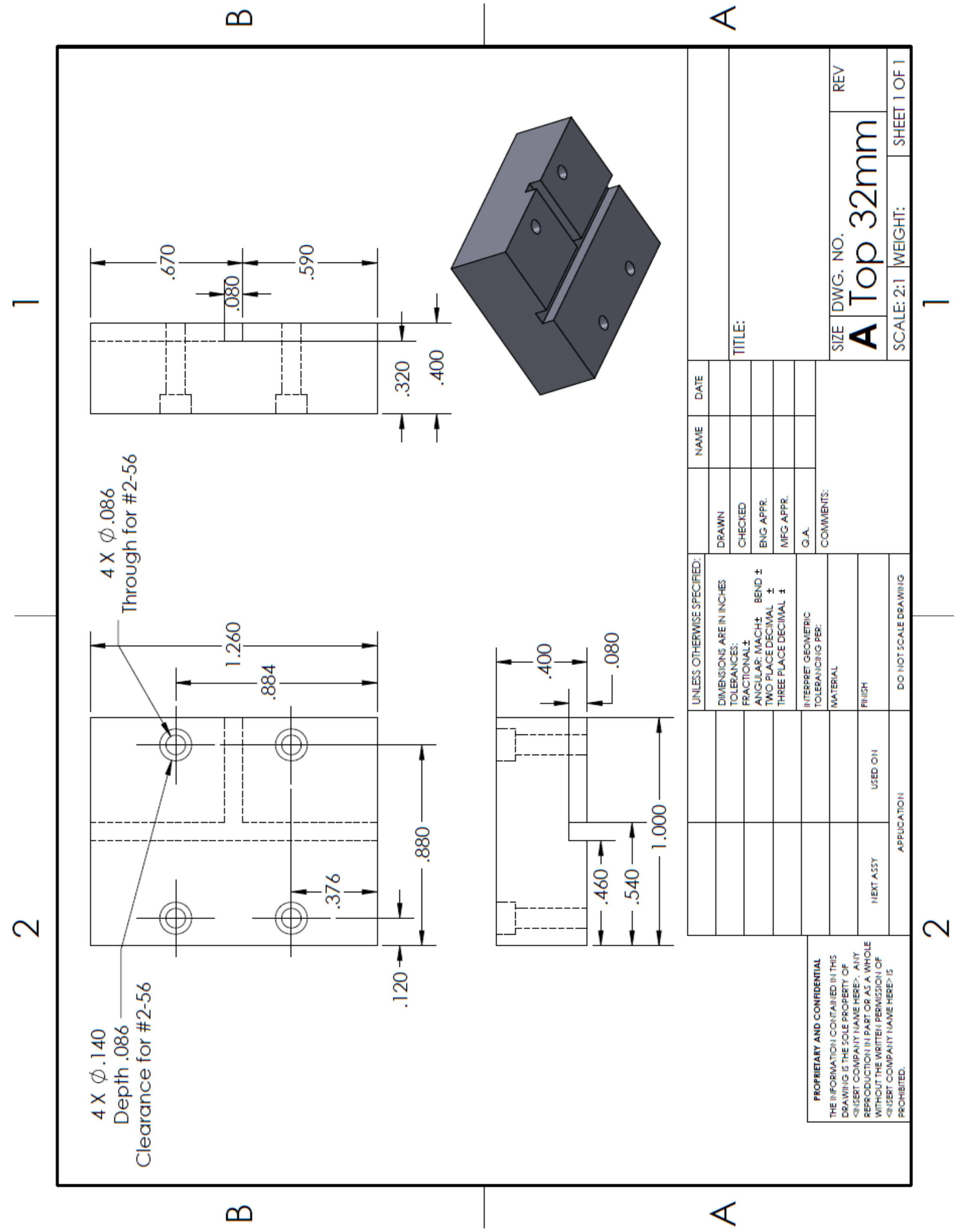
```
        Aistar(z+1) = Aistar(z) + dAistar;
        Ap(z+1)     = Ap(z)   + dAp;
    end

PsDep(E)  =  (abs(As(z+1)^2))*(Abs*ws/2)*(sqrt(e0/u0));  %  Output  signal
power in W (depleted pump approx)
EsDep(E)  =  PsDep(E)*pulsewidth*1000;  %  Output  signal  energy  in  mJ
(depleted pump approx)

PiDep(E) = (abs(conj(Aistar(z+1)))^2)*(Abs*wi/2)*(sqrt(e0/u0)); % Output
idler power in W
EiDep(E) = PiDep(E)*pulsewidth*1000; % Output idler energy in mJ

PpDep(E) = (abs(Ap(z+1)^2))*(Abp*wp/2)*(sqrt(e0/u0)); % Output pump power
in W
EpDep(E) = PpDep(E)*pulsewidth*1000; % Output pump energy in mJ

waitbar(E/size(Epump,2),bar,'Performing integration, please wait');
end
close(bar);

% Non-depleted pump calculation
PpeakNondep = 0.94*(Epump./1000)./pulsewidth;
ApNondep = sqrt(((2*PpeakNondep)./(Abs*wp))*(sqrt(u0/e0)));
AsNondep = As(1).*(cosh(k.*ApNondep.*L));
AiNondep = 1i*As(1).*(sinh(k.*ApNondep.*L));

PsNondep = (abs(AsNondep.^2))*(Abs*ws/2)*(sqrt(e0/u0));
EsNondep = PsNondep*pulsewidth*1000;

difference = EsNondep - EsDep;

% Plot
figure
plot(Epump, EsDep, 'r', Epump, EsNondep, 'b')
title('Depleted vs Non-Depleted Pump Output')
xlabel('Input Pump Energy (mJ)')
ylabel('Output Signal Energy (mJ)')
legend('Depleted','Non-depleted','Difference','location','northwest')
xlim([0 20])
ylim([0 20])
grid on

RunTime          =          [num2str(floor(toc/3600),'%02i')          ':'
num2str(floor(rem(toc,3600)/60),'%02i')                                ':'
num2str(round(rem(rem(toc,3600),60)),'%02i')]
```

APPENDIX G

SOLIDWORKS DRAWINGS FOR PPLN CRYSTAL HOUSINGS


4 X ∅.086
Tap Drill #50
for #2-56
1.880
1.880
.120
.120
2.000
.250
2.000
.250
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC
TOLERANCING PER:
MATERIAL
FINISH
DO NOT SCALE DRAWING
NEXT ASSY
USED ON
APPLICATION
NAME
DATE
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
TITLE:
SIZE
A
DWG. NO.
Coolor
REV
SCALE: 1:1
WEIGHT:
SHEET 1 OF 1
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.


4 X ⌀.086
Clearance for .086
Through
4 X ⌀.140
Clearance for .14
Depth .086
Tapped for #2-56
1.880
1.880
1.254
1.380
.746
.620
.120
.120
2.000
.220
2.000
.220
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL±
ANGULAR: MACH± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
DO NOT SCALE DRAWING
NEXT ASSY
USED ON
APPLICATION
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
NAME
DATE
TITLE:
SIZE
A
DWG. NO.
Base
REV
SCALE: 1:1
WEIGHT:
SHEET 1 OF 1
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.


4 X ∅.140
Depth .086
Clearance for #2-56
4 X ∅.086
Through for #2-56
1.260
.884
.880
.376
.120
.400
.080
.460
.540
1.000
.670
.590
.080
.320
.400
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
DO NOT SCALE DRAWING
NEXT ASSY
USED ON
APPLICATION
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
NAME
DATE
TITLE:
SIZE
A
DWG. NO.
Top 32mm
REV
SCALE: 2:1
WEIGHT:
SHEET 1 OF 1
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.
A
B
1
2