Measuring Low Intensity Light Reflections of High Temperature CO_2 Adsorbed on SiO_2 Near Brewster's Angle

Thomas Ferron
Oregon State University
Department of Physics
Research Associate - Thomas Windom
Advisor - Dr. William Hetherington

May 31, 2013

Abstract

This investigation expands on earlier methods of using surface polarization reflection (SPR) to characterize the first few monolayers of a surface. Past research measured intensities of light reflecting off an SiO_2 optical flat inside a vacuum. Pressure of applied gas was changed to view a change in reflection. In an attempt to remove the need of a vacuum chamber, this experiment was run in the open air. P-polarized light is reflected at Brewster's angle to measure low intensity reflections. Using an SBIG 8300 CCD camera intensities are measured in an open air environment and compared to that of applied CO_2 by the use of a gas jet. SPR theory predicts that higher polarizabilities will cause a larger reflection.

Temperature of the applied gas was increased in order to quantify a difference in reflections based on this change in temperature. Removing ambient particles from the open air by use of a heated gas stream was thought to change the signal to a more consistent level of measurement. The intensity was found to change at higher temperatures but no trend was quantified. Increasing the temperature beyond 323K caused signal fluctuation and was found not to be beneficial in SPR analysis. SPR was found to not have been improved by heating the applied gas.

Acknowledgements

A big thanks to the Oregon State University Physics department for funding this research. This work would not have been possible without the support and insight from my advisor Dr. William Hetherington. A special thanks to Dr. Janet Tate who not only helped develop my writing but also encouraged me to start writing early and often to make this a less painful adventure. Finally thanks to Thomas Windom who was an invaluable research associate as well as having infinite insight on being able to automate and refine code saving countless hours of copy-pasting files.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The motivation behind surface polarization reflection (SPR) is a desire to understand light's interaction with a real material, which contains a bulk and a surface. Fresnel equations predict that at a specific angle based on the refractive indices of two adjacent media, no reflection will occur. Exploring this angle, more formally called Brewster's angle, is the goal of this experiment, more specifically, observing how, in an experimental setting, light exhibits a faint reflection at Brewster's angle. By adsorbing gas molecules to an SiO_2 optical flat, the investigation looks to observe how the intensity of this reflection changes. Using a model developed by William Hetherington and Chang-Yuan Ju, we aim to predict these reflected intensities even when the standard theory of light predicts no reflection will occur.

The original investigation took place to demonstrate a simple technique to analyze a surface, this technique is known as surface polarization reflection. By looking at the reflection caused at Brewster's angle, one would theoretically be able to know what molecules occupy the first few layers of a material. In an earlier experiment reflection intensities have been measured using CO_2, H_2, O_2, and N_2 in a vacuum environment [5]. In order to find a technique that more accurately describes the real world, an open air environment is tested. Originally, this environment was not explored because open air contains many unknown particles that can attach themselves to the surface and change its polarization, as well as the reflection measured. However in order to find an easier way to perform this analysis an open air environment is explored.

This paper explores the opportunity of using open air to conduct these measurements. To expand upon an open air test, in this investigation high temperature gas will be applied in order to attempt at removing these unknown particles. Removing these particles should give a clearer understanding of the reflections that occur near Brewster's angle.

1.2 Theory

1.2.1 Macro Scopic Interactions

The theoretical prediction of light's interaction at Brewster's angle comes from the electromagnetic theory of light. The index of refraction is used to quantify this interaction and is defined by the ratio of the speed of light in a vacuum compared to the speed of light in a bulk material. The speed of light in a medium is

$$v = \frac{1}{\sqrt{\mu\epsilon}} \tag{1.1}$$

and the index of refraction is

$$n = \frac{c}{v} = \sqrt{\frac{\mu\epsilon}{\mu_0\epsilon_0}} = \sqrt{\epsilon_r} \tag{1.2}$$

light therefore moves at different speeds in different materials. By using this information Fresnel equations were developed to describe the reflection and transmission of light as it passes between media. A derivation of this theory will be outlined in Appendix A and the results are shown in Fig. 1.2

The equations describe these interactions with two different polarizations of light, parallel polarization, which will be referred to as P-polarization and perpendicular polarization, which will be referred to as S-polarization. This interaction can be seen in Fig. 1.1, the reflected and transmitted waves are shown as rays to represent the angle they interact with the surface. The angles of transmission and reflection both obey Snell's law and the law of reflection.

Amplitudes of these reflections and transmissions can be seen in Fig. 1.2.

The pertinent equation to our experiment is the curve labelled $r_{\parallel}$. This equation represents the ratio of amplitudes between the initial electric field of the incident and the reflected wave with parallel polarization. This is the only Fresnel equation that can have zero magnitude.

$$r_{\parallel} = \frac{\tan(\theta_i - \theta_t)}{\tan(\theta_i + \theta_t)} \tag{1.3}$$

where θ_i is the angle of incident, and θ_t is the angle of transmission. The angle that corre-

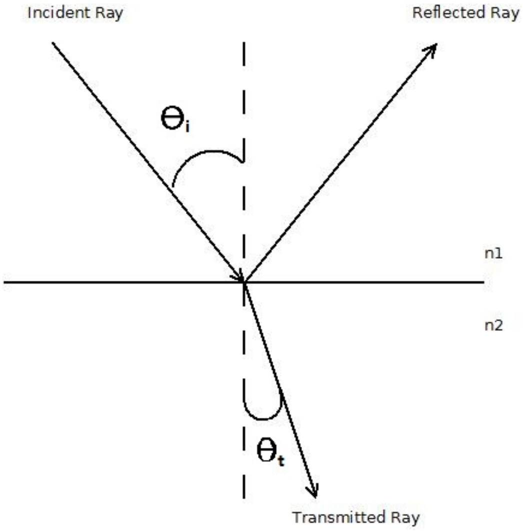


Figure 1.1: Interaction between light and a surface

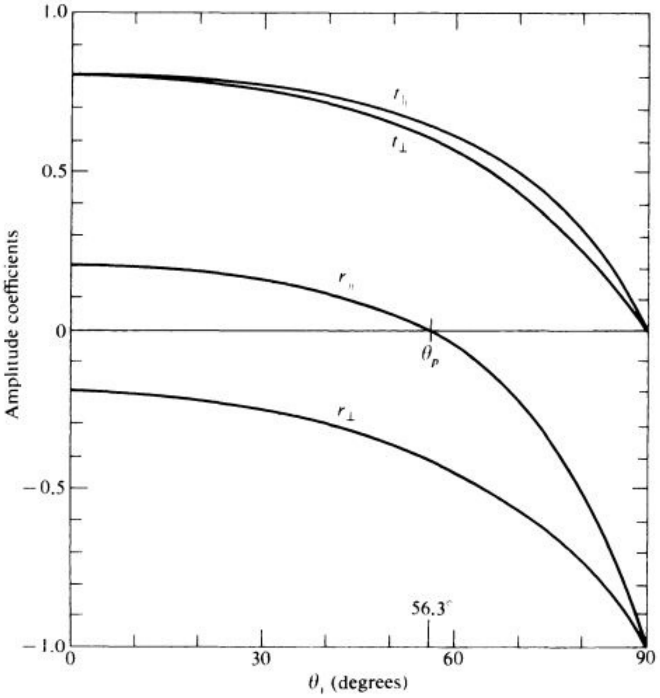


Figure 1.2: Amplitude of reflected and transmitted coefficients of light applied to a glass surface. The measured intensities in this experiment will be the square of the amplitude coefficients. [3]

sponds to zero reflectance is known as Brewster's angle (or Polarization Angle). This angle occurs when the entire incident wave is theoretically transmitted through the interface and will not be reflected. Brewster's angle is given by:

$$\tan(\theta_B) = \frac{n_2}{n_1} \tag{1.4}$$

This model is developed to apply only to bulk material reflections and so the new model developed looks at the surface interactions. The basic understanding of this new model is outlined in the following section.

1.2.2 Micro Scopic Interactions

When light is P-polarized, the electric field associated with the incident light is oscillating parallel to the surface. This causes the excited dipole moments to oscillate in the direction that the reflected wave would propagate. Since dipoles do not radiate in the direction of the dipole's oscillation, no reflection occurs. This is the understanding for a bulk modulus and does not apply to the surface molecules. In a bulk material the electric dipoles are aligned in a specific direction and do not vary. The closer a dipole is to the surface they are able to orient themselves in different directions, allowing for this small reflection.

A model developed by Chang-Yuan Ju predicts that molecules on the outer layers of a surface can be modelled to radiate the appropriate reflected wave. The magnitude of the reflected wave was found to correspond to the polarizibility of the surface molecules. This theory finds that the reflection follows very closely with the Lorentz local field for an isotropic cubic lattice. [6]

$$E_{loc} = \frac{n^2+2}{3}\frac{n\cos(\theta) - \cos(\theta_t)}{n\cos(\theta) + \cos(\theta_t)}E_0 \tag{1.5}$$

This prediction can be seen in Fig 1.3.

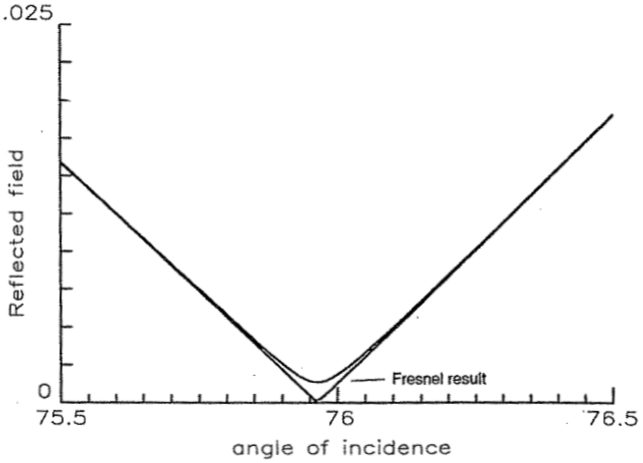


Figure 1.3: SPR prediction compared to Fresnel result [6]

The curves in Fig 1.3 show both the predicted value of the reflected field from Fresnel equations as well as the approximation made by the SPR model. In order to create such a large reflection a material with a refractive index of 4 is used. A high refractive index causes an amplification of the emitted field in order to better measure its effect.

We aim to measure this reflection for a super-polished glass surface composed of fused silica in an open air environment. To further this investigation, we are applying CO_2 by the use of a gas jet and investigating how this reflection changes based on the temperature of the applied CO_2.

Chapter 2

Experimental Design

2.1 Experimental Method

This experiment is designed to create a spread of wave-vectors to reflect off a super-polished glass surface at, and near, Brewster's angle. A diode laser, passing through a focusing lens, with the polished surface placed at the lens focal point interacts with the surface as shown in Fig. 2.1. The reflected beam will contain a cleave in the center where the incident wave-vectors were approaching at Brewster's angle. Using a CCD camera we measure the intensity of the laser at the location of this cleave and compare it to the theoretical reflection intensity predicted by Fresnel equations and Ju's theory.

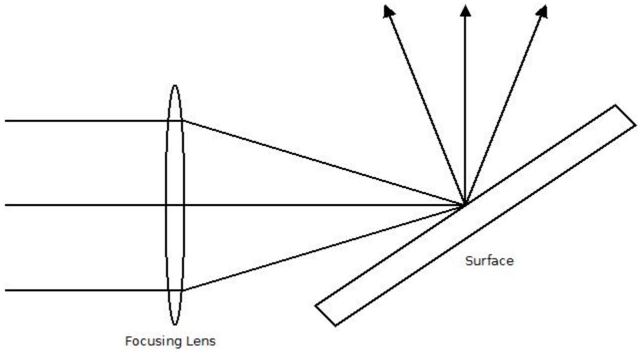


Figure 2.1: Focusing apparatus

In order to measure this reflection under various conditions, a gas jet is used to apply CO_2 molecules to the surface. A heating coil will be used to heat the gas before application. This will allow us to test how the surface molecules are responsible for the reflection. As well as investigating if an open air environment will mask this reflection due to the surrounding particles in the air.

2.2 Experimental Setup

2.2.1 Optical Path

Fig. 2.2 outlines the experimental setup for this investigation. Starting with a 660 nm Panasonic LNCQ05PS diode laser, the beam is moved using mirrors 1 and 2 (from Fig. 2.2) in order to easily adjust it with other optical equipment. The beam is first cleaned by passing it through a Newport Model 900 spatial filter. This filter consists of a microscope objective that Fourier transforms the beam, and a $5\mu m$ aperture is placed at the focal plane to block out all spatial frequencies other than the lowest oscillating mode.

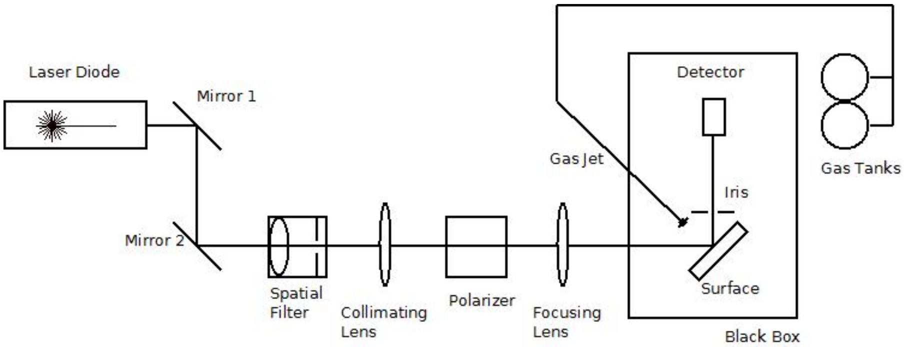


Figure 2.2: Experimental Setup

Following the spatial filter, the beam is collimated using a LAV033 10cm focal length lens. This is placed immediately after the spatial filter in order to collect as much light from the laser as possible since it diverges quickly from the aperture. This causes the lens to clip a small portion of the laser creating a slight diffraction pattern. In order to have the correct polarization for the incident wave, two Thorlabs PBS252 Beam splitting cubes are used to P-polarize the laser.

The PBS252 beam splitting cube is a polarizer designed to pass P-polarized light with no deflection and redirect S-polarized light 90° (see Fig. 2.3). This allows the S-polarized light to be redirected and only pass through P-polarized light required by Fresnel equations. The advertised extinction ratio of the polarizer is $T_p : T_s > 1000 : 1$, where T_p is the intensity of the P-polarized light and T_s is the intensity of the S-polarized light. In order to compound this ratio and essentially eliminate all S-polarized light a second PBS252 is placed immediately after the first polarizer to ensure the intensity caused by the S-polarized light is one part in a million compared, to the P-polarized light.

After the light has been polarized, it gets focused using a KPX118 50cm focal length lens. This will allow us to focus a spread of wave-vectors based on the beam's diameter. The

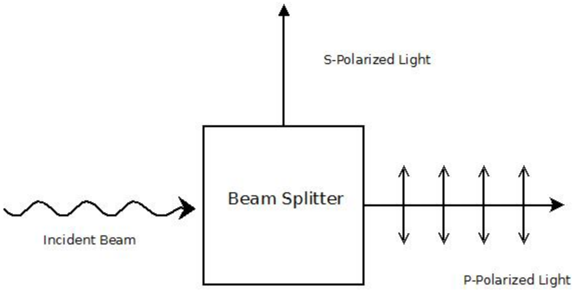


Figure 2.3: Beam splitting cube polarization technique

half angle of the wave-vectors is calculated using the arctan of the distance from the surface to the lens and half the beams diameter giving a half-angle of 2.85°. When the vectors come to a point at the glass surface, the beam is redirected toward the data collection device.

The surface is a 1 inch fused silica substrate with a measured refractive index of 1.456 [7]. Appendix C contains the data specifications for the surface. The important details to note, come from the roughness profile. The thickness of the surface near its center varies by less than 2 angstroms at its maximum difference. This allows us to model this surface as perfectly flat and so we can ignore any reflection due to scattering off the surface itself.

Immediately following the surface is an iris used to block out the reflection caused by the second interface of the glass surface. This is so the two reflections will not interfere at the CCD camera. Between the surface and the CCD camera a third polarizer is added to reduce the intensity of the beam and remove any S-polarized light that may remain.

2.2.2 Gas Jet

A gas jet is used to apply a thin layer of CO_2 molecules to the surface. The jet apparatus consists of a gas cylinder with polyethylene tubing leading from the pressure regulator to the final jet applicator. A micron filter is used to help remove contaminants from the applied gas before it is applied to the surface. A nichrome heating coil surrounds the final applicator in order to increase the temperature the gas as it is applied to the surface. Using a 120V adjustable power supply, the heating coil is able to change the temperature of the gas as it approaches the surface. Temperatures range from 295K to 495K and for this experiment three temperatures will be explored, 295K, 323K, and 373K. A thermistor is placed in between the gas jet and the surface in order to directly measure the temperature.

The gas chosen to test is CO_2. It was chosen because it is only a small percentage of airs composition as well as having a different polarizability. CO_2 has a polarizability of $2.911 \cdot 10^{-24} cm^-3$ and since air is mostly composed of N_2, with a polarizability of $1.7403 \cdot 10^{-24} cm^-3$

[9], CO_2 has a larger polarizability than air. From SPR theory, this suggests that when CO_2 is adsorbed on the surface the reflection will be larger than that of air.

2.3 Data Acquisition

2.3.1 CCD Camera

The camera used is an SBIG STF-8300 containing a KAF-8300 Image Sensor CCD with a resolution of 3326x2504 pixels. This camera allows us to expand past experiments by taking an image of the entire reflected laser. Older experiments used a single photo-diode to image a small area within the reflected beam. CCD's work by creating a charge based on the interactions with incoming photons on each pixel. The quantum efficiency of the KAF-8300 Image Sensor is 54% [8], this corresponds to the percentage of incoming photons that produce a carrier charge. This CCD carries a maximum charge of $25.5ke^-$, and when fully saturated, produces a signal of $64,400ADU$(analog to digital units). The camera contains a gain of $0.37e^-/ADU$, therefore, since each photon corresponds to $\approx 0.54e^-$, each photon then corresponds to a signal of $1.46ADU$.

The camera itself has an inherent noise in the signal that was taken into effect of our measurements. It has a general read noise that corresponds to $\approx$ 9.3 electrons, giving a signal of $\approx$ 25.135. The dark current will correspond to a noise of 0.02 electrons per pixel, since this value is so much less than our read noise we will essentially ignore it. Finally the camera contains an inherent offset in order to not have a negative value for the signal. For a single exposure the offset is set at a signal of 1140. To negate this offset a dark frame image is taken that subtracts an image with a closed shutter to simulate removing all noise. Dark frame images add a signal of 100 to each image in order to remove the possibility of a negative signal.

2.3.2 Analysis Programs

Using CCDOPS, the camera control software, pictures of the laser are able to be taken. The intensity can then be read using a program written by William Hetherington entitled SBIGreader.py. This program images the picture taken by CCDOPS and allows us locate the center of the beam to further analyze it. By using a program written by Thomas Windom entitled Analysis_1_Extraction.py, 10 exposures are averaged, and 300 adjacent rows are summed in order to amplify the small changes we are looking for.

2.3.3 Data Analysis

Two major obstacles exist in order to properly interpret the collected data. The first is to remove the fixed pattern present in our data created by the various optics. Second, to align the minima of comparable data sets as Brewster's angle shifts for the different indices of refraction.

We will assume our data follows the form, $f(\theta)M(\theta) = d(\theta)$. Given $d(\theta)$ as the raw data collected, $f(\theta)$ is the curve that will contain the interesting results we will extract, and $M(\theta)$ is the modulation function that is generated by the laser and optics that create the fixed pattern in the data.

By assuming the above form, a best fit polynomial is generated using the program Analysis _2_Modulation_2.py. This polynomial can be set as the function $f(\theta)$ and will be used with the baseline air measurements to solve for $M(\theta)$.

$$M(\theta) = \frac{d(\theta)}{f(\theta)}$$

This modulation function will be divided out of the CO_2 data in order to compare results using the relation

$$f(\theta) = \frac{d(\theta)}{M(\theta)}$$

. This modulation is reported in chapter 3.

Now that the data no longer contains the fixed pattern it can be shifted freely to align Brewster's angle. By applying different gases to the surface, Brewster's angle shifts based on the change of refractive index. The index of refraction of CO_2 is 1.00045 and for Air it is 1.00027 [9] The shift in the signal can then be calculated using the geometry of Fig. 2.4 such that $d(\tan(\theta_1 - \theta_2)) = l$ where d is the distance from the surface to the camera, l is distance that the Brewster angle will shift, and θ_1 and θ_2 are Brewster's angle for the $CO_2 - Surface$ and $Air - Surface$ interfaces. The total shift caused by CO_2 corresponds to a distance of $34\mu m$. This is a shift of 7 pixels on the camera at $5.4\mu m/pixel$. [1] This is calculated using the arctan of the length of a single pixel divided by the distance between the surface and the camera giving a resolution of 0.00079° per pixel.

When temperature begins to be applied to the surface a second shift in the index of refraction occurs. Fused silica changes index of refraction with temperature shifts according to the Sellmeier formula. [7]

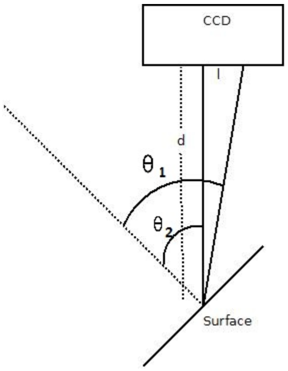


Figure 2.4: Geometry to calculate the changed location for Brewster's angle

$$\frac{dn_{abs}(\lambda, T)}{dT} = \frac{n^2(\lambda, T_0) - 1}{2n(\lambda, T_0)} \cdot (D_0 + 2 \cdot D_1 \cdot \Delta T + 3 \cdot D_2 \cdot \Delta T^2 + \frac{E_0 + 2 \cdot E_1 \cdot \Delta T}{\lambda^2 - \lambda_{TK}^2}) \quad (2.1)$$

The constants for the equation can be found in Appendix C for Schott glass FK3. Since the temperature is not going to be changed by more than 100K the shift will not be very large. Using the Sellmeier formula the change in refractive index causes a shift in Brewster's angle by less than 1 pixel.

Chapter 3

Results and Analysis

3.1 Room Temperature CO_2

In order to compare changes in signal due to high temperature CO_2, the first requirement is to know how CO_2 changes the reflection of light at room temperature. CO_2 was applied to the surface and compared to the reflections caused in open air. The polynomial used to fit the air data was the highest order polynomial that would retain the appropriate shape of data at the minimum curve. Higher orders were able to simulate the data more closely but in some cases caused inappropriate shapes to form that would distort the comparison. The most common polynomial used was an 32^{nd} order polynomial. This polynomial was fit to the air data and using the method outlined in section 2.3.3 enabled the modulation function to be extracted from the data. Fig. 3.1 is a sample modulation function, and Fig. 3.2 compares the best fit polynomial of the air data to the CO_2 data after the modulation function is divided out. Dividing by this modulation function, the CO_2 data was extracted from the fixed pattern imbedded in the beam.

Since there is a shift in indices of refraction, the CO_2 measurements in Fig. 3.3 were made after the CO_2 results were shifted by 7 pixels. This is to align the locations of Brewster's angle in both situations (see section 2.3.3). The difference between the two data sets were summed pixel by pixel to compute the average difference. This summation only occurred in the 20 pixels surrounding the minimum value since that is the location near Brewster's angle that we want to measure. In order to relate different trials this sum was divided by the daily air signal summed over the same range of pixels in order to find the percent increase in signal caused by the CO_2. Fig. 3.3 shows the area that is summed around Brewster's angle. Appendix B contains the intensity changes for each trial. The average increase was found

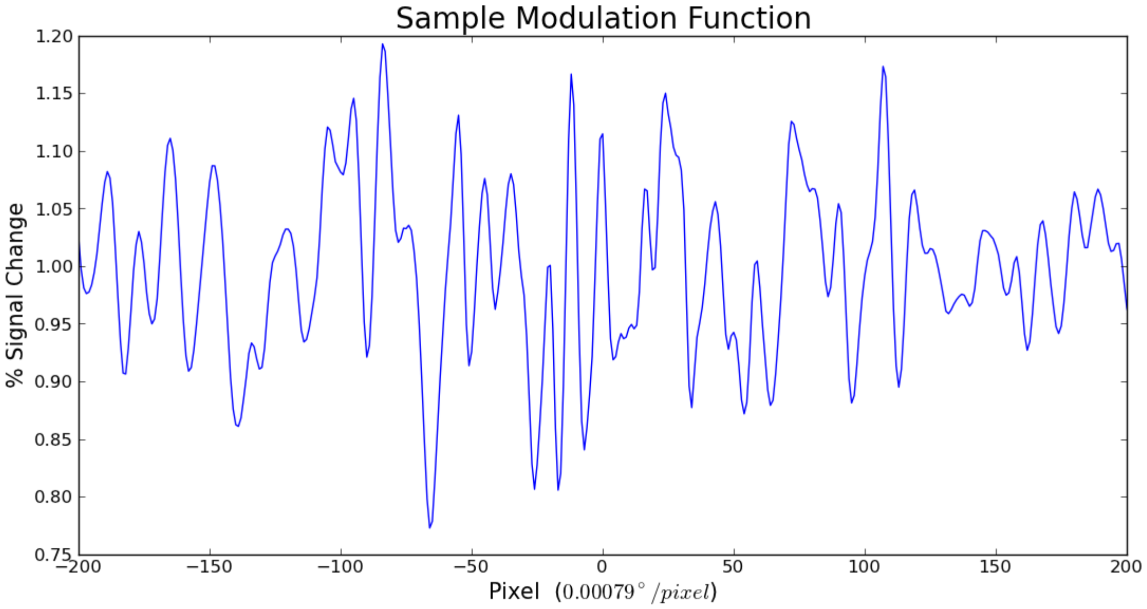


Figure 3.1: Sample Modulation Function

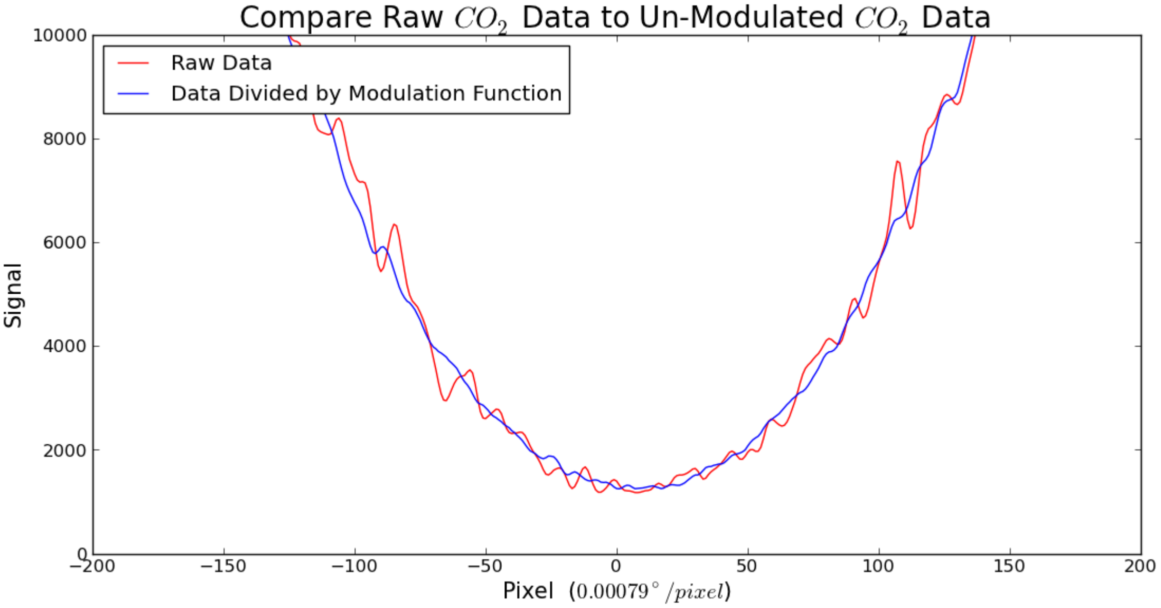


Figure 3.2: Comparison of the same set of data before and after the modulation function has been removed

to be 3.98% ± 1.55%. For a detailed analysis on these results see *SPR: Surface Polarization Reflections from few Mono-layer Adsorbate on* SiO_2 by Thomas Windom.

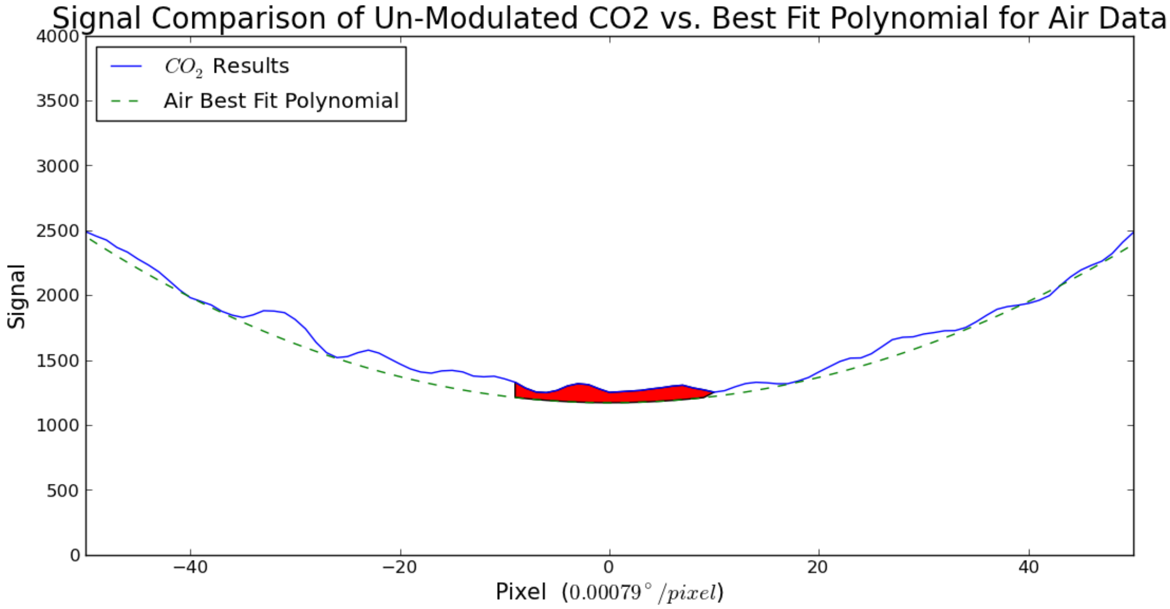


Figure 3.3: Comparison between the un-modulated CO_2 measurements compared to air. The red region is the 0.016° range around Brewster's angle where the surface reflection has a higher intensity than the bulk reflection.

3.2 High Temperature CO_2

3.2.1 High Temperature

Heating the CO_2 was thought to remove ambient particles from the surface, and allow our measurements to only contain information regarding the surface layer and the CO_2. Fig. 3.4 shows that for three different temperatures, 295K (Room temperature), 323K, and 373K the signal did not change in the expected manor. In fact, the only useful data that was able to be extracted was at room temperature, where the percent increase was in the expected range of values to those referenced in section 3.1. The 323K line was shifted more than the expected result calculated from the Sellmeier formula and the signal had a similar increase as room temperature did. The 373K data contained a much larger shift, similarly, not calculated by the Sellmeier formula, a second modulation was also introduced into the data.

3.2.2 High Temperature CO_2 Analysis

Observations were able to be made after running several sets of trials. Each set of data included three consecutive trials consisting of a first baseline measurement in open

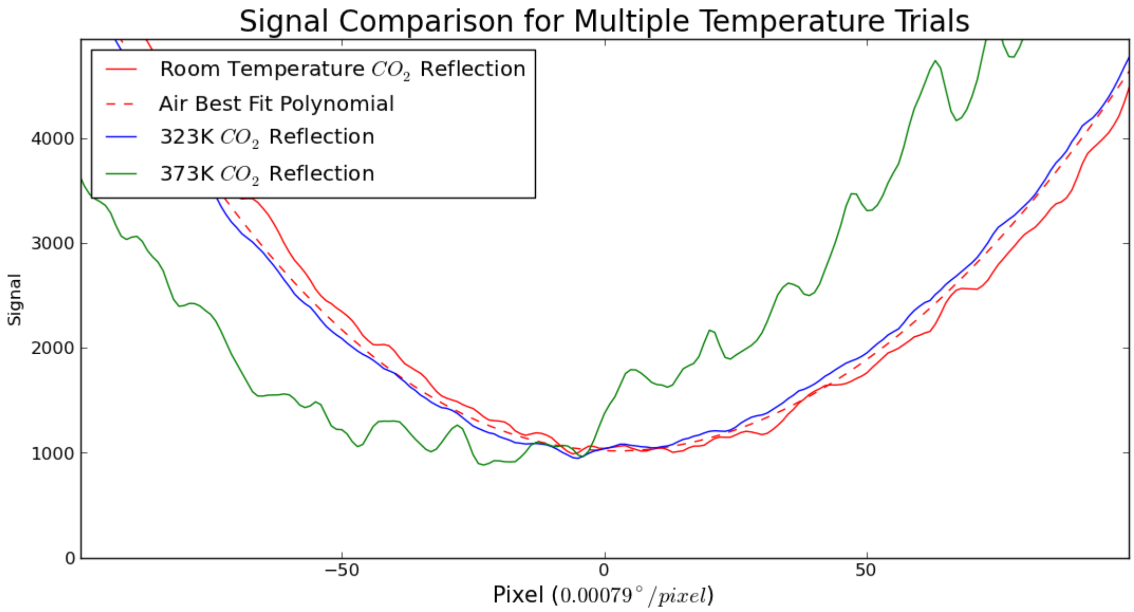


Figure 3.4: Comparison of un-modulated signals where the dotted red line represents the air best fit polynomial.

air, followed by CO_2 measurements at three temperatures, 295K, 323K, and 373K. Since the room temperature intensities fell in the range of values found in the data presented in section 3.1, we can assume that each trial did not effect the following measurements after the temperature returned to 295K. This lets us assume each trial was independent of the trial before. Fig. 3.4 shows that the 323K, measurements did not have any new modulation introduced and so the act of heating the CO_2 to 323K did not change the fixed pattern in the beam. The method of analysis outlined in section 2.3.3 is able to be used to measure the signal increase for this temperature. Fig. 3.5 shows three sets of temperature measurements (with a complete list found in Table 4.1) and no trend was found between trials. Any trend found in a set of data was unable to be reproduced on a different day of data collection. This prevents us from running any statistical analysis on how a temperature increase of 50K effects the measurements.

The second aspect of the data is when the temperature is increased to 373K. A second modulation pattern is introduced after the original modulation has been divided out from the data. Appendix B contains Fig. 4.1 and Fig. 4.2 to show that the initial modulation function changes at this higher temperature. For room temperature, the pattern is fixed based on the location in the beam. At 373K, the pattern changes causing the introduction to the new modulation shown by the green line in Fig. 3.4 This causes a problem with our

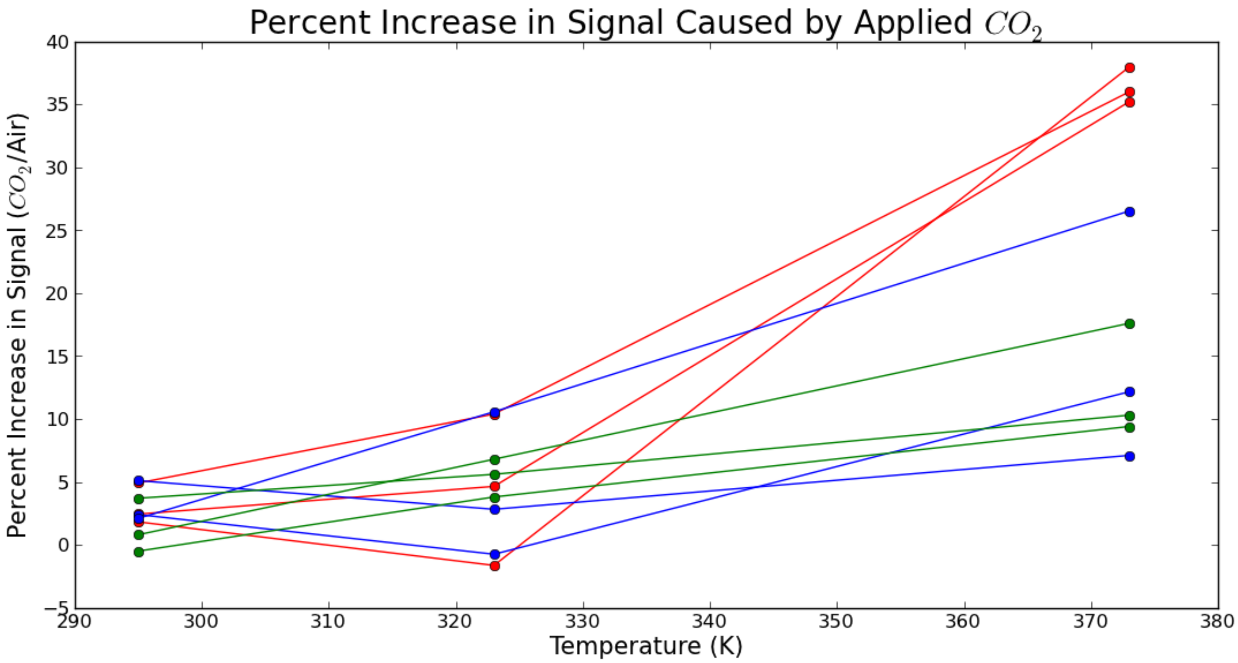


Figure 3.5: Each set of colored lines represent a set of trials taken in succession. A full list of trials can be seen in Table. 4.1

analysis method since we assume a constant modulation for a single set of trials. We are unable to divide out by a common modulation function preventing an accurate comparison of signal. This new modulation is the reason for the large range of values found Fig. 3.5. The unpredictable shifts in signal, caused by this new modulation, artificially increases the percent signal increase at this higher temperature. Also introduced, is a much larger shift in Brewster's angle than anticipated. According to the Sellmeier formula only a single pixel should have occurred at this temperature. The average pixel shift at this temperature was 32 pixels. The reason behind this new modulation and shift is unknown but several origins are possible.

The first possibility is the heating of a nearby polarizer causing the material to expand. By expanding the material that causes the polarization a longer optical path length is introduced and has the possibility of introducing this second modulation. Second heating the optical mount that holds the SiO_2 surface. Our camera is able to measure differences in $7.9 \cdot 10^{-4\circ}$ so any flexing of the mount will cause an artificial shift in Brewster's angle.

Chapter 4

Conclusion

This experiment set out to measure reflections that are caused by only a few layers of molecules on the surface. With our accuracy in measurements any slight change to the environment shows up in our detector. We found that as the temperature is increased more issues began to develop. The open environment contained to many unpredictable elements and caused our measured reflection to be different based on the day that they occurred. Attempting to raise the temperature of the applied gas in order to expunge these ambient particles proved ineffective at changing the signal and at higher temperatures only made it more difficult to remove the fixed modulation introduced into the reflection. It is found that an increase in temperature prevents SPR from concluding anything about a surface in this environment.

This experiment can be improved by increasing the quality of a majority of the equipment used. The end product of our beams profile is not a clean Gaussian function that is required for this sensitive of an experiment. Without this clarity, the analysis is based on specific polynomial fits to enable us to locate the minimums of the data. Only by comparing these minima are we able to come to any evidence for the predictions made through the theory.

The first piece to upgrade is the spatial filter. Proper spatial filtering requires an aperture and objective that fit together to properly isolate the single oscillating mode. We began with a microscope objective and attempted to match it with an aperture. This created a partially filtered beam that was unable to give us the Gaussian profile required. Following the spatial filter the lenses available to us were not ideal for our purposes. The incorrect focal length for our collimating lens forced us to trim a part of our beam as it leaves the spatial filter. More importantly after we focused the beam size with a significant diameter it caused a stigmatism in the focal length making it impossible to place the surface at the focal point.

Without placing the surface at the focal point it was difficult to create the appropriate spread of wave-vectors desired.

It is unknown if the quality of the laser had any impact on any measurements. It may turn out that a different diode or a HeNe would produce a better result but without the highest quality in optical equipment it is unknown if this would improve any signal.

While this initial test proved inconclusive to whether an increase in temperature effects the reflections at Brewster's angle, several more investigations are possible in this domain. The first is to test different gases to apply to the surface. Several possibilities would be O_2 or H_2 which have a smaller polarizability compared to air. A second test would be to coat the surface in a substance that will strongly adsorb to it, such as acetic acid. Then, compare the reflections after room temperature gas is applied compared to hot gas. It may turn out that heating the surface is only required if it contains such a strongly adsorbed molecule to its surface.

Bibliography

[1] *Model STF-8300 M/C CCD Camera Operating Manual.*

[2] David J. Griffiths. *Introduction to Electrodynamics (3rd Edition).* Addison Wesley, 3rd edition, 1 1999.

[3] Eugene Hecht. *Optics (4th Edition).* Addison-Wesley, 4 edition, 8 2001.

[4] William Hetherington. Private discussions, 2013.

[5] Mark Jacobsen. Surface polarization reflection. Undergraduate Thesis, 5 2004.

[6] Chang-Yuan Ju. *Theory and Application of Optical Second Harmonic Generation on Dielectric Surfaces.* PhD thesis, Oregon State University, 1994.

[7] Schott Technical Information. *Temperature Coefficient of the Refractive Index*, 7 2008.

[8] TrueSense Imaging. *KAF-8300 Image Sensor, Device Performance Specification*, 7 2012.

[9] Marvin J. Weber. *Handbook of Optical Materials.* CRC Press, 1 edition, 9 2002.

Appendix A

In order to arrive at the Fresnel equations two pieces of information are required. The first being Snell's Law

$$n_i \sin(\theta_i) = n_t \sin(\theta_t)$$

and the second are the boundary conditions governing electromagnetic interactions at material boundaries [2].

$$\begin{array}{cc} \epsilon_1 E_1^{\perp} = \epsilon_2 E_2^{\perp} & E_1^{||} = E_2^{||} \\ B_1^{\perp} = B_2^{\perp} & \frac{1}{\mu_1} B_1^{||} = \frac{1}{\mu_2} B_2^{||} \end{array}$$

starting with the equation for a wave $\vec{E}(\vec{r},t) = \vec{E_0} e^{i(\vec{k}\cdot\vec{r}-\omega t)}$ where for this case ω is the same for every wave in question. The incident wave interacts at the same point in time and space as the reflected and the transmitted wave, shown in Fig. ??. This causes the k vectors to have equal magnitudes at each interface to keep continuity between the incident, reflected. and transmitted waves. Therefore,

$$\vec{k_I} \cdot \vec{r} = \vec{k_R} \cdot \vec{r} = \vec{k_T} \cdot \vec{r}$$

Plugging this expression into the wave equation and the boundary conditions gives a relation for the incident, reflected, and transmitted portions for the electromagnetic waves. Each side of the equation corresponds for the appropriate side for the boundary and subscripts i,r,and t will be used to denote the incident, reflected, and transmitted waves respectively. For this derivation only the parallel polarized case will be considered as they are what are important to derive Brewster's angle. For a more detailed derivation refer to *Introduction to Electrodynamics* by David Griffiths.

Using the boundary conditions the following relations are constructed.

$$\vec{E_{0_i}} e^{i(\vec{k_i}\cdot\vec{r}-\omega t)} + \vec{E_{0_r}} e^{i(\vec{k_r}\cdot\vec{r}-\omega t)} = \vec{E_{0_t}} e^{i(\vec{k}\cdot\vec{r_t}-\omega t)}$$

$$\frac{1}{\mu_i}(\vec{B_{0_i}}e^{i(\vec{k_i}\cdot\vec{r}-\omega t)}+\vec{B_{0_R}}e^{i(\vec{k_r}\cdot\vec{r}-\omega t)})=\frac{1}{\mu_2}\vec{B_{0_t}}e^{i(\vec{k_t}\cdot\vec{r}-\omega t)}$$

Now, dividing out by the exponential term and knowing that $\vec{B_0}=\frac{E_0}{v}(\hat{k}\times\hat{E})$ we can reduce both of these equations down to relations that only on the electric field. Also in this step we reduce the vector components to their appropriate angle representations, this reduction is made possible by the electric field being parallel polarized with respect to the interface.

$$E_{0_i}\cos(\theta_i)-E_{0_r}\cos(\theta_r)=E_{0_t}\cos(\theta_t)$$

$$\frac{1}{\mu_i v_i}E_{0_i}+\frac{1}{\mu_r v_r}E_{0_r}=\frac{1}{\mu_t v_t}E_{0_t}$$

since the medium for the reflected and incident waves are the same $\mu_i=\mu_r$ and the law of reflection gives us $\theta_i=\theta_r$.

Lastly these equations can be combined to create ratios based on the intensity of the incident electric field $r_{\parallel}=\frac{E_r}{E_i}$ and $t_parallel=\frac{E_t}{E_i}$ and these ratios give the Fresnel equations.

$$r_{\parallel}=\frac{n_t\cos(\theta_i)-n_i\cos(\theta_t)}{n_i\cos(\theta_t)-n_t\cos(\theta_i)}$$

$$t_{\parallel}=\frac{2n_i\cos(\theta_i)}{n_i\cos(\theta_t)-n_t\cos(\theta_i)}$$

Appendix B

Table 4.1: Percent Increase in Signal Caused by Applied CO_2 at Several Temperatures

Trial Set	Trial #	295K	323K	373K
1				
	1	5	3	7
	2	15	5	2
	3	5	3	7
2				
	1	12	7	8
	2	2	11	27
	3	1	7	18
3				
	1	-1	4	9
	2	15	5	2
4				
	1	2	5	35
	2	2	-2	38
	3	5	1	36
5				
	1	34	16	-5
	2	2	5	-4
	3	2	-1	1.4

Table 4.2: Percent Increase in Signal Caused by Applied CO_2 at Room Temperature

Trial #	Percent Signal Increase
1)	4.44
2)	4.83
3)	4.85
4)	5.11
5)	3.02
6)	3.25
7)	6.02
8)	5.28
9)	5.07
10)	0.79
11)	-0.09
12)	6.60
13)	2.05
14)	4.54

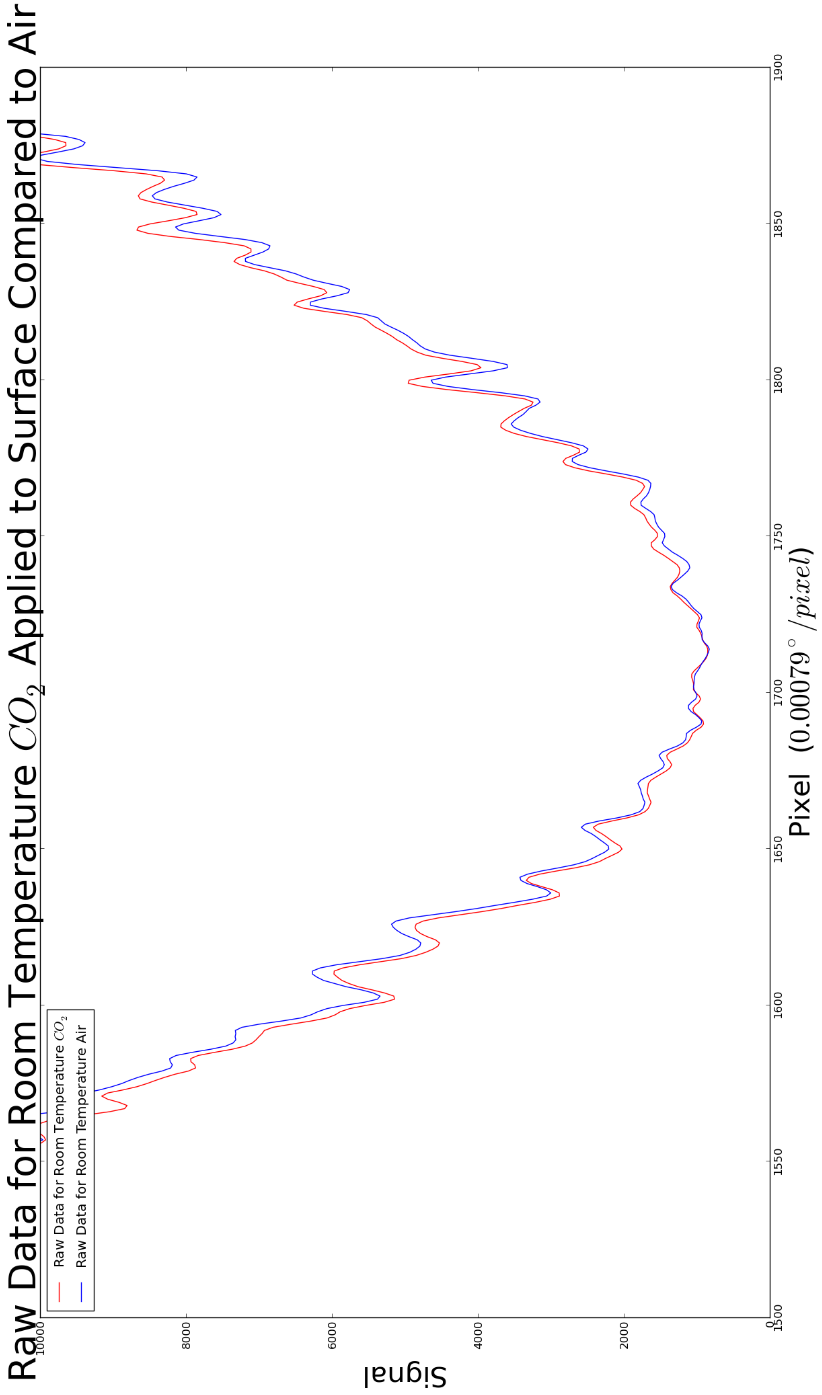


Figure 4.1: Raw data for room temperature results comparing air to CO_2.

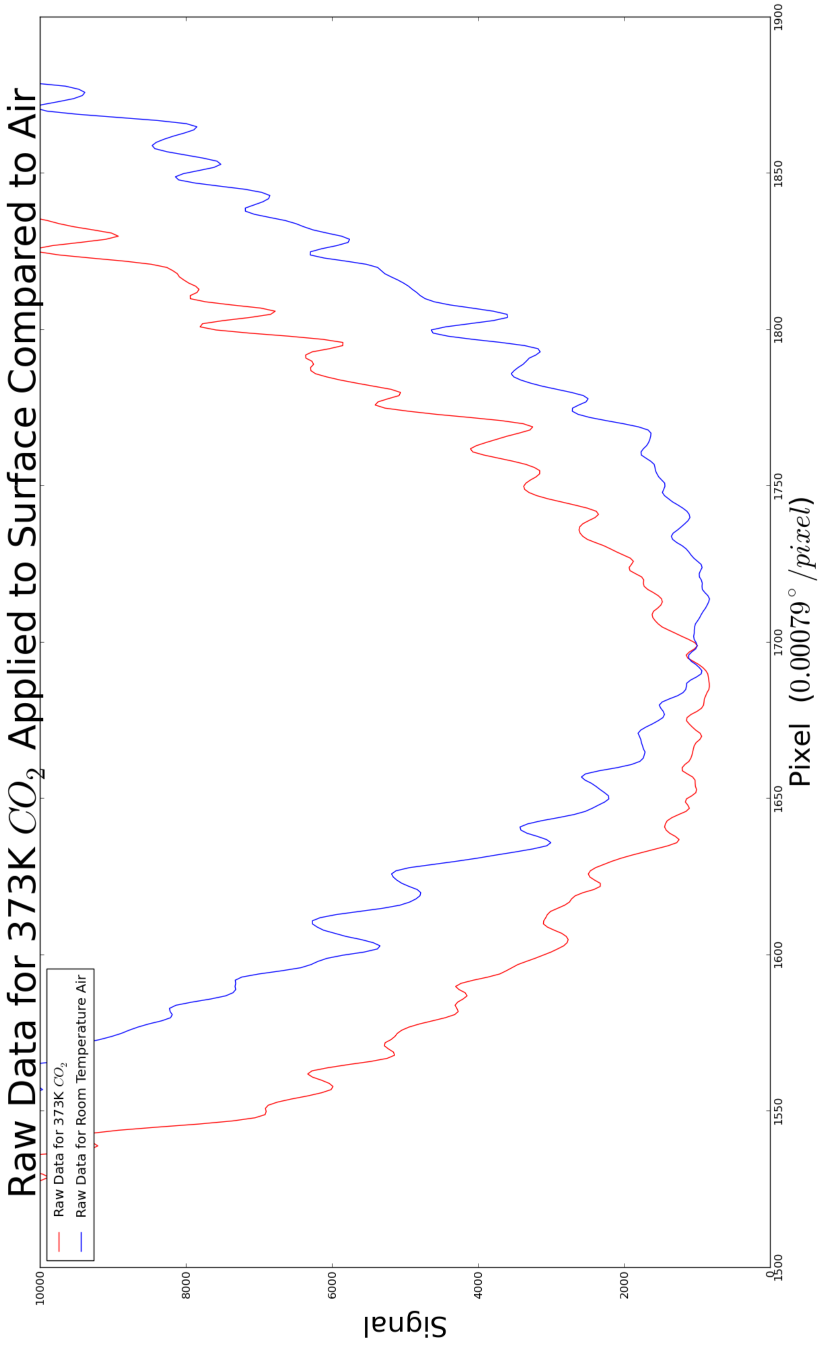


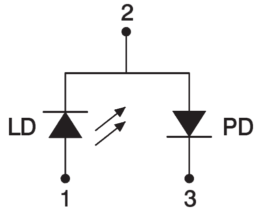
Figure 4.2: Raw Data for room temperature air measurements compared to CO_2 heated to $100°C$

Appendix C

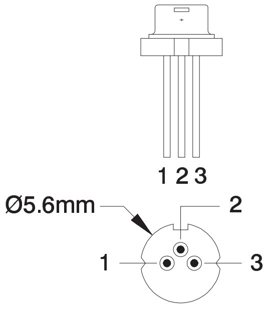
DataSheets

Panasonic LNCQ05PS Laser Diode Specifications

The Panasonic LNCQ05PS is an AlGaInP laser diode with a highly visible red beam. Specifically developed for optical information systems, the 660nm output conforms to standards required by DVD-RAM and DVD-R applications. Using a unique crystal growth technique, the LNCQ05PS combines high output (50mW in the CW mode and 70mW in pulse mode) with a low operating current and good reliability at an extended operating temperature of +60°C. This high power visible diode is very cost effective and therefore should be applicable to other industrial products such as laser communication systems and laser alignment systems. The diode has a Ø5.6mm package.



Internal Circuit



Package Type: Ø5.6mm

Absolute Maximum Ratings (Tc=25 °C)

Characteristic	Symbol	Value	Unit
Optical output power	Po	50	mW
Pulse optical output power	Po (pulse)	70 *	mW
Laser diode reverse voltage	VR(LD)	1.5	V
Photodiode reverse voltage	VR(PD)	30	V
Operating temperature	Topr	-10 to +60	°C
Storage temperature	Tstg	-40 to +80	°C

Notes: *Pulse condition - Less than 50% duty cycle, less than 1µs pulse width, and no bias.

Operating and Electrical Characteristics (Tc=25 °C)

Characteristic	Symbol	Min.	Typ.	Max.	Unit	Test Condition
Threshold current	Ith	20	35	50	mA	CW
Operating current	Iop	60	90	115	mA	CW, Po=50mW
Laser diode operating voltage	Vop	2.0	2.5	3.0	V	CW, Po=50mW
Lasing wavelength	λp	645	660	675	nm	CW, Po=50mW
Beam divergence (parallel)	θ //	7.0	8.5	10.5	deg	CW, Po=50mW, (FWHM)
Beam divergence (perpendicular)	θ ⊥	17.5	22.0	26.5	deg	CW, Po=50mW, (FWHM)
Differential efficiency	η	0.5	0.9	1.2	mW/mA	CW, Po=45mW 45mW / I (50mW) - I (5mW)
Monitor current	Im	–	0.1	–	mA	CW, Po=50mW
Photodiode dark current	ID(PD)	–	–	0.1	µA	VR(PD)=15V

Disclaimer: The laser diode information summarized above is based on the respective diode manufacturer's commercial catalog and/or data sheet specifications. The data is presumed to be accurate; however, it is subject to change without notice. Optima makes no representation as to the accuracy of the information and does not assume any responsibility for errors or omissions contained herein. The user must refer to the manufacturers specifications for details concerning the intended application and operation, diode limitations, and safety.

For current pricing and stock availability please contact:

Optima Precision Inc. 775 SW Long Farm Road West Linn, Oregon 97068 U.S.A.
Phone: (503) 638-2525 Fax: (503) 638-4545 email: optima@optima-optics.com
Website: http://www.optima-optics.com

Revised28Jun01

Data Sheet

SCHOTT

FK3
464658.227

n_d = 1.46450	ν_d = 65.77	$n_F - n_C$ = 0.007063
n_e = 1.46619	ν_e = 65.57	$n_{F'} - n_{C'}$ = 0.007110

Refractive Indices

	λ [nm]	
$n_{2325.4}$	2325.4	1.43972
$n_{1970.1}$	1970.1	1.44498
$n_{1529.6}$	1529.6	1.45039
$n_{1060.0}$	1060.0	1.45557
n_t	1014.0	1.45612
n_s	852.1	1.45834
n_r	706.5	1.46106
n_C	656.3	1.46232
$n_{C'}$	643.8	1.46267
$n_{632.8}$	632.8	1.46300
n_D	589.3	1.46444
n_d	587.6	1.46450
n_e	546.1	1.46619
n_F	486.1	1.46939
$n_{F'}$	480.0	1.46978
n_g	435.8	1.47315
n_h	404.7	1.47625
n_i	365.0	1.48149
$n_{334.1}$	334.1	1.48708
$n_{312.6}$	312.6	1.49217
$n_{296.7}$	296.7	
$n_{280.4}$	280.4	
$n_{248.3}$	248.3	

Constants of Dispersion Formula

B_1	0.973346627
B_2	0.146642231
B_3	0.679304225
C_1	0.00640795469
C_2	0.020565293
C_3	80.4965389

Constants of Dispersion dn/dT

D_0	$-4.90 \cdot 10^{-6}$
D_1	$1.23 \cdot 10^{-8}$
D_2	$-1.19 \cdot 10^{-10}$
E_0	$3.45 \cdot 10^{-7}$
E_1	$7.72 \cdot 10^{-10}$
λ_{TK} [μm]	0.18

Internal Transmittance τ_i

λ [nm]	τ_i (10mm)	τ_i (25mm)
2500	0.650	0.340
2325	0.810	0.590
1970	0.971	0.930
1530	0.988	0.970
1060	0.998	0.995
700	0.997	0.993
660	0.997	0.993
620	0.997	0.993
580	0.997	0.993
546	0.997	0.993
500	0.997	0.993
460	0.996	0.990
436	0.996	0.989
420	0.995	0.987
405	0.994	0.986
400	0.994	0.985
390	0.994	0.984
380	0.992	0.980
370	0.988	0.971
365	0.985	0.964
350	0.954	0.890
334	0.890	0.740
320	0.700	0.410
310	0.510	0.190
300	0.300	0.050
290		
280		
270		
260		
250		

Color Code

λ_{80}/λ_5	33/30
(*= λ_{70}/λ_5)	

Remarks

inquiry glass

Temperature Coefficients of Refractive Index

	$\Delta n_{rel}/\Delta T [10^{-6}/K]$			$\Delta n_{abs}/\Delta T [10^{-6}/K]$		
[°C]	**1060.0**	**e**	**g**	**1060.0**	**e**	**g**
-40/ -20	-0.7	-0.4	-0.1	-2.6	-2.4	-2.1
+20/ +40	-0.4	0.0	0.3	-1.7	-1.3	-1.0
+60/ +80	-0.6	-0.2	0.3	-1.6	-1.2	-0.8

Relative Partial Dispersion

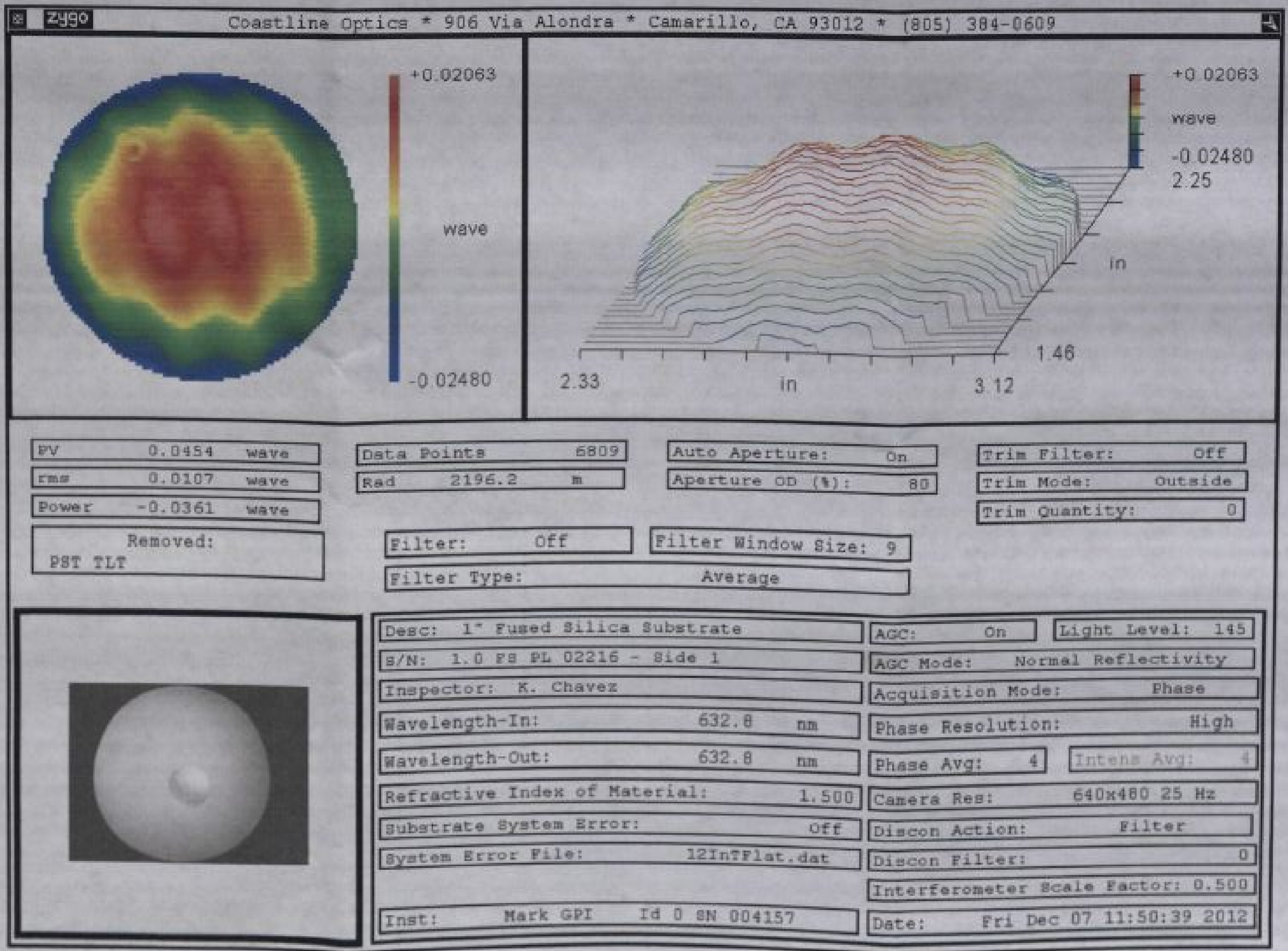
$P_{s,t}$	0.3133
$P_{C,s}$	0.5644
$P_{d,C}$	0.3083
$P_{e,d}$	0.2387
$P_{g,F}$	0.5329
$P_{i,h}$	0.7419
$P'_{s,t}$	0.3112
$P'_{C',s}$	0.6097
$P'_{d,C'}$	0.2571
$P'_{e,d}$	0.2371
$P'_{g,F'}$	0.4736
$P'_{i,h}$	0.7370

Deviation of Relative Partial Dispersions ΔP from the "Normal Line"

$\Delta P_{C,t}$	0.0207
$\Delta P_{C,s}$	0.0082
$\Delta P_{F,e}$	-0.0008
$\Delta P_{g,F}$	-0.0003
$\Delta P_{i,g}$	0.0079

Other Properties

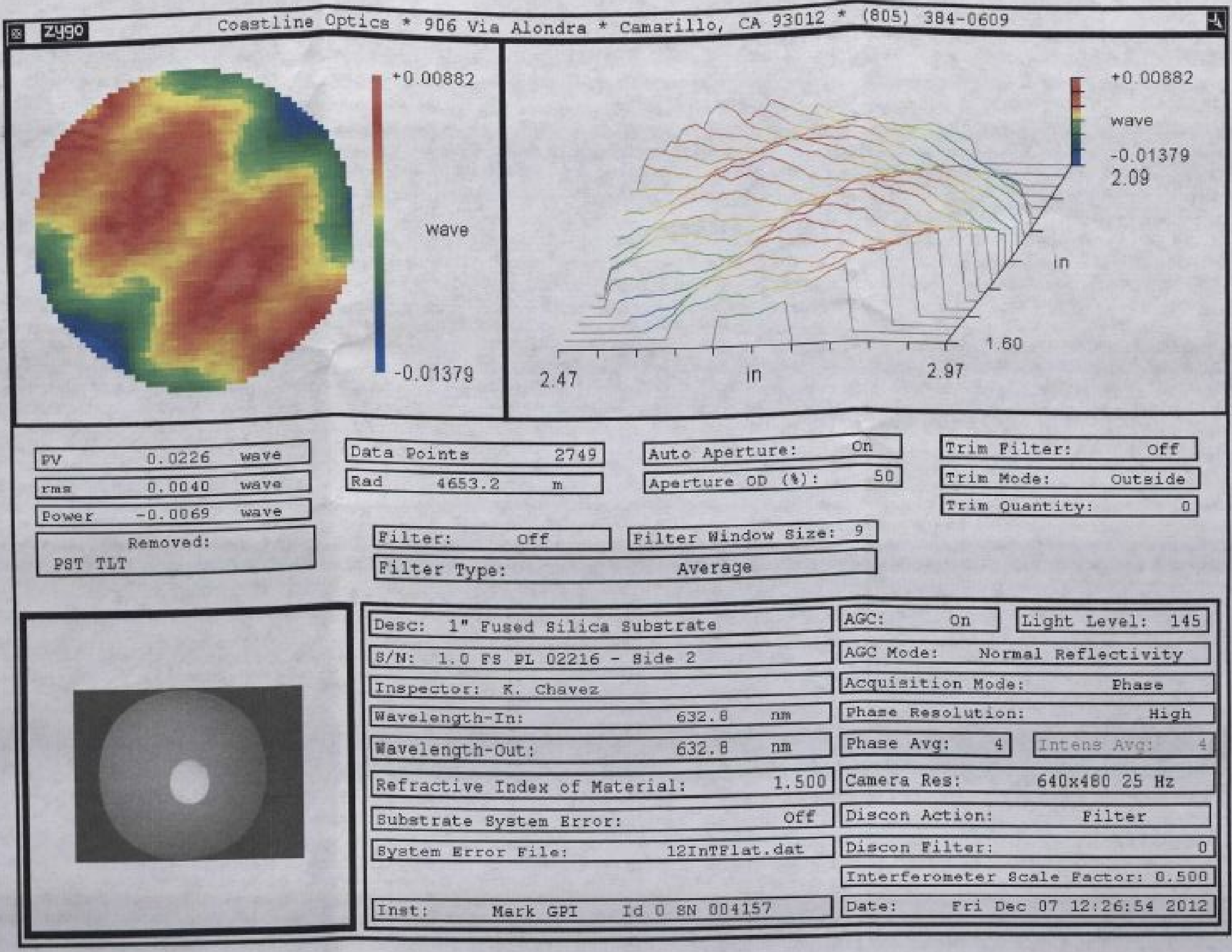
$\alpha_{-30/+70°C}$ [10^{-6}/K]	8.2
$\alpha_{+20/+300°C}$ [10^{-6}/K]	9.4
T_g [°C]	362
$T_{10^{13.0}}$ [°C]	369
$T_{10^{7.6}}$ [°C]	622
c_p [J/(g·K)]	0.840
λ [W/(m·K)]	0.900
ρ [g/cm^3]	2.27
E [10^3 N/mm^2]	46
μ	0.243
K [10^{-6} mm^2/N]	3.71
$HK_{0.1/20}$	380
HG	
B	1
CR	2
FR	3
SR	52.4
AR	2
PR	1



Coastline Optics * 906 Via Alondra * Camarillo, CA 93012 * (805) 384-0609

```
PV        0.0454   wave      Data Points         6809   Auto Aperture:          On   Trim Filter:          Off
rms       0.0107   wave      Rad      2196.2        m   Aperture OD (%):        80   Trim Mode:        Outside
Power    -0.0361   wave                                                            Trim Quantity:          0
       Removed:                Filter:       Off      Filter Window Size:  9
 PST TLT                       Filter Type:                   Average

Desc:  1" Fused Silica Substrate                 AGC:          On    Light Level:  145
S/N:  1.0 FS PL 02216 - Side 1                   AGC Mode:    Normal Reflectivity
Inspector:  K. Chavez                            Acquisition Mode:           Phase
Wavelength-In:                  632.8     nm     Phase Resolution:            High
Wavelength-Out:                 632.8     nm     Phase Avg:     4    Intens Avg:    4
Refractive Index of Material:          1.500     Camera Res:         640x480 25 Hz
Substrate System Error:                  Off     Discon Action:             Filter
System Error File:              12InTFlat.dat    Discon Filter:                  0
                                                 Interferometer Scale Factor: 0.500
Inst:          Mark GPI      Id 0  SN 004157     Date:      Fri Dec 07 11:50:39 2012
```



Coastline Optics * 906 Via Alondra * Camarillo, CA 93012 * (805) 384-0609

PV 0.0226 wave

rms 0.0040 wave

Power -0.0069 wave

Removed:
PST TLT

Data Points 2749

Rad 4653.2 m

Auto Aperture: On

Aperture OD (%): 50

Trim Filter: Off

Trim Mode: Outside

Trim Quantity: 0

Filter: Off

Filter Window Size: 9

Filter Type: Average

Desc: 1" Fused Silica Substrate

S/N: 1.0 FS PL 02216 - Side 2

Inspector: K. Chavez

Wavelength-In: 632.8 nm

Wavelength-Out: 632.8 nm

Refractive Index of Material: 1.500

Substrate System Error: Off

System Error File: 12InTFlat.dat

Inst: Mark GPI Id 0 SN 004157

AGC: On

Light Level: 145

AGC Mode: Normal Reflectivity

Acquisition Mode: Phase

Phase Resolution: High

Phase Avg: 4

Intens Avg: 4

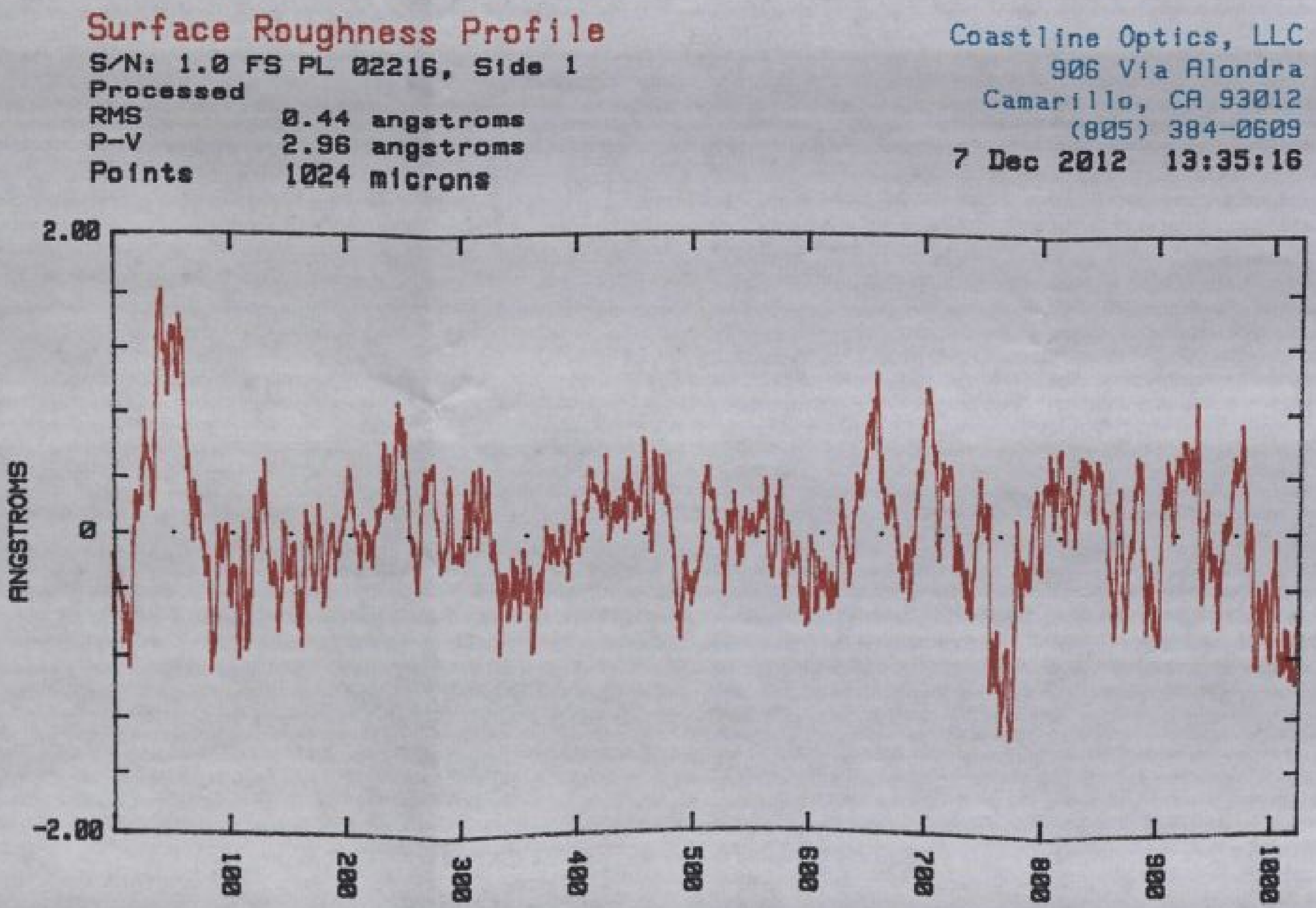
Camera Res: 640x480 25 Hz

Discon Action: Filter

Discon Filter: 0

Interferometer Scale Factor: 0.500

Date: Fri Dec 07 12:26:54 2012



Surface Roughness Profile
S/N: 1.0 FS PL 02216, Side 1
Processed

```
RMS       0.44 angstroms
P-V       2.96 angstroms
Points    1024 microns
```

Coastline Optics, LLC
906 Via Alondra
Camarillo, CA 93012
(805) 384-0609
7 Dec 2012 13:35:16

Appendix D

The following code was used in order to extract and analyze the data. The first program entitled Analysis_1_Extraction.py, and the second entitled Analysis _2_Modulation_2.py.

Analysis_1_Extraction.py

```
”””
A program to plot the same row from multiple SBIG files.
Goal: To compare peaks and valleys to determine if they are
   reproducible or the product of noise in the signal.
written by Thomas Windom (based on Dr. Hetherington's programs for
    analysis of ST6 images).
Resources used:
http://matplotlib.org/users/pyplot_tutorial.html
http://docs.python.org/2/
Original 4−11−13
”””
###


import os #allows us to read files from other directories
from scipy import *
import numpy as np
import matplotlib as mp
import matplotlib.pyplot as plt
import math

def FilesToArray(files): #open the files, convert to a list of
   arrays of data. Each SBIG file has a header of 2048 ascii
   charactars followed by 2532x3352 pixles of data in 16−bit
   integer form.
```

```
        numrows=2532 #number of rows for an SBIG image
        numcols=3352 #number of columns for an SBIG image
        pixels=numrows*numcols #how many pieces of data (pixels,
           16bit integers) we are looking for.
        a=0 #sets initial step for the ”while” loop below.
        global alldata
        alldata=[] #define an empty array which we will store each
            file’s data in.
        # now we begin reading each file and storing the data.
        global background
        background=[]
        while a<len(files): #for each file in the list, we want to
            separate the header from the data and put the data
           into an array.
                afile=open(files[a],’r’) #opens the a(th) file in
                   the list.
                allfile=afile.read(2048).split(’end’)[0] #
                   separates the parameters in the header from the
                    data.
                print(”FileName: ”+files[a])
                print(”Parameters: ”+allfile) #Print the filename
                   and header as each file is read.  Data shows up
                    as a little array symbol.#    print(”
                   Parameters: ”+allfile) #Print the filename and
                   header as each file is read.  Data shows up as
                   a little array symbol.
                #background.append(int(str(allfile[189])+str(
                   allfile[190])+str(allfile[191])+str(allfile
                   [192])))
                #print background
                #if a==1: #print file info for only the first file
                    of each set.
                #       print(”Parameters: ”+allfile) #Print the
                   filename and header as each file is read.  Data
                    shows up as a little array symbol.
                data = np.fromfile(afile, dtype=uint16, count=
                   pixels) #Dr. H’s magic line that does all the
                   work.  uint16 is the key to life’s secrets.
                #print data.shape #used to determine the dimension
                    of the array ”data”.
                #print(data) #Shows the data from this file in a 1
                   D array.
#Now we need to reshape ”data” into a numrows x numcols array.
                data.resize([numrows, numcols])
                #print data.shape #confirms ”data” is now the
```

```
                   correct 2D matrix.
                afile.close #close the file before we mess it up!
#Now it is time to tack this data onto our existing array of data
   from other files.
                alldata.append(data) #appends, or adds, ’data’ as
                   the next element of ’alldata’.
                a=a+1
#Finally, return the entire list of data from all the files. Each
    element is a numrows x numcols matrix for its respective file
   in the list ”files” showing the intensity of the pixels.
        print(”Data retrieval complete. (end of FilesToArray).”)
        return(alldata)

def SaveData(dat_file, array_list) :
    #
    fout = open(dat_file, ’w’)
    for i in range(len(array_list[0])) :
        the_line = ’’
        for j in range(len(array_list)) :
            if j > 0 :
                the_line += ’, ’
            the_line += str(array_list[j][i])
        the_line += ’\n’
        fout.write(the_line)
    fout.close()

def ScaledPlot(files_nogas,files_gas,row,rows_sum,col_center,
   filename,shift,temp):
        lfs=20 #label font size for plots
        tfs=30 #title font size for plots
        numrows=2532
        numcols=3352
        FilesToArray(files_nogas)
        alldata_nogas=alldata
        FilesToArray(files_gas)
        alldata_gas=alldata
#Using the NOGAS files, add them all together, add the rows
   specified together, and average over #files*#rows
        sum_data=np.empty([numrows,numcols])
        a=0
        while a<len(files_nogas):
                onedata=alldata_nogas[a]
                for y in np.arange(numcols):
                        z=0
                        while z<rows_sum:
                                sum_data[row+z,y]=sum_data[row+z,y
                                  ]+(onedata[row+z,y]) #Sum each
                                  element of each file in the row
                                   we are looking at.
                                z=z+1
                    a=a+1
                    print 'air file '+str(a)+' calculated'
        FinalAvg_nogas=[]
        #FinalAvg_nogas=[(sum(x)/(len(files_nogas)*rows_sum))
          *5.006E-05 for x in zip(*sum_data)]#for 1s exposure,
          this is mW/m^2
        FinalAvg_nogas=[(sum(x)/(len(files_nogas)*rows_sum)) for x
           in zip(*sum_data)]
        FinalAvg_nogas=[0 if math.isnan(x) else x for x in
          FinalAvg_nogas] #Replaces "nan" values (0/0 types of
          errors) with a zero.  From http://stackoverflow.com/
          questions/1803516/replace-the-nan-value-zero-after-an-
          operation-with-arrays

#Using the GAS files, add them all together, add the rows
  specified together, and average over #files*#rows
        sum_data=np.empty([numrows,numcols])
        a=0
        while a<len(files_gas):
                    onedata=alldata_gas[a]
                    for y in np.arange(numcols):
                            z=0
                            while z<rows_sum:
                                sum_data[row+z,y]=sum_data[row+z,y
                                  ]+(onedata[row+z,y])  #Sum each
                                   element of each file in the
                                  row we are looking at.
                                z=z+1
                    a=a+1
                    print 'gas file '+str(a)+' calculated'

        FinalAvg_gas=[]
        #FinalAvg_gas=[(sum(x)/(len(files_nogas)*rows_sum))*5.006E
          -05 for x in zip(*sum_data)]
        FinalAvg_gas=[(sum(x)/(len(files_nogas)*rows_sum))for x in
           zip(*sum_data)]
        FinalAvg_gas=[0 if math.isnan(x) else x for x in
          FinalAvg_gas]

#Now there are two sets of data, one for no gas and one for gas.
```

```
Adjust the x values to be angles, then plot everything.
    x = np.arange(numcols)
    #x = x−col_center
    #x = x*0.0007933 #converts to degrees
    xshift = np.arange(shift,numcols+shift)
    #xshift = xshift − col_center
    #xshift = xshift*0.0007933
    plt.plot(x,FinalAvg_nogas,lw=1,color='g')
    plt.plot(xshift,FinalAvg_gas,lw=1,color='b')
    plt.xlabel(r'$\alpha$ ($^\circ$)', fontsize=(lfs))
    plt.ylabel(r'Intensity (mW/m$^2$)', fontsize=(lfs))
    plt.legend([”295K”,”%i K” % (temp+273)], prop={'size':lfs
       })
    plt.title(”Intensity of signal around Brewster's angle. \n
        Hydrogen Gas” , fontsize=tfs)
    plt.tick_params(axis='both', which='major', labelsize=16)
    plt.grid(True)
    plt.figtext(0.01,0.01,”PLOT NOTES: \n Rows ”+str(row)+”−”+
       str(row+rows_sum)+” summed over ”+str(len(files_nogas))
       +” files. \n Start filename: ”+str(files_nogas[0]),
       fontsize=12)
    plt.show()
    print ”Plot closed”
    #np.savetxt(filename+'nogas.csv', FinalAvg_nogas,
       delimiter=”,”)
    #print ”csv file saved.”
    #Exports Data into two CSV Files
    print 'Saving Data...'
    options=[]
    y=0
    candidates=0
    while y<numcols:
            if 39500<FinalAvg_nogas[y]<40500:
                    options.append(y)
                    candidates=candidates+1
            y=y+1
    print (str(candidates)+' candidates found for LHS/RHS')
    if candidates<2:
            print 'ERROR: not enough candidates found for LHS
               and RHS.  Edit code to widen or shift search
               range'
    print 'The following elements are candidates for LHS and
       RHS limits.'
    print options
    LHS=input('Choose element for Left Hand Side')
```

```
        print('LHS='+str(LHS))
        RHS=input('Choose element for Right Hand Side')
        print('RHS='+str(RHS))
        a=LHS
        Save_Datanogas=[]
        Save_Datagas = []
        xnogas = []
        xgas = []
        while a<RHS:
                Save_Datanogas.append(FinalAvg_nogas[a])
                Save_Datagas.append(FinalAvg_gas[a])
                xnogas.append(a)
                xgas.append(a+shift)
                a=a+1
        folder = files_gas[0]
        folder = folder[:-16] #knock the exposurexxx.SBIG off and
           save the output files in the corresponding gas folder.
        SaveData(str(folder)+'Datanogas.csv' , [xnogas,
           Save_Datanogas])
        SaveData(str(folder)+'Datagas.csv', [xgas,Save_Datagas])

        SaveData('./Data_For_Analysis_2/Datanogas.csv' , [xnogas,
           Save_Datanogas])
        SaveData('./Data_For_Analysis_2/Datagas.csv', [xgas,
           Save_Datagas])
        print('')
        print ('Data saved in '+str(folder)+' and ./
           Data_For_Analysis_2')
        print('')
        print ('Continue by running Analysis_2')
        print 'Have a fantastic day!'

#------------------------------------------------------------------------------


#FORMAT OF ENTRY:
”””


 ScaledPlot([ LIST OF ”BASELINE” FILE NAMES, NAME IN SINGLE QUOTES,
     SEPARATED BY COMMAS],
 [LIST OF ”BASELINE” FILE NAMES, NAME IN SINGLE QUOTES, SEPARATED
    BY COMMAS],
ROW TO START WITH (ie 400), ROWS TO SUM OVER (ie 20), COLUMN TO
    CENTER ON (ie 1664 or something), ANYTHING IN SINGLE QUOTES (
    left over from old version, I just put 'test'))
```

```
”””
”””
ScaledPlot([
'../5_6_2/Post_CO2_Air_5/exposure001.SBIG',


],[
'../5_6_2/Post_CO2_Air_6/exposure001.SBIG',

],400,100,1800,'test',0,0)
”””

###########################################3

ScaledPlot([
'../5_14/Air_8/exposure001.SBIG',
'../5_14/Air_8/exposure002.SBIG',
'../5_14/Air_8/exposure003.SBIG',
'../5_14/Air_8/exposure004.SBIG',
'../5_14/Air_8/exposure005.SBIG',
'../5_14/Air_8/exposure006.SBIG',
'../5_14/Air_8/exposure007.SBIG',
'../5_14/Air_8/exposure008.SBIG',
'../5_14/Air_8/exposure009.SBIG',
'../5_14/Air_8/exposure010.SBIG',

],[
'../5_14/CO2_8_100/exposure001.SBIG',
'../5_14/CO2_8_100/exposure002.SBIG',
'../5_14/CO2_8_100/exposure003.SBIG',
'../5_14/CO2_8_100/exposure004.SBIG',
'../5_14/CO2_8_100/exposure005.SBIG',
'../5_14/CO2_8_100/exposure006.SBIG',
'../5_14/CO2_8_100/exposure007.SBIG',
'../5_14/CO2_8_100/exposure008.SBIG',
'../5_14/CO2_8_100/exposure009.SBIG',
'../5_14/CO2_8_100/exposure010.SBIG',
],600,300,0,'test',0,0)
```

Analysis _2_Modulation_2.py

```
import numpy as np
import numpy.linalg as la
import numpy.random as ran
import pylab as plt
import sys
import csv
import os
from response_plot_files import *

# Least-squares fit of data to a polynomial.

def MakeData(coeffs, xrange, xstep, noise_amp, noise_sigma) :
    # coeffs is an array of coefficients for non-negative powers
       of x.  Example: [ 1.0, 0.1] for x + 0.1*x**2.
    x = np.arange(0, xrange, xstep)
    y = np.zeros(len(x))
    n = 0
    for c in coeffs :
        y += c*np.power(x, n)
        n += 1
    noise = ran.normal(0, noise_sigma, len(x)) * noise_amp
    y += noise
    title = '$y = '
    n = 0
    for c in coeffs :
        title += '+ ' + str(c) + ' x^{' + str(n) + '} '
        n += 1
    title += '$'
    return x, y, title

def GetData(data_file, targets=[0, 1]) :
    # Targets is the list of only two data arrays to be used as x
       and y.
    file_in = DataFromFile(data_file, ',')       # Delimiter is
       assumed to be a comma.
    return file_in.data[targets[0]], file_in.data[targets[1]]

def MakeDataCoefficients(order, offset) :
    data_coeffs = np.array([offset])
    for n in range(1, order + 1) :
        coeff = np.power(10.0, -n)
        data_coeffs = np.append(data_coeffs, coeff)
    #print('Data coefficients: ')
    #print(data_coeffs)
    #return data_coeffs

def FitDataToPolynomial(x, y, order) :
    coeffs, residuals, rank, singular_values, rcond = np.polyfit(x
       , y, order, full=True)
    crev = coeffs[:: -1]    # Reverse the order of the coeffs
       array
    #sigma = np.sqrt(residuals[0]/len(x))
    sigma = 0
    #print('sigma = ' + str(sigma))
    z =np.zeros( (order+1, len(x)) )
    ztot = np.zeros(len(x), float)
    for n in range(0, order+1, 1) :
        z[n] += crev[n] * np.power(x, n)
        ztot += z[n]
    #print ztot
    return ztot, z, crev, sigma

def PlotFit(x, y, z, ztot, coeffs, sigma, title) :
    plt.plot(x, y, label='Data')
    coeff_string = 'Fit Coefficients\n'
    n = 0
    #for c in coeffs :
        #coeff_string += str(n) + ':    ' + str(coeffs[n]) + '\n'
        #plt.plot(x, z[n], label='$x^{' + str(n) + '}$ piece')
        #n += 1
    coeff_string += 'sigma = ' + str(sigma) + '\n'
    plt.plot(x, ztot, label = 'Total Fit')
    #plt.legend(loc='best')
    plt.legend(loc='upper left')
    plt.title( title )
    graph_min = np.min(ztot)
    graph_max = np.max(ztot)
    for zn in z :
        zn_max = np.max(zn)
        zn_min = np.min(zn)
        if zn_max > graph_max :
            graph_max = zn_max
        if zn_min < graph_min :
            graph_min = zn_min

    x0 = ztot.argmin()
    y0 = ztot.min()
    #plt.plot(x0,y0, 'rd', label = 'Min Value')
    #print x0 , y0
    plt.text(1, graph_min +1, coeff_string)
    global Poly
    Poly = ztot
    return Poly

def MakeResidualHistogram(data, fit, bins, sigma) :
    # Make a histogram of the abs(error)
    #e2 = np.abs(data - fit)
    e2 = data - fit
    hist = np.histogram(e2, bins=bins)
    # Histogram returns the array of values and an array of bin
       edges, which has one more value.
    # Create bin center.
    bincenters = 0.5*(hist[1][1:]+hist[1][:-1])
    histo_max = np.max(hist[0])
    # Simulate the set of error, but with more samples for less
       noise on the noise.
    #  The mean should be 0.0, but it will fluctuate about that
       from run to run.
    normal_dist = ran.normal(0.0, sigma, len(e2)*3000) * histo_max
        * np.sqrt(2.0*np.pi)*sigma
    normal_hist = np.histogram(normal_dist, bins=bins)
    normal_hist_max = np.max(normal_hist[0])
    plt.plot(bincenters, hist[0], label='Data')
    plt.plot(bincenters, normal_hist[0]*histo_max/normal_hist_max,
        label='Simulation')
    the_title = 'Error Histogram with $\sigma = $' + str(sigma)
    plt.title(the_title)
    plt.legend(loc='best')
    plt.show()
    plt.close()

def Test(shift, order) :
        #order = 32   # polynomial order
        offset = 0  # x^0 coefficient in the data
        xmax = 10.0 # x will range from 0 to xmax
        xstep = 0.01    # in increments of xstep
        noise_sigma = 0.02
        noise_amp = 10.0
        numcols=3352
        bins = np.int(xmax/xstep/10.0)  # for the histogram
```

```
data_coeffs = MakeDataCoefficients(order, offset)
x, y, = GetData('Datagas.csv', targets=[0,1])
x2, y2 = GetData('Datanogas.csv', targets= [0,1])



t=0
graph_title='NOOOOOOOOOOOOOO'
while t<10:
        fit2, fit_n2, coeffs2, sigma2 =
           FitDataToPolynomial(x2, y2, order)
        bigshift=x2[argmin(fit2)]
        x=x-bigshift
        x2=x2-bigshift
        t=t+1
if x2[argmin(fit2)] == 0:
        print"shift completed for Trial 1"
else:
        print"ABORT!!! SHIFT NOT SUCCESSFUL!!!!"

poly_air = PlotFit(x2, y2, fit_n2, fit2, coeffs2, sigma2,
   graph_title)
mod_func1 = y2 / poly_air



Data_gasa = y / mod_func1
Data_air = y2 / mod_func1

a=0
Data_gas=[]
while a<len(x):
        Data_gas.append(Data_gasa[a])
        a=a+1

if shift>0:
        a=0
        while a<shift: #shift gas right
                Data_gas.reverse()
                Data_gas.append(0)
                Data_gas.reverse()
                Data_gas.pop()
                a=a+1
if shift<0: #shift gas left
        a=shift
```

```
            while a<0:
                    Data_gas.append(0)
                    Data_gas.reverse()
                    Data_gas.pop()
                    Data_gas.reverse()
                    a=a+1

    #plt.show()
    print”The following error message is due to supressing the
        first plot:”
    print
       ”..........................................................”

    plt.close()

### Prints the three Modulation Functions based on each trial
   currently being tested, if trials were done on the same day
    these should line up.


    plt.plot(x2, mod_func1, label = 'Modulation Function')
    plt.title('Sample Modulation Function',fontsize =40)
    plt.xlabel('Pixel  ($0.00079^\circ / pixel$)', fontsize =
       30)
    plt.ylabel('% Signal Change', fontsize = 30)
    plt.xlim(−200,200)
    plt.show()
    plt.close()
    print
       ”..........................................................”


    plt.plot(x,y,label = 'Raw Data', color='r')
    plt.plot(x,Data_gasa,label = 'Data Divided by Modulation
       Function',color='b')
    plt.title('Compare Raw $CO_2$ Data to Un–Modulated $CO_2$
       Data ', fontsize = 40)
    plt.xlabel('Pixel  ($0.00079^\circ / pixel$)', fontsize =
      (30))
    plt.ylabel('Signal',fontsize = (30))
    plt.legend(loc='upper left')#, fontsize = 20)
    plt.xlim(−200,200)
    plt.ylim(0,10000)
    plt.show()
    plt.close()
```

```
     print
        "......................................................"

  #Shifts the gas data by this many pixels

     plt.plot(x, Data_gas, label = '$CO_2$ Results', color = 'b
        ')
     plt.plot(x2, Data_air, label = 'Air Best Fit Polynomial',
        color = 'g', ls = '--')
     plt.legend(loc = 'upper left')
     plt.title( 'Signal Comparison of Un-Modulated CO2 vs. Best
         Fit Polynomial for Air Data' , fontsize = 40 )
     plt.xlabel( 'Pixel  ($0.00079^\circ / pixel$)',fontsize =
        30)
     plt.ylabel( 'Signal', fontsize = 30)
     plt.fill_between(x,Data_gas,Data_air, where=abs(x)<10,
        facecolor='r',interpolate=True)
     plt.xlim(-50,50)
     plt.ylim(0,4000)

  ##NOTE: Offset above must be correctly set as well, number in
     the same text value for daily folder.
     #print "zero point is "+str(x[0])+"th element of array x"
     zero_point = int(-1*x[0])
     Diff_1 = []
     a = 0
     while a < len(Data_gas):
             Diff_1.append(Data_gas[a] - Data_air[a])
             a = a+1
  #print 'Data_gas - Data_air at same index', Data_gas[393] -
     Data_air[400]
 # print Data_gas[400], Data_air[400]

     #print 'Prints Diff function at the zero point' , Diff_1[
        zero_point]
     #print 'Value of Data_gas at zero point--->' , Data_gas[
        zero_point], 'Data Value at Index + "Shift" --->',
        Data_gas[zero_point-shift]
     #print 'Value of Data_air at zero point--->' , Data_air[
        zero_point], 'Data Value at Index + "Shift" --->',
        Data_air[zero_point-shift]
        #print x
        #print Data_gas[zero_point - shift] - Data_air[zero_point]
        #print zero_point
        #print Data_air.argmin()

        Int1 = []


    ## VALUE HERE LABELLED UNDER "ZERO POINT IN TXT FILE"   a = '
       Zero Point' - 10
        a = zero_point-10

        while a < zero_point+10:
                Int1.append(Diff_1[a])
                a = a+1
    #print x2
        #print 'This is The Riemann Sum values for the three
           trials ',sum(Int1), sum(Int2), sum(Int3)
    #print Int1, Int2, Int3

        a = zero_point-10

        Base1= []
        while a < zero_point+10:
                Base1.append(Data_air[a])
                a = a+1
        #print 'This is the sum of the Air values from -10 -> 10
           pixel range', sum(Base1), sum(Base2), sum(Base3)

        print 'Ratio of Riemann sum over sum of base values "sum(
           Int) / sum(Base)"'
        print '(Percent signal increase):'
        print '——————————————'
        print str(sum(Int1)/sum(Base1))
        print "Polynomial order used: "+str(order)
        print '——————————————'
        plt.show()
        plt.close()
    #MakeResidualHistogram(y, fit, bins, sigma)


#————————————————————————————————————————————
```

```
if __name__ == '__main__' :
    ##The Values that need to placed into program can be found in the daily info txt document with the data csv's
    ##inputted. The format goes as follows
    ##If CO2 set your shift in the first slow and set the second input at 0, if hydrogen is being looked at use Hydrogen shift
    ##    Text(EXPECTED SHIFT (-LEFT, +RIGHT) , POLY ORDER)
    os.chdir('./Data_For_Analysis_2')
    #Test(-7,8)
    #Test(-7,32)
        #Test(7,8)
    #Test(7,32)
```