Chapter 1

The Michelson Interferometer

1.1 Prelab

In this lab you have to find the position of mirror M1 (see Fig. 1.1) such that the optical path length from the beam splitter to M1 and back is identical to the optical path length from the beam splitter to M2 and back again. This special position insures the path length difference is zero and is referred to as the "zero path length" ZPL. As the position of M1 approaches the position of ZPL, you will observe that the periodicity of the circular interference fringes (which are circular for the lamps - sodium and the white light) becomes larger and they are the largest at the ZPL (see Fig. 1.2). Explain qualitatively why this is. Next, model this situation numerically. Derive an expression for the interference pattern as a function of z from two spherical waves where the sources are separated on the z-axis by a distance d apart (representing the two different beam paths in the interferometer). *Hint: write down the electric field amplitudes (U_1 and U_2) from two spherical wave sources and compute the total intensity* $I = |U_1 + U_2|^2$. Plot the total intensity (i.e. interference pattern) for two different separations d. Comment on the change of the ring periodicity as d gets smaller. Also, for a given d comment on the ring periodicity as you move radially away from the z-axis along x or y.

In this lab, you will also use the interferometer to measure the index of refraction of air by putting a vacuum cell in between the beamsplitter and mirror M2 and measuring the additional phase shift acquired when you slowly introduce air into the evacuated cell. How many times does the beam traverse the cell in the interferometer? Compute the total optical path length difference for light at 632.8 nm with and without air in a vacuum cell 10 cm in length. What is the expected number of fringes that will pass as air is leaked into a cell of this length initially under a perfect vacuum?

Also, before doing this lab, you should prepare a **MATLAB** `.m` file which performs the analysis outlined in section 1.10.2. The code for this **MATLAB** script is spelled out in this section, and you should test it with the sample Fourier spectroscopy data (of a white light spectrum) provided on the optics lab webpage next to the link to the `.pdf` of this document. The sample data is

given in both `.csv` format and as a saved **MATLAB** workspace.

1.2 Introduction

A well known interferometer, operating on the principle of division-of-amplitude, is the Michelson interferometer. The incident light beam is divided into two parts by means of a beam-splitter. The divided beams traverse different paths and then are recombined. Depending on the difference in optical path lengths of the two beams, the recombined beams may be 180° out of phase, producing destructive interference, or in phase, producing constructive interference. Observation of these interference fringes, with changing optical path difference allows one to determine wavelengths, wavelength differences, and/or small differences between the two optical paths.

Use of this interferometer as a spectrometer to measure the power spectrum of an emitting source became feasible with the advent of high speed, large storage capacity, computers which could calculate the Fourier transform of the output irradiance of the interferometer. You will use this instrument to realize such a Fourier transform spectrometer.

1.3 The Michelson Interferometer

Fig. 1.1 illustrates the physical arrangement of the Michelson interferometer. The beam-splitter, oriented at 45° to the incoming light, is partially metallized on one surface reflecting half of the beam to mirror M1, and transmitting the other half of the beam to mirror M2. After reflection from these plane mirrors the two beams are recombined at the beam-splitter, and part of this light passes out of the interferometer towards the CCD camera. Mirror M1 is mounted on a computer controlled stage which can be moved via software or the jog/toggle buttons on the DC servo controller. **You will notice there is a second piece of glass in one arm of the interferometer. What is the purpose of this object?**

1.4 Finding the Fringes

Be sure that the vacuum cell is **not** in the interferometer for this part. Place the sodium lamp at the input of the interferometer and make sure the diffusing screen is in front of it (see Fig. 1.1). Note that the lamp will take a few minutes to warm up. The output of the interferometer will be examined using the CCD camera. In this case use the CCD camera with its lens assembly installed. Also, you should reset the translation stage zero so that your measurements are properly referenced. Open up the **APTUser** program on the desktop. Using the *Settings* button on the GUI make sure the max velocity is 0.5 mm/s for both the move and jog. Next, push the *Home/Zero* button to zero the stage. The stage will move to the end of its travel and the computer will note this as the home or zero position. The interferometer has been set so that the position of zero path difference falls near the centre of the range of motion of the computer-controlled stage that M1 sits on (you should ask the TA about where it might be since the exact position depends on the setup and changes from week to week

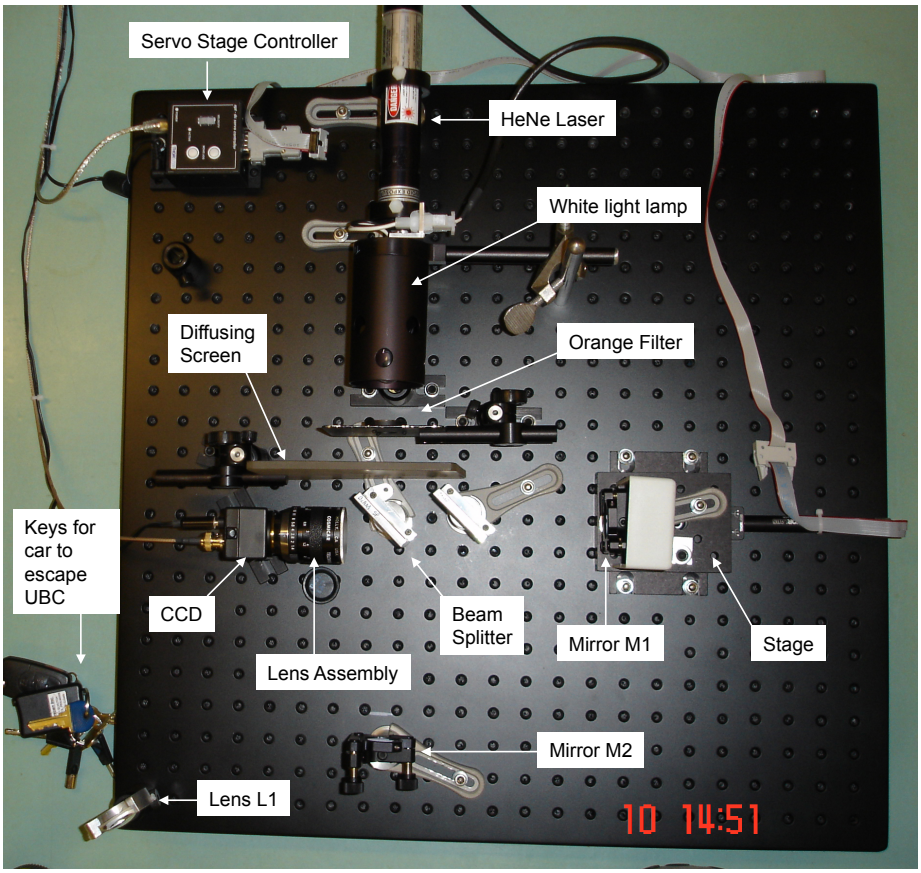


Figure 1.1: The Michelson Interferometer. Note, the sodium lamp is not pictured

due to misalignments). Position the stage near the center of the range (around 10 to 12 mm). Then align the mirrors by adjusting the two knobs of M2 and making the two images of the focussing pin coincide. Fringes should appear on the monitor. You will need to make sure the CCD camera is not saturated by adjusting the aperture on the lens assembly. Fringes of different shapes may be produced by appropriate adjustment of the mirror M2.

1.4.1 Circular Fringes

When the mirrors are effectively parallel to each other (which means they are actually perpendicular since the arms of the interferometer are perpendicular - see Fig. 1.2) circular fringes are produced (see Fig. 1.3). These fringes, normally formed at infinity, may be focussed on a screen (or onto the CCD camera chip) with the use of a lens, as shown in Fig. 1.2.

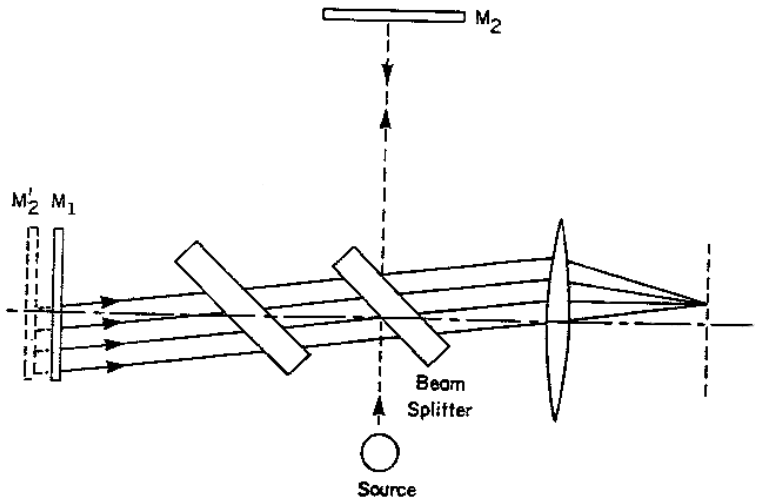


Figure 1.2: Formation of circular fringes. M_2' indicates the effective position of the mirror M_2. When both mirrors are effectively parallel to each other but at slightly different positions, the optical path lengths are different and circular fringes occur. These fringes are focused with a lens on to the CCD camera chip.

1.4.2 Straight Fringes

If the mirrors are slowly angled so that they are not effectively parallel to each other, the centre of the circular fringes moves away and the fringes will appear like a series of curved lines (the edges of a set of concentric circles). When the path length is set to be very near the ZPL, straight line fringes, or "fringes of equal thickness" occur (see Fig. 1.3). They are identical to the fringes formed by an air wedge; the smaller the wedge angle, the broader the fringes.

1.4.3 Changing from One Type of Fringe to Another

When the interferometer is set up, it is unlikely that the mirrors will be aligned or that stage will be already close to the zero path length. Therefore the fringes may be either horizontal or vertical, or anything in between. By adjusting the mirrors the fringes may be changed from one form to another. As the wedge angle between the mirrors is decreased, the fringes become more curved until circular fringes (theoretically) appear.

Find the fringes described above, and practice forming straight and circular fringes until fringes of a specific kind can be produced at will. Note that perfectly straight fringes will only occur right at the ZPL. In practice, of course, the fringes may also be distorted due to imperfections in the flatness of the mirrors. In fact, Michelson and other amplitude division interferometers are used in industrial fabrication plants exactly for this purpose - to verify the flatness or precise curvature of mirrors. Using the **WinTV2000** program on the desktop, record some images of various fringe patterns for your lab book. The GUI button *SNAP* will grab a frame for you. Right click on the captured image to save it

to a file.

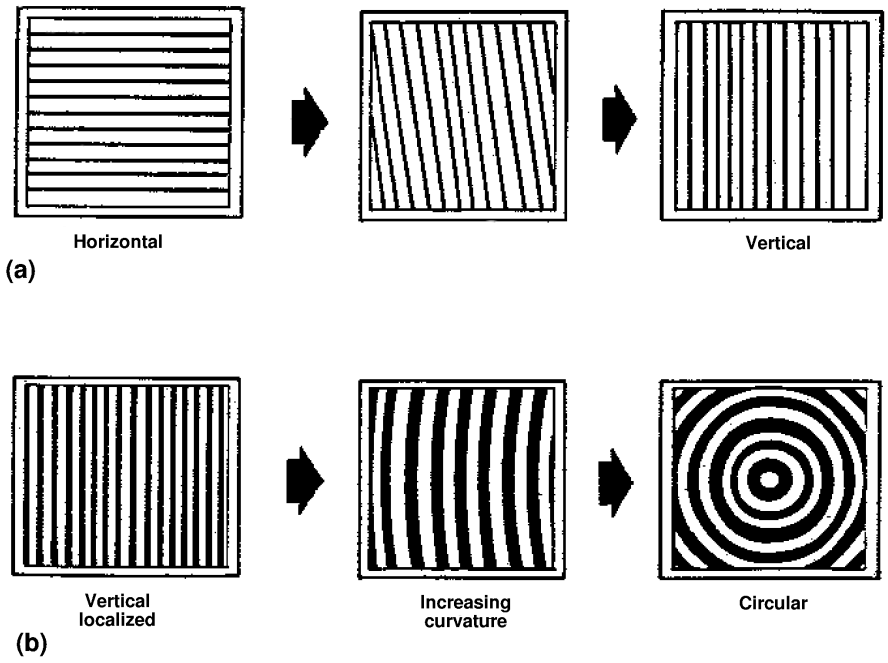


Figure 1.3: (a) Straight fringes and (b) circular fringes which occur when the misalignment of the mirrors is corrected so that they are effectively parallel (see Fig. 1.2).

1.5 The Position Of Zero Path Difference

For some applications in this lab, it will be necessary to locate the position of zero path difference. The following section describes how this is done.

1.5.1 Procedure

Be sure that the vacuum cell is **not** in the interferometer for this part.

If you haven't already done so, you should reset the stage zero so that your measurements are properly referenced. It will take some careful adjustment to find the exact position for the translation stage to achieve a zero path length difference. Therefore, start by setting up the sodium lamp and diffuser and search for the location with the strongest (highest contrast) fringes possible. Do your best to insure the fringes are circular (nicely concentric), well spaced, and clearly defined during your search.

Actually, you will find that there is a series of stage positions where the contrast is high then low and then high again (see Fig. 1.6). Take note of the exact stage positions where the contrast is minimum (or disappears) near the

stage position where the contrast appears best. You will next look systematically and precisely between each of these minima for the ZPL using the white light source.

Another sign that you are near the zero path length difference is that the radius of curvature of the circular fringes is the largest. **Why is that?** At this location, you should be within a fraction of one millimeter from the zero path length difference. To look for changes of the radius of curvature as you move the stage position, you could also look at the curvature of the straight fringes with the mirrors slightly angled.

Be very careful not to bump the equipment at this stage since you can easily knock the system out of alignment and ruin your pre-alignment work!

Now, replace the sodium lamp with the white light source. Note that the Sodium lamp will be hot, so only grab it by the post. You will notice that fringes do not appear. When using the white light source, they will only appear when the arms of the interferometer have equal path length. **Why? What does this tell you about the "coherence length" of the white light source? What is the definition of the "coherence length?"** You must adjust the stage until the path lengths are equal and fringes are observed.

Since finding the white light fringes is like looking for a needle in a haystack, you should cheat by using a narrow band filter with the white light source to find fringes with filtered light and then remove the filter to find the zero path length difference with the white light. **Why would this filter help?** With the narrow band filter in place, slowly scan the stage with the jog buttons to the positions precisely between the minima you found before. When you finally try the correct maxima, the stage will be near the ZPL and fringes will be observed. If you are spending more than 15 minutes without finding the white light fringes, ask a TA to help check you are doing everything correctly. At the point where the fringes appear with the unfiltered white light source, the stage is set for zero path length difference. Write down the stage position for future reference and note the direction you approached the zero-path difference. The latter is important for mechanical backlash in the micrometer screw. Since you will be using a **MATLAB** script to take data, and this script moves the stage forward (i.e. to higher values on the indexed stage), then backwards, then forward again back to the starting point, you should always approach from below your position of zero-path length difference. The fringes will appear and disappear within 2 μm, so you will probably have to adjust the max velocity and step distance of the stage so that you don't go too fast and miss the appearance of the fringes. Also, use the actuator toggle with care since rapid changes in motion of the stage might also misalign the interferometer.

1.6 Calibration Of The Interferometer

Before the interferometer can be used to measure wavelengths it must first be calibrated. By "calibrated" we mean that the distance which the stage actually moves needs to be correlated to the approximate reading provided by the computer.

1.6.1 Procedure

The distance by which the stage travels is found by counting the number of fringes which pass given a certain mirror movement and given a light source whose wavelength is well known. The Helium-Neon Laser (λ =632.8nm) is ideal for this purpose. Slide the lamp up and rotate the diffusing screen and orange filter out of beam path of the laser. When using the laser, place an imaging lens (L1) in the post just after the output of the laser. You should remove the lens assembly on the CCD camera. By adjusting the height and tilt of the imaging lens in the post holder, locate the the main spot of the HeNe laser on the backplane of the CCD camera. Temporarily put a piece of paper just in front of the CCD and look at the pattern cast on this paper. Align the mirror (M2) to see fringes. Now take out the paper and use the monitor to view the CCD output, and further adjust mirror M2 so that about five vertical straight fringes fill the image on the CCD camera (see Fig. 1.3).

Using the monitor to view the CCD output, adjust mirror M2 so that about five vertical straight fringes fill the image on the CCD camera (see Fig. 1.3). You should adjust the camera aperture and attach some neutral density filters to the CCD camera so that it isn't saturated.

The overall plan is to move the stage a certain distance and count the number of fringes that pass. You should therefore take the data, plot it, and count the fringes on the plot. The mirror moves a distance of $\lambda/2$ as one fringe passes a reference point on the CCD Camera. Thus, if N fringes pass through as the mirror moves a distance d, we have,

$$2d = N\lambda \tag{1.1}$$

If the servo controller readings change by ΔD from D_1 to D_2, then the mirror travel is linearly related to the stage displacement through the calibration constant K:

$$d = K\Delta D = K \mid D_2 - D_1 \mid \tag{1.2}$$

To analyze the fringe movement (i.e. to count the fringes), you will use **MATLAB** to control the translation stage (i.e. mirror) position AND to detect the variation of the intensity on a given pixel in the fringe pattern as the fringe pattern moves. Several helpful **MATLAB** files are located in the desktop folder "Michelson files". At this point you should NOT be running the **WinTV2000** program as it will not allow **MATLAB** to access the video card. By reading the comments of the M-files you will be able to tell what each of their functions are.

In order to take data for this section, you will be using the **MATLAB** script called "michelson_timestream.m". You will need to quit the APT user program when you use the "michelson_timestream" file since **MATLAB** will open APT user again with the correct parameters. When this script is executed, it will prompt you to click on the image the location where the fringe pattern is the most visible. The script will then move the stage back by a distance (`span`) specified in mm, then it will move the stage forward by twice that distance (`2 x span`) at a slow velocity (`speed`) specified in mm/s, and then it will move the stage back to the starting point. To use this script, your code will look like this

```
>> data = michelson_timestream(span, speed);
```

where `span` and `speed` are to be chosen by you, and the data will be returned into the array `data`. As an example, consider this code

```
>> data = michelson_timestream(0.01, 0.001);
```

This will move the stage back by 10 μm and then forward by 20 μm at a speed of 1 μm/s (which will take 20 seconds), and then back 20 μm to the starting point. The data will be returned into the array `data` which will have the intensity on the pixel you selected for each frame of the video. Since the video frame rate is 30 frames per second, this means the stage will have moved $\frac{1\ \mu m/s}{30\ \text{frames}/s} = \frac{33\ \text{nm}}{\text{frame}}$.

For a given amount of distance moved by the stage, determine the number of fringes that passed through the reference point (region of interest,ROI) and then calculate the calibration constant K. Repeat this measurement as many times as necessary to obtain consistent results for K and estimate its accuracy.

1.7 Measuring the index of refraction of air

Now that the constant K has been determined, you can use this interferometer to characterize extremely small changes in the optical path length differences between the two arms. You can use this sensitivity to even measure the phase shift induced by the propagation of light through air. You will measure the phase change by counting fringes as air is slowly introduced into one of the interferometer arms. Using the vacuum pump and glass vacuum cell provided, insert the glass vacuum cell between the beam-splitter and M2 and measure the index of refraction of air at 632.8 nm (i.e. with the HeNe laser).

For your measurements, you can use the `vid_capture` **MATLAB** script to capture the intensity on a small area in the CCD image as a function of time. Your code will look like this "`>> d = vid_capture(time);`" where you choose `time` appropriately (try 10 seconds).

To do your measurement, you should first prepare a vacuum in the cell: close the "leak valve" and open the "pump valve" and turn on the vacuum pump. After the pressure in the cell (as read by the meter) has dropped to the minimum on the scale, close the "pump valve" and turn off the pump. Start the data acquisition and slowly open the "leak valve" to let in air while you count fringes. You may need some practice introducing the air fast enough that the pressure rises to atmospheric pressure during the `time` you have set while not rising too quickly so that you miss the passage of fringes due to the finite sampling time of the video capture. Be careful to insert the glass vacuum cell without hitting or misaligning the mirrors (see Fig. 1.4).

Would you have counted the same number of fringes if you had used the Na lamp instead of the HeNe laser for this measurement? Why or why not? What is your value for the index of refraction of air for 632.8 nm light? Does this agree with the "accepted" value? What are your sources of error? Could you use this apparatus as a barometer or thermometer?

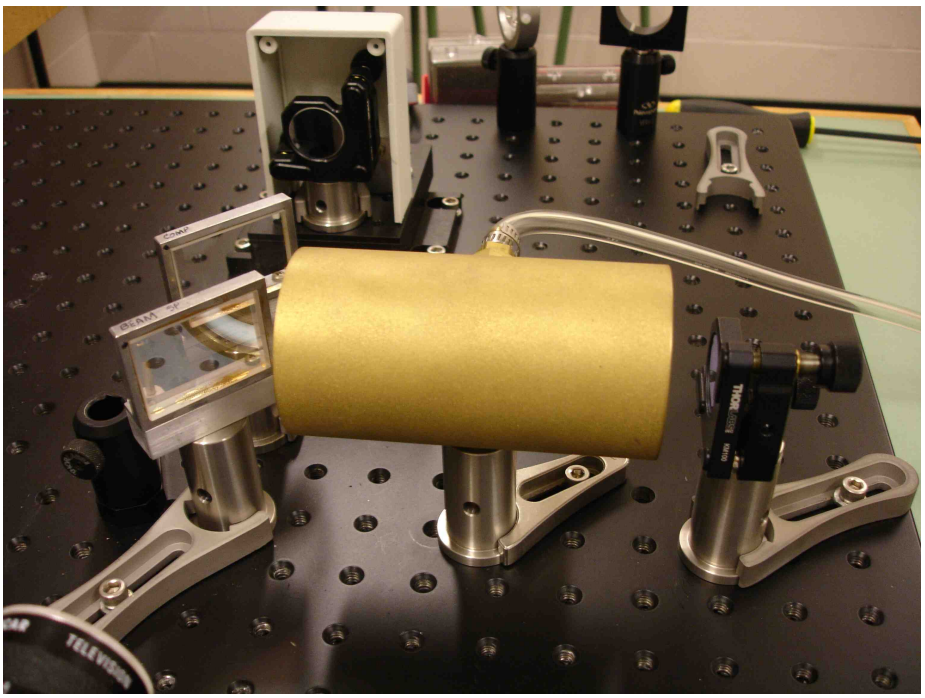


Figure 1.4: Vacuum vessel inserted into M2 arm of interferometer.

1.8 Determination Of Wavelengths

Be sure that the vacuum cell is **not** in the interferometer for this part.

Also, knowing the calibration constant K, an unknown wavelength may be measured by the same procedure; i.e., by noting the stage readings for the passage of a certain number of fringes. Carry out this measurement for the sodium source to get the average wavelength for the two Na D1 and D2 lines. You should use the same span and speed as the previous section. The fringes for these two lines coincide at the position of zero path difference and the measurement should be made near this position. On either side, the fringes become indistinct as they start to overlap. This feature will be studied in the next section.

1.9 Wavelength Differences

In 1887 Michelson and Morley, while studying the "visibility of fringes" in Michelson's interferometer, observed that the H2 line ($n = 3 \rightarrow n = 2$) is not one line but actually a doublet with about $\tilde{\nu} = 0.33$ cm^{-1} separation. The wavenumber, $\tilde{\nu}$, is given in units of inverse centimeters and is the spatial analog of frequency. One can convert to frequency by multiplication with the speed of light, $f = c\tilde{\nu}$. The wavenumber of electromagnetic radiation is defined as $\tilde{\nu} = 1/\lambda$, where λ is the wavelength in vacuum.

In this part of the lab, you will determine the separation between the D1 and D2 lines in sodium using the Na lamp. The fringes from any two lines

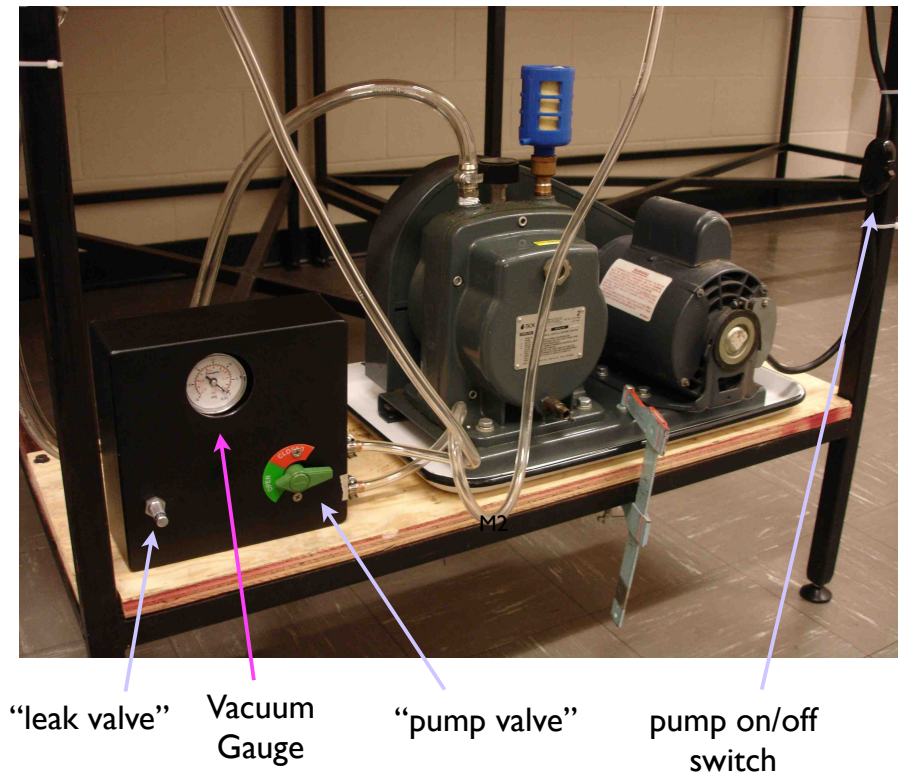


Figure 1.5: Vacuum pump, valves, and vacuum gauge for vacuum cell.

(wavelengths λ_1 and λ_2), such as the Na D1 and D2 lines, always overlap at the position of zero path difference. If the stage is moved away from this position, either forwards or backwards, then a position will be reached (at a distance d) where the bright fringes of one line fall on the dark fringes of the other. The resulting fringe pattern is then very indistinct − the fringes are said to have low visibility. This particular distance is

$$2d = (N+1)\lambda_1 = N\lambda_2 \qquad (\lambda_1 < \lambda_2) \tag{1.3}$$

where N is an integer. The wavelength difference is, therefore;

$$\Delta\lambda = \lambda_2 - \lambda_1 = 2d\left(\frac{1}{N} - \frac{1}{N+1}\right) = \frac{\lambda_1\lambda_2}{2d} \tag{1.4}$$

If λ_1 and λ_2 differ by very little, it is sufficient to replace their product by λ^2, where λ is the average of the two. The expected value of $\Delta\lambda$ ∼0.6nm.

1.9.1 Procedure

Be sure that the vacuum cell is **not** in the interferometer for this part.

Set up a white light source and locate the white light fringes as you did previously. This will locate the stage at zero path difference. Recall that due to backlash the stage position will differ if you approach this position from above

versus from below. Since you will eventually be using a **MATLAB** script to take data, and this script moves the stage forward (i.e. to higher values on the indexed stage), then backwards, then forward again back to the starting point, you should always approach from below your position of zero-path length difference.

Replace the white lamp with the sodium lamp. Now move the stage using the toggle lever on the controller, move the stage over a distance of at least 2 mm noting the stage readings at several positions of minimum visibility. When recording the positions, you should always move the stage in same direction to avoid backlash errors. Repeat this process several times and average the stage readings. You can do this manually or you can use the script from before. Your code will again look like this

```
>> d = michelson_timestream(span, speed);
```

where you choose `span` and `speed` appropriately. A span = 1.0 and speed = 0.01 are probably good starting values for this part.

From the average value of ΔD obtained, and the measured calibration constant, K, calculate d from equation 1.2, and hence obtain a value for $\Delta\lambda$ from equation 1.4. Use the average value of the two Na D1 and D2 lines that you determined above and find the difference between the two lines.

1.10 Fourier Transform Spectroscopy

So far you have investigated the behavior of the interference fringe pattern as a function of the optical path difference for different optical sources. In general the fringe pattern intensity versus the optical path difference (or equivalently versus the time delay between the two interfering beams) is related to the *power spectrum* of the light entering the amplitude division interferometer by a Fourier transform. Therefore one can easily measure the optical power spectrum using the measurement of the intensity variation as a function of stage position.

Let's see how this works [1]. The electric field incident on the camera is the sum of the electric fields $\mathbf{E}_1$ and $\mathbf{E}_2$ arriving from M1 and M2 respectively after having passed through the beam splitter. Assuming for the moment that the input source is a purely *monochromatic* wave, we can characterize these electric fields with a vector amplitude (encoding the strength and polarization of the wave) and a spatial and time dependent phase:

$$\mathbf{E}_1 = \mathbf{A}_1 e^{i(\mathbf{k}_1\cdot\mathbf{r}-\omega t+\phi_1)} \tag{1.5}$$

$$\mathbf{E}_2 = \mathbf{A}_2 e^{i(\mathbf{k}_2\cdot\mathbf{r}-\omega t+\phi_2)}. \tag{1.6}$$

The phase includes terms ($\phi_1 = |\mathbf{k}_1|\, l_1$ and $\phi_2 = |\mathbf{k}_2|\, l_2$) which depend on the optical path lengths from the beamsplitter along path l_1 (encountering M1) and path l_2 (encountering M2). The intensity on the camera is simply the square of the total electric field:

$$I = |\mathbf{E}|^2 = \mathbf{E}\cdot\mathbf{E}^* = (\mathbf{E}_1+\mathbf{E}_2)\cdot(\mathbf{E}_1^*+\mathbf{E}_2^*) \tag{1.7}$$

$$= |\mathbf{A}_1|^2 + |\mathbf{A}_2|^2 + 2\mathbf{A}_1\cdot\mathbf{A}_2\cos\theta \tag{1.8}$$

$$= I_1 + I_2 + 2\sqrt{I_1 I_2}\cos\theta \tag{1.9}$$

[1] This derivation follows that in "Introduction to Modern Optics" by Grant R. Fowles

where we have assumed in the last step that the waves have the same polarization and that $\theta = (\mathbf{k}_1 - \mathbf{k}_2) \cdot \mathbf{r} + \phi_1 - \phi_2$. You might be wondering about a white light source which is probably completely unpolarized. Is this assumption still valid in that case? This assumption is justified since the two fields $\mathbf{E}_1$ and $\mathbf{E}_2$ are copies of the incident field generated by the beam splitter. As long as the beam splitter and the optics which follow preserve the polarization of the light, these two fields will arrive at the detector with exactly the same polarization (whatever it was at the input of the interferometer). **What could the beam splitter and following optics do to change the polarization of the input light?**

If the interferometer is well aligned so that the waves are co-linear ($\mathbf{k}_1 = \mathbf{k}_2$) and the 50/50 beamsplitter generates two waves of the same intensity ($I_1 = I_2 = I/2$), then the interference pattern is simply

$$I(x) = I(1 + \cos kx) \tag{1.10}$$

where $x = l_1 - l_2$ is the path length difference and $k = |\mathbf{k}_1| = 2\pi/\lambda$ is the magnitude of the wavevector.

Okay, now in general, the light illuminating the interferometer will be composed of *many* different frequencies. So, let's model the power spectrum of the incident light as a continuous distribution, $G(\omega)$. Then the total optical power emitted into the interferometer in the frequency range from ω_0 to $\omega_0 + d\omega$ (where $d\omega$ is an infinitesimally small frequency band) is simply $G(\omega_0)d\omega$. For convenience, we can instead use the distribution over wavelength $G(k)$ (where $\omega = ck$). Then the optical power, P, hitting the detector from illuminating the interferometer with this polychromatic source will be the sum of the interference patterns from each monochromatic wave composing $G(k)$. Assuming the interferometer is ideal and has no losses, we have then:

$$\begin{aligned} P(x) &= \int_0^\infty G(k)\,(1 + \cos kx)\,dk & (1.11)\\ &= \int_0^\infty G(k)\,dk + \int_0^\infty G(k)\frac{e^{ikx} + e^{-ikx}}{2}\,dk & (1.12)\\ &= \frac{1}{2}P(0) + \frac{1}{2}\int_{-\infty}^\infty G(k)e^{ikx}\,dk & (1.13) \end{aligned}$$

where $P(0)$ is simply the power measured at zero path length difference. We can rearrange this expression and define the power function $W(x)$

$$W(x) = 2P(x) - P(0) = \int_{-\infty}^\infty G(k)e^{ikx}\,dk. \tag{1.14}$$

We see then that the power function $W(x)$ is the *inverse* Fourier transform of the power spectral density $G(k)$. Inverting this expression we have that the power spectrum is the Fourier transform of the power function

$$G(k) = \frac{1}{2\pi}\int_{-\infty}^\infty W(x)e^{-ikx}\,dx, \tag{1.15}$$

or equivalently

$$G(\nu) = \frac{1}{2\pi}\int_{-\infty}^\infty W(x)e^{-i2\pi\nu x/c}\,dx. \tag{1.16}$$

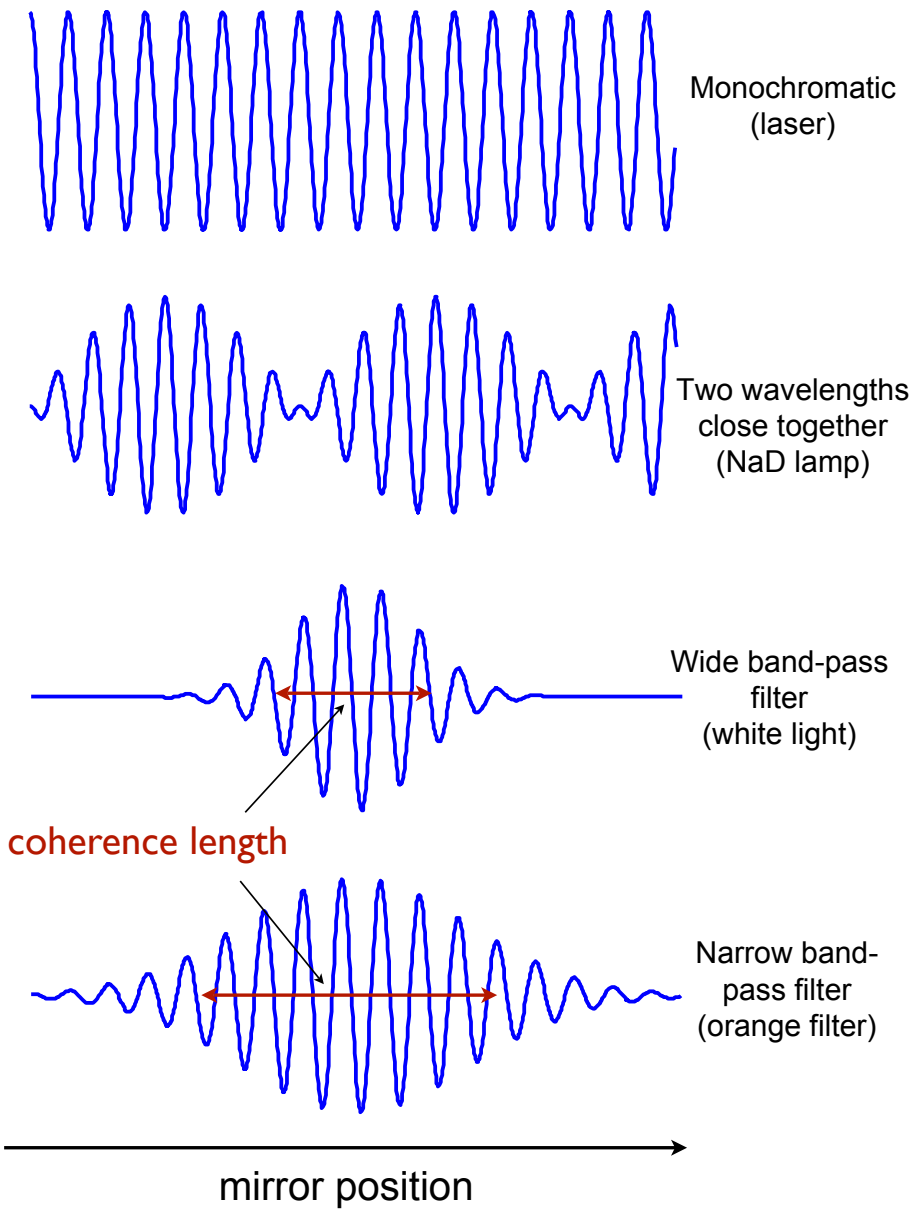


Figure 1.6: Fringe patterns plotted as a function of the mirror position. Notice how the fringe visibility collapses and then revives as a function of the mirror position when the spectrum is discrete. Also, note that if the mirror is moved a distance d, the optical path length between the two arms in the interferometer change by $2d$.

1.10.1 Procedure

Be sure that the vacuum cell is **not** in the interferometer for this part.

In this stage of the experiment, you will use the interferometer to realize

a Fourier transform spectrometer. The idea is to translate the stage around the zero path length position (ZPL), record the fringe pattern, and use it to determine the power function $W(x)$ and then compute numerically the corresponding spectral density $G(k)$ of various optical sources. To do this, you will (as before) use the **MATLAB** script "michelson_timestream.m" which records an array of power values for each video frame (i.e. each stage position). You will use your calibration constant for the motion, K, to deduce the stage position at each frame and therefore generate the power function $W(x)$ (be sure the stage velocities are set the same or else recalibrate K for this part). From this you will numerically compute the Fourier transform to determine the power spectrum of the light.

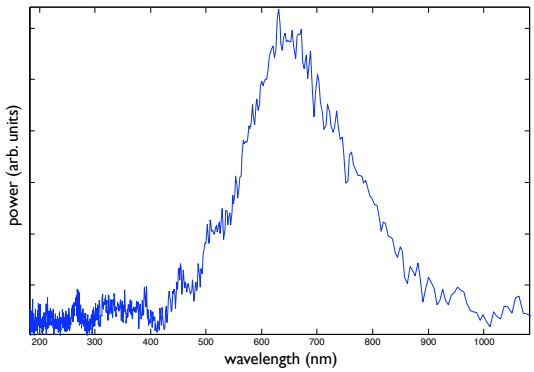


Figure 1.7: Example spectrum from the incandescent bulb measured by the Michelson Fourier transform spectrometer. Your reference white light spectrum should look something like this one. Keep in mind that this measurement of the spectrum is ultimately limited by the wavelength dependent transmission of the optical elements in the Michelson interferometer and by the sensitivity of the CCD detector. See Fig. 1.8 for a typical response curve of a silicon CCD image sensor.

As before, set up a white-light source, find the fringes at zero path length difference, make them vertical, and note the stage setting. Before taking each data run, you should carefully adjust the iris on the camera lens to maximize the fringe brightness without saturating the camera. This will optimize the signal to noise and produce much better data. Start by recording the fringe pattern and computing the spectrum of the white light source. You should get a spectrum similar to that shown in Fig. 1.7. This is your reference spectrum which you will be modifying with a filter. Keep in mind that this measurement of the white light source spectrum is ultimately limited by the transmission of the optical elements in the Michelson interferometer and by the sensitivity of the CCD detector. See Fig. 1.8 for a typical response curve of a silicon CCD image sensor.

To take the data for this section, you will again be using the **MATLAB** script "michelson_timestream.m", only this time, you should use a *very* slow `speed` and a very narrow `span` since the filtered white light has a short coherence length and you will need good spatial resolution (which will determine the

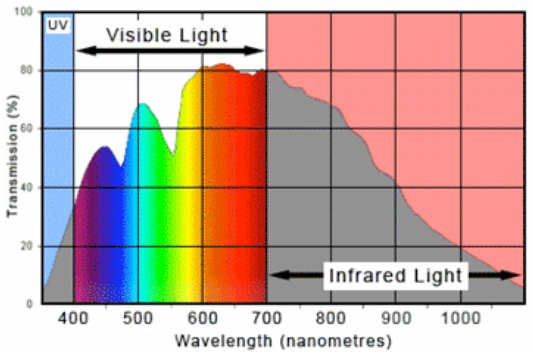


Figure 1.8: The spectral response of a typical silicon CCD image sensor ranges from 400 to 1000 nm.

highest frequency in your spectrum) so you can see the full spectrum without any truncation. Recommended values are `speed = 0.001` and `span = 0.05`. Note that this span might not be wide enough for the narrow band filter but that it already will take around 100 seconds given the speed! So, first check with a higher speed that this is the correct range for your filter and ONLY THEN take your data at the appropriate slow speed. Be aware: if you choose a span smaller than the backlash in the micrometer screw, the stage will not return to where you started. You should therefore insure the span is wide enough that the stage returns to the starting point properly.

Carefully (without disturbing the interferometer alignment) insert the 589 nm interference filter and record the fringe pattern generated. Take your spectrum and normalize it so that the peak of the spectrum has an amplitude of 1. Do the same with the reference spectrum and divide the filtered spectrum by the reference to deduce the transmission function of the filter. **Does the 589-nm interference filter have a Gaussian transmission function? Why or why not?**

You should now take a spectrum of the HeNe laser to check that your calibration of the stage velocity is correct. Is the peak of the spectrum where you would expect it? If not, why not? You should use this information to refine and/or correct the spectra you took of the filtered white light. What is the measured width of the spectral peak of the HeNe output in nm? The linewidth of the HeNe laser is well below 100 MHz. Is the measured width larger than this? If so, why?

1.10.2 Analysis with MATLAB

In order to compute the power spectra from your fringe pattern data, you will need to do some data processing. First, you will need to properly truncate your data because the vector of data, `data`, may include points when the stage was moving backwards and then sitting stationary before and after the slow sweep through the ZPL position. Therefore, find the points `AA` and `BB` which are at the beginning and end of the data and truncate the data. Your code will look

something like the following:

```
>> rawdata = michelson_timestream(span, speed);
>> data = rawdata(AA:BB);
```

Inspect a plot of your data to check that it has been properly truncated. Now, you should transform your vector of data into something equivalent to $W(x)$ (well equivalent up to a constant and a scale factor) by subtracting off the mean and then normalizing the data vector with

```
>> ndata = data-mean(data);
>> ndata = ndata/(max(ndata));
```

Now, to take the Fourier transform, you will use the "Fast Fourier Transform" function (`fft`) which, because of the structure of this super-efficient numerical algorithm, requires that you feed it a vector whose length is equal to a power of 2. Fortunately you can feed the function your vector and tell it the length is longer by rounding up to the next higher power of 2 (and the function will automatically pad your data with zeros at the end to fill in the gap). for this, use some code like the following:

```
>> L = length(ndata);
>> NFFT = 2^nextpow2(L);
>> fftdata = abs(fftshift(fft(ndata,NFFT)));
>> fftdata_oneside = fftdata(NFFT/2+2:NFFT);
>> nfftdata_oneside = fftdata_oneside/max(fftdata_oneside);
```

This will give you the absolute value of the Fourier transform since we aren't interested in the phase factors, only the amplitude versus wavelength. Also, we are only interested in the positive frequency components of the FT, so we will plot `fftdata_oneside`. The last line just normalizes the result by the max value. Why are the array limits `NFFT/2+2` and `NFFT` for the command "`>> fftdata_oneside = fftdata(NFFT/2+2:NFFT);`"? The reason is that the (N/2+1) element is the zero frequency bin and the (N/2) element is actually one of the negative frequency bins. The positive frequencies start at (N/2 + 2).

Now, the data contained in `fftdata` are the values of $|G(k)|$ (technically $|G_n|$ since we just have a list of values indexed by n) where the sample size is $\Delta_k = \frac{2\pi}{N_s \Delta_x}$. N_s = `NFFT` is the number of samples in the data which was "Fast Fourier Transformed" and Δ_x is the step size in, for example, nanometers between images. Remember since Δ_x is defined as the amount the optical path length varied between frames, it is twice the amount the carriage moved between images.

So, you know that each value of $|G_n|$ is paired with a wavevector of $k = n \times \Delta_k$. But you should plot your spectral data in terms of wavelength $|G(\lambda)|$ and so you need to create a vector of λ values corresponding to each point in $|G_n|$. Using the fact that $k = \frac{2\pi}{\lambda}$, we can write that $\lambda = \frac{2\pi}{n\Delta_k}$

```
>> n=1:1:NFFT/2-1;
>> lambda=zeros(size(fftdata_oneside));
>> lambda=NFFT*delta_x./n;
```

```
>> plot(lambda,fftdata_oneside);
```

The little dot (`.`) at the end of `delta_x.` insures that the division to create the vector `lambda` from the vector `n` happens element-wise.

An alternative and more transparent example is the following (given your data file has an EVEN number of elements):

```
>> data = rawdata(AA:BB);
>> data = data - mean(data);
>> N = length(data);
>> spec = abs(fft(data));
```

Now we remove (or just leave out) the zero frequency at index 1, and 'negative' frequencies (N/2+2:N)...

```
>> spec = spec(2:N/2+1);
>> spec = spec / max(spec);
```

And, since we've removed zero frequency, the lambda index starts at 1.

```
>> n = 1:N/2;
>> lambda = N*delta_x./n;
>> plot(lambda, spec);
```