# NAVAL
# POSTGRADUATE
# SCHOOL

**MONTEREY, CALIFORNIA**

# THESIS

## DESIGN OF A BORE SIGHT CAMERA FOR THE LINEATE IMAGE NEAR ULTRAVIOLET SPECTROMETER (LINUS)

by

Rodrigo Cabezas

June 2004

| | |
|---|---|
| Thesis Advisor: | Richard Harkins |
| Co-Advisor: | D. Scott Davis |

| | |
|---|---|
| **REPORT DOCUMENTATION PAGE** | *Form Approved OMB No. 0704-0188* |

| | | |
|---|---|---|
| **1. AGENCY USE ONLY *(Leave blank)*** | **2. REPORT DATE**<br>June 2004 | **3. REPORT TYPE AND DATES COVERED**<br>Master's Thesis |

| | |
|---|---|
| **4. TITLE AND SUBTITLE**: Design of a Bore Sight Camera for the Lineate Image Near Ultraviolet Spectrometer (LINUS) | **5. FUNDING NUMBERS** |
| **6. AUTHOR(S)** Rodrigo Cabezas | |
| **7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**<br>Naval Postgraduate School<br>Monterey, CA 93943-5000 | **8. PERFORMING ORGANIZATION REPORT NUMBER** |
| **9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)**<br>N/A | **10. SPONSORING/MONITORING AGENCY REPORT NUMBER** |

**11. SUPPLEMENTARY NOTES** The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

| | |
|---|---|
| **12a. DISTRIBUTION / AVAILABILITY STATEMENT**<br>Approved for public release; distribution is unlimited. | **12b. DISTRIBUTION CODE** |

**13. ABSTRACT (maximum 200 words)**

The Lineate Image Near Ultraviolet Spectrometer (LINUS) is a spectral imager that works in the ultraviolet region of the spectrum. This thesis describes the latest of several steps in the development of this instrument.

Due to the narrow field of view of the instrument, 2.5 x 0.5 degrees, an accurate pointing method is necessary; also, a scheme of quality evaluation of the post-processed spectral image is desirable. A way to achieve both goals was developed by designing and implementing the layout for two visual cameras, wide and narrow field of view, and a method to capture the images in order to perform the subsequent comparison with the processed spectral image.

Since this is the first time the system is working in full-automated mode, a new wavelength calibration with the emission lines from a platinum hollow cathode lamp was performed and a new response curve for sulfur dioxide ($SO_2$) was taken. Finally, laboratory and outdoor field observations were conducted to test the system integration.

| | |
|---|---|
| **14. SUBJECT TERMS**<br>Sensors, Spectral Imaging, Spectrometer, Remote Sensing. | **15. NUMBER OF PAGES**<br>79 |
| | **16. PRICE CODE** |

| | | | |
|---|---|---|---|
| **17. SECURITY CLASSIFICATION OF REPORT**<br>Unclassified | **18. SECURITY CLASSIFICATION OF THIS PAGE**<br>Unclassified | **19. SECURITY CLASSIFICATION OF ABSTRACT**<br>Unclassified | **20. LIMITATION OF ABSTRACT**<br>UL |

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

# DESIGN OF A BORE SIGHT CAMERA FOR THE LINEATE IMAGE NEAR ULTRAVIOLET SPECTROMETER (LINUS)

Rodrigo Cabezas
Lieutenant, Chilean Navy
B.S., Chilean Naval Polytechnic Academy, 1996

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN APPLIED PHYSICS**

from the

**NAVAL POSTGRADUATE SCHOOL**
**June 2004**

Author: Rodrigo Cabezas

Approved by: Richard Harkins
Thesis Advisor

D. Scott Davis
Co-Advisor

James Luscombe
Chairman, Department of Physics

THIS PAGE INTENTIONALLY LEFT BLANK

# ABSTRACT

The Lineate Image Near Ultraviolet Spectrometer (LINUS) is a spectral imager that works in the ultraviolet region of the spectrum. This thesis describes the latest of several steps in the development of this instrument.

Due to the narrow field of view of the instrument, 2.5 x 0.5 degrees, an accurate pointing method is necessary; also, a scheme of quality evaluation of the post-processed spectral image is desirable. A way to achieve both goals was developed by designing and implementing the layout for two visual cameras, wide and narrow field of view, and a method to capture the images in order to perform the subsequent comparison with the processed spectral image.

Since this is the first time the system is working in full-automated mode, a new wavelength calibration with the emission lines from a platinum hollow cathode lamp was performed and a new response curve for sulfur dioxide ($SO_2$) was taken. Finally, laboratory and outdoor field observations were conducted to test the system integration.

THIS PAGE INTENTIONALLY LEFT BLANK

# TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

# ACKNOWLEDGMENTS

The author would like to thank Dr. Scott Davis for his patience, guidance and assistance in this thesis development, and for the wonderful introduction to the marvelous world of optics. Also, I would like to thank Prof. Richard Harkins for his many long hours of discussion, support and the strong encouragement to pursue my dreams.

Finally, the author would like to extend his deepest gratitude to his wife, Maria Jose, for her unending love and patience. This thesis could not have been completed without her support.

THIS PAGE INTENTIONALLY LEFT BLANK

# I. INTRODUCTION

## A. PROJECT CONTEXT

The design project described in this thesis is the latest step in the development of LINUS, the Lineate Imaging Near Ultraviolet Spectrometer developed at the Naval Postgraduate School (NPS). The instrument is the third generation imaging spectrometer and incorporates experience accumulated from the two previous devices, NUVIS [ref.11] and DUUVIS.

In year 2002, LINUS was deployed into the field for the first time. The deployment included the assessment of the system integration and its projected operational capabilities.

## B. PROJECT OBJECTIVE

The objectives of the thesis research were to design and implement an aiming camera in the visual spectrum and to establish a procedure to compare visual alignment with processed data.

Because the instrument has such a narrow field of view ($2^o$), the ability to align it with the desired scene was difficult. Consequently, many images were discarded because they were out of the region of interest. The incorporation of a bore sight camera solved this problem.

Both objectives were accomplished and tested in the laboratory and demonstrated in the field.

## C. OUTLINE

This thesis is organized into four chapters and three appendices. The following chapter gives a brief description of the physics of an imaging spectrometer and describes the LINUS architecture. Chapter III illustrates the system modification to accommodate the new visual system design, including optical subsystem change, software, and electronic changes. In addition, the experiment setup for the alignments is described.

Chapter IV includes a new replica of the wavelength calibration and sulfur dioxide ($SO_2$) tuning of the instrument, along with the system integration tests in laboratory and field. Conclusions and recommendations are contained in Chapter V. Useful complementary information, such as software code is contained in the appendices.

# II. LINUS

## A. PURPOSE

The purpose of this chapter is to give a brief description of an imaging spectrometer along with the LINUS configuration.

## B. BACKGROUND

Imaging spectrometers combine traditional imaging, like the picture of a camera, with spectroscopy. The first addresses the spatial coordinates while the second deals with the frequency components of the target. Information obtained by this technique is used to discriminate, classify, identify and quantify materials present in the image. Additional features are: sub-pixel target detection (which allows the detection of targets of interest with sizes smaller than the pixel resolution), and abundance estimation, (which allows the detection of concentrations of different elements by the signature spectra present in pixels). Data analysis difficulties require accurate calibration methods to resolve scene pixel non-linearities due to different materials resident in the scene.

Imaging is concerned with the accurate measurement of light intensity over a two-dimensional space. Spatial variations are used to detect scene features and patterns, such as size, shape, color and are used to characterize objects. However, there are some limitations. For instance, objects can be covered with nets, painted in colors to change the highlighted areas, or have additional pieces added to change their appearance. Therefore, imaging is not a perfect method to obtain information.

Spectroscopy, on the other hand is concerned with the study of variations in light intensity as a function of wavelength or frequency. Different materials exhibit different spectral properties due to their atomic or molecular compositions. These characteristics typically include material-specific wavelengths where electromagnetic energy is absorbed (absorption lines or bands) or emitted (emission lines or bands). Spectrometers capable of detecting such spectral characteristics can be used to determine both the materials being observed and some characteristics of their environment. Hence, spectroscopy is a more robust recognition technique.

Spectral imaging integrates those two procedures, producing more valuable data and information from a target. Incident light is detected and recorded according to both position within the image and the wavelength. The resultant data is a three-dimensional group of independent variables, called a hyper-spectral cube. For each spatial element (pixel) of the image, a spectral imager records the intensity over many bands of different wavelengths; the LINUS hyper-cube is shown in Fig 1.

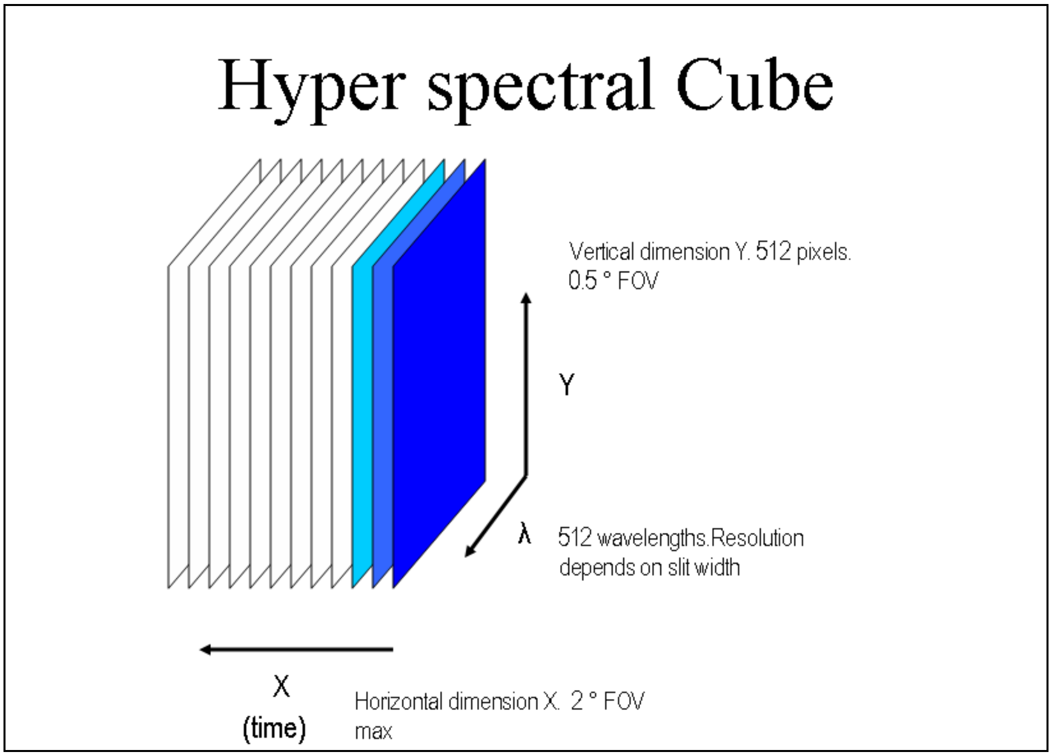


Figure 1. Hyper spectral cube

The benefits of remote sensing in military and civilian applications include:

- Environmental monitoring, like industry stack plumes or volcano activity.
- The ability to defeat camouflage and decoy techniques by examining many regions of the electromagnetic spectrum.
- The possibility that this technology could be used to detect biological or chemical warfare agents.

## C. THEORY

The imaging scheme consists of taking successive slices of the scene by displacing the scanning mirror through the field of view. For LINUS this is performed by rotating the scanning mirror.

The horizontal field of view ( $\theta$ ), imaged at a certain mirror position, depends on the slit width setting and is given by equation (1)

$$\theta = 2\tan^{-1}\left(\frac{w}{2f}\right) \qquad (1)$$

where $\theta$ is in radians, w is the slit width and f=25cm is the focal length of the primary objective lens of the optical system.

The grating disperses the light according to the grating equation (2)

$$m\lambda = d(\sin\theta_i - \sin\theta_o) \qquad (2)$$

where: m = 1 is order of the diffraction set, $\lambda$ is the wavelength of the diffracted light, d is the diffraction grating inter-ruling spacing, $\theta_i$ = incident angle and $\theta_o$ = output angle.

Figure 2 shows that the result is a horizontal dispersion of the incident light corresponding to its wavelength. The horizontal coordinate on the CCD array corresponds to the set of wavelengths of the vertical image strip; each image strip has a one to one (vertical position, Y coordinate) correspondence of the UV light incident on the CCD. Each image coordinate has a match in the CCD array in terms of vertical position and wavelength; the data is stored as a single two-dimensional frame. Then the mirror is moved to get an adjacent vertical slice of the scene. The horizontal scene position (X coordinate) is scanned and stored as the next 2D frame as seen in Fig.1. The hyper-cube is built in time by appending consecutive 2D image frames obtained with small displacements of the scanning mirror until the entire scene has been scanned in the field of view.

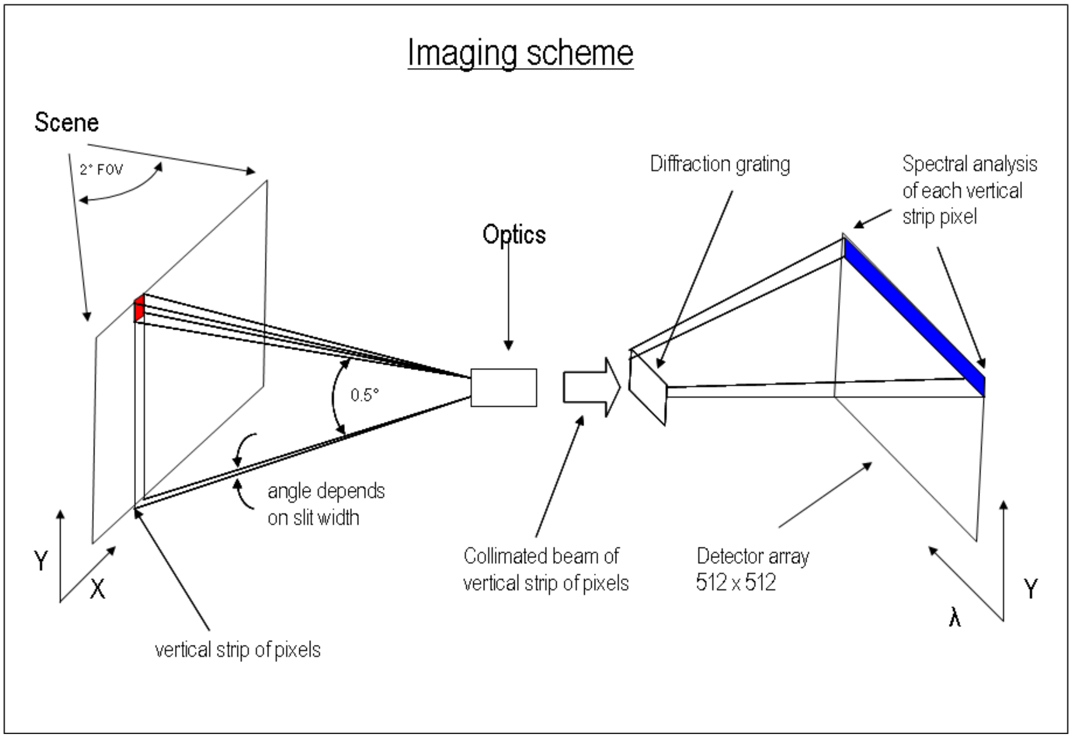


Figure 2. Imaging acquisition

## D. HARDWARE CONFIGURATION

The light radiation from a scene enters the optical aperture and is reflected off the scanning mirror. It passes trough a UV band pass filter and the primary objective lens that focuses the image onto a slit. The slit allows only a thin vertical slice of the scene to continue into the remaining optical path. The vertical slice is focused by a collimator lens on to a diffraction grating, which operates in the first order mode. Finally, the diffracted UV light is focused by the camera objective into the intensified UV camera consisting of a UV-sensitive micro channel plate coupled to a 512 x 512 pixel charge-coupled device (CCD) detector array, as seen in Fig.3.

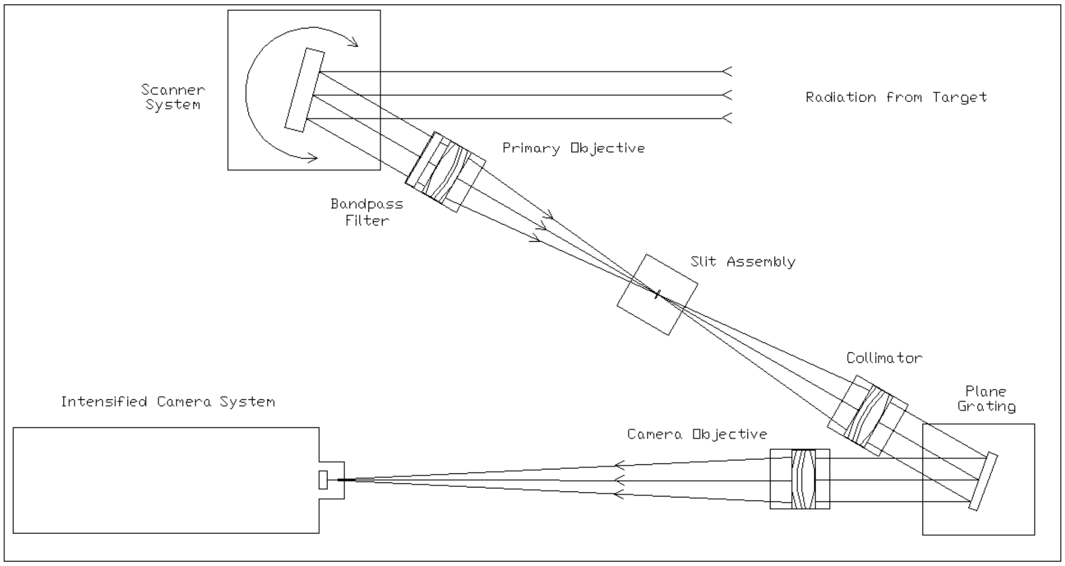


Figure 3. Optical layout [ref. 1]

## E. SOFTWARE

The control software stores the resultant spectral image for later analysis. If the instrument is operated at its full data resolution of 800 horizontal image samples by 512 vertical image samples by 512 wavelength samples by 12 bits (2 bytes) per pixel, the total data storage requirement for one scene will be 800 x 512 x 512 x 2 = 419,430,400 bytes.

The main control software is written in National Instruments Lab View $^{TM}$ and it is integrated in the host computer. In addition, the computer integrates the auxiliary software and the controllers for the devices, as shown in Fig. 4.

A program sample and the operator control panel are shown in Figures 5 and 6 respectively.

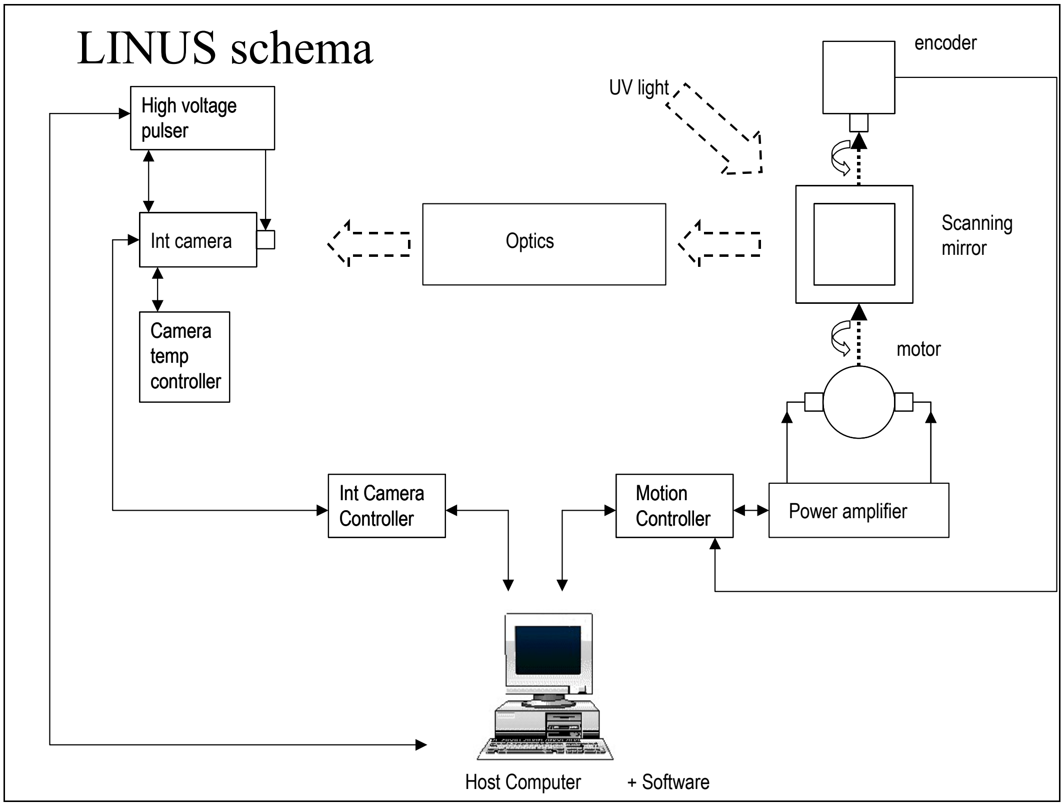


Figure 4. LINUS schema

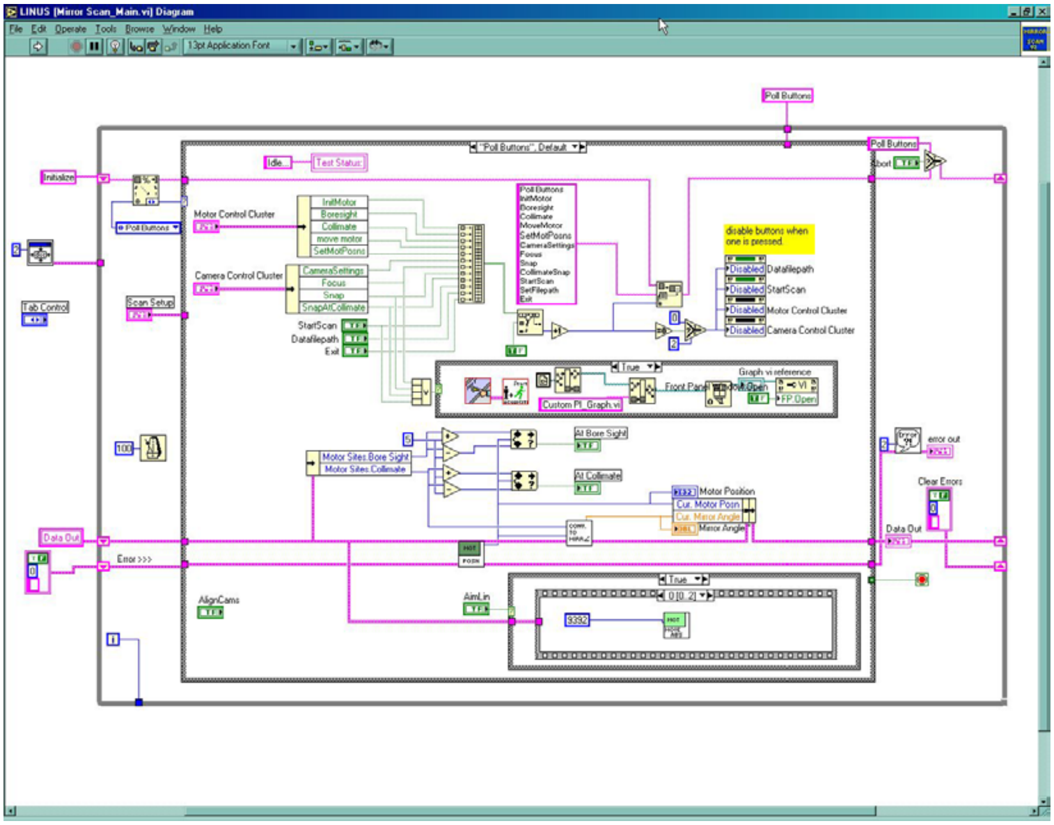
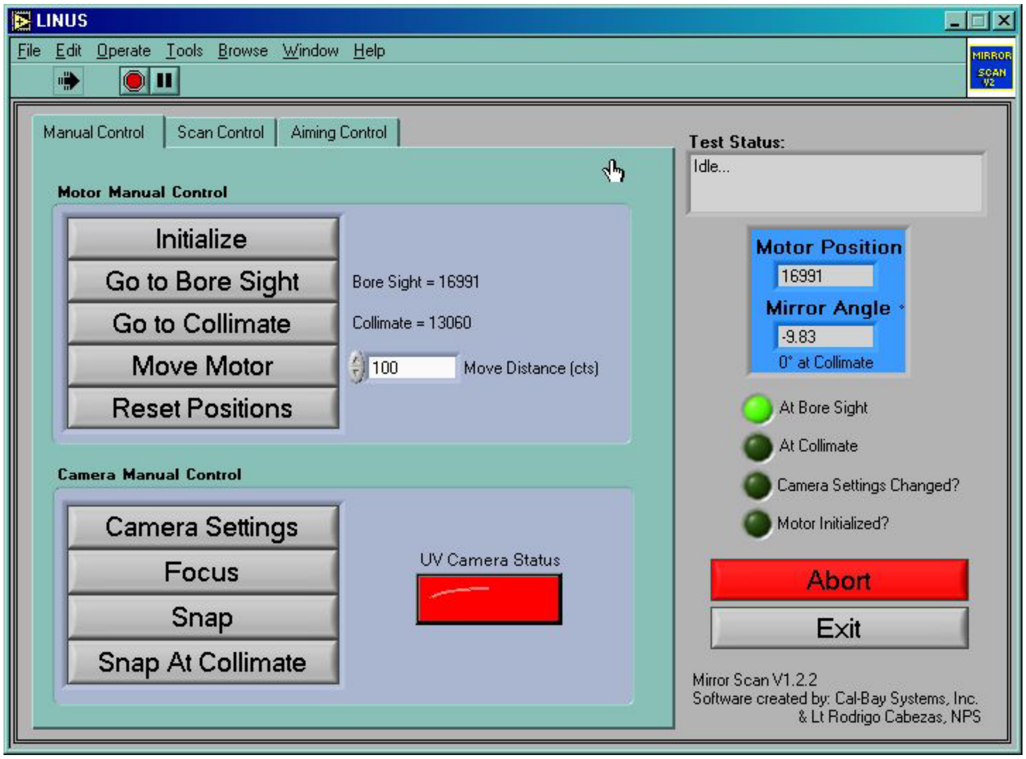


Figure 5. Program sample



Test Status: Idle...

Bore Sight = 16991

Collimate = 13060

Move Distance (cts): 100

Motor Position: 16991

Mirror Angle °: -9.83

0° at Collimate

Mirror Scan V1.2.2
Software created by: Cal-Bay Systems, Inc.
& Lt Rodrigo Cabezas, NPS

Figure 6. Software control panel

THIS PAGE INTENTIONALLY LEFT BLANK

# III. CAMERA DESIGN

## A. PURPOSE

Field measurements with the early versions of LINUS were difficult. Bore sight aiming inaccuracies wasted time and effort. Only after the data was acquired and post-processed did the operator have the answer to where the imager was actually aimed. Because of this, it was decided to integrate a visual camera into LINUS for bore sight aiming purposes, see Figure 15.

During the initial stage of the work, when the visual camera installation was decided, the hardware setup was evaluated and the results were as follows: no growth capacity, slow response, and weight and volume excess for field deployment.

Additionally, a hardware upgrade was required to support integration of the new camera. This process grew into three operating system migrations, five hardware changes and eight main program revisions. Although this effort represented approximately sixty percent of the work, only the final hardware and software versions are addressed in this thesis.

## B. HARDWARE MODIFICATION

The following changes were made to reduce the total size and weight of the LINUS support hardware:

The computer was upgraded from an LCS WINNT i386 with a 40 GB hard drive to a WIN2K Shuttle AMD Athlon with a 100GB hard drive, see Table 1. The Shuttle took up less space, had more memory capacity, was equipped with a fire-wire connection for the new motion controller interface card, and had the capability to host the PCI 1411 frame grabber card for the visual camera.

|  |  |
|---|---|
| Model | Shuttle Technology ® PC |
| Processor | Athlon 2.4 GHz |
| RAM memory | 1 GByte |
| HDD | 110 Gbytes |
| Dimensions | 300 mm x 200 mm x 185 |
| Weight | 2.85 Kg |

Table 1. Host computer characteristics

The motion controller was replaced with the NI FW744 motion controller, See Table 2.

|  |  |
|---|---|
| PID update rate | 62.5 to 500 microseconds /sample |
| Position range | ±$2^{31}$ counts |
| Encoder input | Quadrature, incremental, single ended |
| Weight | 1.7 Kg. |
| Dimensions | 30.7 x 25.4 x 4.3 cm. |

Table 2. Motion controller characteristics [ref. 2]

The motor power amplifier was replaced with the NI MID7654 Servo Power Motor Drive. See Table 3.

|  |  |
|---|---|
| Continuous power output range | 400 w (25% duty cycle) |
| Encoder input | Quadrature, incremental |
| Number of axis | 2 |
| Dimensions | 30.6 x 25.4 x 8.8 cm |
| Weight | 10.2 Kg |

Table 3. Motion power amplifier [ref. 3]

Figure 7 shows the new hardware installed for field deployment and Figure 8 is a schematic of the new LINUS hardware layout

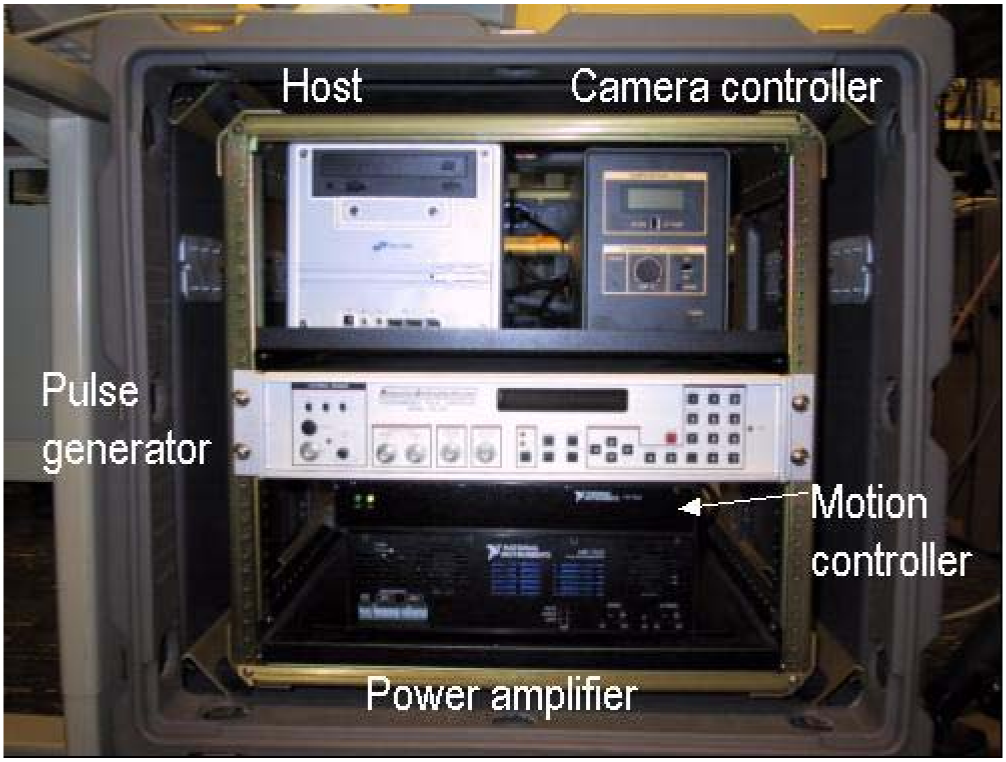


Figure 7. New hardware

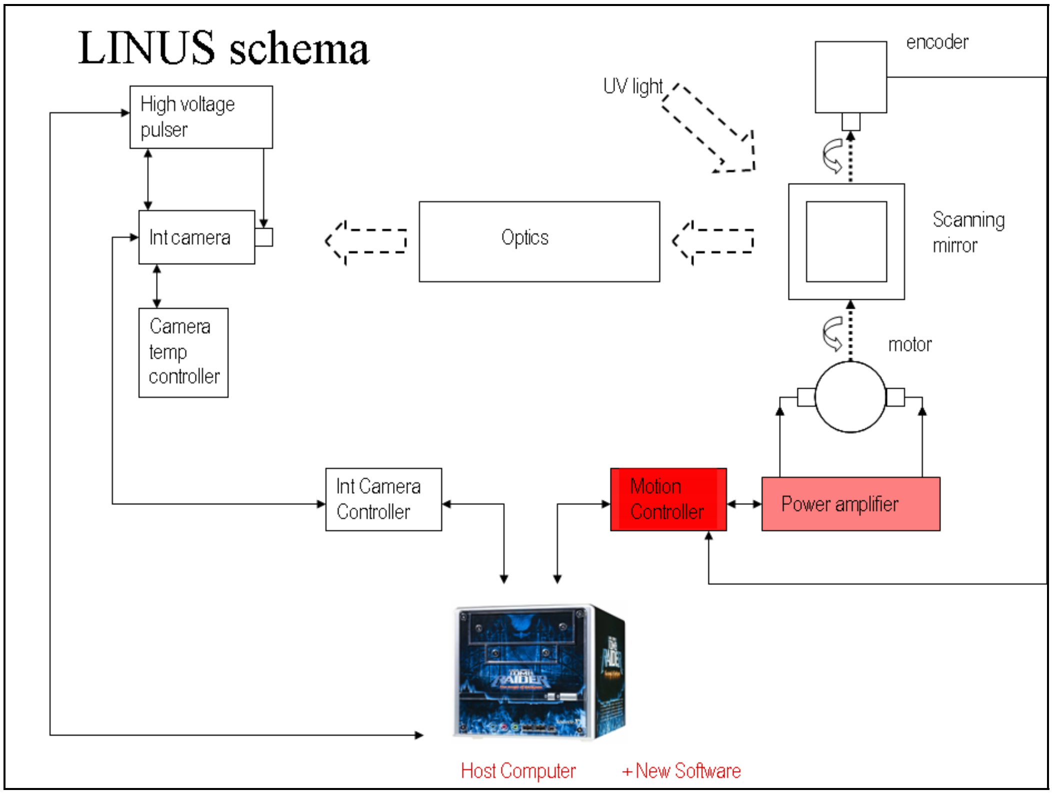


Figure 8. New hardware layout

## C. MOTION CONTROLLER ADJUSTMENT

The power amplifier and motion controller changes required that the servo system be tuned according to the closed control loop modeled in Figure 9. Changes to the transfer function, equation 3, are highlighted in red.

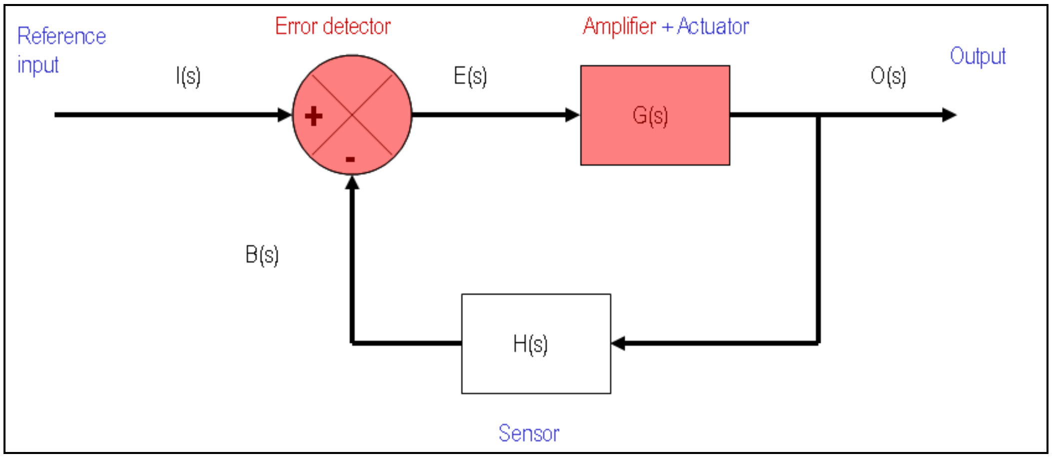


Figure 9. Control loop changes [ref. 4]

The input I(s) is related to the output O(s) by the following equation:

$$\frac{O(s)}{I(s)} = \frac{G(s)}{1+G(s)H(s)} \qquad (3)$$

Notice that the amplifier gain G(s) changed in the numerator and the denominator, altering the closed loop transfer function. Therefore, the scanning sub-system had to be tuned again. See Kompatzki ref [12], for a discussion about PID control and the precision scanning requirements for LINUS.

### 1. PID Tuning Procedure and Results

The automated Lab View PID tuning procedure for the motion controller did not give satisfactory results. Mirror control was generally under damped with settling times between 0.9 to 1.2 seconds. Therefore, manual manipulation of these coefficients was required. This was accomplished using the standard "rule of thumb" procedures for manually tuning a PID control loop as outlined below:

* Set the integral gain (Ki) to zero

* Set the proportional gain (Kp) to a reasonable starting value for your system

* Set the derivative gain (Kd) to twice Kp

* Increase Kp by factors of 1.5 to 2 until the step response yields an overshoot

* Increase Kd by factors of 1.5 to 2 to diminish oscillations and settling time

* If this causes the system to respond slowly, increase Kp and Kd by a factor of 2 until the step response meets your requirements for rise time and settle.

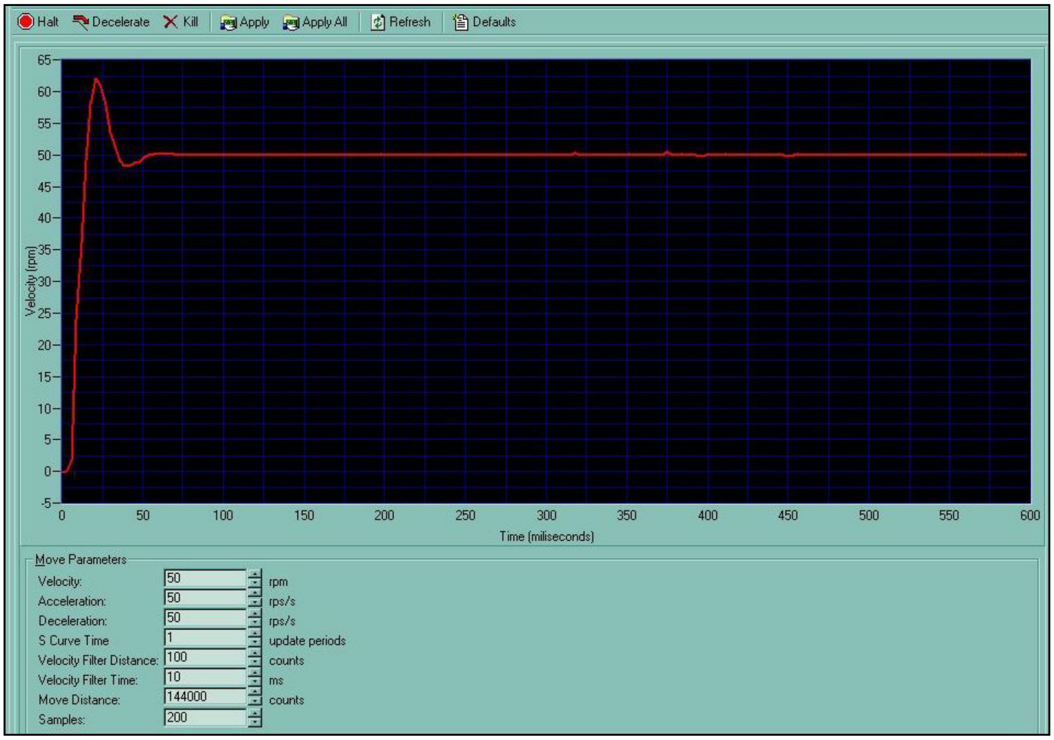
* If there is a final steady state error, apply Ki starting at one and increasing by steps of one until the steady state error is removed.

This procedure produced a stable response with PID parameters listed in Table 4.

| Name | Parameter | Value |
|---|---|---|
| Proportional gain | Kp | 110 |
| Derivative gain | Kd | 600 |
| Integral gain | Ki | 55 |
| Integration limit | Il | 50 |
| Derivative sample period | Td | 3 |

Table 4. PID controller parameters

The dynamic performance of the system is summarized in Figures 10, 11, and 12.

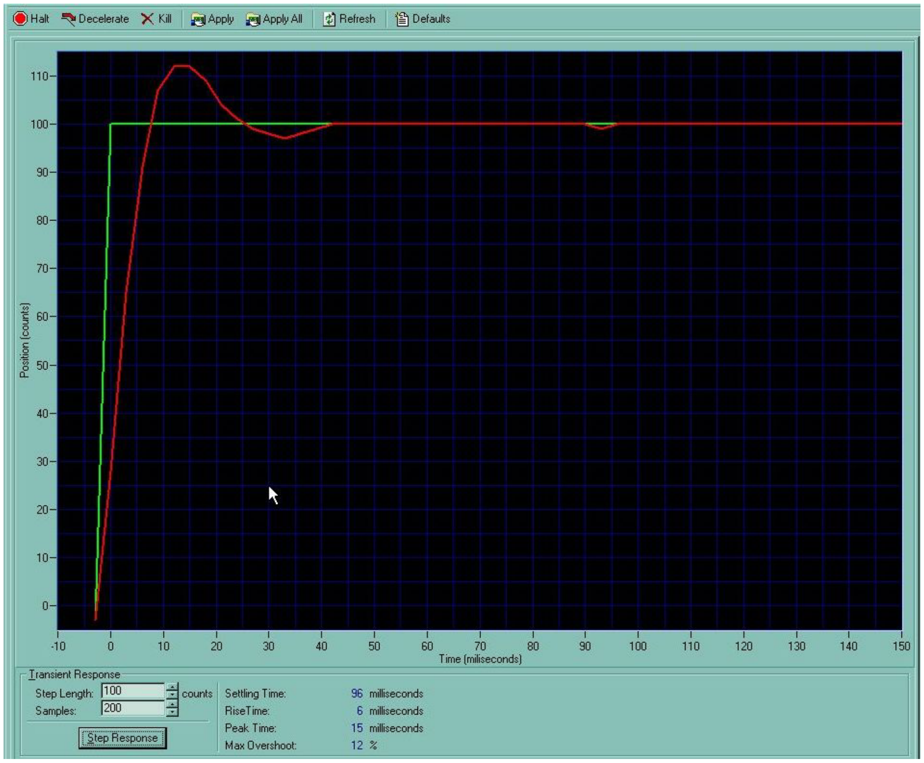


Move Parameters

Velocity: 50 rpm
Acceleration: 50 rps/s
Deceleration: 50 rps/s
S Curve Time 1 update periods
Velocity Filter Distance: 100 counts
Velocity Filter Time: 10 ms
Move Distance: 144000 counts
Samples: 200

Figure 10. Servo velocity response

Figure 10 shows the servo velocity response in milliseconds as a function of axis rotational speed in rpm. For our measurements, this was set to 50 rpm. At this rotational speed, the system settles in about 60ms and that is acceptable for our time requirements.



Step Length: 100 counts
Samples: 200

Settling Time: 96 milliseconds
RiseTime: 6 milliseconds
Peak Time: 15 milliseconds
Max Overshoot: 12 %

Figure 11. Servo step response

Figure 11 shows the servo step response in milliseconds as a function of step counts. A step is defined as 360/144000, which is the resolution of our servo shaft encoder. For this illustration the maximum overshoot was only 12%, the rise time was less than 6ms and the system settled in about 96ms. This is excellent because the minimum time between images for LINUS is, at best, 500 ms.

Although the system is a slightly under damped we leave it like this because it settles faster than if it were critically damped. This did affect our high frequency stability a little as is displayed in the Bode gain and phase plots below.

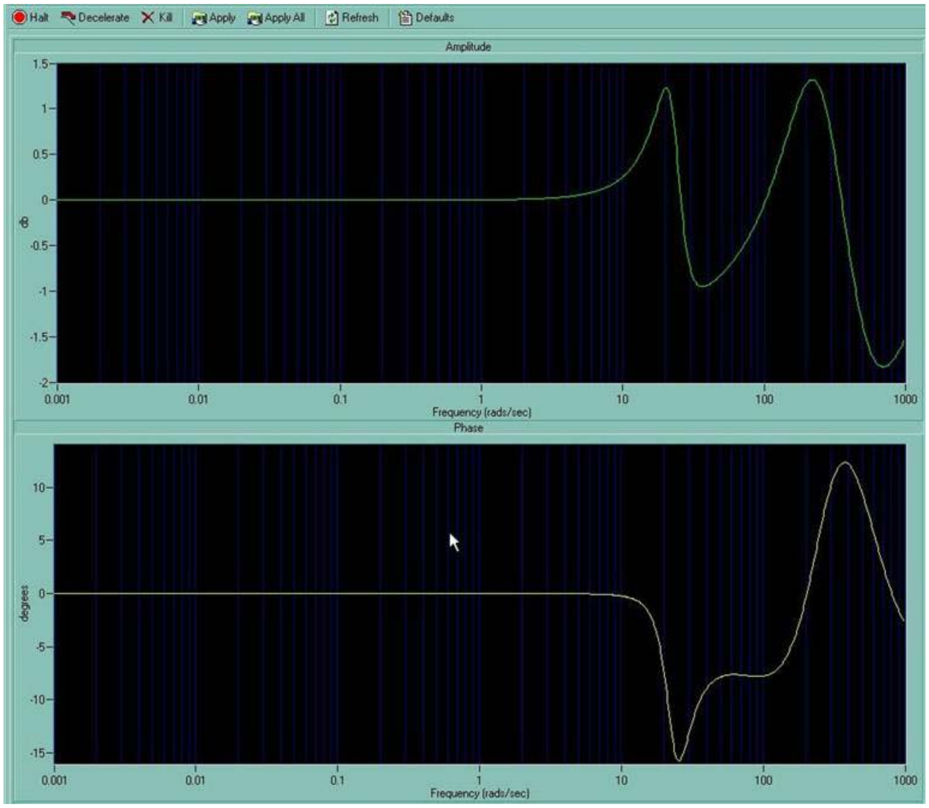


Figure 12. Servo Bode plot (frequency and phase response)

Figure 12 shows the gain and phase Bode plots for our system. The gain and phase response is remarkably stable for lower frequencies. It shows instability at about 25 rads/sec and again at approximately 70 rads/sec. The first peak at 25 Hz correlates easily with the inverse of the period of the overshoot displayed in the step response. We are not able to determine the cause of the peak at 70 hz. In both cases, the instability reflects less than 1.5 Db deviation from zero with the phase shift less that 15 degrees. In theory, this could be easily handled with the judicious use of high frequency filters, but because the effect is relatively small and the step response in the time domain meets our requirements we chose to accept it.

## D. VISUAL CAMERA DESIGN

Two cameras, one wide field of view and one narrow field of view comprise the aiming system selected. The Sony XC-ST70 Black and White CCD camera was selected for implementation into LINUS as the narrow field of view one. It has a 75 mm F 1.4 lens and the main characteristics of the camera are shown in Table 5.

| | |
|---|---|
| Type | CCD |
| Resolution (max) | 768 x 494 pixels |
| Video output | 1.0 Vpp, 75 Ohms |

Table 5. Camera characteristics [ref. 9]

Figure 13 shows a schematic of the how the camera was implemented in the existing LINUS architecture. The idea was to invoke a visual alignment technique that minimally impacted the current spectral imager optical layout designed by Scott Davis [ref.13]. It was decided to position the camera as shown in Figure 14. This choice minimized superstructure changes and allowed video and power cables to be routed through the existing connection junction box, Figure 15.

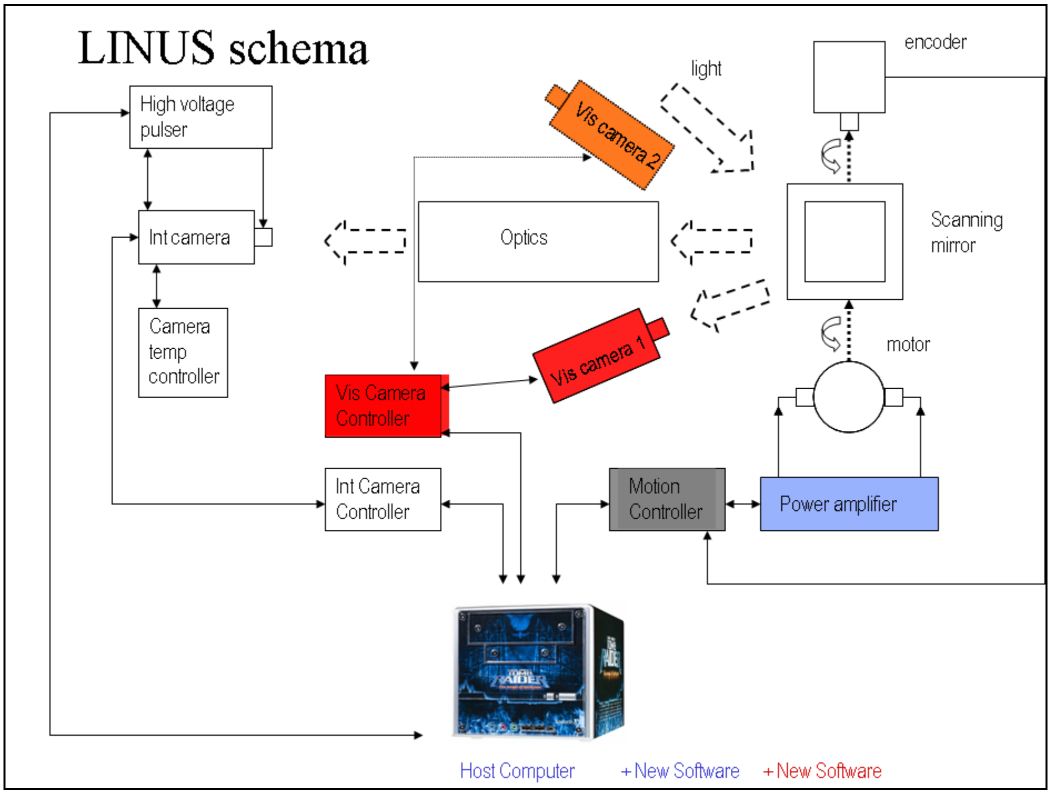


Figure 13. Visual camera layout

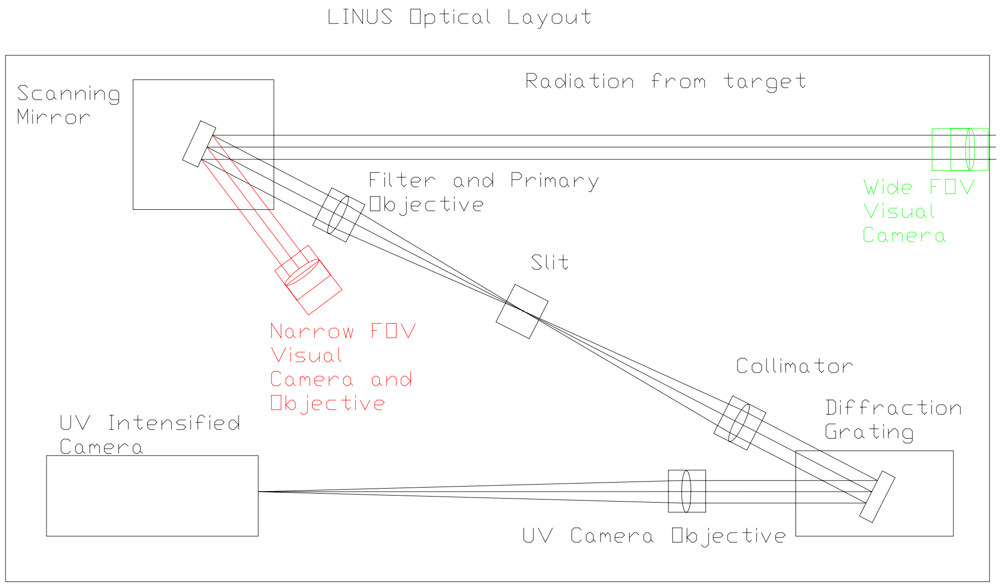


Figure 14. New optical layout

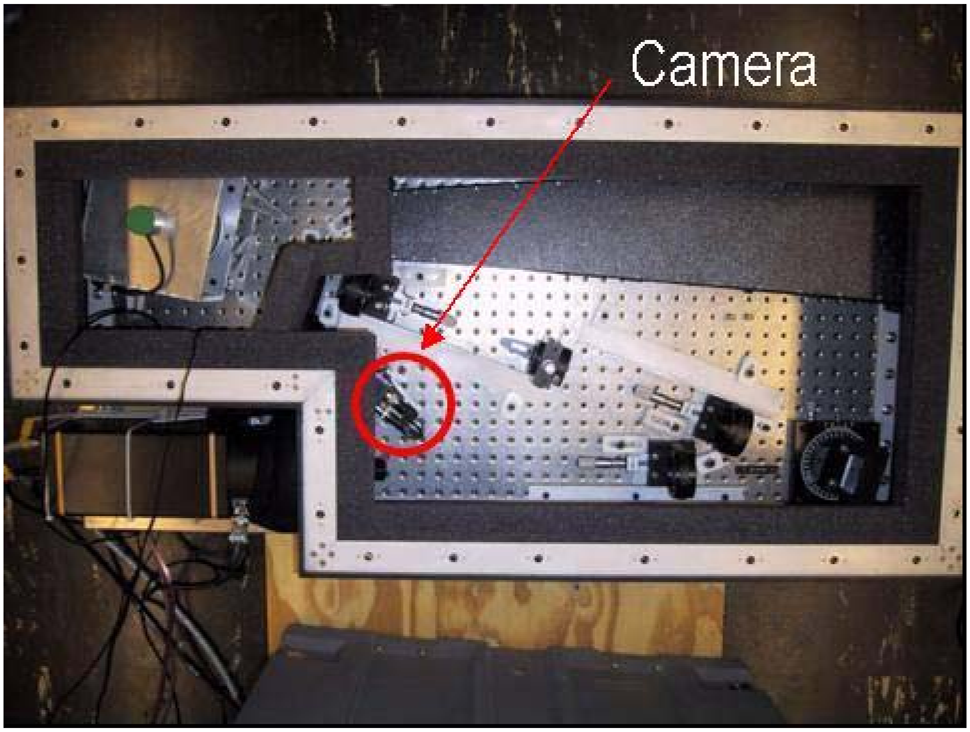


Figure 15. Camera location

The National Instruments IMAQ PCI-1411 was selected as the controller for the visual camera. Camera controller specifications are listed in Table 6.

| | |
|---|---|
| Input formats | RS-170 /NTSC /CCIR |
| Output formats | RGB 32 bit, HSL 32 bit, Luminance 8 bit |
| Interface | PCI |

Table 6. Camera controller characteristics [ref. 10]

The main program was modified by adding two modules to handle the visual camera output. The first module added a call to the frame grabber dynamic link library (DLL), which allowed the data stream to be presented directly to the screen. It also included a routine to move the scanning mirror to the proper alignment position for imaging. This presented, to the operator, a clear field-of-view image that was centered for aiming purposes. The second module created a routine such that one NTSC frame from the image stream could be stored and saved as an eight-bit bitmap file for later comparison against the actual spectral image.

### E. CAMERA ALIGNMENT

The setup for the camera alignment consisted of an optical bench, a class 1 laser with two opposed beams, a platinum lamp and a target as shown in Fig. 16 and 17.

In order to get a stable alignment, the optical subsystem was dismounted from the tripod and placed on three bricks on the floor. The optical bench was positioned in front of the optical aperture with the hyperbolic mirror aligned with the scanning mirror at bore sight angle, as in Fig. 18. The alignment started by leveling and checking the height of all the components. Both laser beams were checked in height and position with respect to the center of the optical aperture and the scanning mirror.

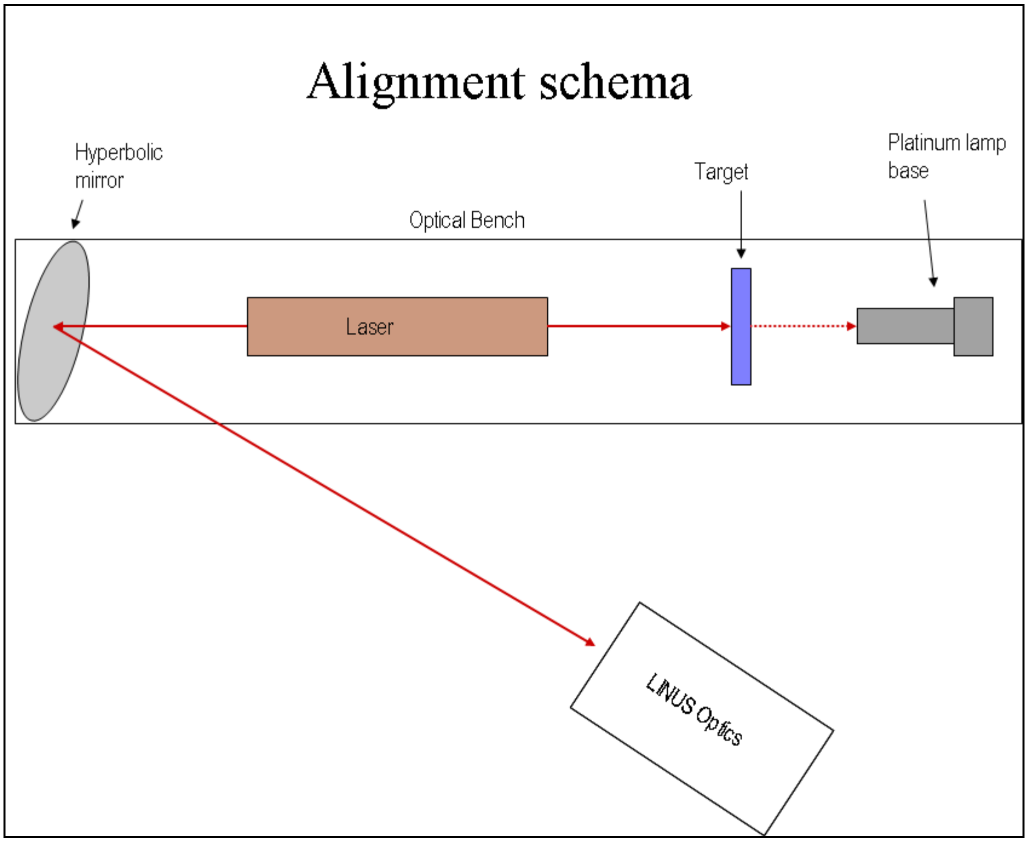


Figure 16. Alignment setup

The consistent height of the laser beam was checked across the optical path inside the optics system in order to determine if it was the appropriate arrival angle from the hyperbolic mirror. The second laser beam allowed alignment of the target and the platinum lamp optical mount.

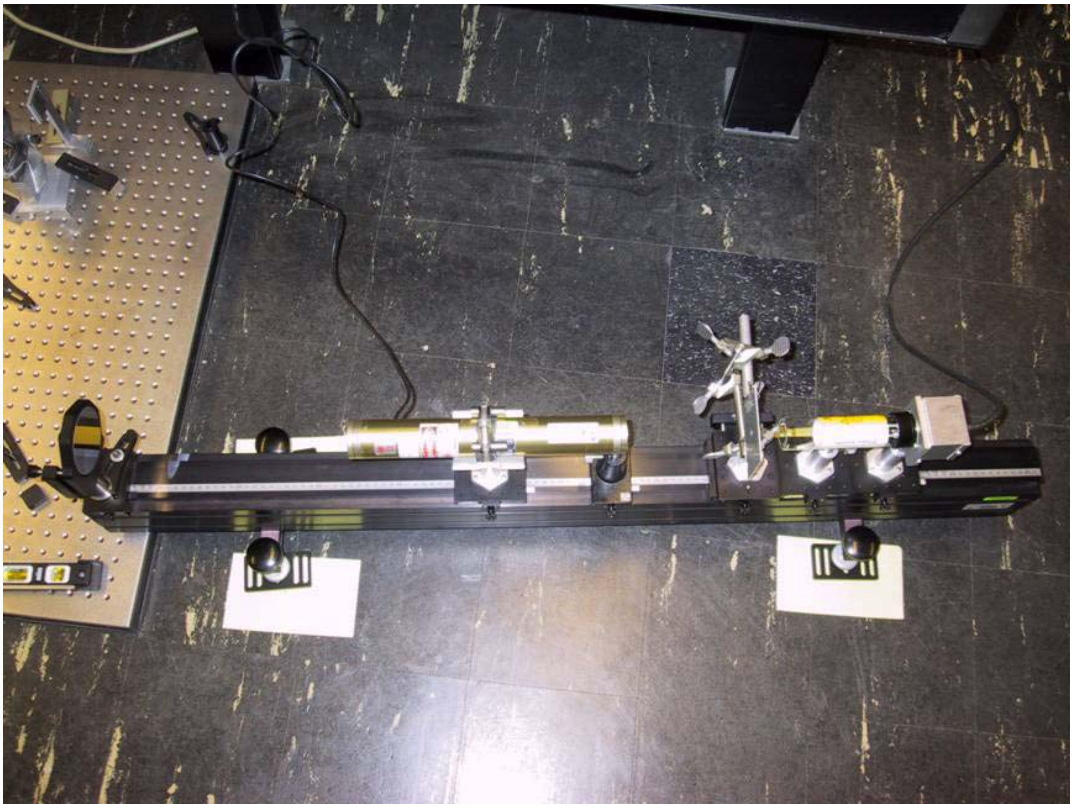


Figure 17. Alignment setup

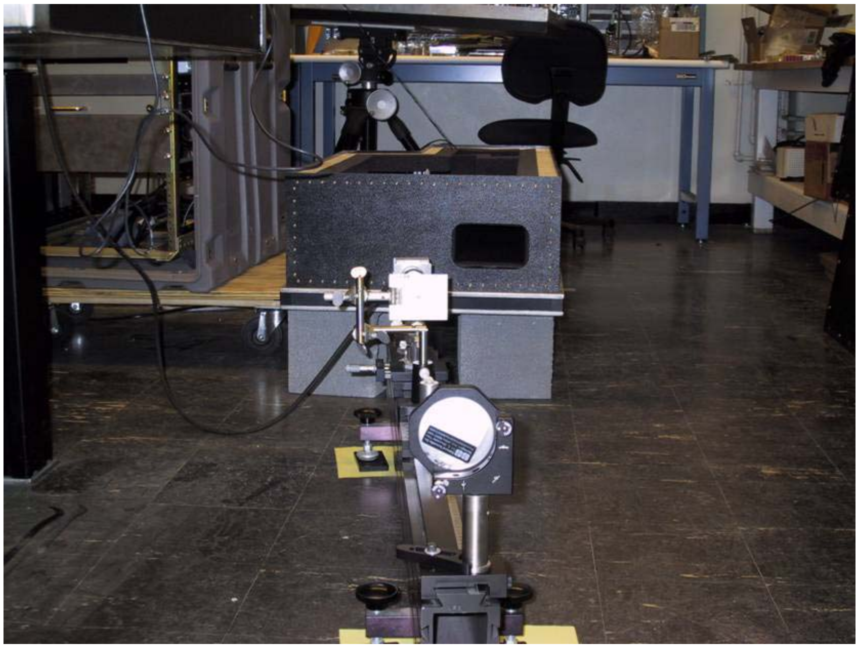


Figure 18. Alignment setup

### 1. Visual Alignment

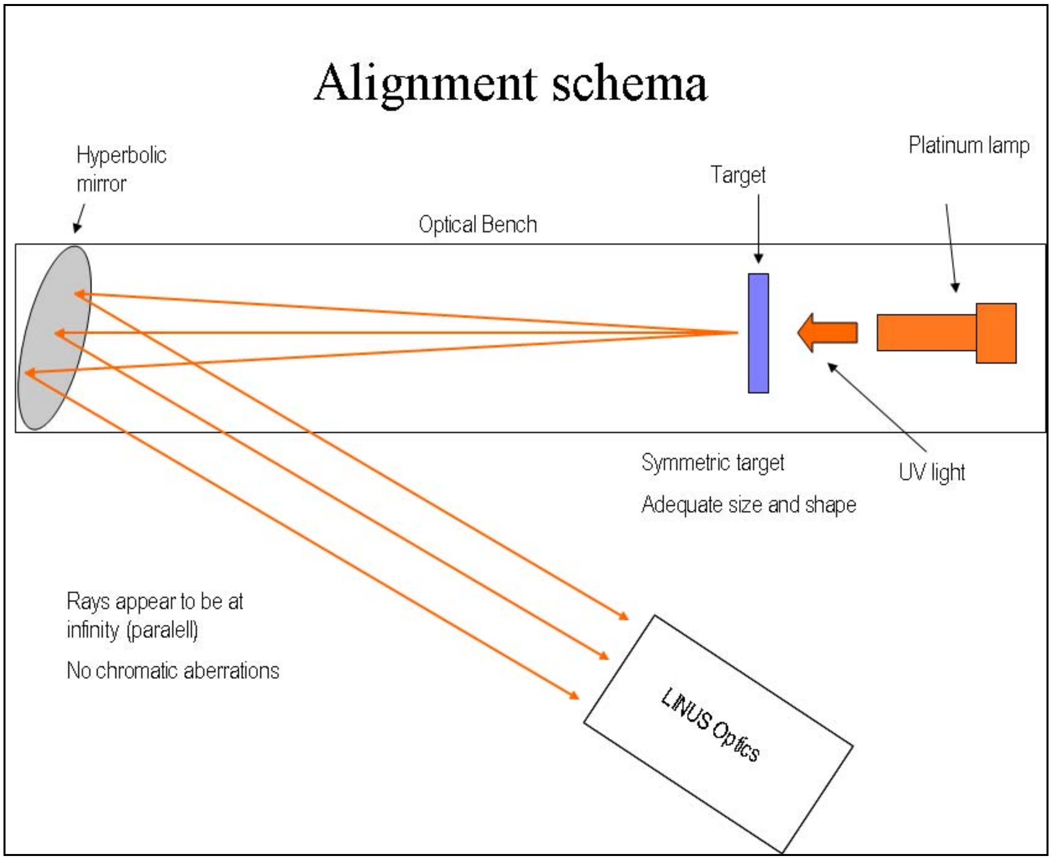


Figure 19. Alignment setup

When the optics were properly aligned, the visual camera was installed and checked for placement with the laser beam in order to determine the proper scanning mirror position and to obtain the vertical height by getting the laser beam in the center of the visual image.

The hyperbolic mirror reflects the light rays as parallel, then the camera objective was focused at infinity, obtaining the picture in Fig. 20.

Figure 20. Target visual picture

Since the platinum lamp emits light in the visual spectrum, a second image was taken using the setup in Fig. 19, with the laboratory in darkness, obtaining a similar result as the Fig. 20. In addition, it is a quick check to align the lamp hollow cathode with the target.

### 2. Ultraviolet Alignment

The setup was the same one used as in the visual camera alignment, shown in Fig. 20. Since the instrument is calibrated for the detection of SO2 in the 300 nm range, a hollow cathode platinum lamp was used as a source of ultraviolet light. The filter used for this purpose was a band pass filter whose curve is shown in Fig.21, with a 50% bandwidth span between 293-304 nm. The filter is modeled according the Gaussian equation (4)

$$f(\lambda) = 15.17\exp(-(\lambda - (298.43/4.48)^2)/2) \qquad (4)$$

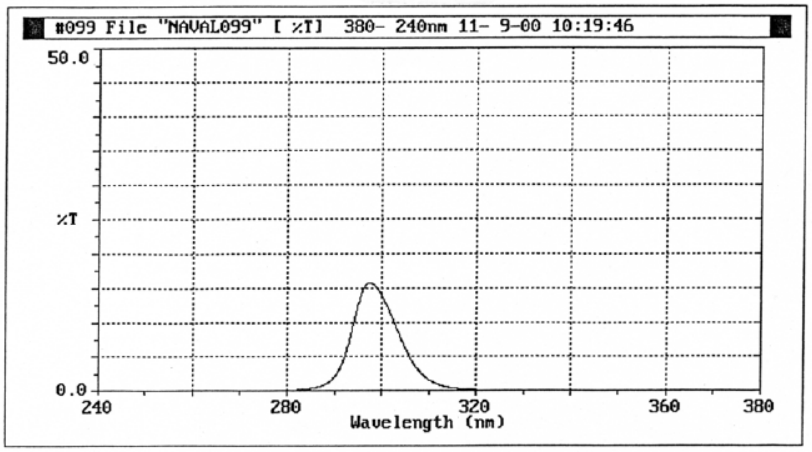


Figure 21. UV filter response [ref. 6]

The platinum spectrum is obtained (Fig 22 [ref.7]) and is compared visually with the image shown in Fig 23. The corresponding peaks are aligned moving the diffraction grating to get the 2998 (Å) peak at the center of the array, approximately pixel 256. Notice that the correspondence has to be calibrated, and then the platinum lamp calibration is performed to obtain the actual transformation from pixel location to wavelength.

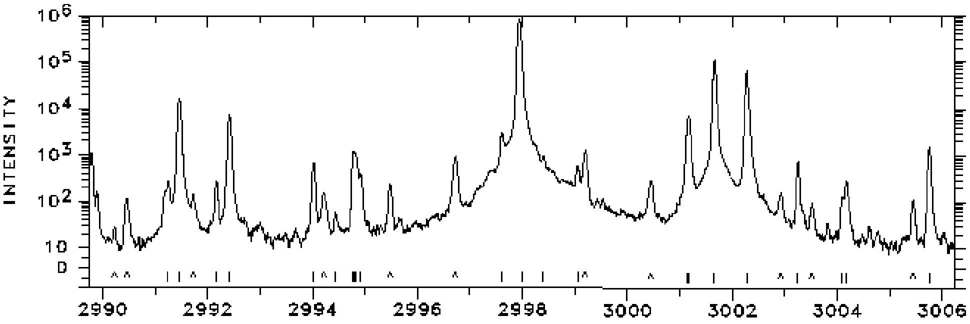


Figure 22. Platinum spectrum 299.0-300.6 nm [ref. 7]

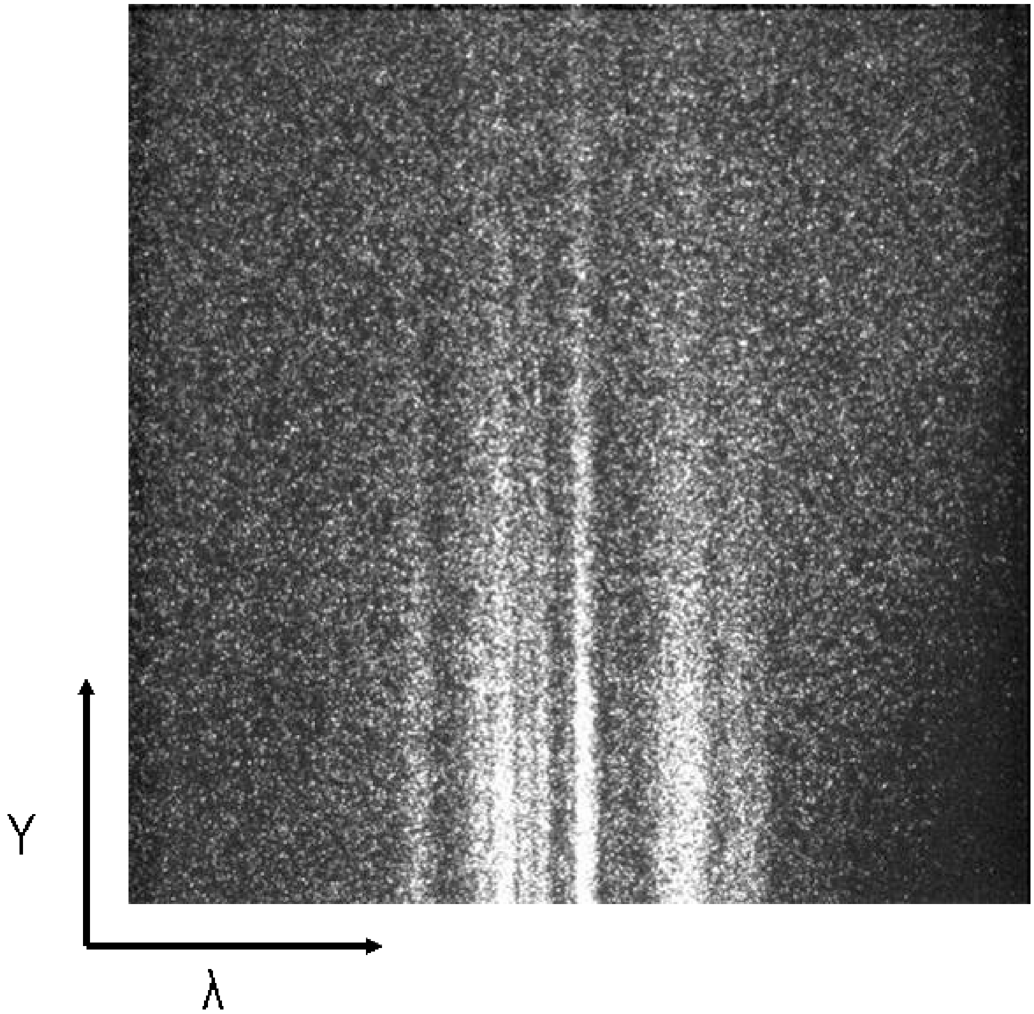


Figure 23. Image wavelength aligned

With the spectrum wavelength v/s pixel location (λ coordinate) visually aligned, the next step is to align the Y coordinate.

The placement is obtained by moving the hyperbolic mirror vertically up and down in order to get a balanced intensity in the Y-axis, as shown in Fig 24.

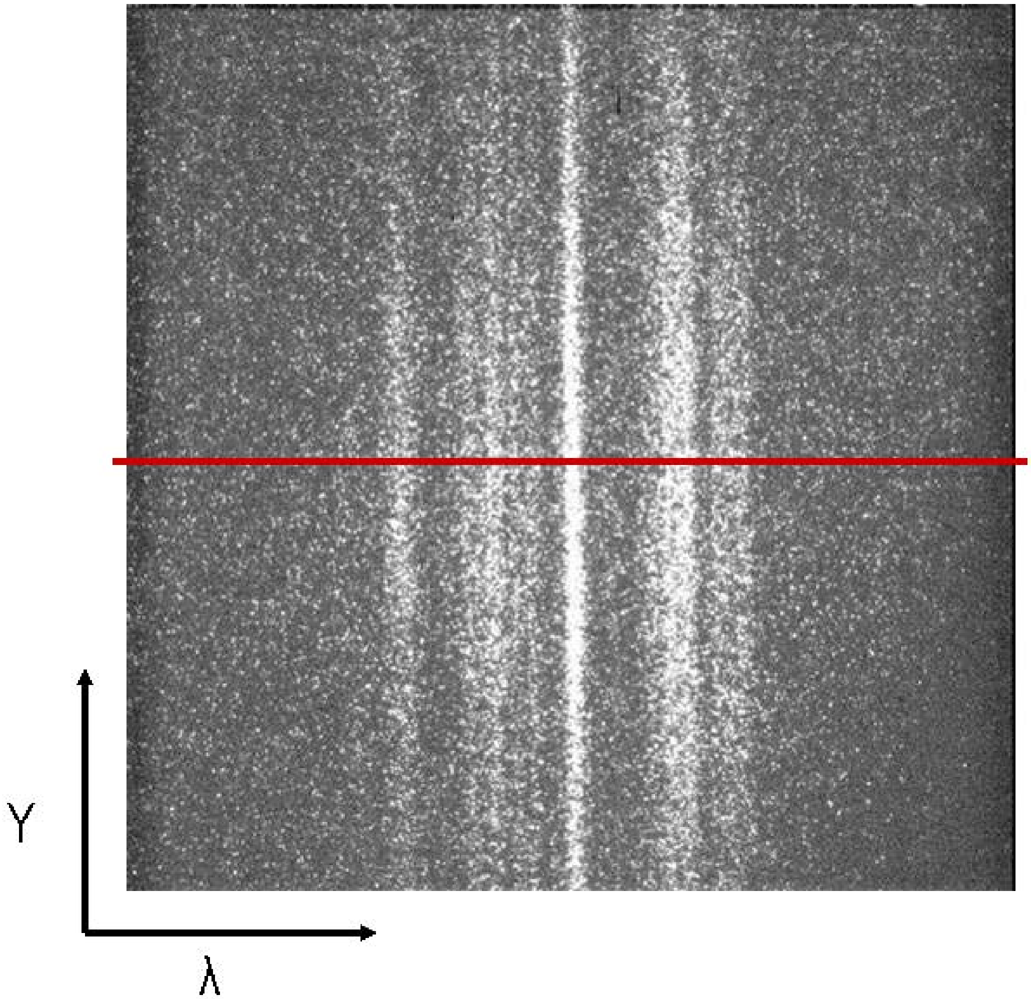


Figure 24. Image Y centered

The red line is located at pixel 256 in the vertical direction. The vertical line shows half of the intensity for each vertical direction from the red line.

With both Y and λ coordinates aligned, the next step is the most difficult: align the X coordinate. There is no direct and instantaneous way to align this coordinate, because the UV radiation is not visible to the human eye and the data collected by the

CCD array is not directly related to physical coordinates (X and Y). The only way to look at the image is to post process the hyper-cube data.

The first way to align the hyperbolic mirror and the internal optical system was formulated by using the diffraction pattern produced by the laser beam going through the slit, as shown in Fig.25. The diffraction pattern with the slit width 0.1 mm last for about 20 counts, then the precision of the alignment is ±10 counts.

During this procedure, an error in the collimate angle was discovered; this error was about 94 counts, or one quarter of a degree. This error was not noticed during the camera's original development, so probably no data set taken had been completely accurate due to this error. The calibration data is acquired at the collimate angle with plus and minus a few counts displaced; the field or laboratory test data is taken with about one degree of field of view. Hence, the error could be important. Chapter IV will try to check for this error.

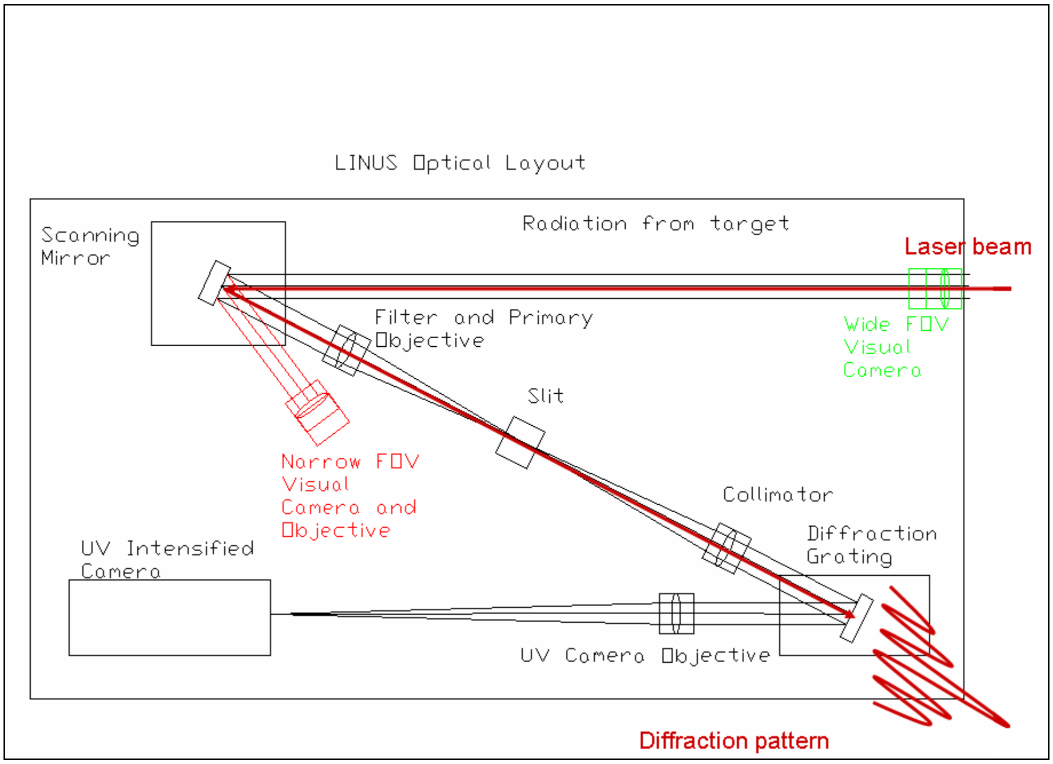


Figure 25. X alignment by laser

A second method was created in order to check and confirm the first one. During the collection of the frames (Y stripes) in the X direction (or time), due to the symmetric characteristics of the target, the central frame has to show the highest intensity peak, as shown in Fig.26.

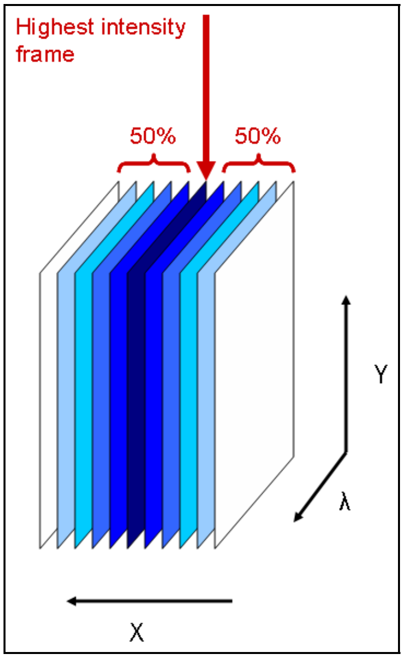


Figure 26. X alignment by intensity

Once the laser aligned the optical system, a check was performed with the data. Fig 27 shows the center frame from a hyper-cube, and an early frame as reference.

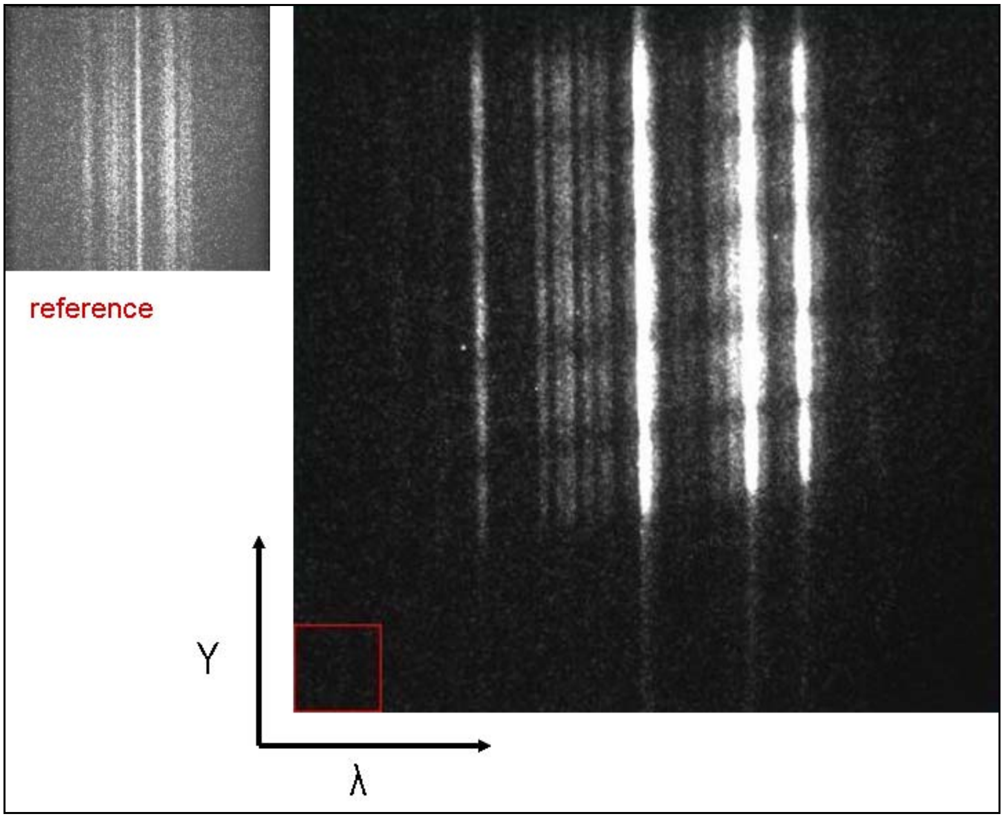


Figure 27. Image X aligned

The reference frame shows a lot of noise and the signal barely above it; meanwhile the center frame shows the highest intensity peaks obscuring the noise in the surroundings. In addition, five vertical peaks give some characteristics of the target used, a cross in this case.

Finally, the post-processed image, X and Y coordinates from the hyper-cube, is shown in Fig 28, verifying the alignments.

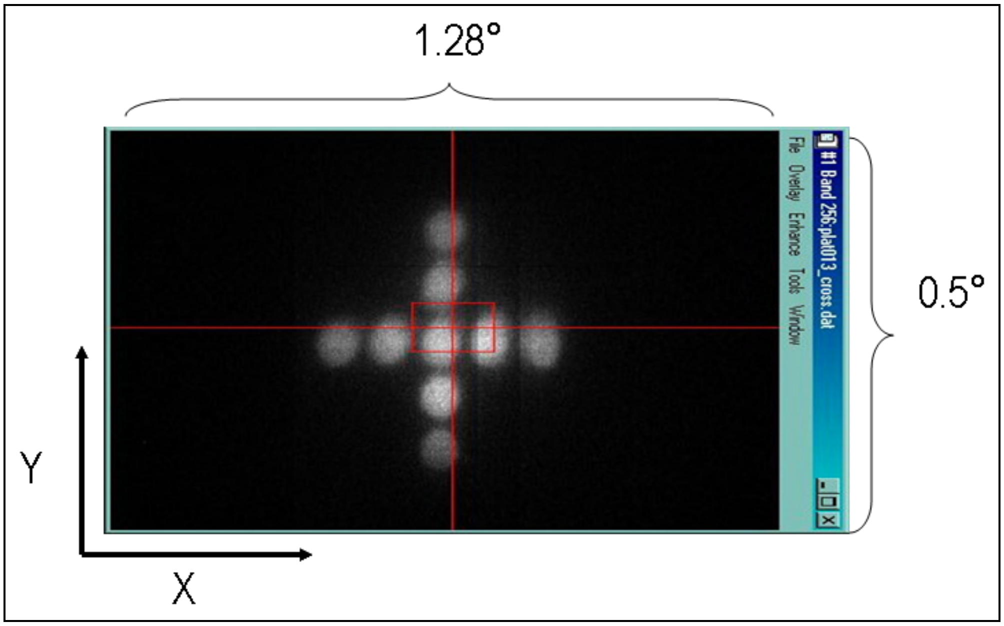


Figure 28. Processed X, Y image, aspect ratio corrected

The image is not completely aligned with the red cross in the center of the picture. However, the error is acceptable.

Figure 29 shows the three dimensions of the hyper-cube, from the data taken while aligning with the cross target. Figure 30 shows a comparison between the visual image and the UV image corrected in aspect ratio.

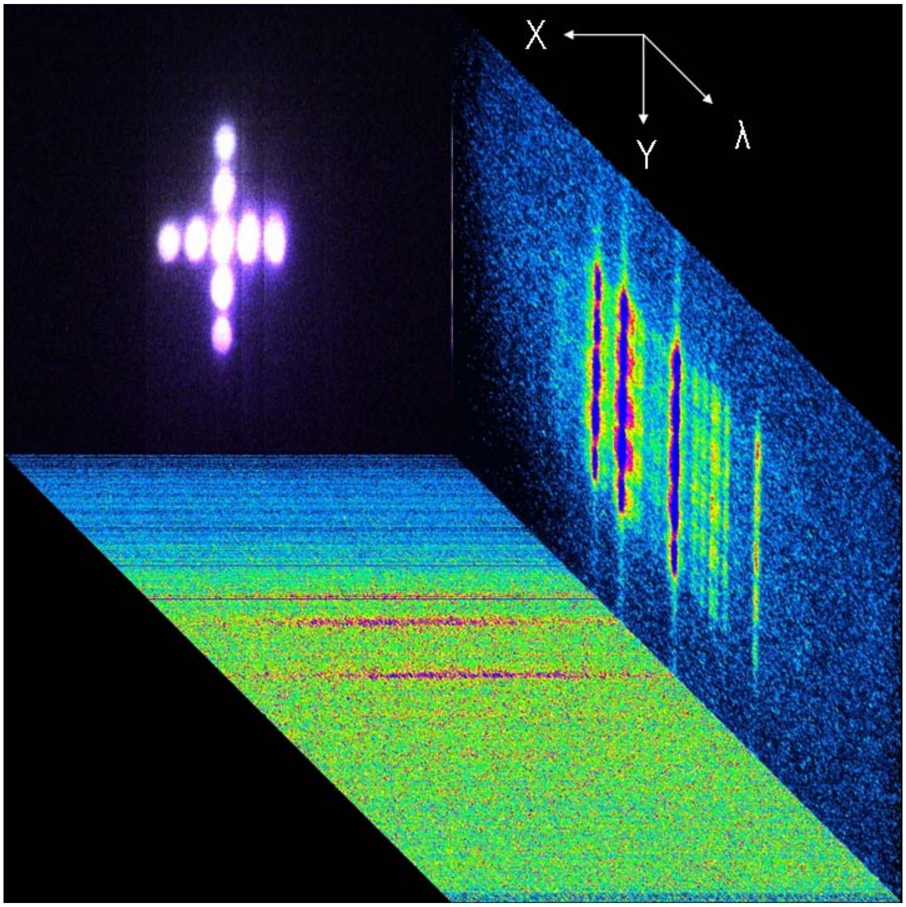


Figure 29. 3D Hyper cube

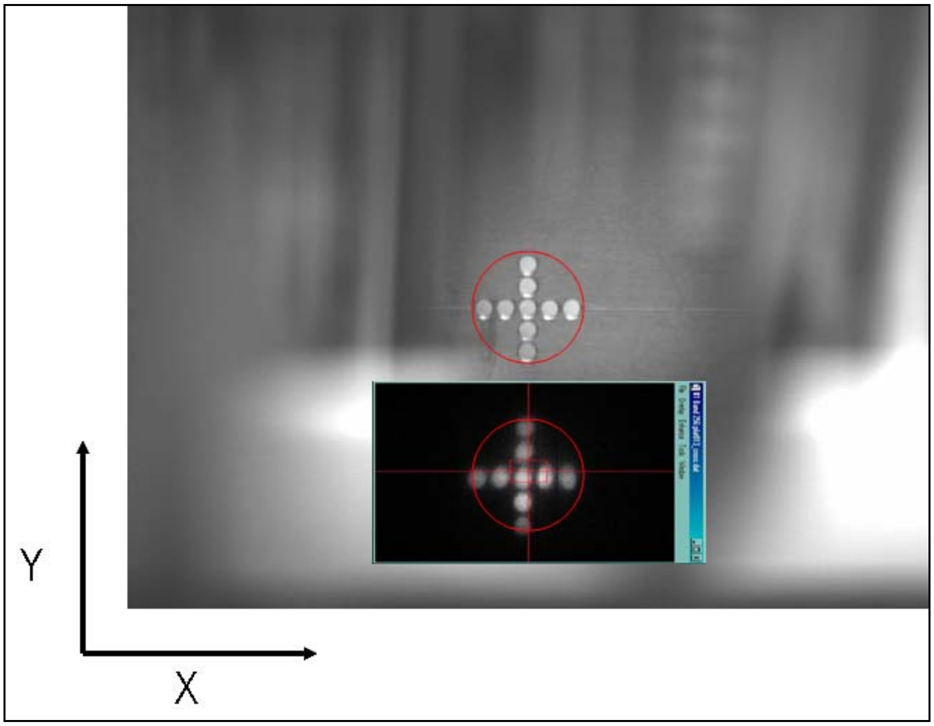


Figure 30. Visual and UV images comparison

# IV. CALIBRATION

## A. PURPOSE

The purpose of this chapter is to check the results performed by Gray [ref. 8] and to test the system integration.

## B. PLATINUM CALIBRATION

The hyper-cube data is recorded in pixels; then, in order to obtain information from them, they have to be calibrated or converted from pixel coordinates to wavelengths.

The data has to be compared with a known ultraviolet light source. The source used is a hollow cathode platinum lamp, as described in chapter four, whose spectrum is well characterized. This characterization was obtained from NIST [ref. 7], in the form of an ASCII file in order to be used by a computer code.

The calibration setup was the same used for UV alignment as described in Fig 19. The data acquisition was set to acquire the image with the scanning mirror in a fixed position at collimate, to ensure the maximum intensity of the scene is reflected. The parameters are shown in Table 7.

| | |
|---|---|
| Counts | 0 |
| Samples | 200 |
| Integration time | 10 s |
| Mirror position | 12965 counts |

Table 7. Data Acquisition parameters

The image obtained is similar to the one shown in Fig. 24; a strong correlation is observed performing a comparison by simple inspection with the platinum spectra from Fig.22.

Due to the filter response, the spectrum region of interest is around 300 nm (290-310 nm), and then the five strongest peaks from the Platinum NIST data around that region were selected. Those peaks are found at 2929.7894, 2955.7255, 2997.962, 3042.6318 and 3064.711 Å

The data was processed by an IDL $^{TM}$ program finding the correspondent peaks and pixel coordinate. In addition, it correlated the data with the NIST standard. Finally, the correlated data was modeled using a linear regression in MS Excel $^{TM}$, with the following results:

| | |
|---|---|
| Adjusted R squared | 0.9936 |
| Standard error | 4.54 |
| Intercept | 2776.58 |
| Coefficient | 0.8299 |

Table 8. Regression statistics

According to the statistics, the correlation is very good and the expected error is low in comparison with the wavelength values. Then, the calibration equation (5) is:

$$\text{Wavelength} = 2776.58 + 0.8299 * \text{Pixel Number} \qquad (5)$$

However, because of the few data points, this result has to be used carefully; the boundaries for interpolation are from 2929 to 3065 Å, and the extrapolation outside these limits is uncertain.

Figure 31 shows the data in blue and the linear regression in pink.

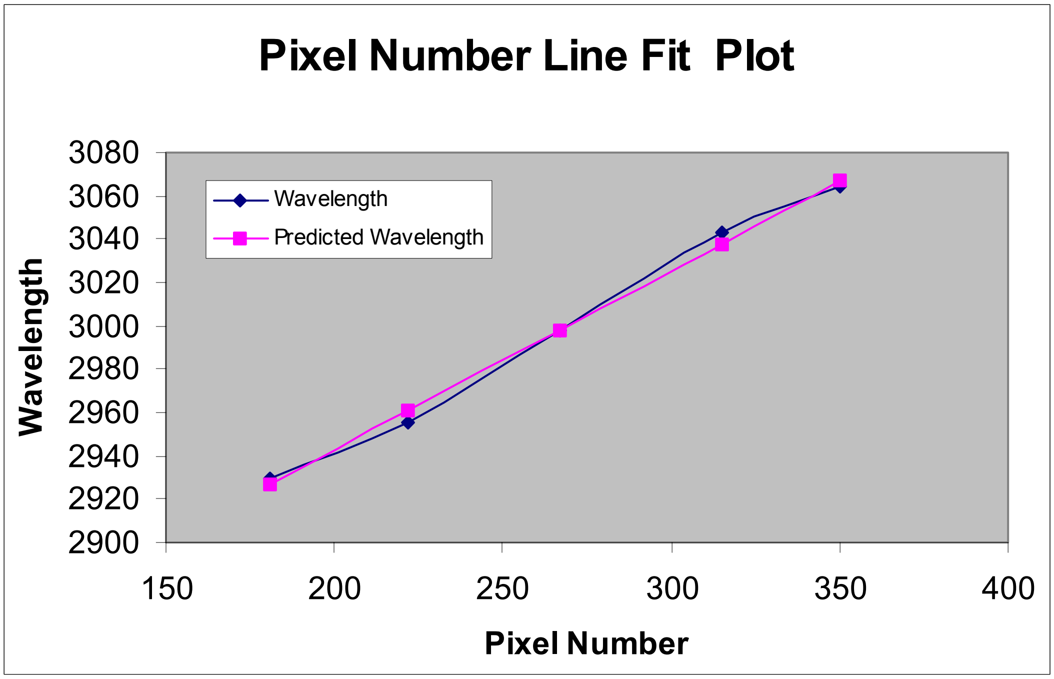


Figure 31. Data linear regression

## C. SULFUR DIOXIDE CALIBRATION

Because of time constrains, only one SO2 concentration data is presented. To obtain a detailed description about the SO2 calibration method see Gray [ref.8].

The differences between the work performed by Gray [ref.8] are:

- The use of a larger gas chamber to obtain a longer light path, due to the low SO2 gas mixture concentration, 0.11 %.
- Different instrument settings, see Table 9.

|  |  |
|---|---|
| Slit width | 0.055 mm |
| Integration time | 1 s |
| MCP voltage | 800 V |
| Frames per file | 30 |
| SO2 concentration | 0.11 % |
| SO2 pressure | 710 mm Hg |

Table 9. SO2 calibration settings

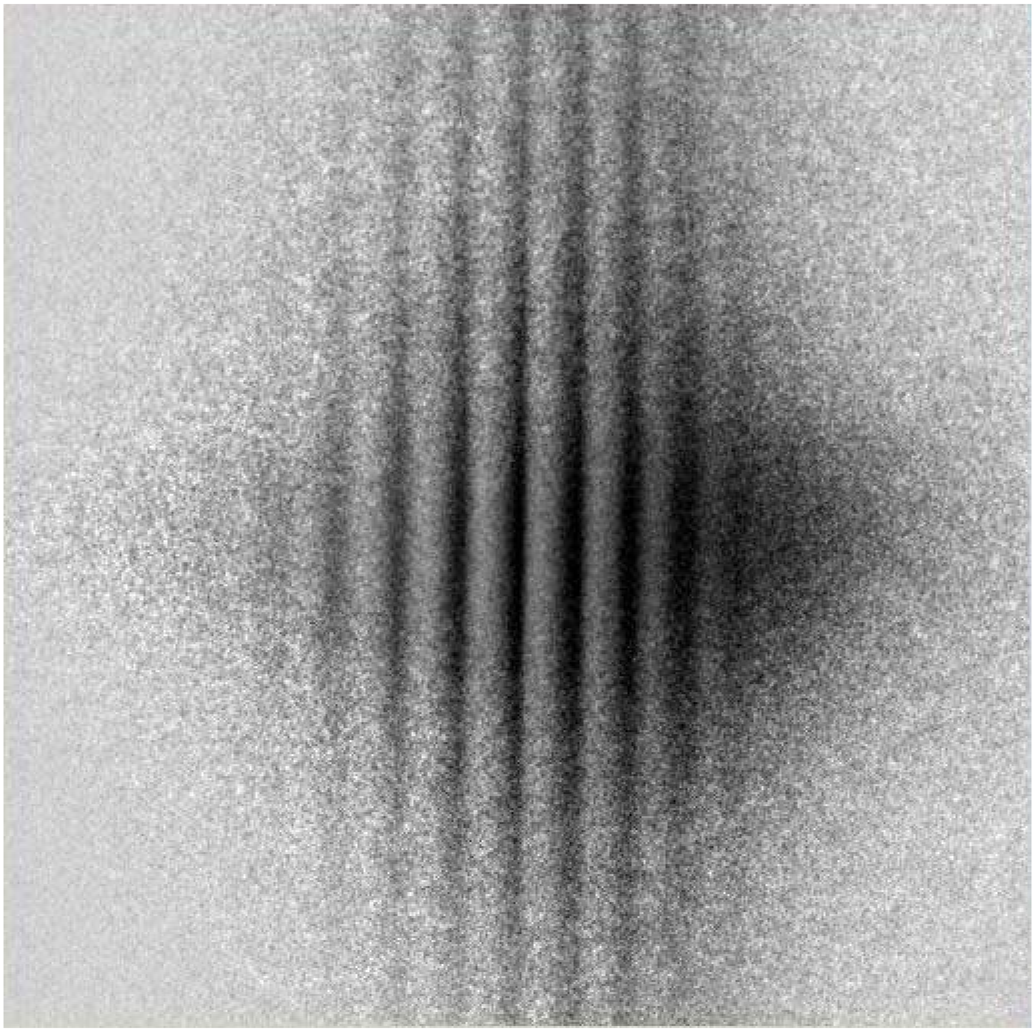


Figure 32. SO2/ vacuum ratio

The data sets obtained were one vacuum set and one SO2 set. They were processed in an IDL $^{TM}$ program and the ratio of the two is presented in Fig.32. This ratio was processed again and wavelength calibrated with the results from Section B, eq (5), obtaining the curve shown in Fig. 33.

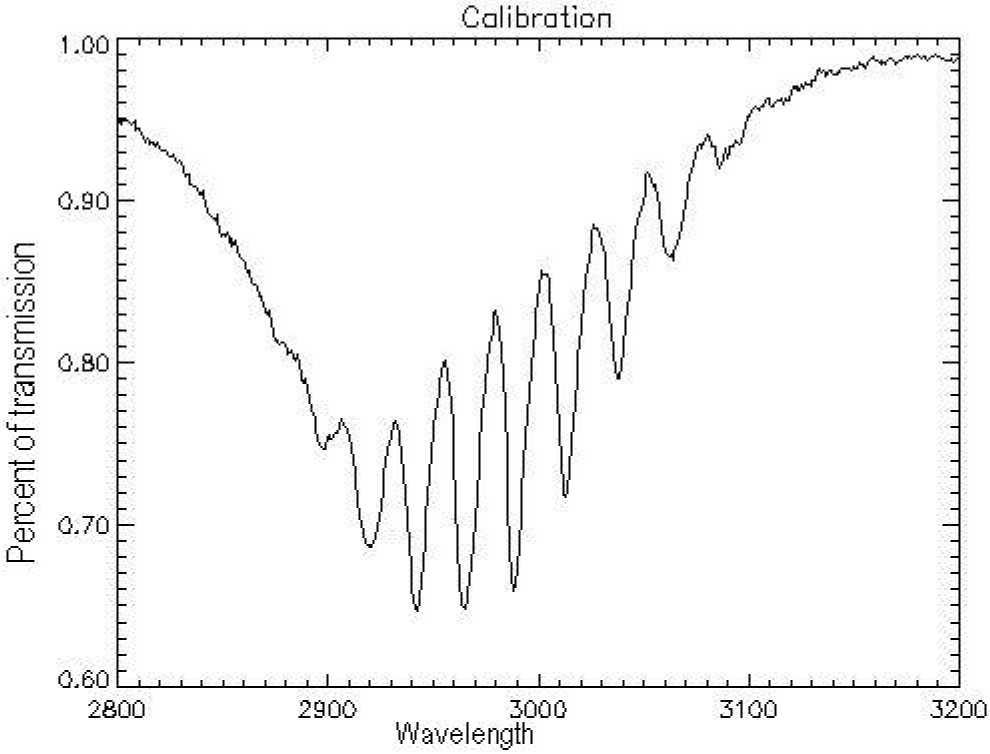


Figure 33. SO2 Calibration

The picture shows the expected peaks in the SO2 spectrum.

THIS PAGE INTENTIONALLY LEFT BLANK

# V. CONCLUSIONS AND RECOMENDATIONS

## A. CONCLUSIONS

1. A method for visual image and UV spectral image alignment was developed and tested successfully in laboratory.

2. The objective to reduce the weight and size of the system was achieved. The integration of the new hardware reduced by the overall weight of LINUS by approximately 60% and the size by 45%.

3. Because of the complexity of LINUS, System integration of the optics, software, and motion controller was difficult. Lab View allowed for the automation of the camera including the precision control of the scanning mirror for imaging and aiming.

## B. RECOMMENDATIONS

1. Install the wide field of view camera and test in the lab and the field.
2. Finish the SO2 partial pressure measurements and calculate the curve of growth.
3. Fully deploy and test LINUS in the field.

THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX A: LABVIEW CODE

## A. LINUS NEW PROGRAM

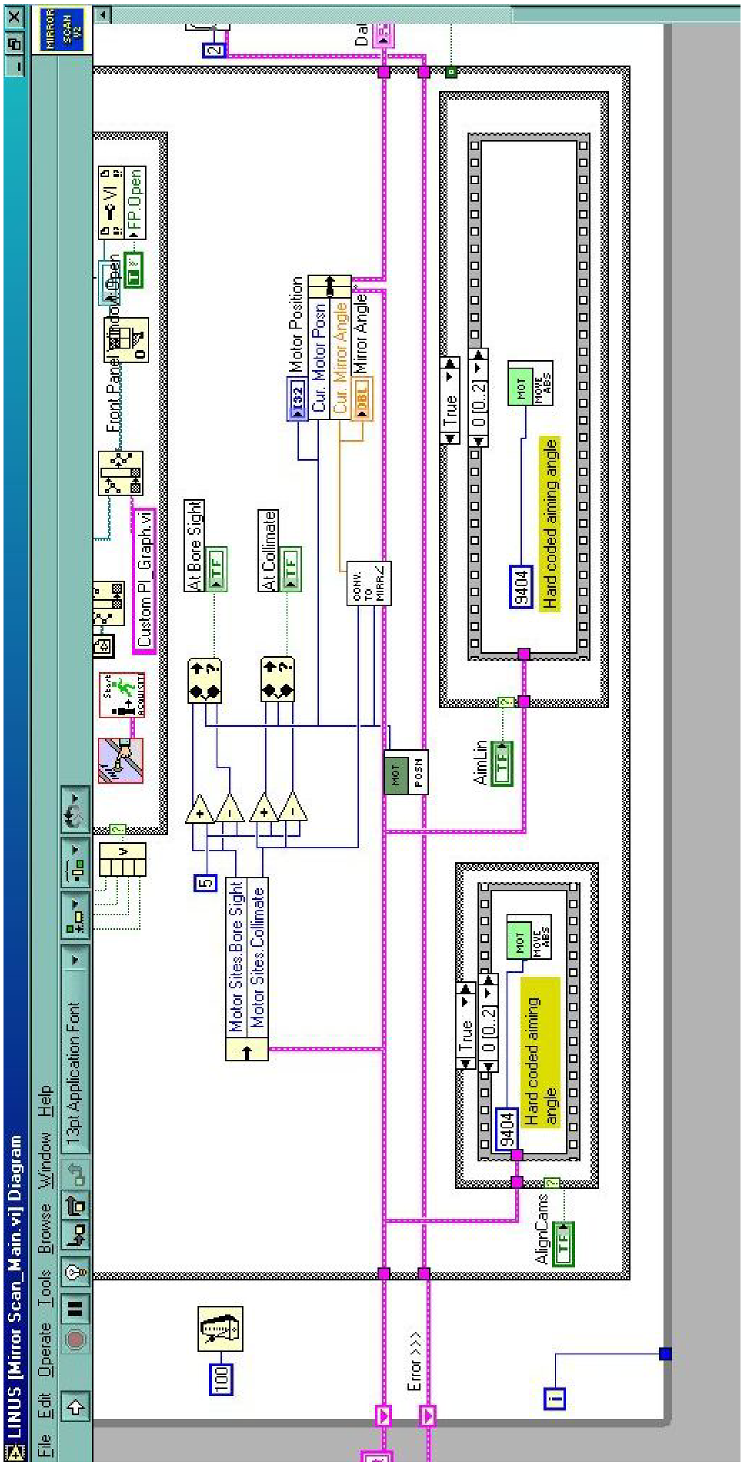


Figure 34. LINUS new code

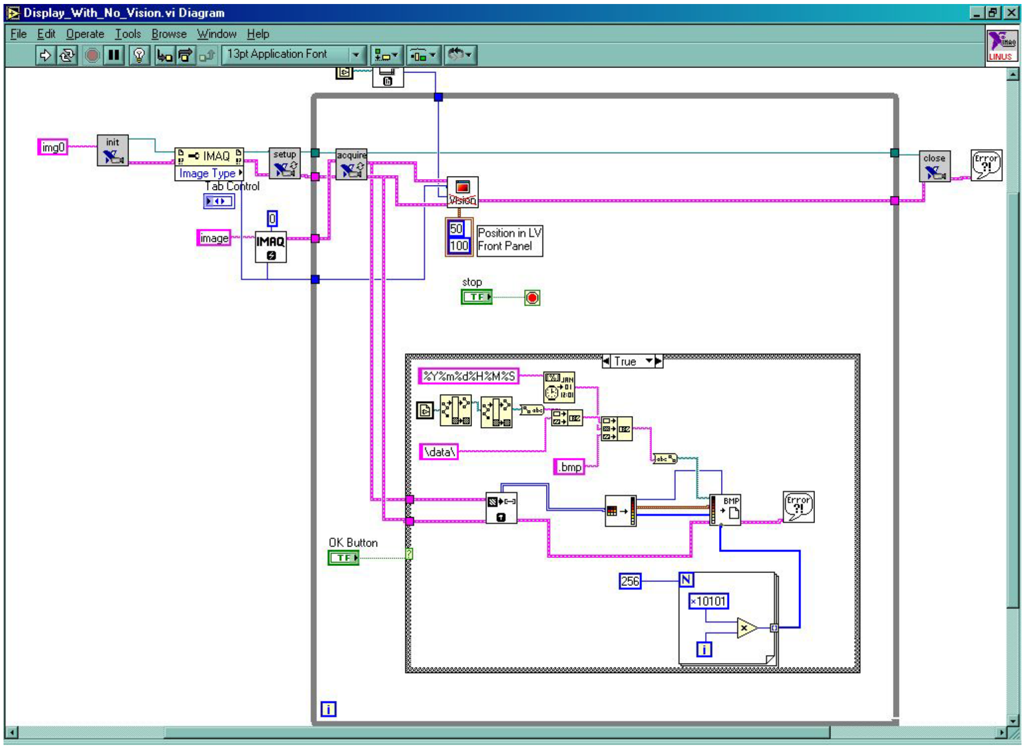


Figure 35. Aiming program sequence 1

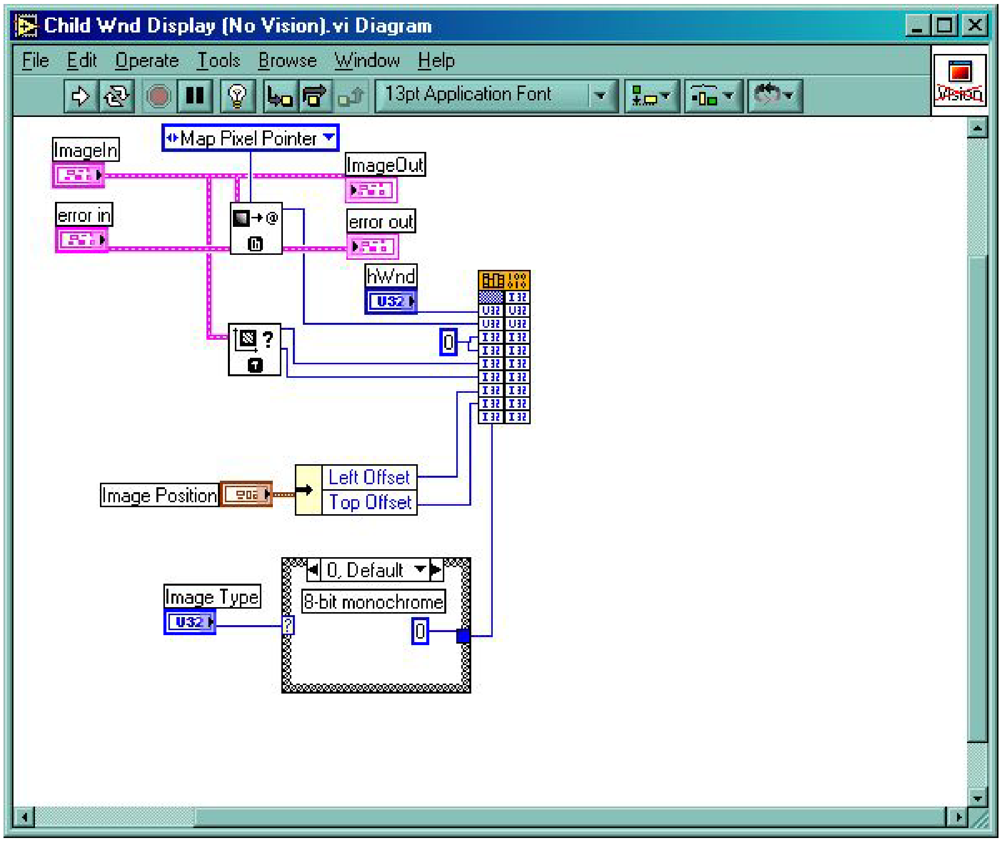


Figure 36. Image display sequence

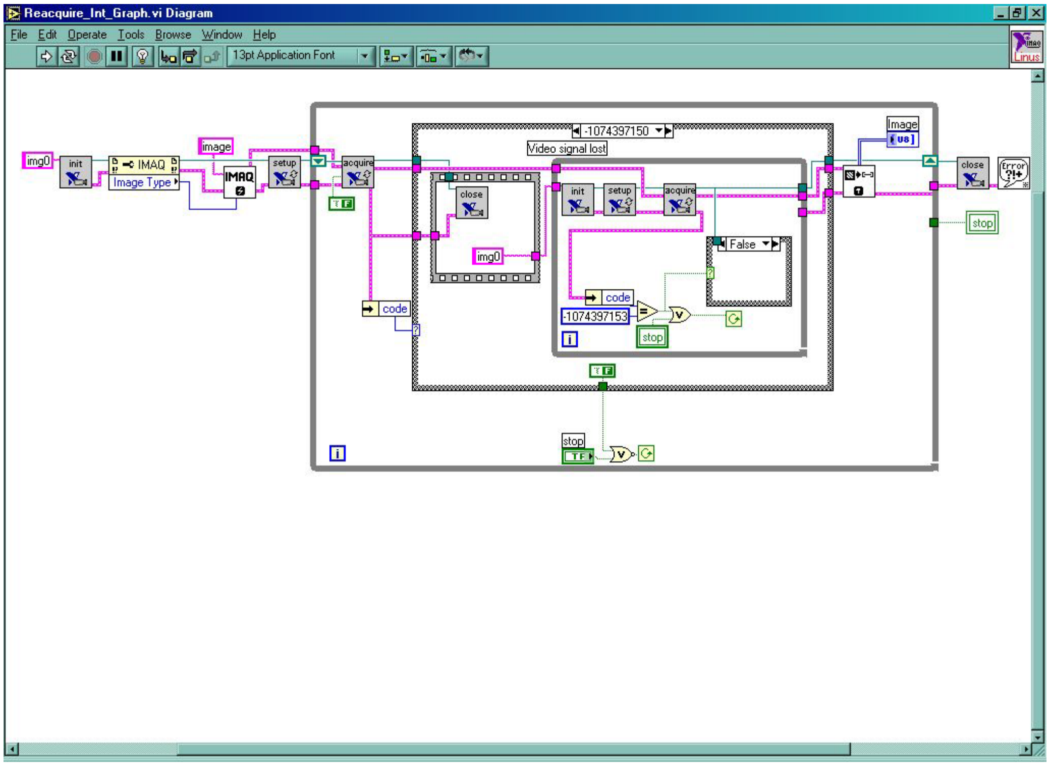


Figure 37. Image display sequence 2

THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX B: IDL CODE

## A. PLATINUM CALIBRATION CODE

This code is used to process the hyper-cube data obtained from a platinum source and compare the spectral data with a known source provided by NIST.

### 1. Calibration Program

```
************************************************************
; CALIBRATION PROGRAM FOR LINUS
;***********************************************************
; Created by Prof R.C. Olsen, 09/2002
; Modifications:
; 25-06-2004      Cabezas     Allows to handle the new LabView
;                       ........Hypercube
;
;***********************************************************

;***********************************************************
; Sub routine: nist filter and calibration data
;           input file: calibration file from NIST
;***********************************************************
pro nist, x_nist, yy

pt_file = 'C:\A_Linus\idl\Platinum_nist_cal.txt'
openr, 1, pt_file
index = 0
hdr = ' '
while not eof(1) do begin
readf, 1, hdr
;print, hdr
index = index + 1
endwhile
close, 1

print, index
lines = index - 1
wave = 3100.
wave_n = 3100.
inten = 100L   ; integer

openr, 1, pt_file
data = fltarr( 2, lines)
for i = 0,lines-1 do begin
 readf, 1, wave, wave_n, inten
 data(0, i) = wave
 data(1,i) = inten
endfor
close, 1

x_nist = data(0,*)
y_nist = data(1,*)
plot, x_nist, y_nist, yrange = [00, 1e6]
```

```
      ; Gaussian filter response function
      ctr = 2985.3
      width = 46.56
      amp = .15403

      pt_filter = amp * exp( -0.5 * ( (x_nist-ctr)/width )^2 )
      plot, x_nist, pt_filter, xrange = [2800, 3200]

      yy = y_nist * pt_filter
      plot, x_nist, yy, yrange = [00, 1e5], xrange = [2800, 3200]
      width = 3
      yy = smooth( yy, width)
      plot,  x_nist,  yy,  yrange  =  [00,  1e5],  xrange  =  [2850,  3150],
xstyle = 1

      return
      end

      ;*******************************************************
      ; Main program
      ;*******************************************************

      disk = 'c:\'
      wdir = disk + 'A_Linus\data\'
      cd, wdir
      dir = disk + 'A_Linus\data\'
      n_file ='Platinum_nist_cal.txt'

      data_dir = disk + 'A_Linus\data\'

      set_plot, 'win'
      window, 6,  xsize = 800, ysize = 400, xpos = 1,   ypos = 1
      window, 16, xsize = 800, ysize = 512, xpos = 1,   ypos = 300
      window, 1,  xsize = 800, ysize = 600, xpos = 500, ypos = 10

      samples = 512     ; Camera pixels
      lines   = 200     ; Samples (steps)
      bands   = 513     ; Camera pixels plus info line

      ;data_cube_06 = uintarr( 512, 513, 100)
      data_cube_06 = uintarr( samples, bands, lines)

      files = file_search('*.dat')
      ifile = 9
      file = files(ifile)
      print, file
      openr, 1, file
      result = fstat(1)
      sz = result.size & print, sz
      readu, 1, data_cube_06
      close, 1

      wset, 6

      tvscl, data_cube_06(0:511 ,1:512, lines/2)
      stop ;***** 1
```

```
line_06 = total ( data_cube_06[*, 1:512, *], 3) /lines
n1 = 171
n2 = 340
line_06 = total( line_06(n1:n2, *),1) /(1+n2-n1)

;****************************************888
wset, 16

;set_plot, 'cgm'
;device, /close, file = dir + 'Pt_cal_with_NIST_overlay_1.cgm'

xx = indgen(512)
plot, xx, line_06*1.5, yrange = [0, 1500], psym = 0, $
 xstyle = 1, ystyle = 1, xrange =[100, 400]
;oplot, xx, line_16, psym = 0, color = 255

max1 = max(line_06)   ; this is the 2997.9622 line
peak1 = !c  ; should be ab out 260

; 3045 angstrom line
; 3042.6318 - approx column 320
subset = line_06( 300:330)
max2 = max(subset)
peak2 = !c + 300
; 3064.711  - approximately 350
subset = line_06( 340:360)
max3 = max( subset)
peak3 = !c + 340
;  2929.7894 - approximately 180
subset = line_06( 150:200)
max4 = max(subset)
peak4 = !c + 150

print, peak1, max1
print, peak2, max2
print, peak3, max3
print, peak4, max4
plots, [peak1, peak1], [0, max1]*1.5
plots, [peak2, peak2], [0, max2]*1.5
plots, [peak3, peak3], [0, max3]*1.5
plots, [peak4, peak4], [0, max4]*1.5

;device, /close
set_plot, 'win'
wshow

stop ;****2

y11 = [ 2929.7894, 2997.962, 3042.6318, 3064.711]
x11 = [peak4, peak1, peak2, peak3]

radius = 1.2
circle = 2* !pi * findgen(9)/8
usersym, radius*sin(circle),  radius* cos(circle), /fill

!x.range = [ 0, 511]
```

```
      !y.range = [ 2800, 3200]
      !p.psym = -8
      !x.style = 1
      !y.style =1
      !p.title = 'Platinum Calibration - June, 2004'
      !x.title = 'Column (pixel)'
      !y.title = 'Wavelength (Angstroms)'
      !p.charsize = 1.5
      !x.thick = 2
      !y.thick = 2
      red = 255
      green = 255* 256L
      blue = green* 256L
      cyan = blue+ green
      white = red+ green + blue

      degree = 1
      result = poly_fit( x11, y11, degree, yfit)
      degree = 2
      result2 = poly_fit( x11, y11, degree, yfit2)
      print, result
      intercept = result(0)
      slope = result(1)
      a = string( intercept, format = "(f7.2)")
      b = string ( slope, format = "(f6.4)")
      str = '!4k!3 = ' + a + ' + '+ b + '* column'

      wset, 1
      plot, x11, y11
      xx = findgen(512)
      yfit2 = result2(0) + result2(1)* xx + result2(2) * xx^2
      yfit = result(0) + result(1)* xx
      oplot, xx, yfit, color = red, psym = 0
      oplot, xx, yfit2, color = green, psym = 0
      xyouts, 50, 3160, str, size = 1.8

      wset, 16
      radius = 0.8
      circle = 2* !pi * findgen(9)/8
      usersym, radius*sin(circle),  radius* cos(circle), /fill
      plot,  xx,  (yfit2-  yfit)/10,  yrange  =  [-10,  3]/10.,  ytitle  =
'wave(nm)', psym = 0
      set_plot, 'cgm'
      device, /close, file = dir + 'pt_cal_lin_minus_quad_fit.cgm'
      !p.font = 0
      ; loadct, 0
      plot, xx, (yfit2- yfit)/10, yrange = [-10, 3]/10., $
        ytitle = 'wave(nm)', psym = 0, yminor = 1, thick = 2, color = 2

      ;xyouts, 100, -0.8, 'Quadratic Fit Minus Linear Fit'
      device, /close
      set_plot, 'win'

      stop
      ;****** 3 **********************************************
      wset, 6
      nist, x_nist, yy
```

```
set_plot, 'cgm'
device, /close, file = dir + 'Pt_cal_with_NIST_overlay_2.cgm'

;oplot, x_nist, yy*1200/max(yy)
wave = intercept + slope* findgen(512)

line_06 = line_06 - min(line_06)
!x.title  ='Wavelength'
!x.range = [2900,3100]
!y.range = [10,1e3]
radius = 0.6
circle = 2* !pi * findgen(9)/8
usersym, radius*sin(circle),  radius* cos(circle), /fill

n1 = where( x_nist ge 2900)
n1 = min( n1)
n2 = where( x_nist le 3100)
n2 = max(n2)
print, x_nist(n1), x_nist(n2), n2, n1, n2-n1

n1 = where( wave ge 2900)
n1 = min( n1)
n2 = where( wave le 3100)
n2 = max(n2)
print, wave(n1), wave(n2), n2,n1, n2-n1

yy3 =yy*max(line_06)/max(yy)
yy3 = smooth (yy3, 5)

order = 2
yyy = line_06
help, yyy
coef = poly_fit( wave, yyy,  order, l6)
l66 = yyy - l6
plot_io, wave, l66 , psym = 3, /nodata
oplot, x_nist,yy3+10  , psym = 0
oplot, wave, l66+40,  psym =-8, color = 255
;oplot, wave, l6, color =  blue, psym = 0
device, /close
set_plot, 'win'

end

;**********************************************
```

## 2. Data Cube Visualization Program

```
;*************************************************************
; DATA CUBE VISUALIZATION PROGRAM FOR LINUS
;*************************************************************
; Created by Prof R.C. Olsen, 02/2004
; Modifications:
;
; 25-06-2004      Cabezas     Allows to handle the new LabView
;                       ........Hypercube
;
;*************************************************************

dir = 'c:\A_Linus\data\'

cd, dir
files = file_search('*.dat')
file = files(3)     ; change the number in parentheses to get different file

print, file
openr, 1, file

samples = 512     ; Camera pixels
lines   = 10      ; Samples (steps)
bands   = 513     ; Camera pixels plus info line


tmp = intarr( samples, bands)
tmp2 = fltarr( samples, lines)

for i = 0, lines -1 do begin
;forrd, 1, tmp
readu, 1, tmp
tmp = swap_endian(tmp)
tmp2 (*,i) = tmp (*, 253)
endfor

close, 1

window, 0, xsize = samples/2, ysize =lines/2
tmp3 = rebin( tmp2, samples/2, lines/2)
tmp4 =  bytscl( tmp3, min = 0, max = 500)
tv, tmp4

;window, 1, xsize = samples, ysize =lines
;tmp5 = rebin( tmp2, samples, lines)
;tmp6 =  bytscl( tmp5, min = 0, max = 500)
;tvscl, tmp4

h1 = histogram ( tmp3, omin = mini, omax = maxi)
nele = n_elements( h1)
x1 = indgen(nele) + mini
window, 2               ; optional
plot, x1, h1, psym = 10       ;optional
```

```
end
;****************************************************
```

### 3. Calibration Input File Sample

This file is obtained from NIST [ref.7] and is used as an input for the calibration program. The following is a few lines sample.

| Wavelength | Wavenumber | Intensity | Shape | Classification | Code |
|---|---|---|---|---|---|
| 2846.34 | 35122.5 | 34 | | | |
| 2846.52 | 35120.3 | 33 | Pt II | 106434-71314 | K |
| 2846.86 | 35116.1 | 120 | | | |
| 2848.32 | 35098.1 | 18 | | | |
| 2849.15 | 35087.9 | 150 | Pt I | 16983-52071 | N |
| 2849.94 | 35078.1 | 31 | | | |
| 2850.41 | 35072.4 | 110 | Pt II | 110257-75184 | K |
| 2850.6 | 35070 | 100 | Pt II | 106434-71364 | K |
| 2851.16 | 35063.1 | 78 | Ne III | | L |
| 2851.23 | 35062.3 | 43 | | | |
| 2852.1238 | 35051.293 | 0 | Mg I | | |
| 2852.87 | 35042.1 | 37 | Pt II | 110408-75365 | K |
| 2853.0972 | 35039.335 | 3800 | Pt I | 13496-48535 | E |
| 2853.3729 | 35035.95 | 810 | Pt I | 68716-33680 | N |
| 2853.5092 | 35034.275 | 510 | | | |

| | | | | | |
|---|---|---|---|---|---|
| 2853.84 | 35030.2 | 190 | | | |
| 2854.14 | 35026.5 | 52 | | | |
| 2855.79 | 35006.3 | 74 | | | |
| 2858.0244 | 34978.931 | 2200 | Ne II | | G |

## B. SO2 CALIBRATION CODE

This code is used to calibrate the instrument to the SO2 response.

### 1. SO2 Calibration Program.

```
      ;************************************************************
      ; CALIBRATION PROGRAM FOR LINUS
      ;************************************************************
      ; Created by Prof R.C. Olsen, 09/2002
      ; Modifications:
      ; 03-06-2004
      ; 25-06-2004      Cabezas     Allows to handle the new LabView
      ;                       ........Hypercube
      ;
      ;************************************************************


      dir1 = 'C:\A_Linus\data\'
      cd, 'C:\A_Linus\data'

      vfiles  =  findfile(  dir1  +  '0608_deut_v*.dat',  count=count_v)
;count1
      sfiles  =  findfile(  dir1  +  '0608_deut_s*.dat',  count=count_s)
;count2

      print, count_s
      print, count_v

      v_size = file_info(vfiles)


      index = sort( vfiles)
      vfiles = vfiles(index)
      index = sort( sfiles)
      sfiles = sfiles(index)

      samples = 512     ; Camera pixels
      bands   = 513     ; Camera pixels plus info line

      lines = v_size.size / samples
      lines = lines / bands
```

```
      lines = lines / 2 ;lines   = 30     ;  (steps)
      print, v_size.size

      print, lines
      data = intarr( samples, bands, lines)

      window, 0, xsize = 512, ysize = 512

      data_cube_v = uintarr( samples, bands-1, lines, count_v)
      data_cube_s = uintarr( samples, bands-1, lines, count_s)

      for ifile = 0, count_v-1 do begin
       file = vfiles(ifile)
       openr, 1, file
        readu, 1, data
       close,1
        data_cube_v(*,*,*, ifile) = data(*,1:512,*)
      ;tv, bytscl(data, min = 40, max = 500), order = 1
       print, ifile,' ', file,' ', min(data), max(data)
      ; xyouts, 10, 490, string(ifile), color = 255, /device
      endfor
      stop
      for ifile = 0, count_s-1 do begin
       file = sfiles(ifile)
       openr, 1, file
       readu, 1, data
       close,1
        data_cube_s(*,*,*, ifile) = data(*,1:512,*)
      tv, bytscl(data, min = 40, max = 500), order = 1
       print, ifile,' ', file,' ', min(data), max(data)
       ;xyouts, 10, 490, string(ifile), color = 255, /device
      endfor
      stop

      ;*****
      for j = 0, count_v -1 do begin
            for i = 0, lines -1 do begin
                  data_cube_v(*,*,i,j)                                =
swap_endian(data_cube_v(*,*,i,j))
            endfor
      endfor
      ;*****
      ;*****
      for j = 0, count_s -1 do begin
            for i = 0, lines -1 do begin
                  data_cube_s(*,*,i,                j)                =
swap_endian(data_cube_s(*,*,i,j))
            endfor
      endfor
      ;*****


      v500    =    total(data_cube_v(    *,*,    *,    0),3)/lines    ;    +
data_cube_v(*,*,1)   ) /2
      s500    =    total(data_cube_s(    *,*,    *,    0),3)/lines    ;    +
data_cube_s(*,*,22) ) /2
      s501 = total(data_cube_s( *,*, *, 1),3)/lines
```

```
      window, 1, xsize = 512, ysize = 512, title = 'SO2'
      tv, bytscl( s500, min = 40, max = 500)
      stop
      window, 1, xsize = 512, ysize = 512, title = 'Vacuum'
      tv, bytscl( v500, min = 40, max = 500)

      rat = float(s500) /float(v500)
      rat1 = float(s501) /float(v500)
      window, 2, xsize = 512, ysize = 512, title = 'Ratio1'
      tv, bytscl( rat, min = 0.5, max = 1.1)
      stop
      window, 2, xsize = 512, ysize = 512, title = 'Ratio2'
      tv, bytscl( rat, min = 0.5, max = 1.1)
      wset, 0
      stop
      sum = reverse(total( rat, 1))
      sum1 = reverse(total( rat1, 1))
      ;sum2 = total(rat,2)
      plot, sum/512,  /ynozero , title ='This is dimension 1'
      stop
      window, 3
      plot,  sum/512,  /ynozero,  color=red,  title  ='This  is  dimension
12';/ynozero
      oplot, sum1/512, color = 255
      stop

      ;window, 4
      ;plot, sum2/512, /ynozero, color=red, title ='this is dim 2'
      ; 10 sets of measure

      stop
      rat_sum = fltarr( 512, count_s)
      for i = 0, count_s-1 do begin
            s =  (data_cube_s( *,*,*, i)) ;+ data_cube_s(*,*,i+1) ) /2
      ;v =  (data_cube_v( *,*, i) + data_cube_v(*,*,i+1) ) /2
      ;rat = float( s) /float(v)
            rat = float( s) / float(v500)

            window, 2 , xsize = 512, ysize = 512, title = 'Ratio'
            tv, bytscl( rat, min = 0.5, max = 1.1)

            wset, 0
            sum = reverse(total( rat, 1));**  <-----------   dimension
1
            plot, sum/512., yrange = [0.6, 1.1]
            rat_sum (*, i) = sum(*)/512

      endfor

      stop
      wset, 2

      tv, bytscl( congrid( rat_sum, 512,128), min = 0.7, max = 1)

      window, 1
      x = [ 30, 50, 75, 100, 150, 200, 250, 300, 350, 400]
      radius = 0.7
```

```
      circle = 2*!pi*findgen(9)/8
      usersym, radius*sin(circle), radius*cos(circle), /fill

      plot,  x,  100*rat_sum  (  256,  *),  /ynozero,  color=red,  title  =
'Curve of Growth - Centerline', $
        xtitle  =  'SO!d2!n  Gas  Pressure  (mm  Hg)',  ytitle  =  'Percent
Absorption',  $
         psym = -8

      wshow
      stop
      ;ints = total ( rat_sum, 1)  &  help, ints
      ;plot, x, ints/512, /ynozero, title = 'Curve of Growth', $
        ;xtitle  =  'SO!d2!n  Gas  Pressure  (mm  Hg)',  ytitle  =  'Percent
Absorption',  $
         ;psym = -8
      ;***************************************************************
****
      ;openr, 1, 'f:\linus\wave.dat'
      ;i = 1
      ;f = 2.
      wave = fltarr(512)
      ;for ii = 0, 511 do begin
      ;readf, 1, i, f
      ;wave(ii) = f
      ;endfor
      ;close, 1

      intercept=2776.58
      slope=0.8299
      wave = intercept + slope* findgen(512)
      ;***************************************************************
************

      set_plot, 'win'

      ;plot,  wave,  reverse(1000*rat_sum  (  *,  9)),  /ynozero,  title  =
'Calibration', $
      ;  xtitle = 'Wavelength', ytitle = 'Percent Absorption',  $
      ;   psym = -8, xrange = [ 2800, 3200]
      plot, wave, (100*rat_sum ( *, 0)), title = 'Calibration', $
        xtitle = 'Wavelength', ytitle = 'Percent Absorption',  $
         psym = -8, xrange = [ 2800, 3200],/nodata
      stop
      for i = 0, count_s-1 do begin
      y = 100*rat_sum(*, i)
      width = 9
      y = smooth( y, width)
      oplot, wave, (y); y
      endfor

      stop;last
      ;plot,   wave,   100*rat_sum   (   *,   0),   /ynozero,   title   =
'Calibration', $
      plot, wave, 100*rat_sum ( *, 0), /ynozero, title = 'Calibration',
$
        xtitle = 'Wavelength', ytitle = 'Percent Absorption',  $
```

```
   psym = -8, xrange = [ 3060, 3140]
oplot, wave, 100*rat_sum ( *, 0), color = 255

end

;*************************************
```

# APPENDIX C: ENVI PROCEDURE

The ENVI procedure is a way to visualize the hypercube data.

From the ENVI menu, choose File, Open Image File and select the desired .DAT data file; then insert into the following popup menu the parameters.

Parameters to see (X,Y) coordinates:

| | |
|---|---|
| Samples | 512 |
| Lines | (Number of samples) i.e: 400 |
| Bands | 513 |
| Data Type | Unsigned int |
| Byte order | Network (IEEE) |
| File type | ENVI Standard |
| Interleave | BIL |

Parameters to see (Y, λ) coordinates:

| | |
|---|---|
| Samples | 512 |
| Lines | 513 |
| Bands | (Number of samples) i.e: 400 |
| Data Type | Unsigned int |
| Byte order | Network (IEEE) |
| File type | ENVI Standard |
| Interleave | BSQ |

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF REFERENCES

[1] D.S. Davis, R.M. Harkins, R.C. Olsen, *The LINUS UV Imaging Spectrometer*, Proceedings of SPIE 5093, pp 748-757, 2003.

[2] National Instruments, *FW7344 Motion Controller User Guide*, August 2001

[3] National Instruments, *MID-7654/7652 Servo Power Motor Drive User Guide*, June 2001.

[4] Ogata, Katsuhiko, *Modern Control engineering,* 4th ed., Prentice Hall, 2002.

[5] National Instruments, *Servo tuning procedure.*

[6] Omega Optical Company, *300 nm Filter data sheet*, September 2000.

[7] NIST, http://physics.nist.gov/PhysRefData/platinum/graphs/2990-3006.html

[8] J. Gray, *Design And First Operations Of The Lineate Imaging Near-Ultraviolet Spectrometer (Linus,),* MS Thesis, Naval Postgraduate School, December 2002.

[9] Sony, *CCD Black and White video Camera Module manual*, p/n 3-867-357-04.

[10] National Instruments*, IMAQ PCI/PXI-1411 User Manual*, November 2000.

[11] Hooks, T.A., *Design construction and operation of the NPS Ultraviolet Imaging Spectrometer (NUVIS*), MS Thesis, Naval Postgraduate School, 1997.

[12] Kompatzki, R. C. *Design and development of the image scanner for lineate imaging near ultraviolet spectrometer (LINUS),* MS Thesis, Naval Postgraduate School, December 2001.

[13] Blackwell, Jerry S., *Experimental Determination Of Paraxial Ray Transfer Matrices And Cardinal Points Of Complex Optical Systems By Means Of Finite Conjugate Imaging,* MS Thesis, Naval Postgraduate School, December 2001.

THIS PAGE INTENTIONALLY LEFT BLANK

# INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. David S. Davis, Code PH DV
   Naval Postgraduate School
   Monterey, California

4. Richard Harkins, Code PH RH
   Naval Postgraduate School
   Monterey, California

5. James Luscombe, Code PH
   Naval Postgraduate School
   Monterey, California

6. Rodrigo Cabezas
   Chilean Navy
   Valparaiso, Chile