NTNU

Norwegian University of
Science and Technology

Development of a spectrometer system to remotely sense mesospheric temperature.

Frank Terje Berge

Physics
Submission date: September 2011
Supervisor: Patrick Joseph Espy, IFY

Norwegian University of Science and Technology
Department of Physics

Development of a spectrometer system to remotely sense mesospheric temperature

Frank Terje Berge

September 30, 2011

Acknowledgements

This report is the product of FY3900, Master Thesis in Physics, as a partial fulfillment of the requirements to achieve a Masters degree at the Institute of Physics, NTNU. I would like to thank Prof. Patrick J. Espy for all his help and assistance. I would also like to thank Amund Gjendem for all his practical assistance and Thomas Korneliussen for proofreading this thesis.

Abstract

At nighttime the strongest source of radiation is the hydroxyl nightglow. By measuring the different lines within a band it is possible to calculate the temperature of the mesosphere more than 85km above ground. In order to do this a spectrometer system has been calibrated and software has been developed. The software include a program to control the spectrometer and automatically take data spectra through the night, and a program to process the result and calculate the temperature. Two measurements were done, the (3,1) and (4,2) bands the night of 31. march 2011 and the (7,4) and (8,5) bands the night of 6. may 2011. Both of them were done at NTNU Trondheim (latitude: 63.4 degrees and longitude: 10.4 degrees). The temperature found was 201.85K +/− 3.55K for the (3,1) band, 205.11K +/− 17.94K for the (4,2) band, 199.63K +/− 29.86K for the (7,4) band and 196.37K +/− 8.41K for the (8,5) band. Since the (7,4) and (8,5) bands were measured later than the (3,1) and (4,2) bands, they should be colder. The predicted temperature was also calculated using a program developed at The British Antarctic Survey that uses the MSIS-E-00 [Picone et al., 2003] model for the background atmosphere. The predicted temperatures were 202.3K for the (3,1) band, 201.7K for the (4,2) band, 176.5K for the (7,4) band and 176.4K for the (8,5) band. The predicted temperature and measured temperature are almost the same for the (3,1) and (4,2) bands while it is about 20-25K lower for the (7,4) and (8,5) bands.

Contents

1. Introduction

It was Meinel who first discovered the upper atmospheric nightglow of hydroxyl. This radiation is caused when hydroxyl molecules cascades to lower vibrational levels. The brightest emissions are those with $\Delta v = 2$, which means that the molecule cascades down two vibrational levels. The OH-layer is found at about 87±3 km above ground, and has a half-width of 5-8km [Baker and Stair, 1988]. Here ozone and atomic hydrogen react and the result is hydroxyl and O_2. The hydroxyl resulting from this reaction can only have high vibrational levels, v=6 to v=9, with maximum population at v=8. As the hydroxyl molecules cascade to lower vibrational levels light is emitted at wavelengths from $0.4\mu m$ to $4\mu m$. This is the strongest radiation from the night sky and most can be observed from the ground. Since Meinel's discovery many researchers have made ground based, air and rocket borne observations of the intensities of individual OH vibrational-rotational bands. One result of these studies came when Krassovsky [Krassovsky, 1972] discovered that the OH emission was not spatially uniform. He found wave-like variations in the band intensity with a period of about one hour. This led to the discovery of large scale gravity waves in the upper atmosphere.

1.1. Temperature of the OH-layer

Briefly each transition from a vibrational level is accompanied by simultaneous transitions between closely spaced rotational levels (as detailed in section 2.3). Since these levels are closely spaced, collisions between molecules populate the levels according to a Boltzmann distribution. The relative intensity of the rotational lines is given by:

$$I_J = S_J N_0 \cdot \nu^3 e^{-\frac{\epsilon_J}{kT}} \ , \tag{1.1}$$

where I_J is the relative intensity, S_J is the line strength, N_0 is the population and ϵ_J is the energy of the initial state. This equation can be rewritten as:

$$ln(\frac{I_J}{S_J}) = -\frac{\epsilon_J}{kT} + ln(N_0) \ . \tag{1.2}$$

Note that we assume ν to be constant over a band and that the throughout this thesis the term intensity is used generically, and in most cases the correct term would be photon radiant flux. From eq.1.2 we see that if we plot $ln(\frac{I_J}{S_J})$ against $\frac{\epsilon_J}{k}$ and fit a line through those points, the slope of that curve will be $\frac{-1}{T}$. This gives us a unique way of measuring the temperature of the mesosphere at 90km above ground. In order to measure the intensity of the hydroxyl rotational lines we have used a spectrometer and let it acquire data through the night, then averaged the data and used eq.1.2 to find the temperature.

The vibrational-rotational lines that have been used are the (3,1), (4,2), (7,4) and (8,5) bands. The (3,1) and (4,2) bands are the brightest bands, but are affected by H_2O absorption and a rapidly changing instrument sensitivity. Therefore the (7,4) and (8,5) have also been used in order to improve the results.

1.2. Progress of this text

First the theory is presented in the next chapter. Here the origin of the hydroxyl glow is briefly explained along with the quantum physics which causes it. In the end the calculations used are shown. Chapter 3 gives the spesifications of the equipment and shows how the instrument was calibrated. The next chapter explains the software that was developed to make it easier for the reader to understand what the software does. Chapter 5 show the results of the measurements and the predicted temperature using software made by The Britsh Antarctic Survey. The last chapter is the conclusion.

2. Theory

In order to understand the method used to calculate the temperature of the OH in the mesosphere, a basic understanding of the quantum mechanical transitions is needed. First the atmospheric temperature profile is presented, along with the OH-layer and the chemical reactions that involve OH. Then the quantum mechanics of a diatomic molecule is briefly explained. After which comes a small section about the Meinel lines and their nomenclature. Finally the calculation method is given.

2.1. The Atmospheric Structure

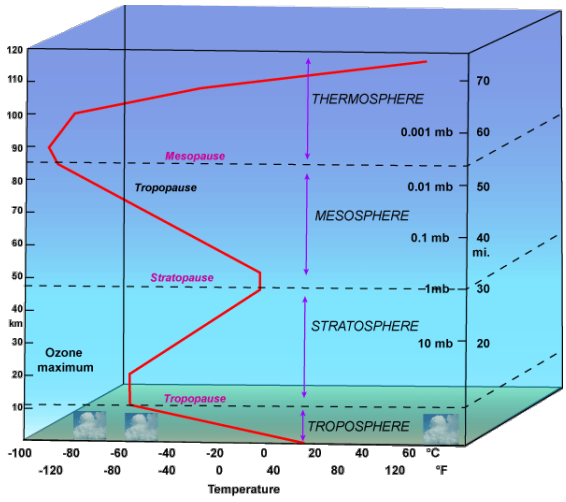

Figure 2.1.: Typical vertical structure of atmospheric temperature. Source:[web, b]

Figure 2.1 shows the structure of the atmosphere and typical temperatures of the different layers. The atmosphere is divided into several horizontal layers based on the temperature gradients. The first layer, the troposphere, starts at ground level and is bounded by the tropospause at about 15km. The temperature gradient here is negative as the ground is heated by sunlight and it gets cooler as you move away from it. Above the tropospause is the stratosphere which is bounded by the stratospause at about 45km above ground. In the stratosphere the temperature gradient is positive. This is caused by the heating associated with the absorption of solar ultraviolet radiation by ozone. The ozone layer is widely spread over the atmosphere from ground level to about 70km above

ground [Andrews, 2000], but the concentration peaks at approximately 25km. Above the stratosphere is the mesosphere which is bounded by the mesopause at 90km. In the mesosphere the temperature gradient is negative again. This is because of decreased solar heating and increased radiative cooling to space. The outer layer of the atmosphere is the thermosphere where the temperature gradient is again positive. This is caused by dissociation and ionization of O_2 and N_2 by solar ultraviolet light. The pressure here is so low that due to the low collision frequency, the dissociated atoms do not thermalize and their high velocity is interpreted as high temperature. The temperatures here can therefore reach as high as 2000K, but because of the low pressure there is little conduction of heat.

2.2. The OH-layer

The night time OH-layer has a typical half-width of 5-8km and is centered at a mean altitude of 87±3 km [Baker and Stair, 1988]. The exited state of hydroxyl can be produced by this reaction [Marsh and Smith, 2006]:

$$H + O_3 \rightarrow OH^* + O_2 \tag{2.1}$$

Since the production of OH^* is directly proportional to how much it radiates, it is also possible to infer the highly variable atomic oxygen density by measuring the intensity of the hydroxyl airglow. Near the mesopause the ozone concentration is dependent on the balance between the production of ozone through recombination of atomic oxygen and the destruction of ozone through reactions with hydrogen [Marsh and Smith, 2006],

$$[O_3] = \frac{k_{O+O_2+M}[O][O_2][M]}{k_{H+O_3}[H]} \tag{2.2}$$

However this is only true at night since at daytime the rate of photo-dissociation of ozone is much bigger than the rate of reaction 2.1. The production rate of exited hydroxyl is [Marsh and Smith, 2006],

$$P(OH^*) = k_{O_3+H}[O_3][H] \tag{2.3}$$

By substitution of eq.2.2 into eq.2.3 we get,

$$P(OH^*) = k_{O+O_2+M}[O][O_2][M] \tag{2.4}$$

From eq.2.4 it's clear that the production of hydroxyl is proportional with the atomic oxygen and the diatomic oxygen concentrations. Atomic oxygen is mainly produced by destruction of ozone [Dikty et al., 2010],

$$O_3 + h\nu \rightarrow O_2 + O \tag{2.5}$$

which requires light. Therefore the production of atomic oxygen only occures at daytime, but recombination time are long enough that the atomic oxygen does not vanish during the night (except at lower the lowest altitudes).

2.3. Diatomic Radicals

This section is a short introduction to the quantum mechanics of diatomic radicals and is based on [Herzberg, 1971]. When a radical is split off a parent molecule, it often has one or more unpaired electrons, which means it has nonzero spin. This has caused many authors to define a free radical as a system with nonzero spin. This is a simple definition, but it also mean that chemically stable molecules such as O_2, NO, NO_2, ClO_2 could be considered as free radicals while highly reactive and short-lived molecules such as $C_2, C_3, CH_2CHF, CF_2, HNO$ in their singlet states (S=0) would not be free radicals. Therefore many scientists use a looser definition of free radicals: a free radical is any species that has a short lifetime in the gaseous phase under ordinary laboratory conditions.
The energy of a molecule can be approximated as the sum of three parts,

$$E = E_e + E_v + E_r \tag{2.6}$$

where E_e is the electronic energy, E_v is the vibrational energy and E_r is the rotational energy. Since these energies are widely spread, with $E_e >> E_v >> E_r$, the frequencies, and therefore the time scales of the motions are very different. Thus we may treat each separately.

2.3.1. Rotation

The $dumbbell - model$ is a simple model for the rotation of a diatomic molecule. In this model the diatomic molecule is described as two points of masses m_1 and m_2 with a massless rod of length r between them. If we use classical mechanics the energy of such a system would be:

$$E_r = \frac{1}{2} I\omega^2 \ , \tag{2.7}$$

where I is the moment of inertia and ω is the angular velocity. The moment of inertia with respect to the centre of mass is:

$$I = m_1 r_1^2 + m_2 r_2^2 = \mu r^2 \ , \tag{2.8}$$

where r_1 and r_2 is the distance from the centre of mass to the two masses and $r = r_1 + r_2$. μ is the reduced mass given by

$$\mu = \frac{m_1 m_2}{m_1 + m_2} \ . \tag{2.9}$$

Since the angular momentum of the system is:

$$P = I\omega \ , \tag{2.10}$$

the rotation energy can also be written as:

$$E_r = \frac{P^2}{2I} \ . \tag{2.11}$$

However in wave mechanics the rotational energy can only take certain descrete values. These can be obtained by solving Schrödinger equation of a rigid rotator. The result is:

$$E_r = \frac{h^2}{8\pi^2\mu r^2} J(J+1) \ , \tag{2.12}$$

where J is the rotational quantum number, which can take the values 0,1,2,.... It is usual to write eq.2.12 in the form of term values, measured in cm^{-1}. These term values are the energy values divided by hc, or:

$$F(J) = \frac{E_r}{hc} = BJ(J+1) \ , \tag{2.13}$$

where B is the rotational constant,

$$B = \frac{h}{8\pi^2 c\mu r^2} \quad [cm^{-1}] \ . \tag{2.14}$$

Because of the centrifugal force, if the molecule is not completely rigid there will be a slight change in the internuclear distance when the molecule is rotating. This causes a small correction term:

$$F(J) = BJ(J+1) + DJ^2(J+1)^2 + ... \ , \tag{2.15}$$

where D is approximated by:

$$D = \frac{4B^3}{\omega^2} \ . \tag{2.16}$$

Normally this correction term can be ignored since D is much smaller than B, but this is not true for hydroxyl.

2.3.2. Vibration

The vibrations of a diatomic molecule can be approximately represented by the model of the harmonic oscillator. This is a mechanical system of a moveable object connected to a fixed position by a spring. The moveable object is then subject to a restoring force proportional to the positional deviation from the equilibrium position. The motion of the two nuclei in the diatomic molecule can be reduced to the motion of a single particle with μ as mass, where μ the reduced mass from eq.2.9. The displacement, x, of the ocsillator is in this case $r - r_e$ where r_e here is the equilibrium position and r is the distance between the nuclei. The potential energy is then:

$$V = \frac{1}{2}kx^2 = \frac{1}{2}k(r - r_e)^2 \ , \tag{2.17}$$

where k is the force constant which is given by:

$$k = 4\pi^2\mu\nu_{osc}^2 \ . \tag{2.18}$$

Substituting put eq.2.17 into the wave equation you get:

$$E_v = h\nu_{osc}(v + \frac{1}{2}) \ , \tag{2.19}$$

where v is the vibrational quantum number and can take the values 0,1,2,....The vibrational energy is then, in term values:

$$G(v) = \frac{E_v}{hc} = \omega(v + \frac{1}{2}) \ , \tag{2.20}$$

where ω is the vibrational frequency given by:

$$\omega = \frac{\nu_{osc}}{c} \quad [cm^{-1}] \ . \tag{2.21}$$

If we look at eq.2.20 we see that the energy when v=0 is not zero, but $\frac{1}{2}\omega$. The molecule is not strictly a harmonic oscillator, but an anharmonic oscillator. That means the potential function is not a parabola, but a curve that can be approximated by the Morse function given by:

$$V = D_e[1 - e^{-\beta(r-r_e)}]^2 \ , \tag{2.22}$$

where β is:

$$\beta = \sqrt{\frac{2\pi^2 c\mu}{D_e h}} \ . \tag{2.23}$$

Using this potential function in the wave function we get:

$$G_0(v) = \omega_0 v - \omega_0 x_0 v^2 + \omega_0 y_0 v^3 - ... \ . \tag{2.24}$$

Here we have taken the lowest level (v=0) as the point of zero energy and not the the lowest potential energy. Generally, $\omega_0 x_0$ and $\omega_0 y_0$ are much smaller than ω_0, as a result the spacing between the levels will gradually decrease unlike the harmonic oscillator where it's constant.
Since rotation and vibration of the molecule will take place at the same time, the rotational constant will change with different vibrational levels. This is because the rotational constant is dependent on r, the distance between the atoms. It can be shown that the rotational constant on different vibrational levels is given by:

$$B(v) = B_e - \alpha_e(v + \frac{1}{2}) + ... \ , \tag{2.25}$$

where B_e is the rotational constant at equilibrium position and α_e is a constant.

2.3.3. Rotation-vibration spectra

If you consider a transition from one vibrational level to another vibrational level, and at the same time includes transitions from one rotational state to another we get a rotation-vibration band. The OH-Meinel bands are examples of such bands. A energy-level diagram can be seen in fig.2.2. To get the wave numbers of the lines of such bands one must add together the term values of the rotation and the vibration. The vibrational part is:

$$\nu_v = G(v') - G(v'') \ , \tag{2.26}$$

where v' and v'' are the vibration levels of the upper and lower state. The rotational part is:

$$\nu_r = F_{v'}(J') - F_{v''}(J'') \ . \tag{2.27}$$

By adding the vibrational part and the rotational part and substituting in eq.2.13 we get:

$$\nu = \nu_v + B'J'(J'+1) - B''J''(J''+1) \ , \tag{2.28}$$

where B' and B'' are the rotational constants of the upper and lower state and J' and J'' are the corresponding rotational quantum numbers. According to the selection rule, $\Delta J = J' - J'' = 0, \pm 1$. If $\Delta J = 0$ we get the Q-branch and by renaming $J' = J''$ as J we get:

$$\nu_q = \nu_v + (B' - B'')J + (B' - B'')J^2 \ . \tag{2.29}$$

If $\Delta J = 1$ we get the R-branch and by renaming J'' as J and $J' = J + 1$ we get:

$$\nu_r = \nu_v + 2B' + (3B' - B'')J + (B' - B'')J^2 \ . \tag{2.30}$$

If $\Delta J = -1$ we get the P-branch and we do the same renaming, only this time $J' = J - 1$, and we get:

$$\nu_p = \nu_v - (B' + B'')J + (B' - B'')J^2 \ . \tag{2.31}$$

From these equations we see that if B">B', then ν_q will decrease with J though very slowly. The R-branch wavenumber ν_r will first increase with J until J^2 becomes large enough and then it will start to decrease. Finally ν_p will always decrease with J. The dumbbell-model does not take the electrons into account. The correction for this problem leads to splitting of the rotational levels which depends on the angular momentum Λ, which can take two values, $\frac{1}{2}$ or $\frac{3}{2}$.
In 1950 Meinel observed the night sky for the first time with moderate resolution in the infrared and found what he believed was a new electronic band system for OH. It was soon discovered that it was in fact part of the rotation-vibration spectrum of OH.

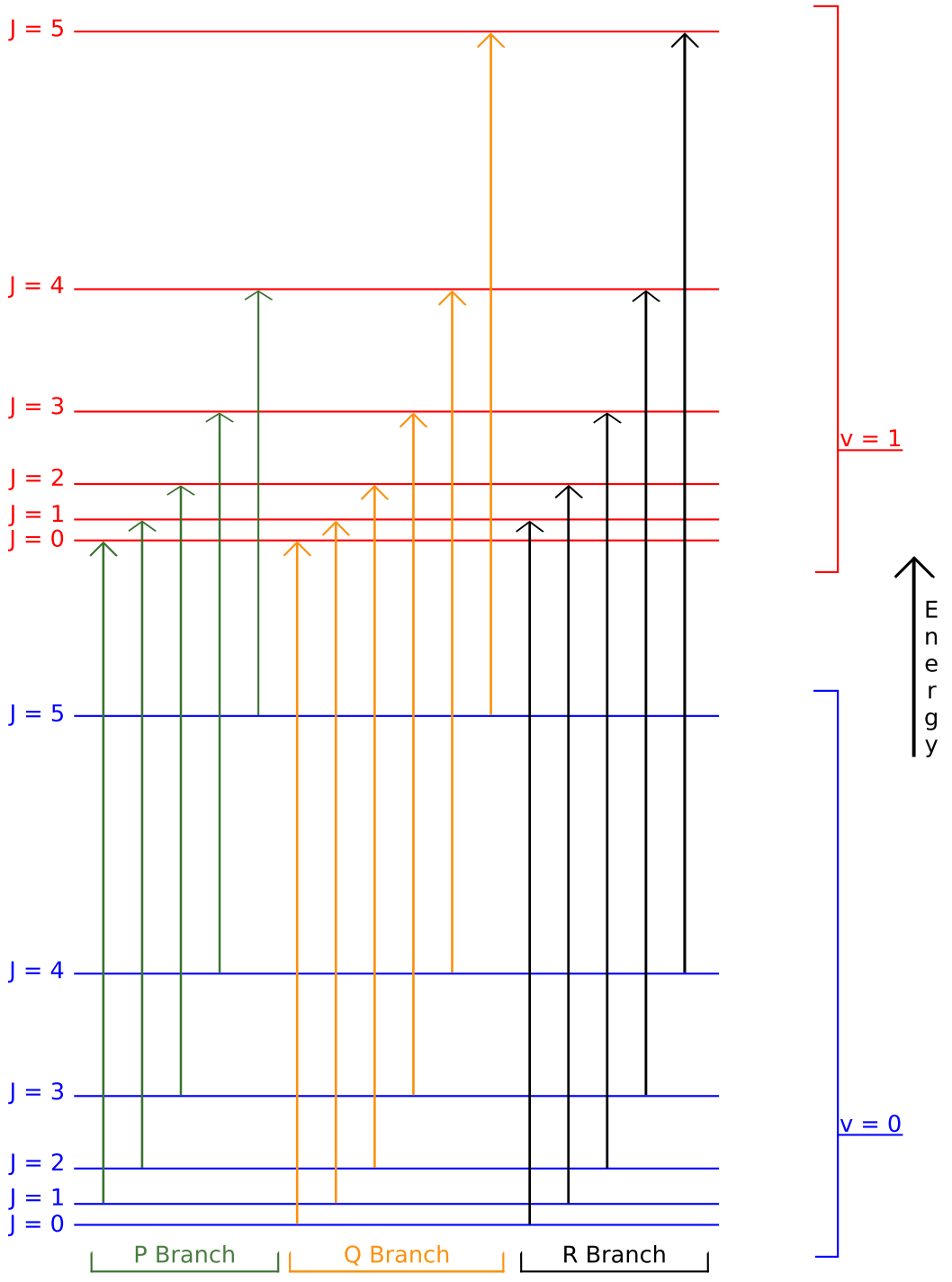

Figure 2.2.: Energy level diagram showing possible rotational transitions for a v=0 to v=1 vibrational transition. Note that this is caused by absorption not emission. Source:[web, e]

2.4. Meinel Lines

Reaction 2.1 gives off 3.30eV, which is just enough energy to excite the ninth vibrational level (v=9)[Herzberg, 1971][Kvifte, 1961]. Labratory investigations show that reaction 2.1 in fact gives the OH radical in vibrational state v=6 to v=9 [Sivjee, 1991]. An OH radical can cascade down to a lower vibrational state and at the same time change rotational state. A photon is then released whose wavelength depends on the energy reduction of the OH radical. Lower vibrational states of OH are also believed to be caused by collisional quenching. Its uncertain how this happens, but two ways are proposed [Sivjee, 1991],

$$OH^*(v') + M \rightarrow OH^*(v'-1) + M \tag{2.32a}$$

$$OH^*(v') + M \rightarrow OH^*(v'=0) + M \tag{2.32b}$$

where M is O_2, N_2 or O. Eq. 2.32a gives a stepwise reduction of vibration level while eq.2.32b gives a "sudden death" where it instantly reaches vibrational level zero.
Figure 2.3 shows a typical spectra of the OH Meinel(3,1) and (4,2) bands. The (3,1) notation means its a transition from vibrational state v=3 to v=1, and similary (4,2) is a transition from v=4 to v=2. Each has three rotational branches, the P-branch, the Q-branch and the R-branch. The different lines of the R-branch are impossible to make out with our resolution, as the energy of the rotational transitions first increases with increasing J numbers and then starts to decrease. This causes the lines to overlap, as we can see in figure 2.3. The energy of the rotational transitions in the Q-branch is almost the same. There is a small change in energy since B changes (eq.2.25), but with our resolution its just one big line. In the P-branch however the lines are easy to seperate, until they become blended with the R-branch of the next vibrational-rotational band. The notation for each line is $P_{1,2}(J'')$. Since this is the P-branch, $\Delta J = J' - J'' = -1$ which means $J'' = J' + 1$. Since J=0 does not exist in a Π state, the first line in the P-branch is $P_2(2)$, and since $J' = 1$ that makes $J'' = 2$. The subscript in $P_{1,2}(J'')$ is either 1 or 2 and refers to the ${}^2\Pi_{\frac{3}{2}}$ and ${}^2\Pi_{\frac{1}{2}}$ states. Its the doublet nature of the ground state (${}^2\Pi$), which causes each band to consist of two subbands. $P_1(J'')$ which refers to the ${}^2\Pi_{\frac{3}{2}}$ is the strongest as can be seen in figure 2.3. This is because the energy levels in the ${}^2\Pi_{\frac{3}{2}}$ state are lower, which means it has a higher radiating population. Studying these lines we see why the OH radical is so useful. If we look at the rotational and vibrational constants of the OH radical [Herzberg, 1971],

B_e	18.867 cm^{-1}
α_e	0.708
γ_e	0.00207
ω_e	3739.94
$\omega_e x_e$	86.350
r_e	0.9707 $\mathring{A}$

Here we see that B_e is large compared to the rotational constant of other diatomic molecules like O_2, NO and N_2, which are around $2cm^{-1}$. This causes the lines to spread out and make them much easier to study.

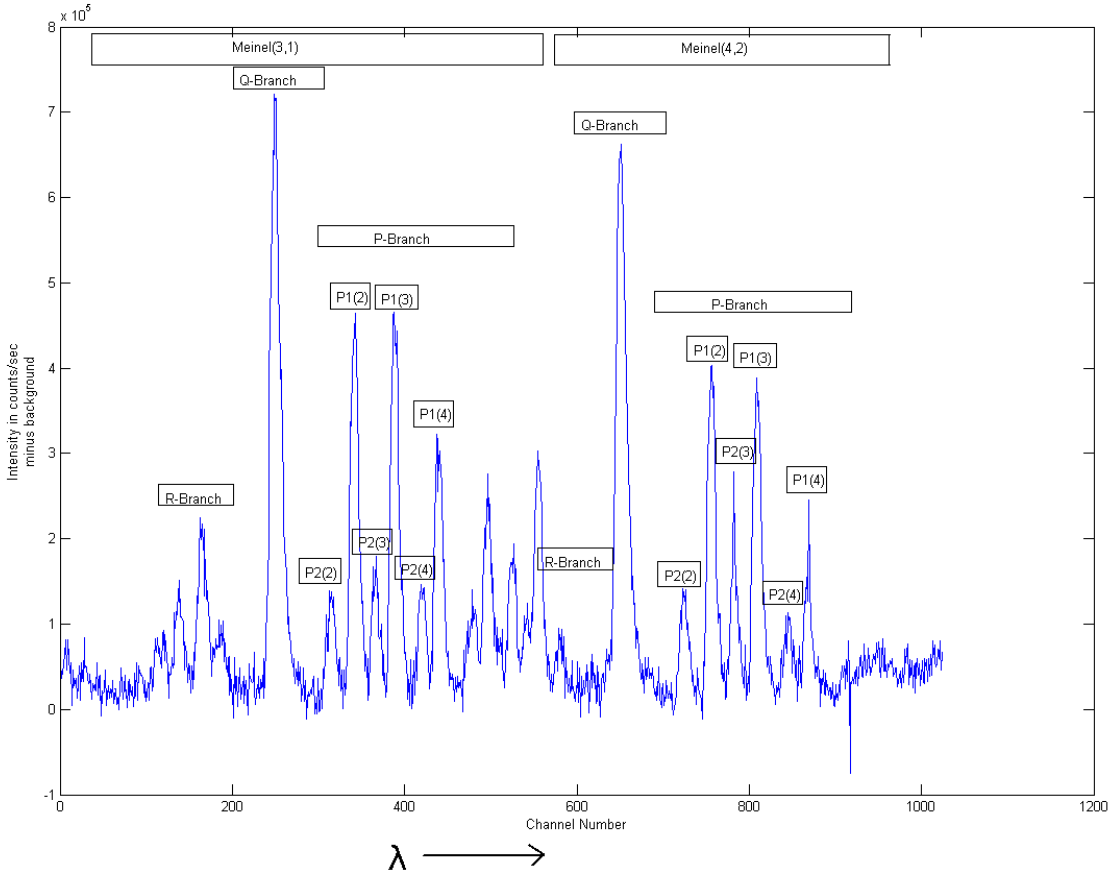

Figure 2.3.: A scan of the nightsky over Trondheim taken 31.march. The x-axis show channel number, the wavelengths are approximatly from 1450nm to 1650nm. The textboxes shows the different bands, branches and lines

2.5. Temperature of the OH-layer

The equation used for calculating the temperature is:

$$I_J = S_J N_0 \cdot \nu^3 e^{-\frac{\epsilon_J}{kT}} \ , \tag{2.33}$$

where I_J is the measured intensity, S_J is the line strength, N_0 is the total population, ϵ_J is the energy of the initial state, ν is frequency, and k is Boltzman's constant. If you take the natural logarithm of eq.2.33 you get,

$$ln(I_J) = ln(S_J \cdot \nu^3) + ln(N_0) - \frac{\epsilon_J}{kT} \tag{2.34}$$

We can rewrite eq.2.34 as, if we assume ν to be constant over a band:

$$ln(\frac{I_J}{S_J}) = -\frac{\epsilon_J}{kT} + ln(N_0) \ . \tag{2.35}$$

Which means if we integrate over each line in the P-branch and divide by the corresponding line strength, then take the natural logarithm of that and plot against $\frac{\epsilon_J}{k}$, the slope of the fitted curve through those points will be $\frac{-1}{T}$. The calculation of the line strengths and the energy at the initial state was done by Patrick Espy using a computer program made by E.E. Whiting[Whiting, 1973] and is described in detail in [Espy, 1986]. The numbers used for the line strength and the initial energy is given in the table below,

Band	(3,1)		(4,2)	
	$\epsilon_J\ [cm^{-1}]$	S_J	$\epsilon_J\ [cm^{-1}]$	S_J
$P_2(2)$	10300.41	3.9839e+011	13377.61	3.4350e+011
$P_1(2)$	10172.30	4.9798e+011	13248.92	4.2855e+011
$P_2(3)$	10354.21	7.3932e+011	13429.00	6.3772e+011
$P_1(3)$	10247.07	9.0706e+011	13320.76	7.8114e+011
$P_2(4)$	10443.29	1.0852e+012	13514.12	9.3671e+011
$P_1(4)$	10352.45	1.2943e+012	13421.92	1.1156e+012
$P_2(5)$	10567.02	1.4404e+012	13632.41	1.2443e+012
$P_1(5)$	10488.78	1.6789e+012	13552.73	1.4469e+012
$P_2(6)$	10724.72	1.8041e+012	13783.23	1.5599e+012
$P_1(6)$	10656.31	2.0624e+012	13713.39	1.7795e+012
Band	(7,4)		(8,5)	
	$\epsilon_J\ [cm^{-1}]$	S_J	$\epsilon_J\ [cm^{-1}]$	S_J
$P_2(2)$	21628.20	7.5506e+011	24042.11	6.2772e+011
$P_1(2)$	21497.99	9.3453e+011	23911.58	7.7574e+011
$P_2(3)$	21672.33	1.3866e+012	24083.74	1.1537e+012
$P_1(3)$	21560.82	1.6907e+012	23971.24	1.4043e+012
$P_2(4)$	21745.52	2.0180e+012	24152.80	1.6805e+012
$P_1(4)$	21649.13	2.3977e+012	24055.03	1.9934e+012
$P_2(5)$	21847.34	2.6601e+012	24248.91	2.2172e+012
$P_1(5)$	21763.10	3.0897e+012	24163.10	2.5712e+012
$P_2(6)$	21977.31	3.3129e+012	24371.63	2.7647e+012
$P_1(6)$	21902.84	3.7780e+012	24295.52	3.1483e+012

3. Calibration

3.1. The Spectrometer

The system consist of the Andor Shamrock sr-163 equiped with the Andor iDus InGaAs camera and an adjustable slit accessory with a shutter attached. For improved cooling capability, a Koolance Exos-2 watercooler was installed. With the iDus camera follows the Andor Solis software.

3.1.1. Andor Shamrock sr-163

The Andor Shamrock sr-163 spectrograph is based on a Czerny-Turner optical layout, and features a 163mm focal length and a numerical aparture of f/3.6.[sha,]. It comes with a interchangable grating for different wavelength and range requirements and a micrometer wavelength drive to change the angle of the grating. The grating in our system is the SR1-GRT-0600 grating with 600 lines/mm. The blaze is 1600nm and the nominal dispersion is 7.11nm/mm. The maximum recommmended wavelength of this grating is 1715nm. The spectrometer is equipped with an added accessory the SR1-ASM-0020 adjustable slit.

3.1.2. Andor iDus InGaAs

The Andor iDus InGaAs DU491A-1.7 features a thermo-electric cooler which can reach -90 °C.
Technical Specifications:

Active Pixels	1024 x 1
Pixel Well Depth (e^-, typical)	Me^- Typical: HDR = 170, HS = 5
Read Noise (e^-, typical)	580 e^-
Readout Rate per pixel kHz (μs)	100 (10)
Max Spectra Per Sec	97
Pixel Size (μm)	25(W) x 500 (H)

Source: [web, a]

3.1.3. Andor Solis

The Andor Solis software makes it easier to take scans. It has several useful tools, and is easy to use. I developed my own software which is described in chapter 4, but that is designed for taking scans during a specific time period that makes it impractical when you only want to take one acquisition. This program was therefore used frequently for

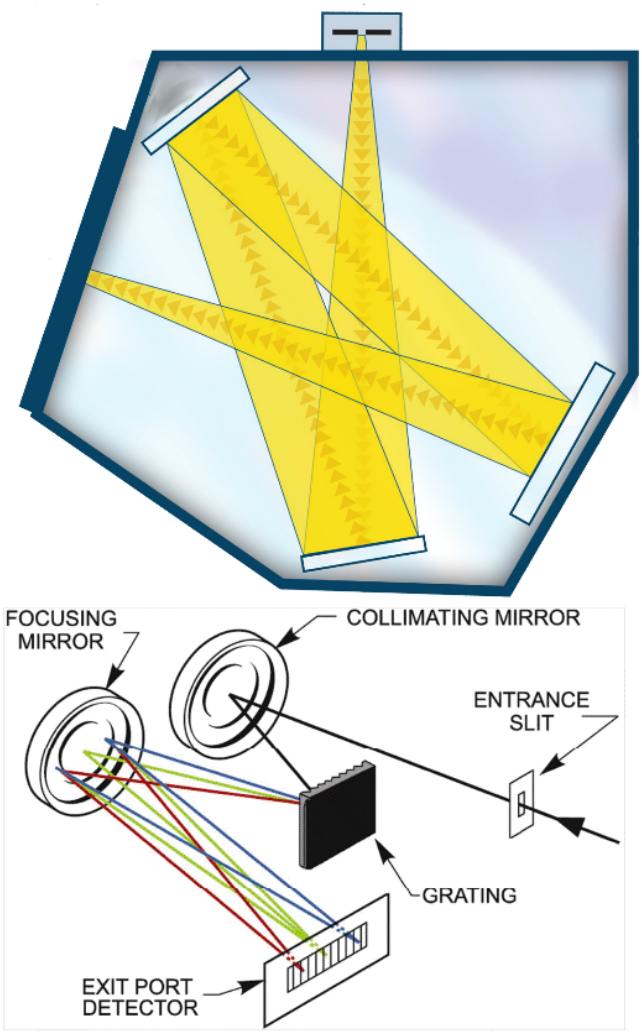

Figure 3.1.: The optical layout of the Andor Shamrock sr-163. [sha,]

calibration and testing, and it can export data to .asc files which can then be loaded in matlab.

3.1.4. Shutter

The Melles Griot 04 UTS 201 is part of the UltraThin Electronic Shutter series. It has five blades with a maximum speed of 1/60s. The apparture is 25.2mm. It is normally closed and has a solenoid voltage of 12Vdc. The resistance is 50Ω, the inductance is 54.9mH and the peak pulse current is 1A[web, c]. The controller used is Melles Griot 04 ISC 850 which will generate the shutter control voltage when commanded by a standard, TTL, logic level.

3.1.5. Koolance Exos-2

The Koolance Exos-2 was fitted to the iDus camera to reduse the time it takes to reach the desired temperature. It has two 120mm fans and a built in pump and reservoir, and since the iDus already has water cooling blocks in place installation is very easy. The water cooler reduces the cooldown time by several minutes and lowers the temperature one can reach, but it also increases the time it takes to stabilize.

3.2. Precalibration

3.2.1. CCD-Temperature

As mentioned the iDus InGaAs camera features a thermo-electric cooler which can cool down the CCD-chip to -90 °C. However the colder it gets the more energy is required to excite electrons across the band gap of the diodes. We can see this clearly in fig.3.4 where the intensity suddenly reaches zero at higher wavelengths. We define the wavelength where the intensity is halved as $P/2(\lambda)$. This wavelength will decrease as the temperature decreases. But since noise also decrease with temperature the optimal temperature is one that gives a high enough $P/2(\lambda)$ and low noise. We looked at the background levels, fixed pattern noise and thermal noise. The fixed pattern noise is caused by a slight difference in the offset of each channel. We found the fixed pattern noise by taking a background and looking at the standard deviation. This was calculated using Andor Solis which gives the mean and standard deviation after a scan. To find the thermal noise we took a background and subtracted by another background and looked at the standard deviation of the result. We took about 8 scans and averaged to find the result which is given in the table below:

Temperature	BG level	Fixed Pattern Noise	Thermal Noise	$P/2(\lambda)$
-50 °C	1019.0	139.0	14.8	1629
-55 °C	1013.6	137.5	14.5	1626
-60 °C	1009.8	135.8	14.6	1622
-65 °C	1006.8	134.3	14.6	1619

The background level, fixed pattern noise and thermal noise are all in counts while $P/2(\lambda)$

is in nm. As we see from the table the differences are small, but we chose -60 °C as the temperature for the CCD-chip since this gave us better wavelength coverage through the (4,2) hydroxyl band while reducing noise.

3.2.2. Testing the background levels

We tested the background levels to see if it changed over time. We let the spectrometer take backgrounds through the night by covering the slit and letting it run. "Accumulation mode" was used which means several scans are taken and then added together. We took six scans with ten seconds integration time without pause in between. The scans started when the solar zenith angle was 90 ° and stopped at sunrise when the solar zenith angle reached 90 ° again. The reason for this was to test the software which will be explained in section 4.1. It ran for three days and the result is shown in fig.B.1, fig.B.2 and fig.B.3. The plot shows the average intensity over each of the 1024 channels for each scan. The result from the first night is very good. It it has a maximum of $1.87 \cdot 10^4$ and a minimum of $1.856 \cdot 10^4$. However shortly after it started again for the second night it jumps to $2.3 \cdot 10^4$ and stabilizes. The reason for this is unknown as the conditions should be the same for each night. The third night we see that it has fallen down again to about the same levels as the first night. This shows us that using a shutter to get backgrounds through the night is necessary if we want accurate results. We also noticed that the first scans after turning the spectrometer on had lower intensity. After some tests we saw that it takes about five scans before you get normal results. Even after just uninitializing and initializing driver of the iDus camera the first scan is slightly off, but not as bad as if you power it off and then on again. Therefore it is important to discard the first scans, especially if it has been powered off before use.

3.3. Channel to wavelength calibration

We wanted to find a way to convert channel number to wavelength. We started by defining three overlapping bandwidths,

Region	Short	Mid	Long
Micrometer setting	730	790	832
Bandwidth	∼1330-1530nm	∼1450-1650nm	∼1540-1750nm

Here the micrometer setting refer to the micrometer which turns the angle of the grating. We then took several scans using different sources to try to find out exactly were they started to overlap. By comparing lines which appeared in two bandwidths we found out that the mid-section is shifted 624 channels to the right. That means that channel one in the mid-section is channel 625 in the short section. The long-section starts almost were the short-section ends. There is a gap of about 30 channels between them so they do not overlap. However the first channel on the long section corresponds to channel 532 on the mid section. To make it easier to calculate the wavelength we defined the "super-channel-number" which starts at the begining of the short-section and ends at the end of the long-section. In this way, only a single expression is needed to convert channel number to wavelength.

Region	Short	Mid	Long
Super-channel-numbers	1-1024	625-1648	1057-2080

To find the relation between the channelnumber and wavelength we took several scans in all the three sections using flourescence lines from a HeNe-laser,a Na-lamp and a standard fluorescent lamp. All three sources gives spectral lines in the 1330nm-1700nm region. The fluorescent lamp contains mercury and therefore shows some very useful mercury lines in this region. Using a preliminary estimate of the wavelength scale based on grating parameters we could then indentify spectral lines using [Outred, 1978].

Super-Channel-Number	Wavelength [nm]	Type
113	1357	Hg
163	1367	Hg
952	1523	Ne
987	1529,5	Hg
1570	1637,4	Na

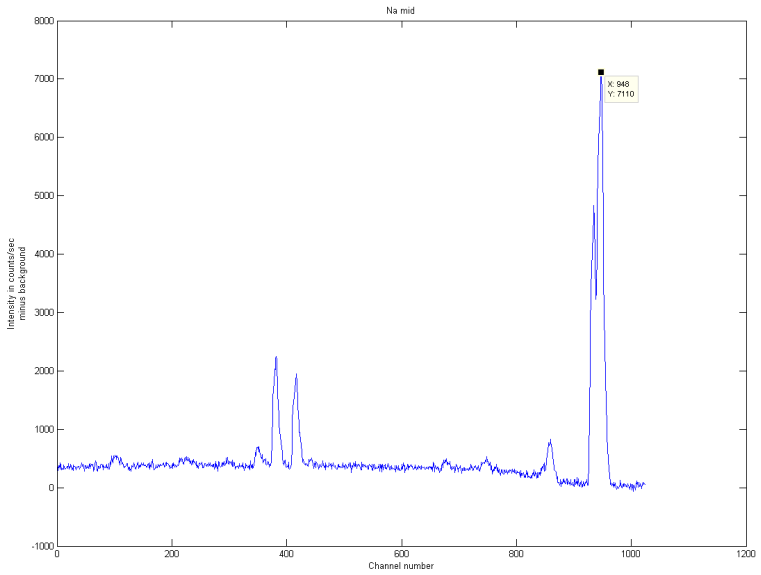

Figure 3.2.: A scan using a Na-lamp in the ”mid”-wavelenghts

The sodium spectral line can be seen in fig.3.2 while the neon spectral line and the other three mercury lines can be seen in the appendix in fig.B.4 and fig.B.5. Using these numbers and the matlab function ”polyfit” with n=2, we then found this relation between the super-channel-number and wavelength,

$$Wavelength[nm] = -0.86627 \cdot 10^{-6} \cdot SCN^2 + 0.2071 \cdot SCN + 1333.6 \qquad (3.1)$$

where SCN is the super-channel-number. A plot of this curve can be seen in fig.3.3.

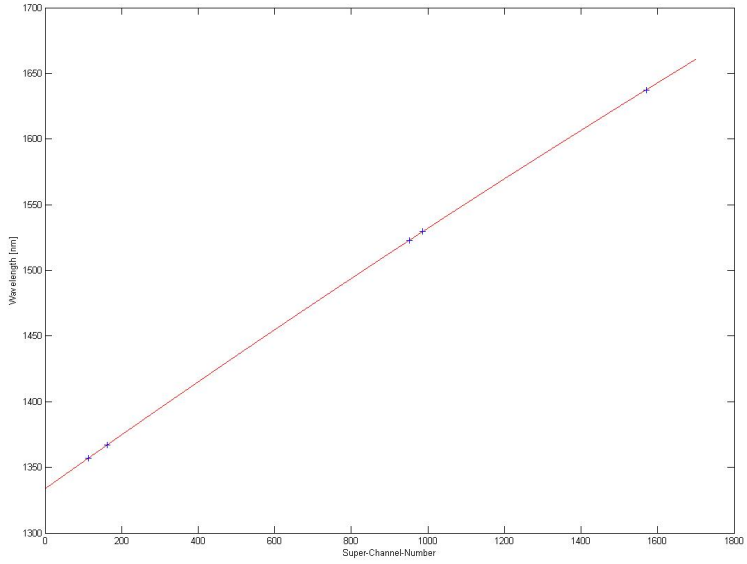

Figure 3.3.: A plot of the five spectral lines and the curve found with polyfit.

3.4. Finding the response curve

In order to know the absolute intensity of a source, the intrument response to a source of known brightness must be measured. This will result in a conversion between instrument signal and absolute photon radiance at each wavelength. The lamp used as standard was a 200-watt quartz-halogen tungsten coiled, filament lamp (model 220H) and is called m-255. This lamp was calibrated by Optronic Laboratories, 26. march 1980. The spectral irradiance was measured 50cm from the lamp using 6.50 amperes dc. The total irradiance was measured to be $6.41\frac{mW}{cm^2}$. The spectral irradiance at different wavelengths was given in a table which can be seen in table 1 in the appendix. We picked out all the points in the 1000-2000nm region and fitted a curve through those points using polyfit in matlab. This is the calibration curve and can be seen in fig.B.6.
The other equipment used was:
Shunt Resistor, Fluke Y5020 Current shunt
Power Supply, Bentham 605 S.N. 93143
Voltmeter, Hewlett Packard 34401A S.N.3146A13844
The m-255 and the shunt resistor were connected in series to the power supply. The voltmeter measured the voltage over the shunt resistor. The shunt value was 0.0099762Ω. The power supply was set to 6.5A and the voltmeter showed 65.006mV. To reduce the input intensity into the instrument the lamp was moved to a distance of 5.75m, and two apertures were used to eliminate scattered light. This modifies the manufacturers calibration from $C \ \ [\frac{\mu w}{cm^2 \cdot nm}]$, to $C \cdot (\frac{0.5}{5.75})^2 \ \ [\frac{\mu w}{cm^2 \cdot nm}]$. In order to provide an extended

source calibration, the light was directed onto a Lambertion reflector screen, and the instruments field of view was filled by the illuminated portion of this screen. The screen has been calibrated by Labsphere INC. and has a reflectance factor varying between 0.983-0.989 in the wavelength area 1000-1800nm. After reflection from the screen, the lamp intensity is scattered such that it falls off as $\frac{1}{cos(\theta)}$, where θ is the normal to the screen. Integrating the reflected radiance from this extended source over the scattering hemisphere modifies the manufacturers calibration from $C \cdot (\frac{0.5}{5.75})^2 \ \ [\frac{\mu w}{cm^2 \cdot nm}]$, to $\frac{C}{\pi} \cdot (\frac{0.5}{5.75})^2 \ \ [\frac{\mu w}{cm^2 \cdot nm \cdot sr}]$. Scans were taken in the wavelength region 1180-1750nm. The result can be seen in figure.B.7 and figure.B.8. The response curve was then calculated using,

$$R = \frac{I}{\frac{C}{\pi} \cdot (\frac{0.5m}{5.75m})^2} \qquad [\frac{c/s}{\mu W/(cm^2 \cdot nm \cdot sr)}] \tag{3.2}$$

where R is the response curve, I is the measured curve and C is the calibration curve from Optronic Laboratories. We wanted the result in photons per second and kilorayleighs instead of microwatts. The energy of one photon is,

$$E_{ph} = \frac{1.9878 \cdot 10^{-10} \ \ \mu J \cdot nm}{\lambda} \tag{3.3}$$

Multiplying the response curve with this gives you the response curve in $\frac{c/s}{(ph/s)/(cm^2 \cdot nm \cdot sr)}$.

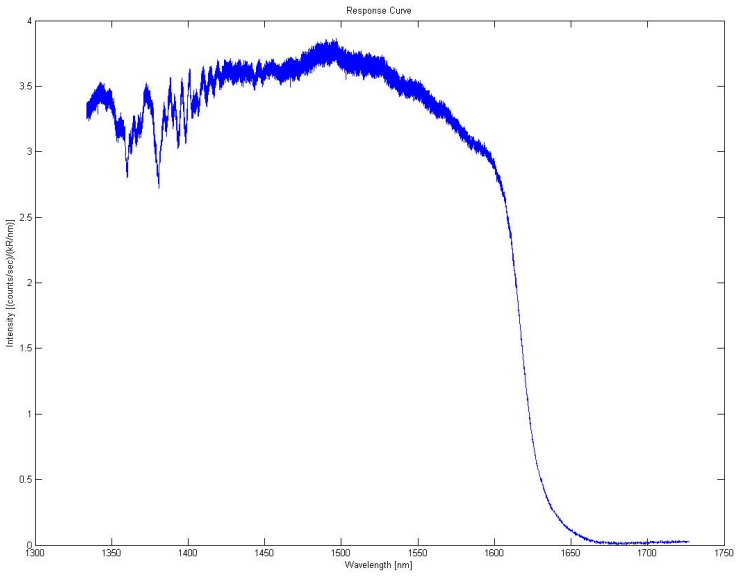

Figure 3.4.: The response curve in the 1330-1750nm region.

One kilorayleigh is,

$$1kR = 10^9 \ [\frac{ph}{cm^2 \cdot sec \cdot sphere}] = \frac{1}{4\pi} \cdot 10^9 \ [\frac{ph}{cm^2 \cdot sec \cdot sr}] \tag{3.4}$$

So to get the response curve in kilorayleighs we have to multiply with eq.3.3 and then with eq.3.4. The factor you have to multiply with is 0.0497 divided by the wavelength in nm and you get the response curve in $\frac{c/s}{kR/nm}$. The response curve in kilorayleighs can be seen in fig.3.4 in the 1330-1750nm region and fig.B.9 in the 1180-1380nm region.

3.5. Transfer standard

We calibrated the spectrometer against a transfer standard whose absolute calibration is tied to a similar absolute source from a different manufactor. The standard we used is a low brightness source (LBS) from Stockholm. This has been used to calibrate instruments from Australia, Korea and for the British Antarctic Survey, so if we calibrate our spectrometer against this source then we can compare data with them. We put the LBS right in front of the spectrometer and took scans with five seconds integration time. The data was then divided by the response curve and the result can be seen in fig.B.11.

3.6. Noise spectrum

We also looked at the noise spectrum of the spectrometer. This was done by taking a background and then subtracting by another background. This was then divided by a piece of the response curve. The piece that was chosen was approximatly 1350nm to 1550nm of fig.3.4 because this was a fairly flat section of the response curve, and it was 1024 points long so it fitted the data. The mean of the result was 0.0940kR/nm and the standard deviation was 0.5211. This means that a signal of 94R/nm will give a signal to noise of 1, and is called the noise equivalent signal.

4. Software

4.1. Automated acquisition

The main reason for this program is to take data during the night without requiring manual input. The Andor software is not capable of that so developing our own software was needed. First an attempt was made using Andor Basic, Andor's own programing language, but it lacked the necessary flexibility for our needs. Therefore it was decided to use Matlab and the Andor software developer's kit (sdk). The program is in the appendix, where main.m is the script which is run and awty.m, scan.m and savedata.m are the functions main.m uses. Main.m starts with opening the schedule file. This file contains all the input data the program needs.
Example of the schedule file:

1	Write a comment here!				
2	63.4				
3	10.4				
4	50				
5	C: \\Data\				
6	90	130	10	60	10
7	140	180	20	100	8

The first line is for writing a comment, it is not used by the program for control. The second line is the latitude of the instrument's position and the third line is the longitude. The forth line is the altitude which is not used for program control. The fifth line is where you want your data to be saved. The next lines contains details on how you want to aquire data. The first two numbers are the solar zenith angle interval of when you want to take data. You can have as many intervals as you want. The last three numbers are integration time, accumulation time and the background duty cycle. The integration time is the number of seconds the light is integrated in the detector. A number of these integrations are read out and integrated in computer memory to achieve the total accumulation time. Thus the accumulation time should be a multiple of the integration time. Figure 4.1 is taken from the Andor manual and shows

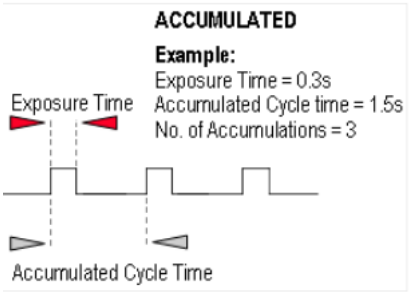

Figure 4.1.: This picture explains the definitions of the accumulation cycle. Source:[sha,]

the definitions used. Since we don't want any breaks between the accumulations the exposure time, which is the integration time, is the same as the accumulated cycle time. They will not be exactly the same, but the differance is only a matter of milliseconds. The accumulation time is therefore the integration time times the number of accumulations. The background duty cycle is how often you want to take backgrounds. In the example above there are two intervals. When the solar zenith angle is 90 ° the program begins taking 10 second scans, and six of these are added in memory before writing to disk. This gives a total accumulation time of 60 seconds per data spectrum. Every 10th data spectrum will be taken with the shutter closed, yielding a background spectrum. When the solar zenith angle reaches 130 ° it takes a break until it reaches 140 °. Then it takes five twenty-second scans which are added together for a total accumulation time of 100s per data spectrum. Now every eighth data spectrum is a background. Once the solar zenith angle reaches its highest angle and starts to drop, the process repeats itself backwards.

The next step in main.m is to load the drivers and initialize the instrument. If it successfully initializes, then it will return the errorcode 20002. Afterwards the shutter is closed, the temperature is set to -60 °C and a while-loop will run until the GetTemperature-function returns the errorcode 20036 which means the temperature has stabilized. Unfortunately the driver gives the errorcode too soon, the temperture will drift for some minutes after the driver claims it has stabilized. This isn't a problem unless it starts taking data immediately after it has cooled down. If so the first minutes of data should be discarded. Another thing that should be noted is that the first scan after the instrument has been powered off is corrupt. Therefore main.m takes a dummy scan for thirty seconds which is then discarded.

Next in main.m is the main while-loop which checks if it is time to scan. If so, it scans and saves the data, and then repeats. The while-loop is meant to run for as long as desired. It can run for several days if the schedule file is the same for each day. Since the instrument has to be cooled off and uninitialized, simply terminating the while-loop is a bad idea. Therefore a stop-button is made before the while-loop is begun. The code is,

```
f=figure('Position',[50 600 1 50]);
u=uicontrol('string','stop','callback','delete(get(gcbo,''parent''))');
while ishandle(f)
...
end
```

The while-loop will run as long as the stop-button is not pushed. This is just a simple trick to stop the eternal while-loop whenever you want.

The main while-loop starts by calling awty.m, the "are we there yet" function. It takes in the latitude, longtude and the solar zenith angle intervals. It calculates the solar zenith angle using an algorithm made by Roland Leigh `(RL40@le.ac.uk)` which has

been modifided for matlab. The hardware clock has to be set to GMT. It then checks which interval it is in, if any, and returns it along with the solar zenith angle. The variable ”check” is used as a boolean determining if its in an interval and should take data or not. If not the program is paused for thirty seconds, while the time is displayed, else the scan is started. The scan.m function takes in all the acquisition details from the schedulefile and starts the scan. If it is not taking a background scan the shutter is opened. A while-loop then runs, trying to aqcuire the data, until the scan is complete. It then creates a vector with the data and the acquisition details. The first six columns store the time when the scan completed, in year, month, day, hour, minute and milliseconds. Columns seven to eleven store the acquisition details: number of accumulations, integration time, solar zenith angle, tag (zero for background and one for regular signal), and the temperature. The next columns from 12-1035 store the counts at each channel. Note that the vector returned from driver, stored in ”tempdata”, is reversed before is is stored. This is because the driver returns the data from channel 1024 to 1. After the shutter is closed (if open) the data is then returned to main.m, where the latest background is subtracted from the data and displayed. After that the data is saved using savedata.m. Savedata.m takes in the data, the savelocation from the schedulefile and the longitude. It then calculates the number of days since 1.january, and the time of local noon in GMT. Since we are taking data during the night we let the day start and stop at local noon for convenience sake. Therefore if the time is before local noon, one day is subtracted. It then saves the data in the ”daymat” matrix, where all the data from the same day is stored. The matrix is then saved in the given savelocation with the name, yyyy-ddd.mat . After the acquisition is done and the stop-button is used, main.m turns off the cooler and let a while-loop run until the temperature is above -20 °C then the driver is uninitialized.
In figure.B.10 the time delay between each scan during a night is shown.

4.2. Calculating the temperature

The calculations were done using matlab. The program is shown in the appendix, and consist of four files where tempregmain.m is the main matlab script that is run and startinput.m, integration.m and tempreg.m are the functions it uses. Tempregmain.m loads the datafile and run the functions.
The first function is startinput.m, which start with taking in input from the user. This is because we didn’t have the shutter at the time we took the data. Therefore we let the spectrometer take scans before it got dark enough and let it run long after sunrise. The start and end input are required to cut out the interval were it is dark enough and the hydroxyl nightglow is clearly visible. Without the shutter we didn’t get any backgrounds during the night so we covered it up and took backgrounds either before we left in the afternoon, in the morning or both. The bg input is the scan you want to use as background. With the shutter installed it should pick the latest background as background, since we already know from section 3.2.2 that the background levels are unstable and can change through the night. This can be easily be done since each

data spectrum is tagged if it is a background or not. Without the shutter we found the background by averaging all the data spectra to see how it changes. This also makes it easy to pick out the start and end points. After the input is typed in it averages all the data spectra and subtract the background. The resulting data spectrum has a tendency of being curved due to residual solar scatter. A curve is therefore calculated in order to flaten the data spectrum. This is done by dividing the data spectrum into intervals of 50 pixels, since each data spectrum is 1024 pixels you get 21 intervals. The lowest point in each interval is found and the curve is found by fitting it through those points. The averaged data spectrum is then plotted along with the curve. After the data spectrum is subtracted by the curve and the result is plotted.
Next function is the integration.m function which integrates each line. It takes in the flattened data spectrum from startinput.m and output the values from integrating each OH line. It start with the array called lines. This contains the points where you want to integrate, i.e. the channel number of the peak of each line you want to integrate. Before the integration the data spectrum is divided by the response curve and the result is plotted. The integration is done by adding the values to the left and to right of the points in the array 'lines'. It can only go 7 channel numbers to the left and 14 channel numbers in total. It starts by adding the values on the left and stops adding when the next value is below half the value of the peak or until it has added 7 values. It then does the same on the right side. The sum is then divided by the number of points that where added and is converted to nm by dividing by 0.2071 from eq.3.1.
The last function calculates the temperature using the method described in section 2.5. The first two arrays, E and S, are the initial energy and the line strength of each line, which can be found the table in section 2.5. It then calculates the logarithm of the integration value divided by the line strength for each line and plots the result against the initial energy divided by boltzman's constant, 0.695 $cm^{-1} \cdot K^{-1}$[web, d]. A line is then fitted through the points of each band using several functions which can be found in section C.9. These functions are taken from [Bevington, 1969] and calculate the slope, intercept, χ^2 and the reverse of the slope which is the temperature. The result is then printed on screen.

5. Results

5.1. Measurements

Measurements of the (3,1) and (4,2) bands were done 31. March 2011. Since we did not have the shutter at this time we set the acquisition program to only take regular data spectra. The number of integrations was set to 6 and the integration time was set to 10 seconds which means the total accumulation time was 60 seconds. The averaged intensity of each data spectrum is shown in figure B.12. The spectrometer was covered up the first 250-300 scans. Then the cover was removed and the sunlight was bright enough to completely saturate the data spectrum. At data spectrum number 380 it gets dark and the average intensity is lower than when it was covered up. This indicates that the average background noise went down through the night or we failed to completely cover it. The spectrometer was placed inside a room looking through an open window. Therefore the temperture inside the room rapidly declined which may have affected the spectrometer. The temperature sensor on the CCD-chip however only showed -59K or -60K. At the 900th data spectrum the sun rises and it becomes saturated again until it is covered again at the 1124th data spectrum. The program described in section 4.2 was used to calculate the temperature. The first data spectrum was number 382 and the last was 881. Data spectrum 1125 was chosen as background since this had about the same averaged intensity as the chosen interval. Figure B.13 shows the average of the data from 382 to 881 subtracted by 1125 and the curve calculated to flatten the spectrum. Figure B.14 shows the flattened spectrum. Figure 5.1 shows the result in figure B.14 divided by the response curve which can be seen in figure 3.4. The arrows show the lines that were used in the calculation of the temperature. The first 6 lines are the $P_2(2)$, $P_1(2)$, $P_2(3)$, $P_1(3)$, $P_2(4)$, $P_1(4)$ lines of the (3,1) band and the last 4 are the $P_2(2)$, $P_1(2)$, $P_2(3)$, $P_1(3)$ lines of the (4,2) band. Figure 5.2 shows the plot of the natural logarithm of the integrated value of each line divided by the line strength plotted against the initial energy divided by Boltzman's constant.

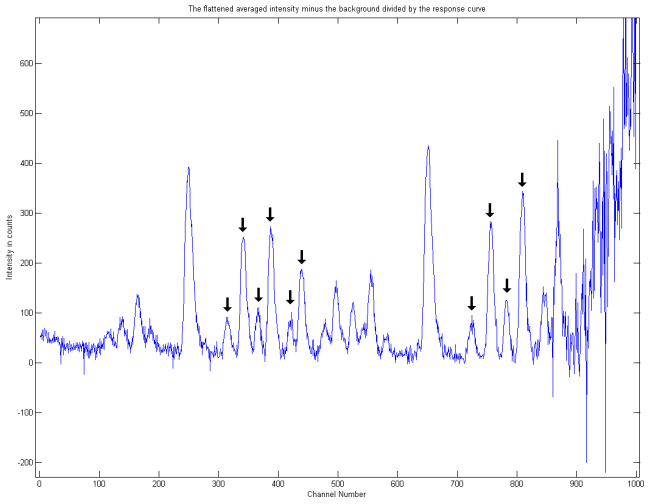

Figure 5.1.: The result from figure B.14 divided by the response curve in figure 3.4. The arrows shows the lines that were used in the calculation of the temperature.(31. march 2011)

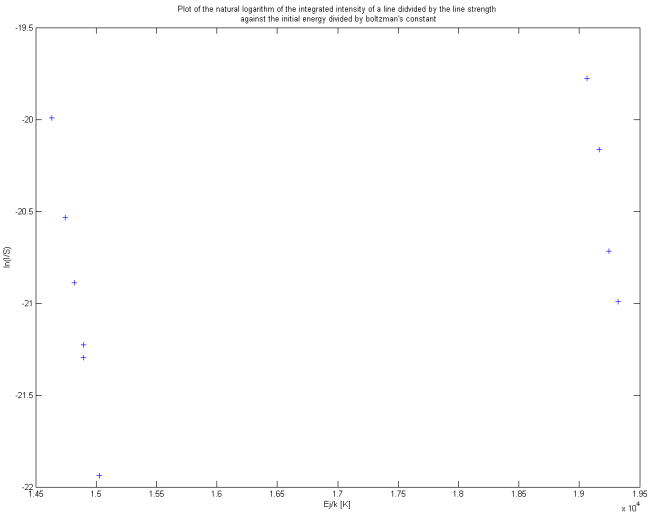

Figure 5.2.: The plot of the natural logarithm of the integrated value of each line divided by the line strength plotted against the initial energy divided by Boltzman's constant.(31. march 2011)

The output is displayed in the table below,
Fit to Ln(y)=Ln(a)+b*x $\longrightarrow$ Ln(I/S)=-1/T*(Ej/k)

Band	Slope	+/-	Intercept	+/-	χ^2
(3,1)	-0.0050	8.7074e-5	52.5185	1.2919	0.0028
	T [K]	+/-	b	+/-	
(3,1)	201.8562	3.5479	10601	447.1072	
Band	Slope	+/-	Intercept	+/-	χ^2
(4,2)	-0.0049	4.2648e-4	73.1868	8.1881	0.0135
	T [K]	+/-	b	+/-	
(4,2)	205.1191	17.9436	15012	2992.8	

Here the slope is -1/T, the intercept is the value of ln(I/S) if the initial energy was zero, χ^2 is the goodness of fit and b is the initial energy divided by Boltzman's constant when ln(I/S) is zero.

The same was done for the (7,4) and (8,5) bands. Another measurement was done in the 1180-1380nm region 6.may 2011. Since we did not have the shutter this time either the setup was very similar to the previous measurement. The number of integrations was set to 30 while the integration time was still 10s which gives a 5 minutes accumulation time. The reason for this change is that the (7,4) and (8,5) are much weaker than the (3,1) and (4,2) bands and therefore a longer accumulation time is needed. The averaged intensity of each data spectrum can be seen in figure B.15. This time the instrument was not covered up at the beginning, only for a short period at the end. The average intensity is rapidly rising at the end even though it is covered. As mentioned the reason for this may be that the instrument was not completely covered or it may be caused by changing room temperature. No matter the reason data spectrum 231 was chosen as background and data spectrum 110 and 140 was chosen as start and end points. Again the software from section 4.2 was used to calculate the temperature. Figure B.16 show the average of the data spectra from 110 to 140 subtracted by 231 and the curve calculated to flatten the spectrum. Figure B.17 show the flattened spectrum. Figure 5.3 show the result in figure B.17 divided by the response curve which can be seen in figure B.9. The arrows in this figure show the lines that were used in the calculation of the temperature. Only 3 lines were used from the (7,4) band, $P_1(2)$, $P_1(3)$, $P_1(4)$ and 6 lines from the (8,5) band, $P_2(2)$, $P_1(2)$, $P_2(3)$, $P_1(3)$, $P_2(4)$, $P_1(4)$. The reason why only 3 lines were used from the (7,4) band is because the resolution was not good enough. The resolution is constant in wavelength, but the spacing of the rotational lines is constant in frequency. This means that the spacing between rotational lines becomes smaller at shorter wavelengths. Thus a higher resolution is needed to separate the lines. At the (7,4) band the resolution is not good enough so the lines overlap and the $P_2(n)$ lines blends into to the $P_1(n)$ lines. At the (8,5) band it is just good enough to distinguish the lines. The $P_1(4)$ is much lower than it should be because of H_2O absorption. Therefore the value from integrating this line was multiplied by two. Figure B.18 shows the plot of the natural logarithm of the integrated value of each line divided by the line strength plotted against the initial energy divided by Boltzman's constant.

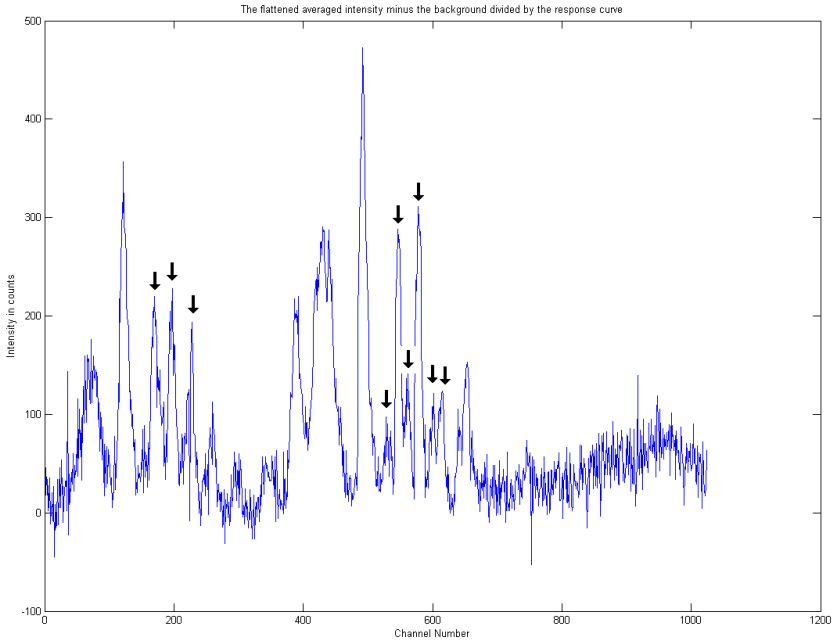

Figure 5.3.: The result from figure B.17 divided by the response curve in figure B.9. The arrows shows the lines that were used in the calculation of the temperature.(6. may 2011)

The output is displayed in the table below,

Fit to Ln(y)=Ln(a)+b*x $\longrightarrow$ Ln(I/S)=-1/T*(Ej/k)

Band	Slope	+/-	Intercept	+/-	χ^2
(7,4)	-0.0050	7.4928e-4	134.0881	23.2527	0.0134
	T [K]	+/-	b	+/-	
(7,4)	199.6318	29.8609	26768	8646	
Band	Slope	+/-	Intercept	+/-	χ^2
(8,5)	-0.0051	2.1818e-4	154.8034	7.5454	0.0142
	T [K]	+/-	b	+/-	
(8,5)	196.3700	8.4134	30399	2784.1	

5.2. Calculations using MSIS

To test the results we used a program to predict the temperature of the mesosphere. The program we used is a steady state hydroxyl model developed at The British Antarctic Survey that uses the MSIS-E-00 [Picone et al., 2003] model for the background atmosphere. We used the model to calculate the temperature on the 31. March 2011 and 6. May 2011. The input was daynumber (89 for 31.March and 124 for 6.May), latitude (63.4), longitude (10.4) and an adjustment factor for the MSIS atomic oxygen, which was set to 1. The maximum solar zenith angle was set to -5 degrees meaning it averages through the night when the sun is more than 5 degrees below the horizon. The predicted temperature profile of the higher atmosphere can be seen in figure B.19 for 31.march and in figure B.22 for 6.may. The predicted intensity of each OH vibrational state can be seen in figure B.20 for 31.march and in figure B.23 for 6.may. The vibrational states are $OHx = OH^*(v' = x)$. The predicted OH concentration of each vibrational state can be seen in figure B.21 for 31.march and in figure B.24 for 6.may. The predicted temperatures is given in the table below,

31. march 2011 daynumber 089,

Band	Predicted temperature	Measured temperature
(3,1)	202,3	201,85
(4,2)	201,74	205,1

6. may 2011 daynumber 124,

Band	Predicted temperature	Measured temperature
(7,4)	176,53	199,63
(8,5)	176,36	196,37

As we can see the predicted temperature and measured temperature of the (3,1) and (4,2) bands are almost the same. The (3,1) band is especially good considering the very low χ^2 and uncertainty. For the (7,4) and (8,5) bands the predicted temperatures are about 20-25 degrees colder than measured. The measured temperatures the (7,4) and (8,5) bands have a rather high uncertainty so that may be the reason for the deviance, but both the predicted temperatures and the measured temperatures gets colder towards the summer. Changes in the model temperature profile in this altitude range could lead to an uncorrect model temperature. That is, the temperature profile for day 89 is insensitive to small changes in the modeled height of the the OH-layer, while for day 124 the temperature structure could lead to small differences in height causing large temperature changes. Similary , since there are few measurements that goes into the empirical MSIS model in this altitude range, there could be model discrepancies during this rapid transition to summer conditions. Also the smaller spacing between the lines for the (7,4) and (8,5) bands could lead to the intensity of lines being overestimated, leading to a large error in the temperatures.

6. Conclusion

The purpose of this master thesis was not to accurately measure the temperature of OH-layer in the mesosphere, but to develope a system that is capable of accurately measuring the temperature of OH-layer in the mesosphere. In this thesis I have evaluated an integrating spectrometer as to its capability and suitability for making systematic measurements of the OH nightglow in order to infer mesospheric temperatures. This includes the characterization of the intrumental resolution as a function of slit width, calibrating the wavelength scale, and calibrating the absolute signal response. I have also performed studies of the behaviour of the instrumental response and noise response as a function of temperature, and developed an integrated software package that allows systematic and semi-autonomous acquisition of night glow spectra.
The instrument generally seems capable of routinely measuring the hydroxyl night glow of the (3,1) and (4,2) bands at the current resolution. However the instability of the background level that was described in section 3.2.2 should be further investigated. The inclusion of the shutter could mitigate the effects of the changing background. Unfortunately the shutter arrived to late so we did not have time to use it, but it was installed and tested. Though it has not been properly mounted on the instrument, it works as intended with the software. With that said the best way to solve this problem is not just using a shutter, but indentify the problem causing it and solve it. Also including ν^3 in eq. 2.35 would give more accurate results.
As for the (7,4) and (8,5) bands, the lower signal levels and water absorption make these upper state levels, where the OH is directly populated, more challenging. The instruments resolution could be improved to distinguish the lines, by narrowing the slit width. This would lower the signal levels proportionally and therefore the integration time will need to be lengthened to maintain the same signal to noise ratio. Since these bands are far from the cutoff frequency, the detector could be cooled of more to reduce the noise while maintaining the signal response. It is beyond the scope of this thesis to perform these trade off studies, but what has been done will allow them to be done, and result in optimum performance of the instrument for nigth glow studies.

Bibliography

[web, a] Andor technology. http://www.andor.com/.

[web, b] apollo.lsc.vsc.edu. http://apollo.lsc.vsc.edu/classes/met130/notes/chapter1/vert_temp_all.html.

[web, c] Melles griot ultrathin electronic shutters. https://www.cvimellesgriot.com/Products/UltraThin-Electronic-Shutters.aspx.

[web, d] physics.nist.gov. http://physics.nist.gov/cgi-bin/cuu/Value?kshcminv.

[web, e] Rotational spectroscopy. http://www.zampwiki.com/?t=Microwave_spectroscopy.

[sha,] *Users Guide, Andor Shamrock sr-163.*

[Andrews, 2000] Andrews, D. G. (2000). *An Introduction to Atmospheric Physics.* Cambridge.

[Baker and Stair, 1988] Baker, D. and Stair, A. (1988). Rocket measurements of the altitude distributions of the hydroxyl airglow. Technical report. Physica Scripta 37, 611.

[Bevington, 1969] Bevington, P. R. (1969). *Data Reduction and Error Analysis for the Physical Sciences.* McGraw-Hill.

[Dikty et al., 2010] Dikty, S., Schmidt, H., Weber, M., von Savigny, C., and Mlynczak, M. G. (2010). Daytime ozone and temperature variations in the mesosphere: a comparison between saber observations and hammonia model. Technical report, Institute of Environmental Physics, Bremen, Germany and Max-Plack Institute for Meteorology, Hamburg, Germany and NASA Langley Research Center, Hampton, VA, USA.

[Espy, 1986] Espy, P. J. (1986). *A Spectroscopic Investigation of the Infrared Molecular Band Systems of N_2 and N_2^+ Resulting From Low Energy Electron Impact Excitation.* PhD thesis, Utah State University.

[Herzberg, 1971] Herzberg, G. (1971). *The Spectra and Structures of Simple Free Radicals.* Dover Publications, INC.

[Krassovsky, 1972] Krassovsky, V. (1972). Infrasonic variations of the oh emission in the upper atmosphere. Technical report. Ann. Geophys. 28, 739.

[Kvifte, 1961] Kvifte, G. (1961). Relative population of oh x $^2\pi$ levels. *Planet. Space Sci.*, 5:158–162. Department of Physics, Agricultural College of Norway.

[Marsh and Smith, 2006] Marsh, D. R. and Smith, A. K. (2006). Saber observations of the oh meinel airglow variability near the mesopause. Technical report, National Center for Atmospheric Research, Boulder, Colorado.

[Outred, 1978] Outred, M. (1978). Tables of atomic spectral lines from the 10000$\AA$ to 40000$\AA$ region. Technical report, The Department of Physics, the Polytechnic of North London.

[Picone et al., 2003] Picone, J., Hedin, A., Drob, D., and Aikin, A. (2003). Nrl-msise-00 empirical model of the atmosphere: Statistical comparisons and scientific issues. *Geophys. Res.*

[Sivjee, 1991] Sivjee, G. G. (1991). Airglow hydroxyl emissions. Technical report, Department of Mathematics and Physical Sciences Embry-Riddle Aeronautical University.

[Whiting, 1973] Whiting, E. (1973). Computer program for determining rotational line intensity factors for diatomic molecules. Technical report, NASA Report TN-D-7268.

Appendices

A. Tables

Table 1.

Wavelength [nm]	Spectral Irradiance $[\frac{\mu W}{cm^2 \cdot nm}]$
300	.0321
320	.0620
350	.143
370	.223
400	.402
450	.834
500	1.41
550	2.05
600	2.70
650	3.28
700	3.79
750	4.18
800	4.43
900	4.62
1000	4.54
1100	4.27
1200	3.87
1300	3.46
1400	3.06
1500	2.67
1600	2.33
1700	2.10
1800	1.82
1900	1.59
2000	1.38

B. Plots

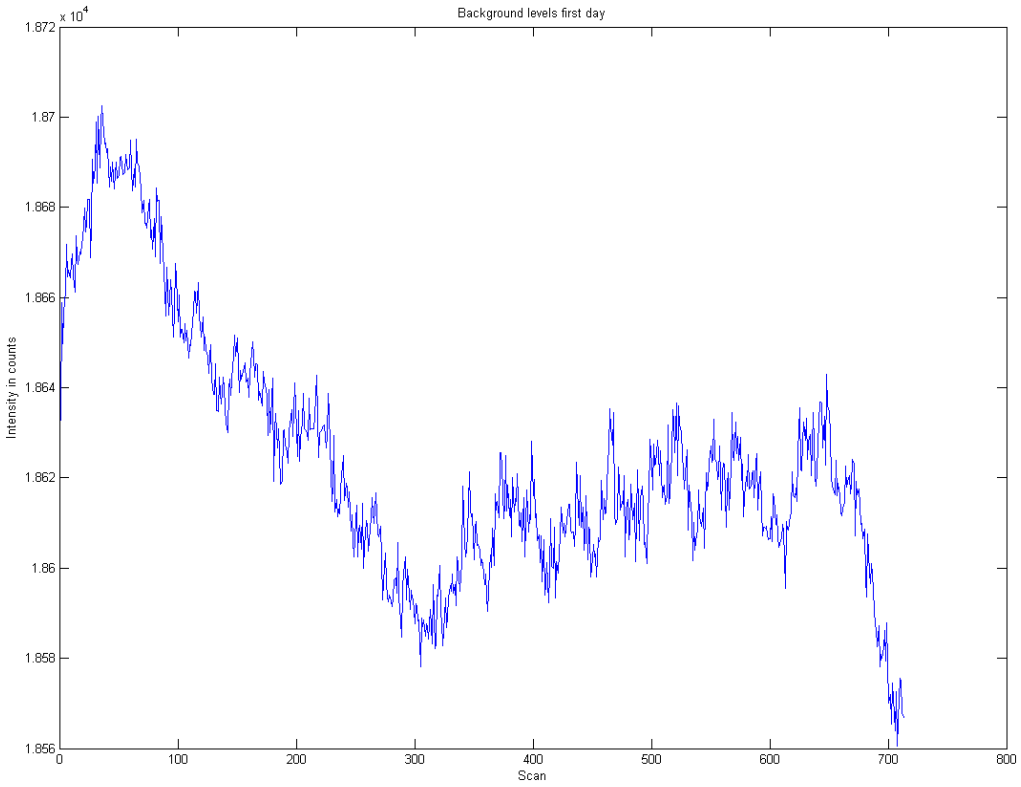

Figure B.1.: Plot showing the average background level and how it changes during the night of day 1. This was done by covering the slit of the spectrometer and letting it run through the night. The count of each channel was then averaged. The accumulation mode was used where several scans are taken and then added together. Here six scans with ten seconds integration time was added together without pause in between. Which means each accumulation was 60 seconds long. The first scan was set to begin when solar zenith angle was 90 degrees. The x-axis the number of the scan or minutes since the first scan. The y-axis is the intensity in counts.

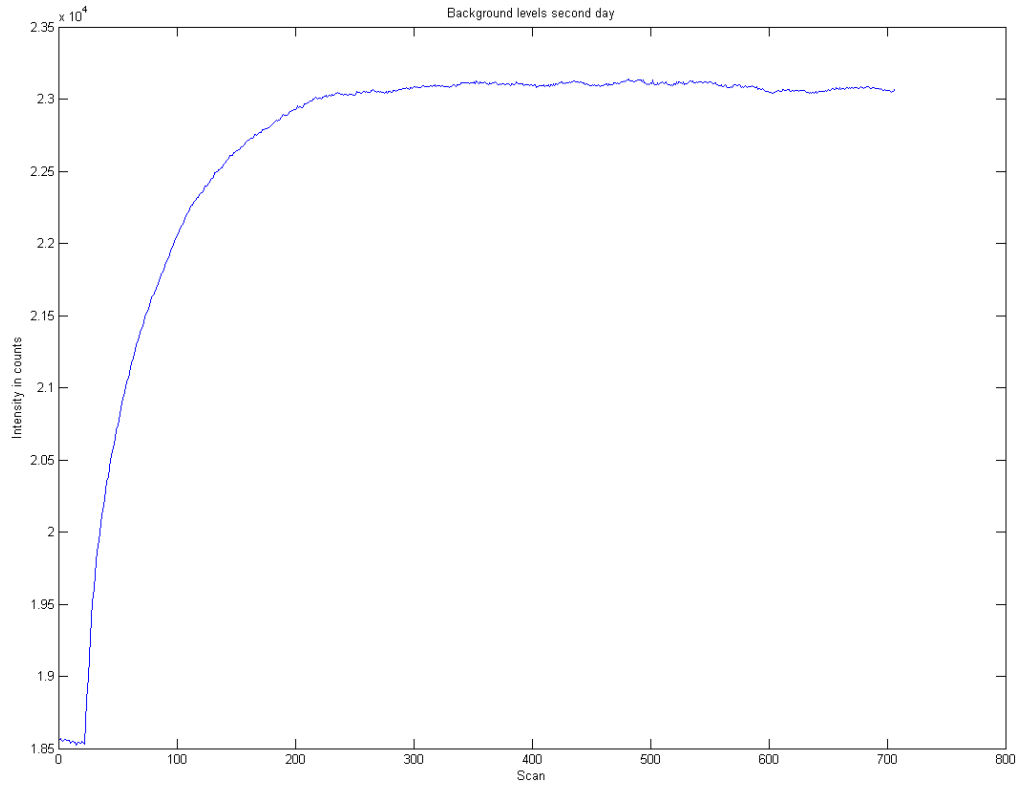

Figure B.2.: Plot showing the average background level and how it changes during the night of day 2. This was done by covering the slit of the spectrometer and letting it run through the night. The count of each channel was then averaged. The accumulation mode was used where several scans are taken and then added together. Here six scans with ten seconds integration time was added together without pause in between. Which means each accumulation was 60 seconds long. The first scan was set to begin when solar zenith angle was 90 degrees. The x-axis the number of the scan or minutes since the first scan. The y-axis is the intensity in counts.

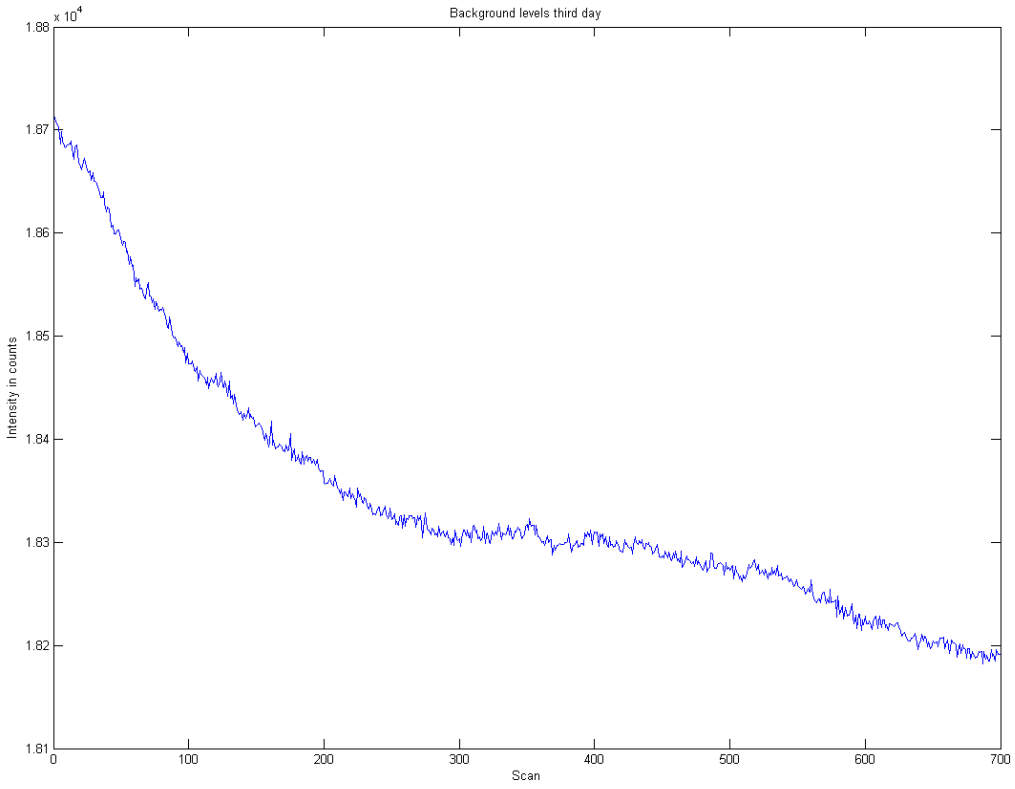

Figure B.3.: Plot showing the average background level and how it changes during the night of day 3. This was done by covering the slit of the spectrometer and letting it run through the night. The count of each channel was then averaged. The accumulation mode was used where several scans are taken and then added together. Here six scans with ten seconds integration time was added together without pause in between. Which means each accumulation was 60 seconds long. The first scan was set to begin when solar zenith angle was 90 degrees. The x-axis the number of the scan or minutes since the first scan. The y-axis is the intensity in counts.

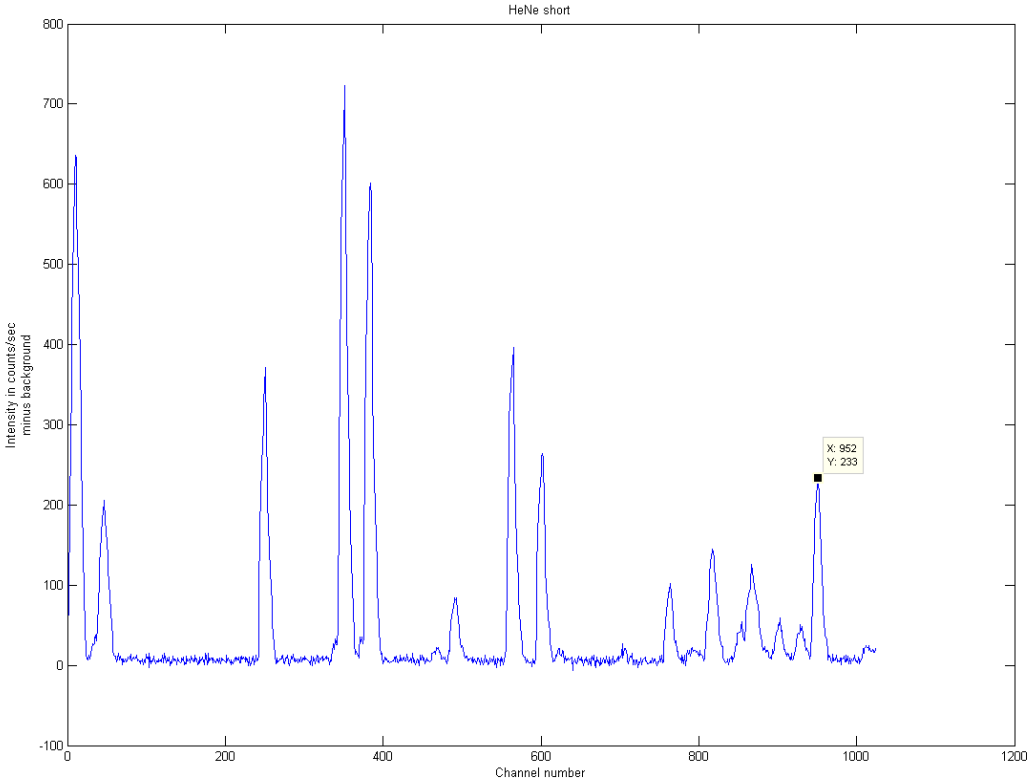

Figure B.4.: A scan using a HeNe-laser in the ”short”-wavelenghts.

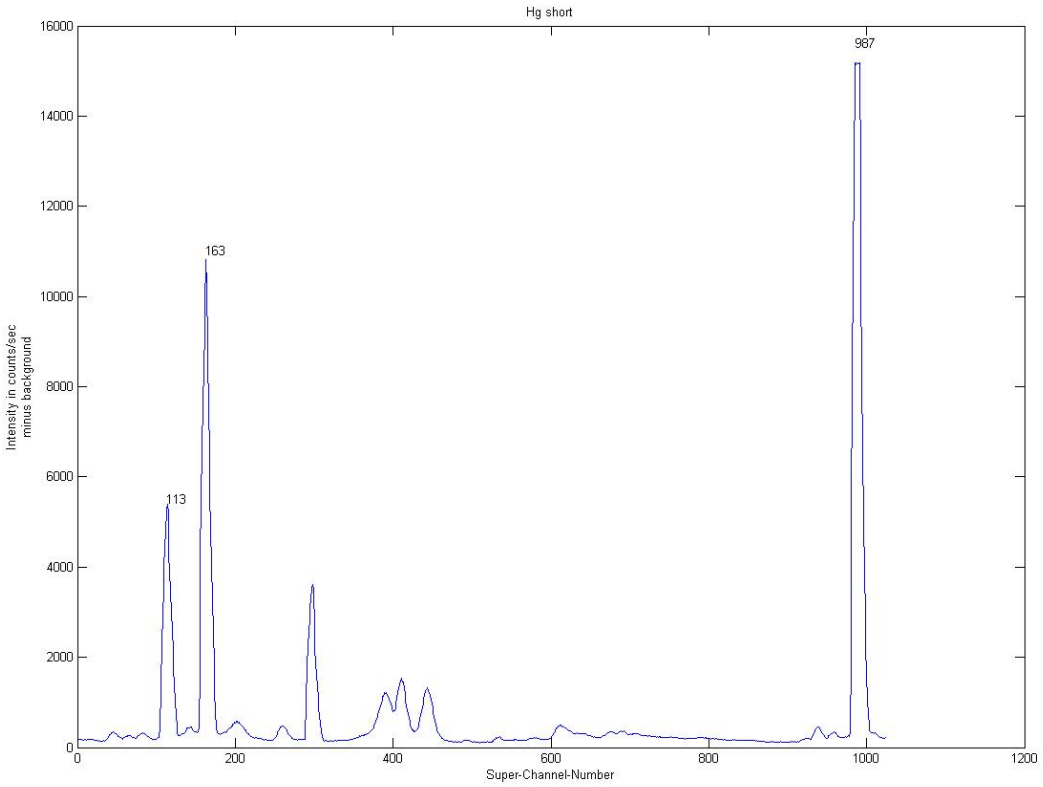

Figure B.5.: A scan using a fluorescent lamp in the "short"-wavelenghts. The channel number is shown at the top of each spectral line which was used in the calibration. The integration time should have been shorter as the 1529.5nm line (channel number 987) is oversaturated, but this scan was done just for illustrational purpose.

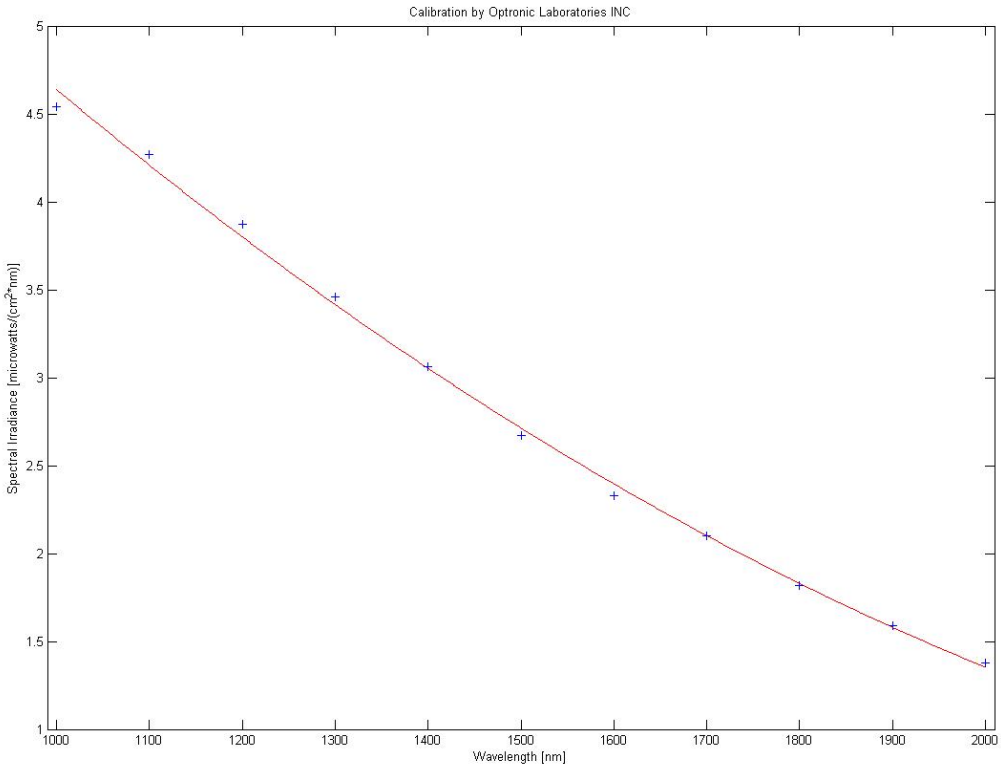

Figure B.6.: A plot showing the selected points from the calbration report and the curve fitted through using polyfit in matlab

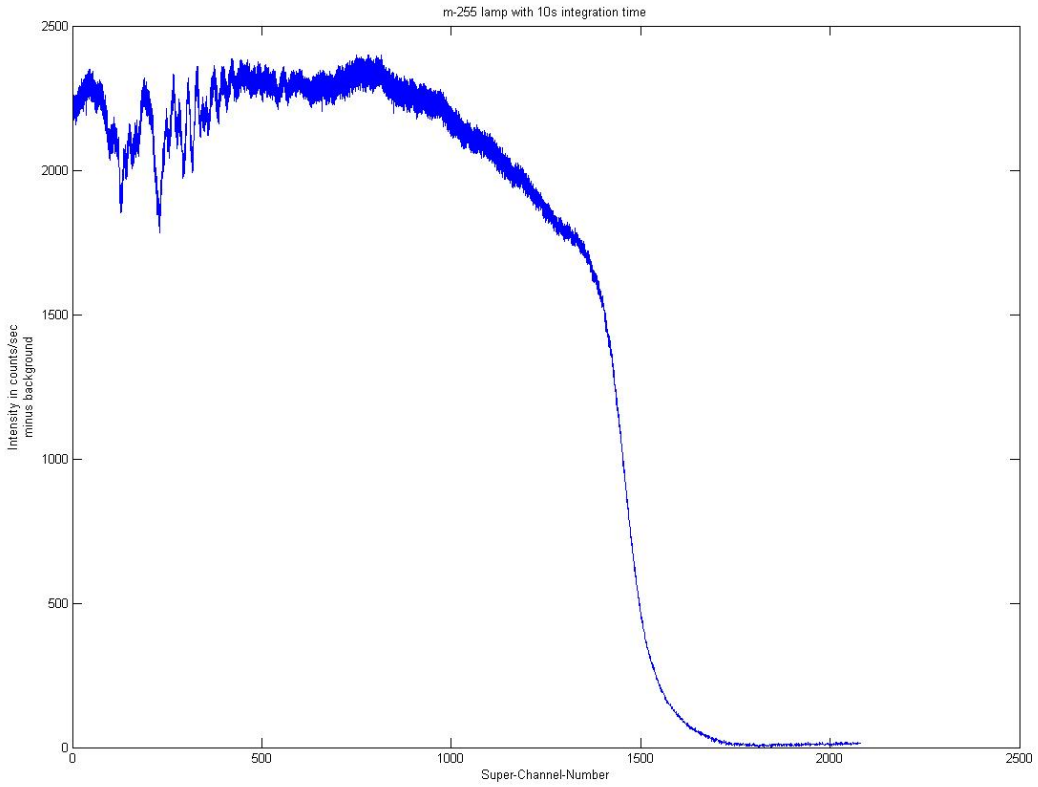

Figure B.7.: A scan of the m-255 lamp using 10s integration time. This was made by taking three scans in the short, mid and long sections and putting them together using the ”super-channel” definition.

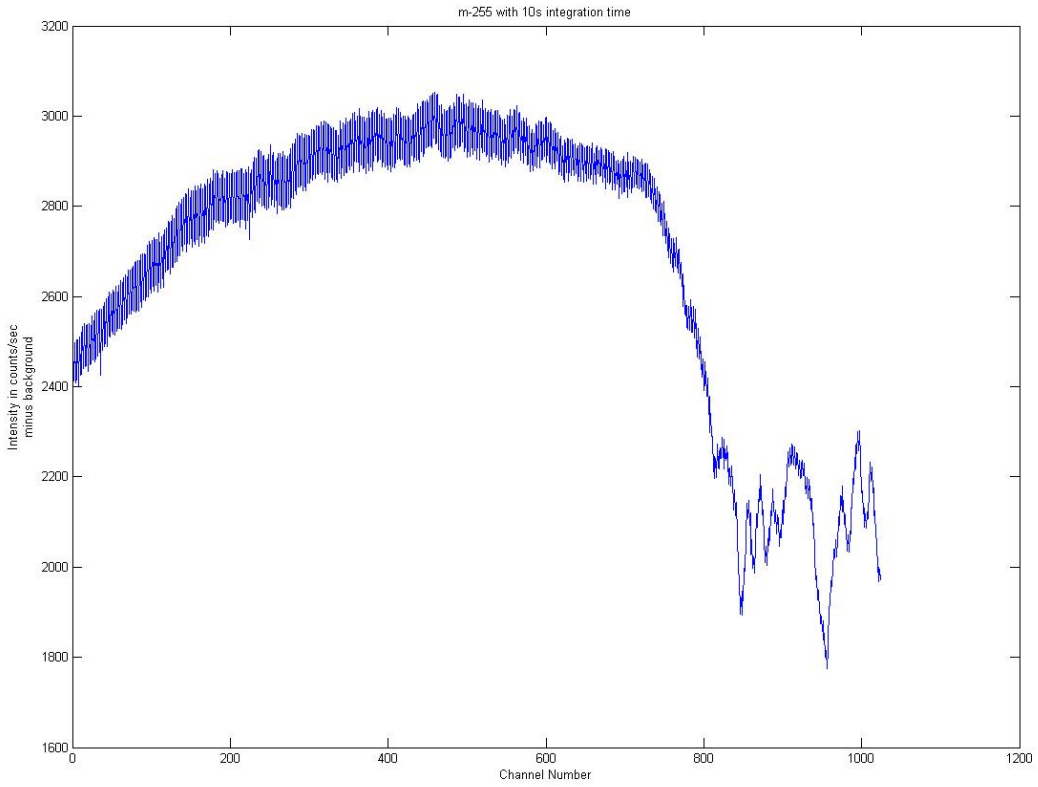

Figure B.8.: A scan of the m-255 lamp using 10s integration time. This was taken with the micrometer setting at 655. You can see that this scan is overlapping fig.B.7

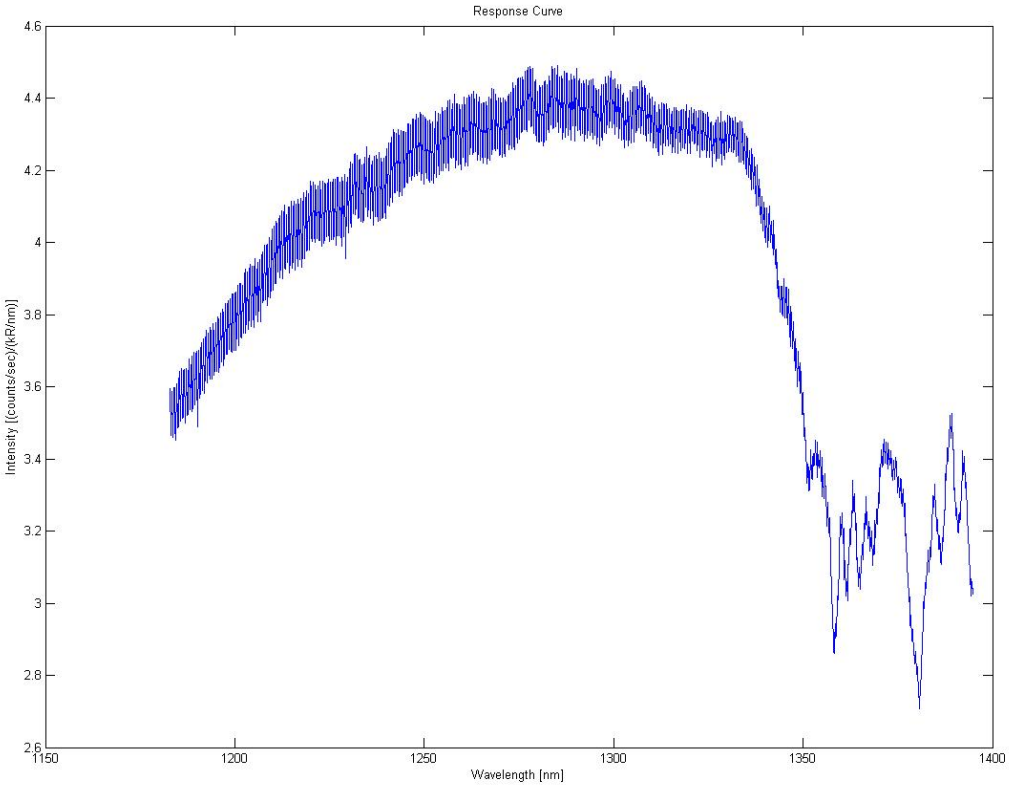

Figure B.9.: The response curve in the 1180-1380nm region.

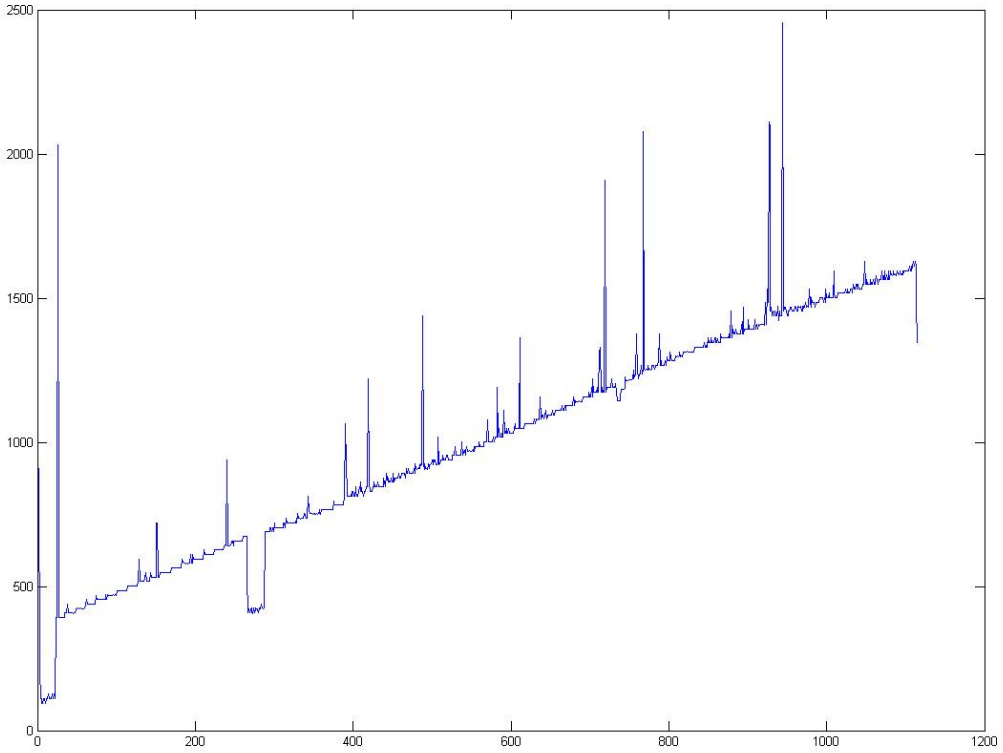

Figure B.10.: A plot of the timedelay between each scan during a night. The y-axis is milliseconds and the x-axis is number of data-vectors in the daymat-matrix. The computer used is fairly old, but the delay is never larger then approximately 1.5seconds (except a few spikes) and is linear to the size of the daymat-matrix. The small dips is propably caused by user interaction. Its seems the delay increase by a couple hundred milliseconds when windows goes to sleep-mode, and runs the program in the background.

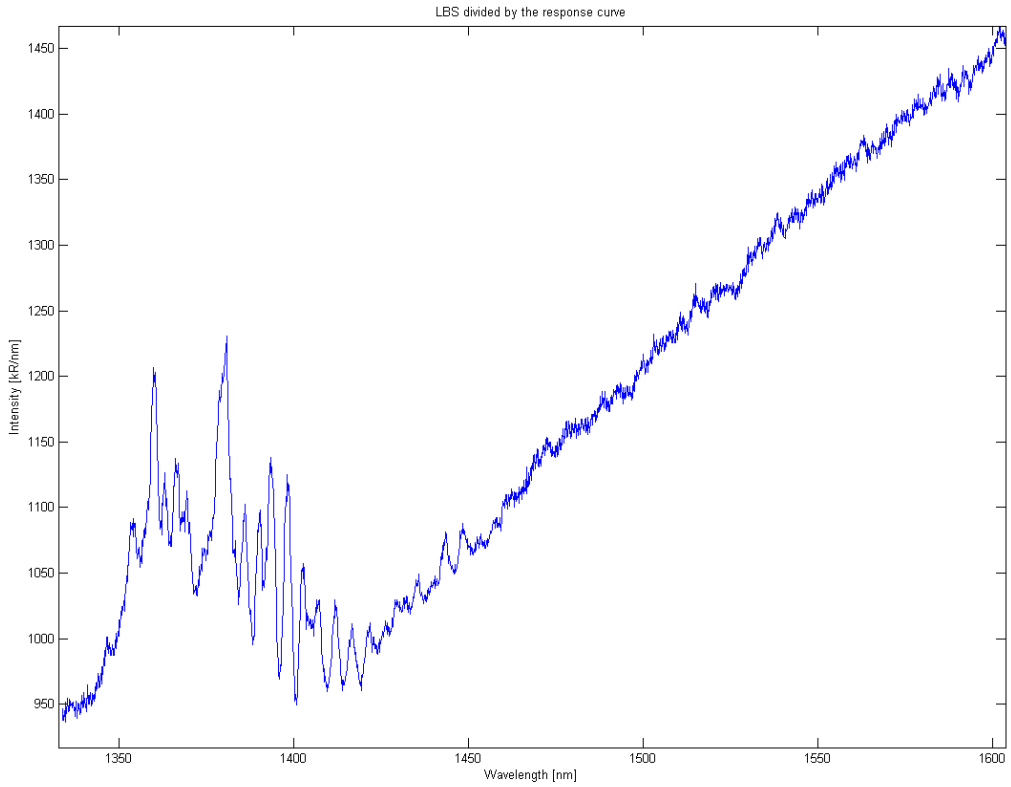

Figure B.11.: The plot shows a scan of the LBS divided by the response curve from fig.3.4.

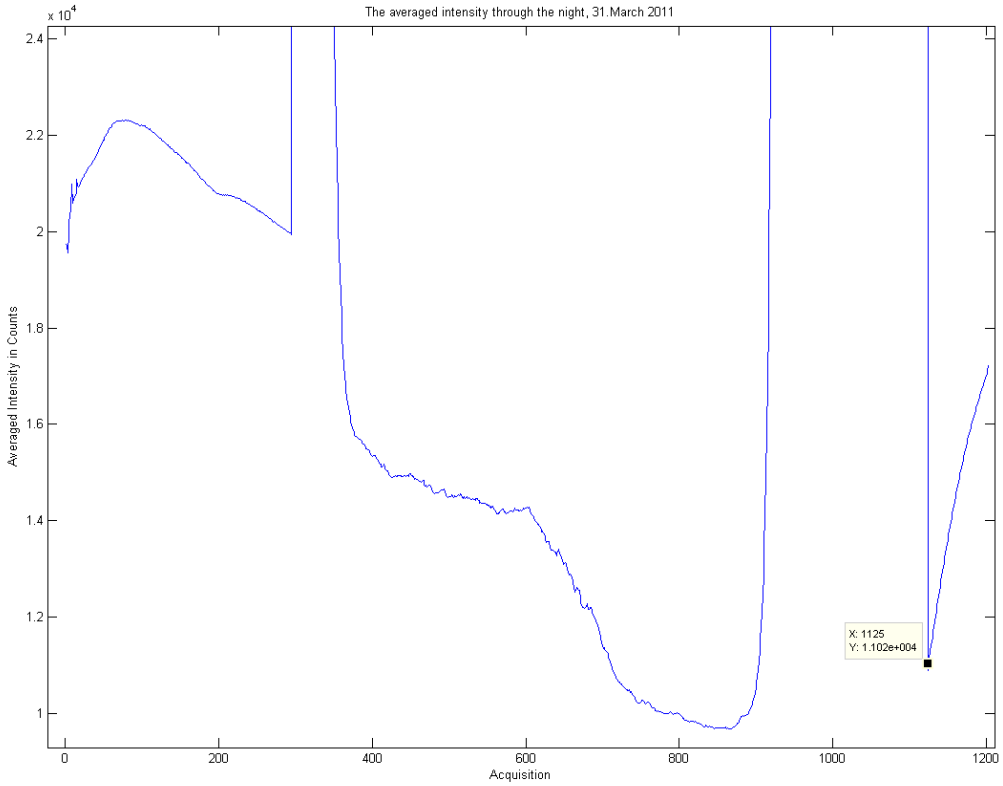

Figure B.12.: The averaged intensity of each data spectrum 31. march 2011. The 1125th data spectrum was chosen as background.

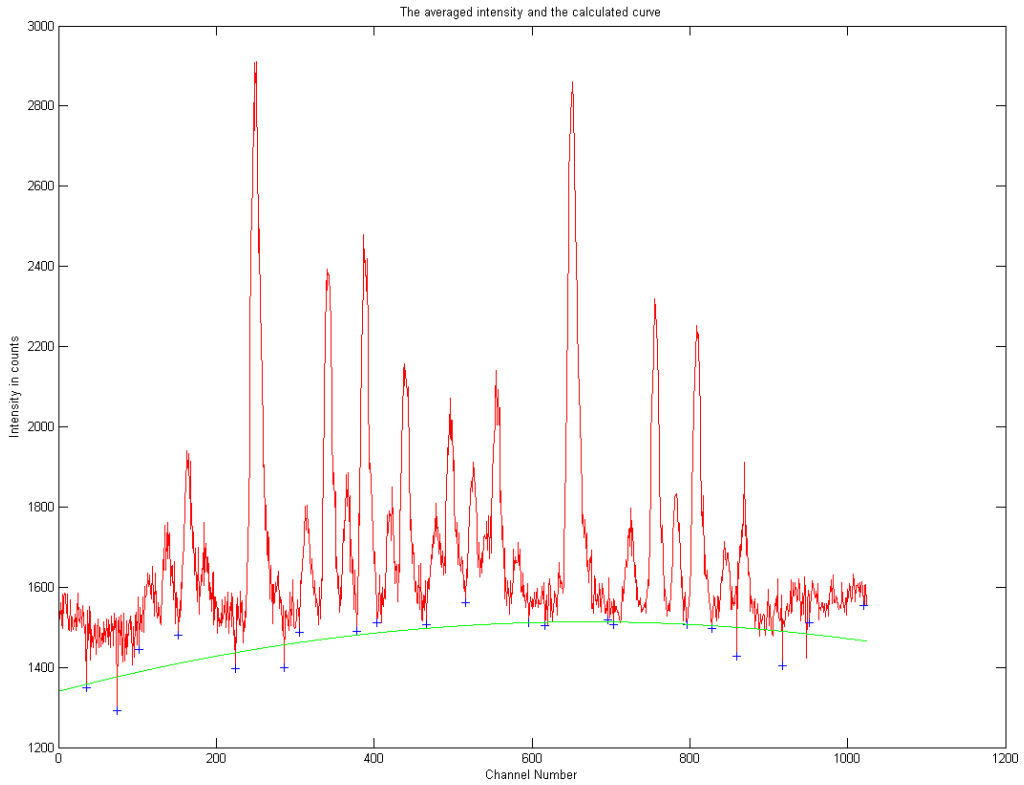

Figure B.13.: The averaged intensity of data spectrum 382 to 881 using 1125 as background. The green curve is the calculated curve described in section 4.2.(31. march 2011)

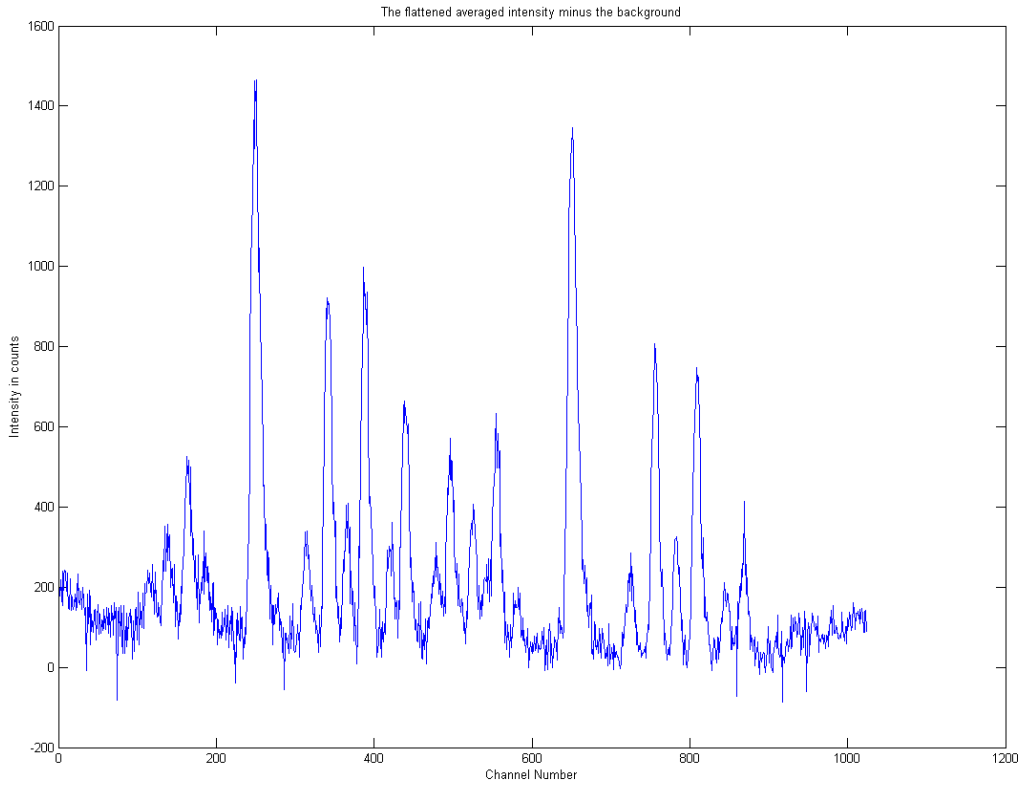

Figure B.14.: The average subtracted the calculated curve.(31. march 2011)

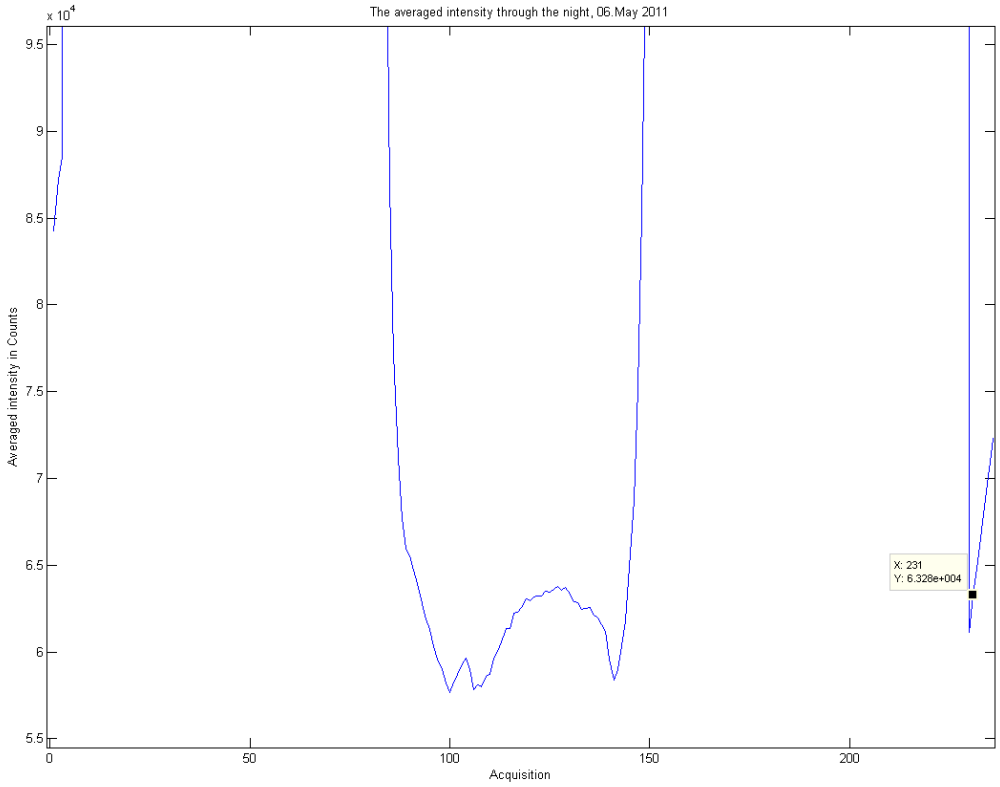

Figure B.15.: The averaged intensity of each data spectrum 6. may 2011. The 231th data spectrum was chosen as background.

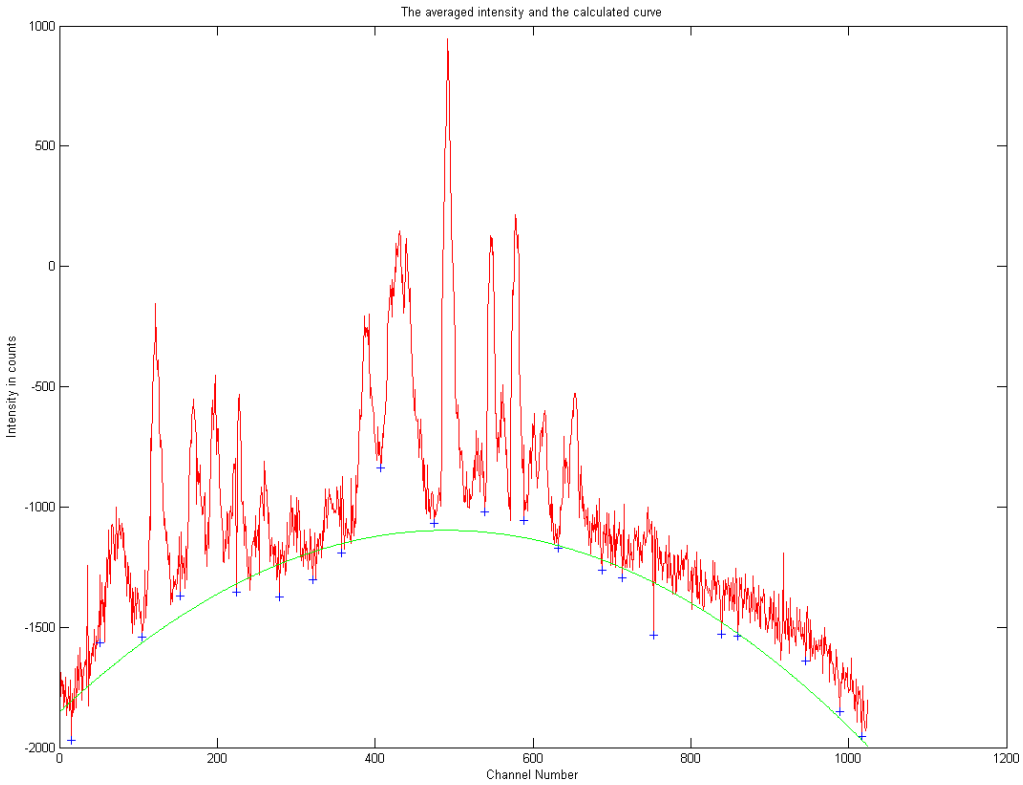

Figure B.16.: The averaged intensity of data spectrum 110 to 140 using 231 as background. The green curve is the calculated curve described in section 4.2. (6. may 2011)

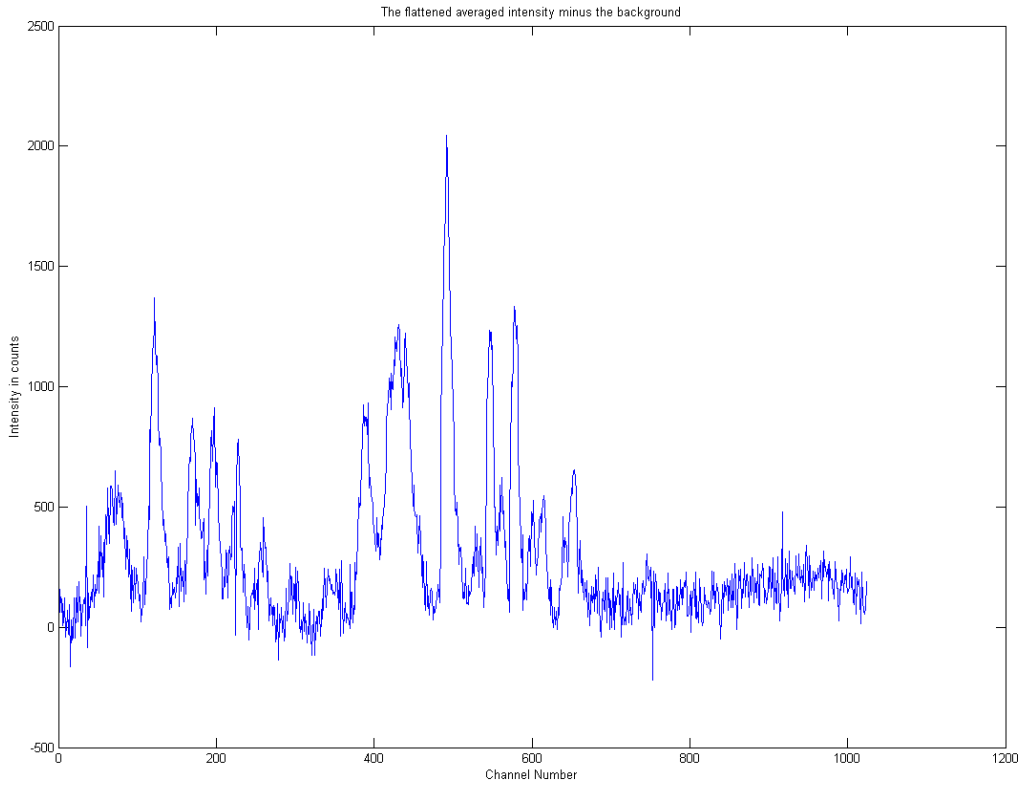

Figure B.17.: The average subtracted the calculated curve.(6. may 2011)

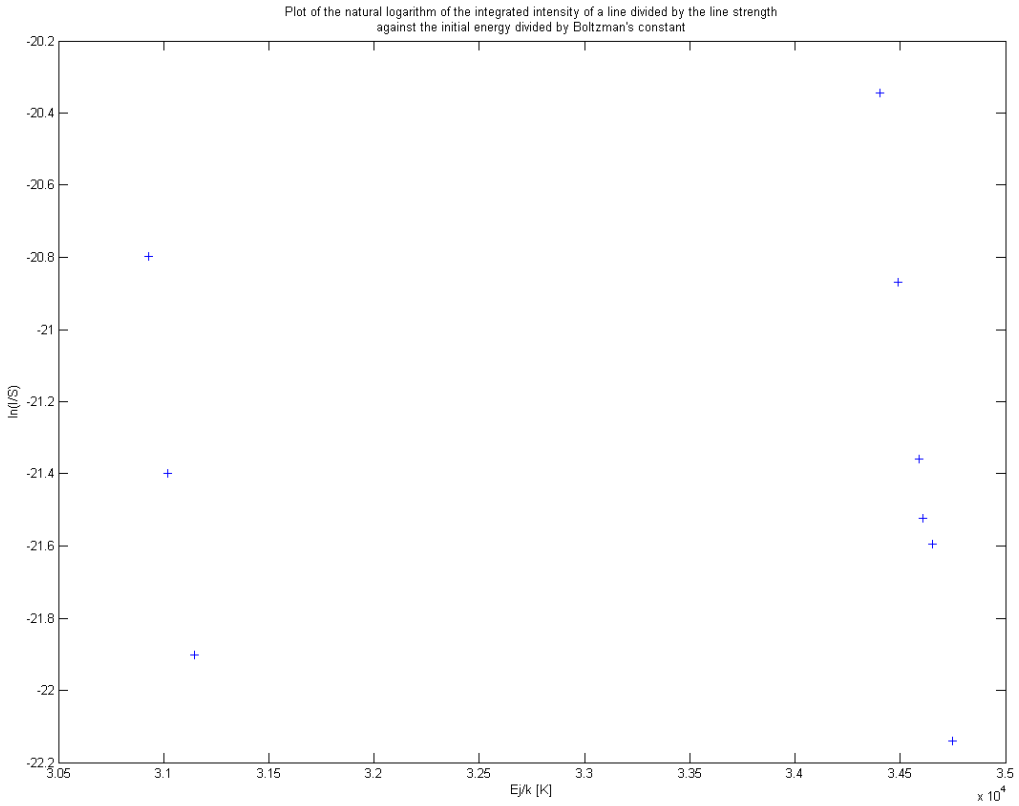

Figure B.18.: The plot of the natural logarithm of the integrated value of each line divided by the line strength plotted against the initial energy divided by Boltzman's constant.(6. may 2011)

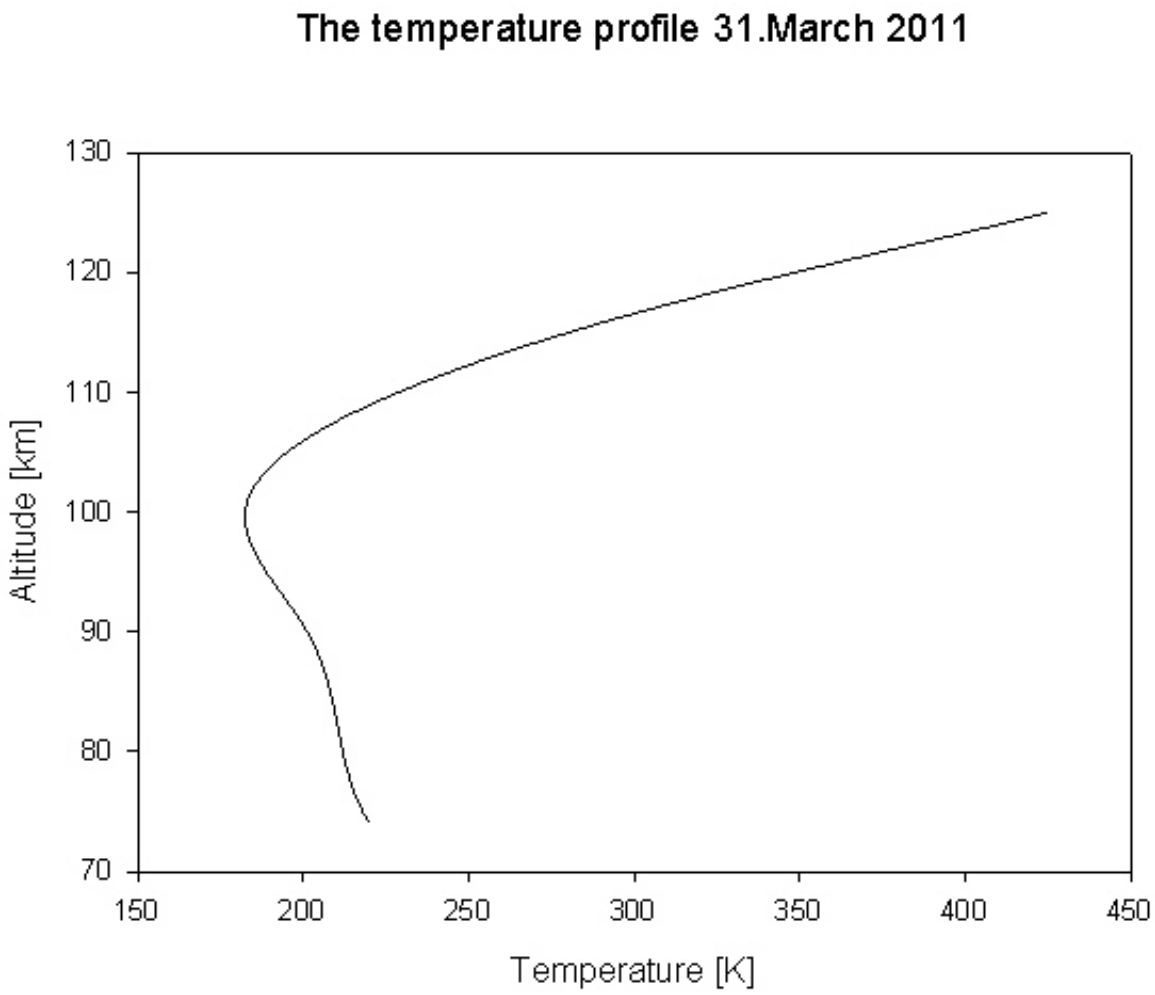

Figure B.19.: Predicted temperature profile 31. march 2011.

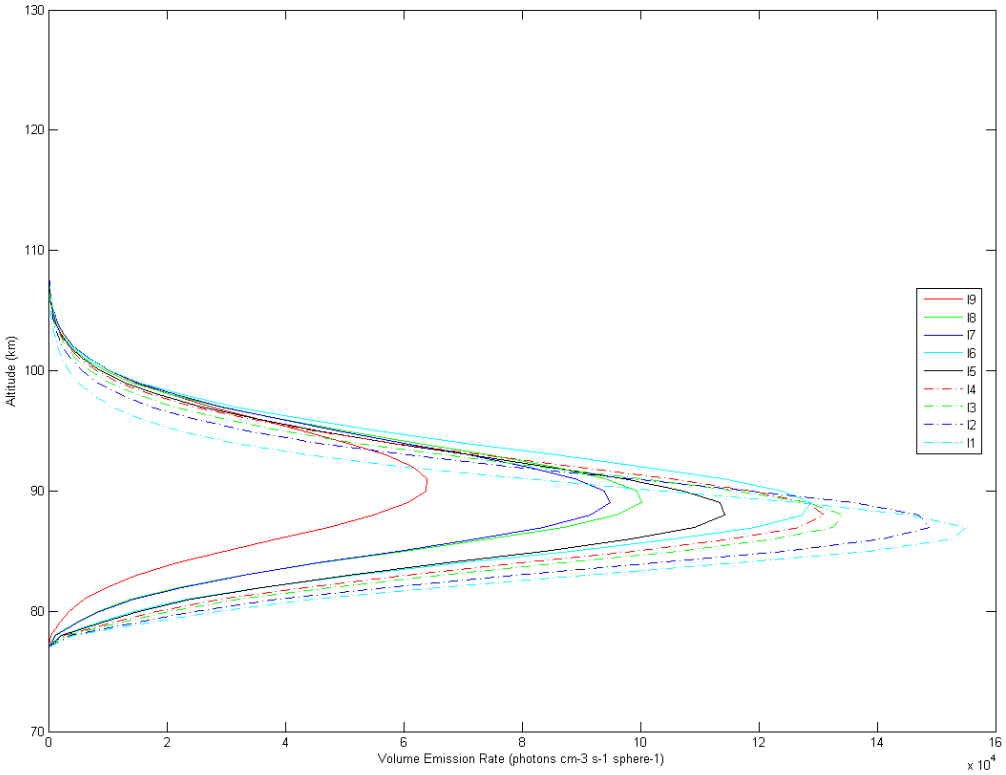

Figure B.20.: The predicted intensity of each vibrational state of the OH molecule, 31. march 2011.

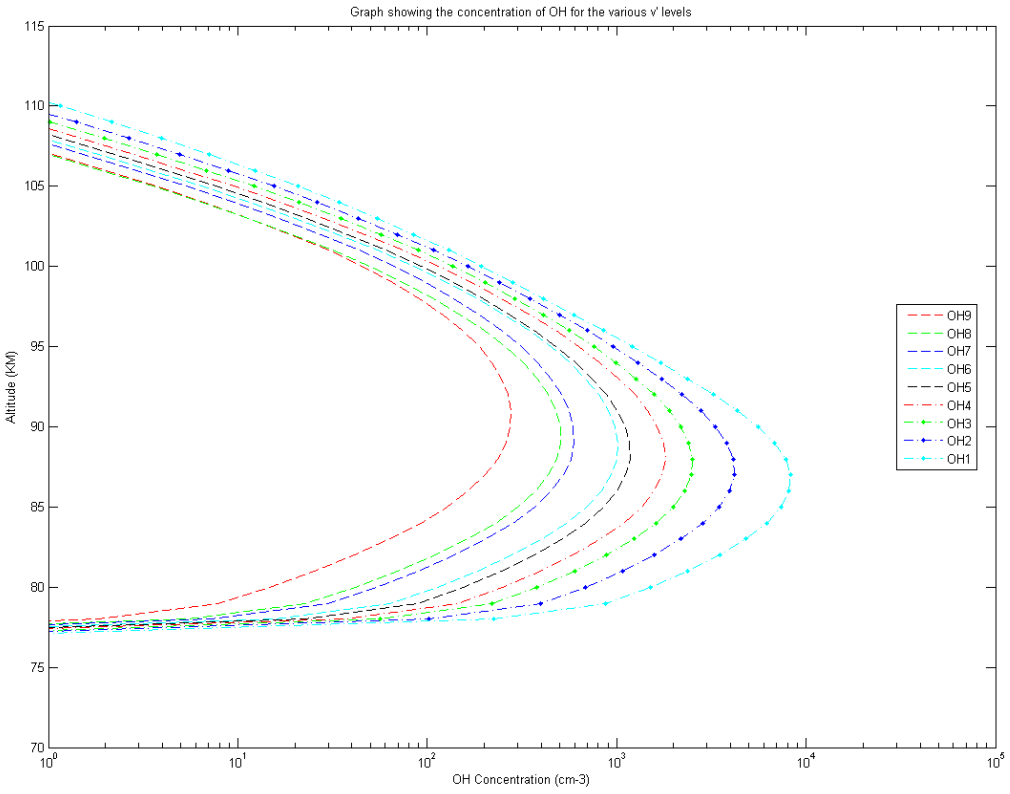

Figure B.21.: The predicted concentration of each vibrational state of the OH molecule, 31. march 2011.

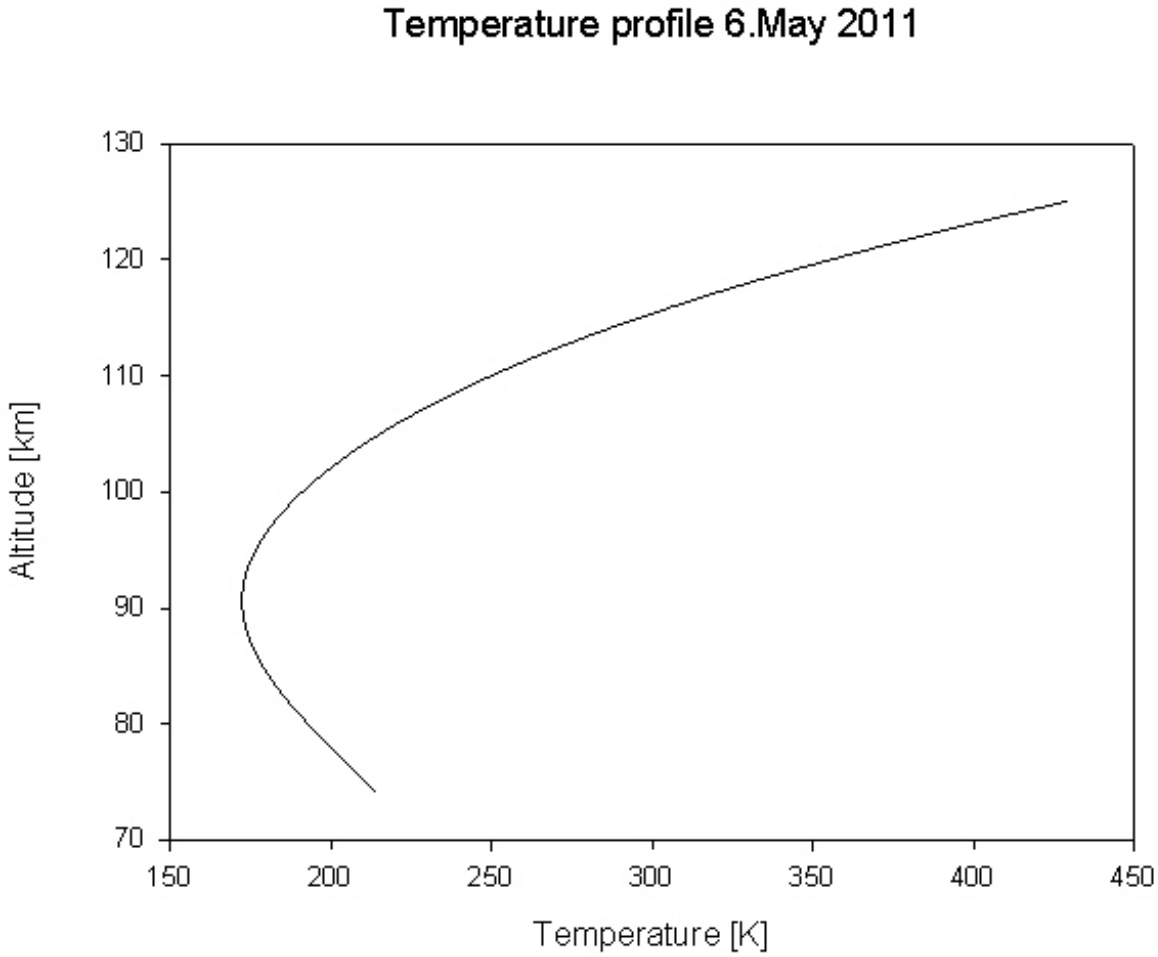

Figure B.22.: Predicted temperature profile 6. may 2011.

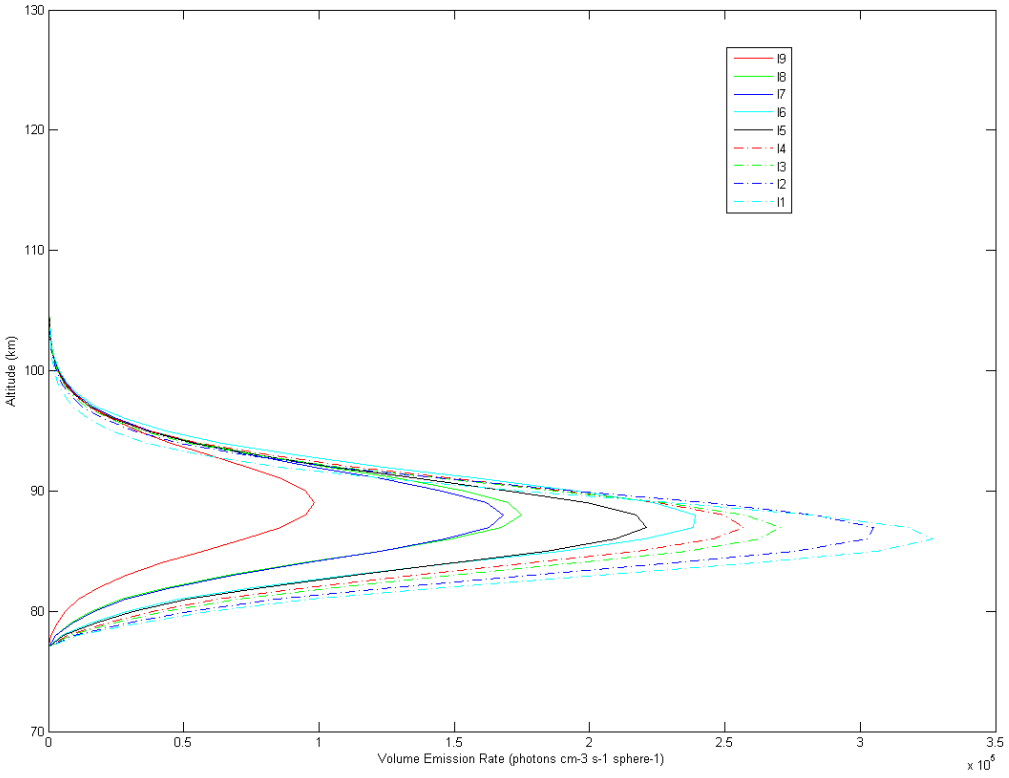

Figure B.23.: The predicted intensity of each vibrational state of the OH molecule, 6. may 2011.

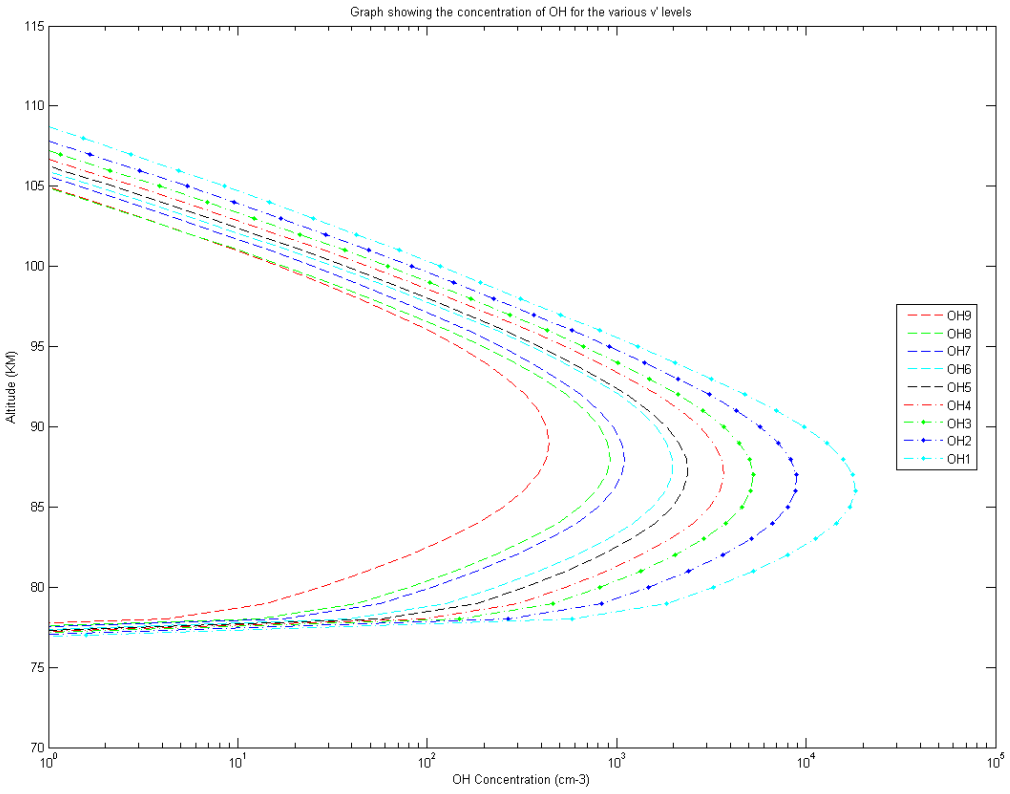

Figure B.24.: The predicted concentration of each vibrational state of the OH molecule, 6. may 2011.

C. Code

C.1. Main.m

```
addpath('C:\Program Files\Andor SOLIS\Drivers\')
fid=fopen('schedule.txt');
comment=fgetl(fid);
lat=str2num(fgetl(fid));
long=str2num(fgetl(fid));
alt=str2num(fgetl(fid));
savelocation=fgetl(fid);
[SZA1,SZA2,int,AT,BGDC]=textread('schedule.txt','%d %d %d %d %d','headerlines',5);
loadlibrary('ATMCD32D');
[a,b]=calllib('ATMCD32D','Initialize','C:\Program Files\Andor SOLIS\Drivers');
    if(a==20002)
        disp('Initialized!')
    else
        disp('Failed to initialize!')
    end
a=calllib('ATMCD32D','SetShutter',0,2,2,2);
temp=-60;
a=calllib('ATMCD32D','SetTemperature',temp);
a=calllib('ATMCD32D','CoolerON');
a=0;
while(a~=20036)
    [a,b]=calllib('ATMCD32D','GetTemperature',temp);
    disp('Temperature is: ')
    disp(b)
    if(a==20036)
        disp('Temperature is stabilized!')
    end
    if(a~=20036)
        pause(30)
    end
end
f=figure('Position',[50 600 1 50]);
u=uicontrol('string','stop','callback','delete(get(gcbo,''parent''))');
bgsig=0;
garbage=scan(0,1,[10 20],[30 60],0);
interval=0;
bg=zeros(1,1024);
while ishandle(f)
    previousinterval=interval;
    [check,sza,interval]=awty(SZA1,SZA2,lat,long);
    if(previousinterval~=interval)
```

```
        bgsig=0;
    end
    if(check==1)
        data=scan(bgsig,interval,int,AT,sza);
        if(bgsig==0)
            bg=data(12:1035);
        else
            figure(2)
            plot(1:1024,data(12:1035)-bg(1,:))
        end
        bgsig=bgsig+1;
        if(bgsig>=BGDC(interval))
            bgsig=0;
        end
        savedata(data,savelocation,long);
    end
    if(check==0)
        pause(30)
        bgsig=0;
        time=clock;
        times=strcat('Waiting to acquire data... Time: ',datestr(time),'
    sza:',num2str(sza));
        disp(times)
    end
end
a=calllib('ATMCD32D','CoolerOFF');
while(b<-20)
    [a,b]=calllib('ATMCD32D','GetTemperature',temp);
    disp('Temperature is: ')
    disp(b)
    if(b>=-20)
        disp('Temperature is above -20')
        disp('Starting shutdown in 30 seconds!')
    end
    pause(30)
end
a=calllib('ATMCD32D','ShutDown');
```

C.2. awty.m

```
function [check,SZA,interval] = awty(sza1,sza2,lat,long)
time=clock;
month=time(2);
day=time(3);
hour=time(4);
min=time(5);
sec=time(6);
Pi=3.14128;
GMT=hour+(min/60)+(sec/3600);
longitude=long;
latitude=lat;
latitude=latitude/57.296;
darray=[0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 303, 334];
dn=darray(month)+day-1;
N=(2*Pi*dn)/365;
EQT=0.000075+(0.001868*cos(N))-(0.032077*sin(N))-(0.014615*cos(2*N))
-(0.040849*sin(2*N));
th=Pi*((GMT/12)-1+(longitude/180))+EQT;
delta=0.006918-(0.399912*cos(N))+(0.070257*sin(N))-(0.006758*cos(2*N))
+(0.000907*sin(2*N))-(0.002697*cos(3*N))+(0.00148*sin(3*N));
cossza=(sin(delta)*sin(latitude))+(cos(delta)*cos(latitude)*cos(th));
SZA=(atan(-cossza/sqrt(-cossza*cossza+1))+ 2*atan(1))*57.296;
interval=0;
check=0;
for i=1:length(sza1)
    if(SZA>=sza1(i) && SZA<=sza2(i))
        check=1;
        interval=i;
    end
end
return
```

C.3. scan.m

```
function data = scan(bgsig,interval,int,AT,sza)
if(bgsig==0)
    tag=0;
else
    tag=1;
end
if(tag==0)
    disp('Taking background, total accumulation time is:')
    disp((round(AT(interval)/int(interval))*int(interval)))
else
    disp('Taking signal, total accumulation time is:')
    disp((round(AT(interval)/int(interval))*int(interval)))
    a=calllib('ATMCD32D','SetShutter',0,1,2,2);
end
ec=calllib('ATMCD32D','SetAcquisitionMode',2);
ec=calllib('ATMCD32D','SetExposureTime',int(interval));
ec=calllib('ATMCD32D','SetTriggerMode',0);
ec=calllib('ATMCD32D','SetNumberAccumulations',round(AT(interval)/int(interval)));
ec=calllib('ATMCD32D','SetAccumulationCycleTime',int(interval));
ec=calllib('ATMCD32D','StartAcquisition');
tempdata=1:1024;
ec=20072;
pause(int(interval)/2)
while(ec==20072)
[ec,tempdata]=calllib('ATMCD32D','GetAcquiredData',tempdata,1024);
end
[ec,realtemp]=calllib('ATMCD32D','GetTemperature',-60);
time=clock;
data=1:1035;
data(1)=time(1);%year
data(2)=time(2);%month
data(3)=time(3);%day
data(4)=time(4);%hour
data(5)=time(5);%min
data(6)=time(6)*1000;%millisec
data(7)=round(AT(interval)/int(interval));
data(8)=int(interval);
data(9)=sza;
data(10)=tag;
data(11)=realtemp;
for i=1:1024
    data(11+i)=tempdata(1025-i);
end
if(tag==1)
   a=calllib('ATMCD32D','SetShutter',0,2,2,2);
end
return
```

C.4. savedata.m

```
function savedata(data,path,long)
time=clock;
month=time(2);
day=time(3);
hour=time(4);
min=time(5);
darray=[0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 303, 334];
dn=darray(month)+day-1;
if(long<=180)
    diff=long/15;
else
    diff=(long-360)/15;
end
tln=12-diff;
newtime=hour+(min/60);
if(newtime<tln)
    dday=dn-1;
else
    dday=dn;
end
ddaystring=num2str(dday);
yearstring=num2str(time(1));
namestring=strcat(yearstring,'-',ddaystring);
fullpath=strcat(path,namestring,'.mat');
a=exist(fullpath,'file');
first=0;
if(a==2)
    load(fullpath,'-mat')
else
    daymat=zeros(1,1035);
    first=1;
end
[l,w]=size(daymat);
if(first)
    daymat(1,:)=data;
else
    daymat(l+1,:)=data;
end
save(fullpath,'daymat')
clear daymat
return
```

C.5. tempregmain.m

```
addpath('M:\public_html\master\prog\Resultat 31-03-11')
load 2011-89.mat
vector=startinput(daymat);
intv=integration(vector);
tempreg(intv)
```

C.6. startinput.m

```
function vektor=startinput(daymat)
[l,w]=size(daymat);
start=input('Start:');
endp=input('End:');
bg=input('bg:');
if(start<0 || start>=l || endp<1 || endp>l || bg<0 || bg>l)
   disp('Invalid entry points!')
   exit
end
sumv=zeros(1,1024);
for i=start:endp
    sumv=sumv+daymat(i,12:1035)-daymat(bg,12:1035);
end
sumv=sumv/(endp-start);
plot(sumv,'red')
hold
lowp=zeros(1,21);
lowpx=zeros(1,21);
lowp=lowp+10^19;
c=0;
intv=1;
for i=1:1024
    if(sumv(1,i)<lowp(1,intv))
        lowp(1,intv)=sumv(1,i);
        lowpx(1,intv)=i;
    end
    c=c+1;
    if(c==50)
        c=0;
        intv=intv+1;
    end
end
plot(lowpx,lowp,'+')
p=polyfit(lowpx,lowp,2);
lowpp=zeros(1,1024);
lowpp(:)=polyval(p,1:1024);
plot(lowpp,'green')
disp('Press any key to continue...')
pause
vektor=zeros(1,1024);
for j=1:1024
    vektor(1,j)=sumv(1,j)-polyval(p,j);
end
hold
plot(vektor)
disp('Press any key to continue...')
pause
end
```

C.7. integration.m

```
function intgv = integration(vektor)
lines=[313 342 365 389 422 440 725 755 783 808];
l=length(vektor);
addpath('M:\public_html\master\');
respreg_kr % calculates the response curve
for k=1:l
   vektor(k)=vektor(k)/nycc(k,2); %nycc is the response curve
end
plot(vektor)
disp('Press any key to continue...')
pause
p=length(lines);
intgv=zeros(1,p);
for j=1:p
    half=vektor(lines(j))/2;
    midl=vektor(lines(j));
    count=1;
    tl=lines(j)-1;
    tr=lines(j)+1;
    while(vektor(tl)>half && count<7)
        midl=midl+vektor(tl);
        count=count+1;
        tl=tl-1;
    end
    while(vektor(tr)>half && count<14)
        midl=midl+vektor(tr);
        count=count+1;
        tr=tr+1;
    end
    intgv(j)=midl/(count*0.2071);
end
return
```

C.8. tempreg.m

```
function tempreg(intgv)
E=[10300.41 10172.30 10354.21 10247.07 10443.29 10352.45 13377.61 13248.92 ...];
S=[3.9839*10^11 4.9798*10^11 7.3932*10^11 9.0706*10^11 1.0852*10^12 ...];
final=zeros(2,10);
final(1,:)=E(:)/0.6950356;
final(2,:)=log(intgv(:)/S(:));
plot(final(1,:),final(2,:),'+')
start=[1,7];
stop=[6,10];
nptsv=[6,4];
addpath('M:\public_html\master\prog\f2matlab\fitaexp');
mwt=0;
for i=1:2
    p=final(1,start(i):stop(i));
    in=final(2,start(i):stop(i));
    sig=zeros(1,nptsv(i));
    sig(:)=sig(:)+1;
    npts=nptsv(i);
    [npts,p,in,sig]=sort3(npts,p,in,sig);
    [icpt,slp,sigicp,sigslp,chi2,q]=fit(p,in,npts,sig,mwt);
    disp(' Fit to Ln(y)=Ln(a)+b*x ')
    disp(' Slope    = ')
    disp(slp)
    disp(' +/- ')
    disp(sigslp)
    disp(' Intercept= ')
    disp(icpt)
    disp(' +/- ')
    disp(sigicp)
    disp(' Chi**2   = ')
    disp(chi2)
    if(slp~=0)
            revslp=1/slp;
            delrsl=sigslp/slp^2;
            revic=-icpt/slp;
            drevic=abs(sigicp/slp)+abs(icpt*sigslp/slp^2);
            disp(' If x and y axis are reversed, then')
            disp('    y=mx+b+>')
            disp(' m=')
            disp(revslp)
            disp(' +/- ')
            disp(delrsl)
            disp(' b=')
            disp(revic)
            disp(' +/- ')
            disp(drevic)
    end
end
return
```

C.9. Functions

```
function betacfresult=betacf(a,b,x)
itmax=100;
eps=3.0e-7;
am=1;
bm=1;
az=1;
qab=a+b;
qap=a+1;
qam=a-1;
bz=1-qab*x/qap;
for  m=1:itmax;
    em=m;
    tem=em+em;
    d=em*(b-m)*x/((qam+tem)*(a+tem));
    ap=az+d*am;
    bp=bz+d*bm;
    d=-(a+em)*(qab+em)*x/((a+tem)*(qap+tem));
    app=ap+d*az;
    bpp=bp+d*bz;
    aold=az;
    am=ap/bpp;
    bm=bp/bpp;
    az=app/bpp;
    bz=1;
    if(abs(az-aold)<eps*abs(az))
        betacfresult=az;
        return
    else
        error('A or B too big, or ITMAX too small');
    end
end
```

```
function betairesult=betai(a,b,x)
if( x<0 || x>1 )
    error('bad argument X in BETAI')
end
if( x==0 || x==1 )
    bt=0;
else
    bt=exp(gammln(a+b)-gammln(a)-gammln(b)+a*log(x)+b*log(1-x));
end
if(x<(a+1)/(a+b+2))
    betairesult=bt*betacf(a,b,x)/a;
    return
else
    betairesult=1-bt*betacf(b,a,1-x)/b;
    return
end
```

```
function [a,b,siga,sigb,chi2,q]=fit(x,y,ndata,sig,mwt)
sx=0;
sy=0;
st2=0;
b=0;
if(mwt~=0)
    ss=0;
    for  i=1:ndata;
        wt=1/(sig(i)^2);
        ss=ss+wt;
        sx=sx+x(i)*wt;
        sy=sy+y(i)*wt;
    end
else
    for  i=1:ndata;
        sx=sx+x(i);
        sy=sy+y(i);
    end
    ss=(ndata);
end
sxoss=sx/ss;
if(mwt~=0)
    for  i=1:ndata;
        t=(x(i)-sxoss)/sig(i);
        st2=st2+t*t;
        b=b+t*y(i)/sig(i);
    end
else
    for  i=1:ndata;
        t=x(i)-sxoss;
        st2=st2+t*t;
        b=b+t*y(i);
    end
end
b=b/st2;
a=(sy-sx*b)/ss;
siga=sqrt((1+sx*sx/(ss*st2))/ss);
sigb=sqrt(1/st2);
chi2=0;
if(mwt==0)
    for  i=1:ndata;
        chi2=chi2+(y(i)-a-b*x(i))^2;
    end
q=1;
sigdat=sqrt(chi2/(ndata-2));
siga=siga*sigdat;
sigb=sigb*sigdat;
else
    for  i=1:ndata;
        chi2=chi2+((y(i)-a-b*x(i))/sig(i))^2;
    end
    q=gammq(0.5*(ndata-2),0.5*chi2);
end
return
```

```
function gammlnresult=gammln(xx)
cof=[76.18009173,-86.50532033,24.01409822,-1.231739516,
.00120858003,-.536382*10^-5];
x=xx-1;
tmp=x+5.5;
tmp=(x+0.5)*log(tmp)-tmp;
ser=1;
for j=1:6
    x=x+1;
    ser=ser+cof(j)/x;
end
gammlnresult=tmp+log(2.50662827465*ser);
return


function gammqresult=gammq(a,x)
if(x<0.||a<=0.)
    error('Invalid arguments in routine GAMMQ')
end
if(x<a+1)
    gamser=gser(a,x);
    gammqresult=1-gamser;
else
    gammqresult=gcf2(a,x);
end
return


function gammcf=gcf2(a,x)
itmax=100;
eps=3.0e-7;
gln=gammln(a);
gold=0;
a0=1;
a1=x;
b0=0;
b1=1;
fac=1;
for  n=1:itmax;
    an=(n);
    ana=an-a;
    a0=(a1+a0*ana)*fac;
    b0=(b1+b0*ana)*fac;
    anf=an*fac;
    a1=x*a0+anf*a1;
    b1=x*b0+anf*b1;
    if(a1~=0)
        fac=1/a1;
        g=b1*fac;
        if(abs((g-gold)/g)<eps)
            gammcf=exp(-x+a*log(x)-gln)*g;
            return
        end
```

```
        gold=g;
    end
end
error('A too large, ITMAX too small')


function gamser=gser(a,x)
itmax=100;
eps=3.0*10^-7;
gln=gammln(a);
if(x<=0)
    if(x<0)
        error('x less than 0 in routine GSER')
    end
    gamser=0;
    return
end
ap=a;
sum=1/a;
del=sum;
for n=1:itmax;
    ap=ap+1;
    del=del*x/ap;
    sum=sum+del;
    if(abs(del)<abs(sum)*eps)
        gamser=sum*exp(-x+a*log(x)-gln);
        return
    end
end
error('A too large, ITMAX too small');


function [n,arrin,indx]=indexx(n,arrin)
indx=zeros(1,n);
for  j=1:n
    indx(j)=j;
end
l=(n/2)+1;
ir=n;
dummy=0;
while(dummy==0)
    if(l>1)
        l=l-1;
        indxt=indx(l);
        q=arrin(indxt);
    else
        indxt=indx(ir);
        q=arrin(indxt);
        indx(ir)=indx(1);
        ir=ir-1;
        if(ir==1)
           indx(1)=indxt;
           return
```

```
        end
    end
    i=l;
    j=l+l;
    while(j<=ir)
        if(j<ir)
            if(arrin(indx(j))<arrin(indx(j+1)))
                j=j+1;
            end
        end
        if(q<arrin(indx(j)))
            indx(i)=indx(j);
            i=j;
            j=j+j;
        else
            j=ir+1;
        end
    end
    indx(i)=indxt;
end


function [r,prob]=pearsn(x,y,n)
tiny=1.0e-20;
ax=0;
ay=0;
for  j=1:n
    ax=ax+x(j);
    ay=ay+y(j);
end
ax=ax/n;
ay=ay/n;
sxx=0;
syy=0;
sxy=0;
for  j=1:n;
    xt=x(j)-ax;
    yt=y(j)-ay;
    sxx=sxx+xt^2;
    syy=syy+yt^2;
    sxy=sxy+xt*yt;
end
r=sxy/sqrt(sxx*syy);
df=n-2;
t=r*sqrt(df/(((1-r)+tiny)*((1+r)+tiny)));
prob=betai(0.5*df,0.5,df/(df+t^2));
return


function [n,ra,rb,rc]=sort3(n,ra,rb,rc)
[n,ra,iwksp]=indexx(n,ra);
wksp=zeros(1,n);
for  j=1:n
```

```
    wksp(j)=ra(j);
end
for  j=1:n
    ra(j)=wksp(iwksp(j));
end
for  j=1:n
    wksp(j)=rb(j);
end
for  j=1:n
    rb(j)=wksp(iwksp(j));
end
for  j=1:n
wksp(j)=rc(j);
end
for  j=1:n
rc(j)=wksp(iwksp(j));
end
return
```