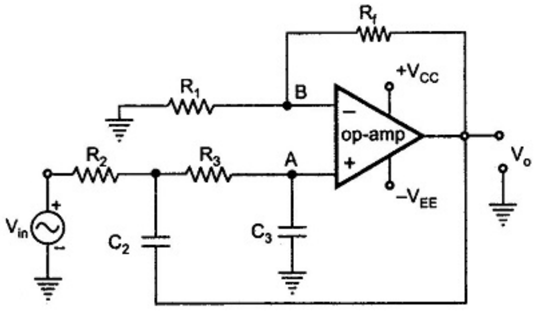
Interferometry Session 6

Wednesday, November 2, 2022 11:58 AM

Welcome to Martin's Lab Book. The work in the following section was completed on: **Friday, January 27th, 2023** **9am – 12pm** in a synchronous manner becoming of a lab workbook.
Aims
• Start FFT'ing all of the data.

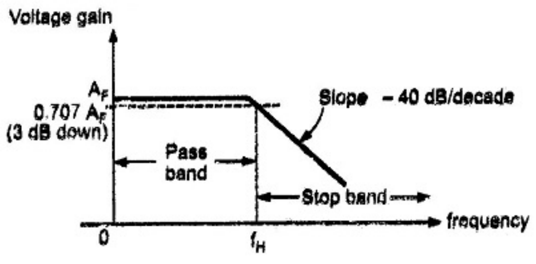
Preamble: Butterworth Filters



https://www.eeeguide.com/wp-content/uploads/2016/09/Second-Order-Low-Pass-Butterworth-Filter-001.jpg

$$\frac{V_o\,(s)}{V_{in}\,(s)} = \frac{A}{s^2 + 2\,\xi\,\omega_n\,s + \omega_n^2}$$

where A = overall gain, ξ = damping of system, ω_n = natural frequency of oscillations.
This results in the frequency response, as follows:



A second order Butterworth filter is a type of low-pass filter that has a smooth frequency response and a maximally flat magnitude response in the passband. The cut-off frequency of 1Hz means that the filter allows frequencies lower than 1Hz to pass through, while attenuating frequencies higher than 1Hz.

In Python, a second order Butterworth filter can be implemented using the **scipy.signal** library, specifically the **butter** function. The function takes in parameters such as the filter order, the cut-off frequency, and the desired filter type (low-pass in this case), and returns the filter coefficients in the form of numerator and denominator polynomials. These coefficients can then be used in conjunction with the **lfilter** function to filter a signal.

A Butterworth filter is implemented using the following given code:

```
filter_order = 2
freq = 1 #cutoff frequency
sampling = 50 # sampling frequency
sos = signal.butter(filter_order, freq, 'hp', fs=sampling, output='sos')
filtered = signal.sosfilt(sos, y1)
y1 = filtered
filtered = signal.sosfilt(sos, y2)
y2 = filtered
```

Preamble: Micro-step Conversions

Task 10
This is our measurements for Finito 3.
This yields **96.4pm per micro-step**.

Starts at –6,018,362	12.82mm
Ends at 70,674,203	5.42mm

This yields **37.8pm per micro-step**.

Starts at 70,674,203	12.82mm
Ends at –55,758,255	17.60mm

Together, if we average this data, this yields **59.97pm per micro-step**.

Task 11
This is our measurements for Finito

Starts at –55,758,255	17.60mm
Ends at [unknown]	[unknown]

In all, this

Task_13_1_ML
This was our first run with the notch and bandpass filters, but the Hg lamp further away, as well as with less shielding provided by the beam stoppers placed around.

Starts at	8.33mm
Ends at	> END

Task_13_2_ML
This was a second run with the filters, however the Hg lamp was in closer proximity.

Starts at	5.66mm
Ends at	20.37mm

Task 12

Task 12a - Using the Hg green line to correct the yellow doublet
You should follow the program analysis.py for the steps in this analysis. This program should not be used as a black box but more as a template or starting point. There are many ways that you could make these corrections what we suggest here is only one possible way.

- Reconfigure your apparatus using the second detector and another beam splitter so that the interferogram of the green line falls on one detector and the yellow doublet on the other. It may be easier to align this with the green laser and then replace it with the Hg lamp as the Hg lamp with filter is so weak.
- Take a long interferograms.
- Using the green line (wavelength 546.0nm) correct for the positions where the data were taken. The easiest way to do this is to use the crossing points as we did in callibrate.py to set the absolute distance scale for each half wavelength. Then correct the position of the points in that half wavelength then move on to the next half wavelength building on the already corrected point. So the position of each point now is just:

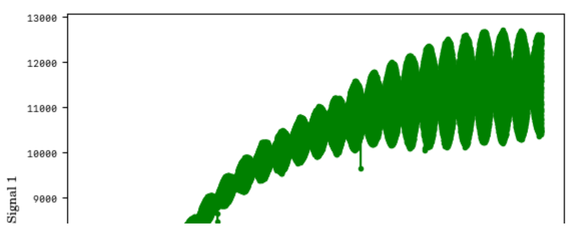
$$x_{\text{corr}} = x_{\text{uncore}}^{\text{init}} + \frac{\lambda_{\text{fit}}}{\lambda_{\text{true}}}\left(x_{\text{uncorr}} - x_{\text{uncorr}}^{\text{init}}\right)$$

  where x_{corr} is the new corrected position, x^{init}_{core} is the initial uncorrected *x* position, λ_{fit} and λ_{true} are the true wavelengths from the number of µsteps the true wavelength of the green light. By doing this you are essentially stretching or compressing the chunk of data that you have fitted to correct for irregularities in the stage's movement.
- Now that you have a corrected set of positions for where the data were actually taken. You now need to produce a "re-sampled" data set for the yellow doublet where the spacing between points is regular. Probably, the simplest way of doing this is to fit a cubic spline to your corrected but now unevenly spaced data and then call the cubic spline function with the regular space points that where you now want your regular points.
- Take the Fourier transform of this interferogram. Does what you see make sense?

Green and yellow filters
Task_12_yellow_doublet(_1).txt

Starts at [variable]	25.30mm
Ends at [variable]	5.46mm

Task_12_yellow_doublet.txt



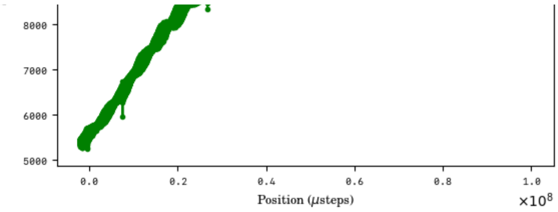


Figure 1: Pre

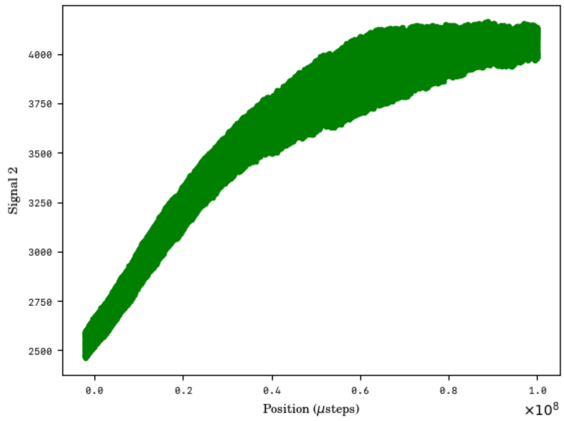


Figure 1: Pre

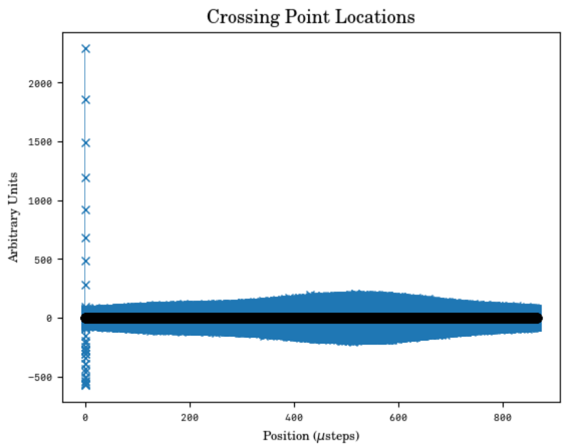


Figure 1: Pre

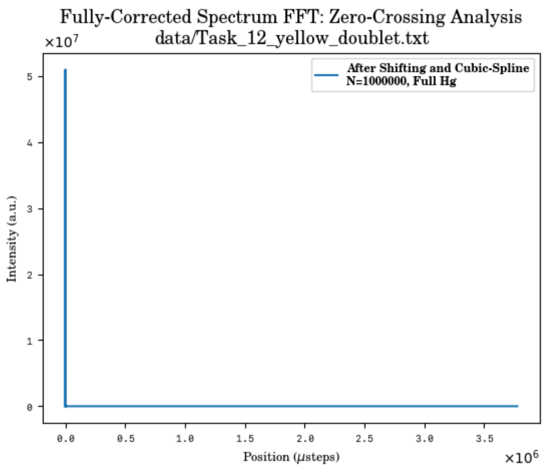
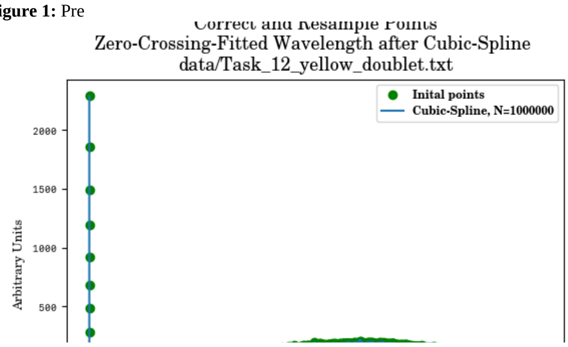


Figure 1: Pre



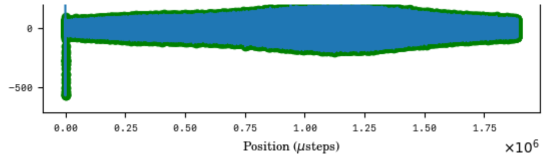


Figure 1: Pre

Task_12_yellow_doublet_1.txt

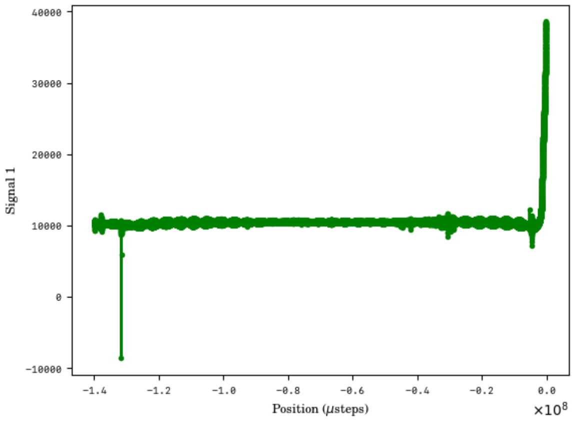


Figure 1: Pre

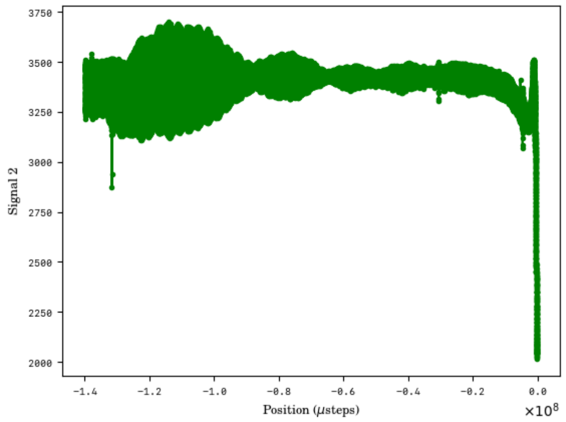


Figure 1: Pre

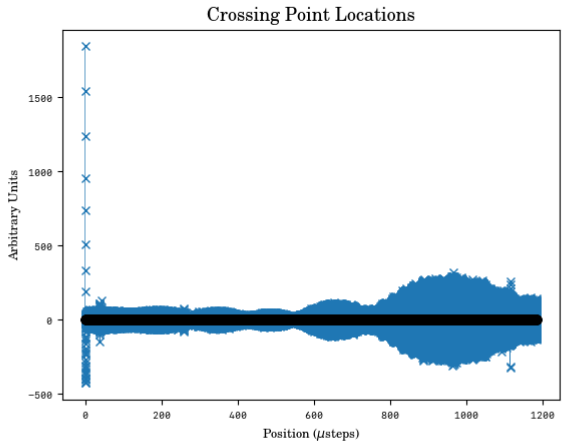
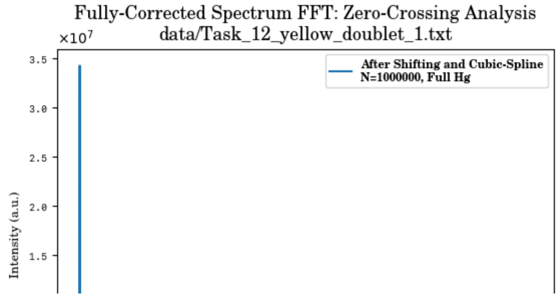


Figure 1: Pre



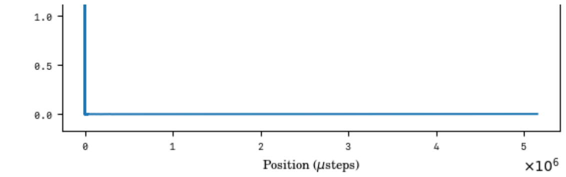


Figure 1: Pre

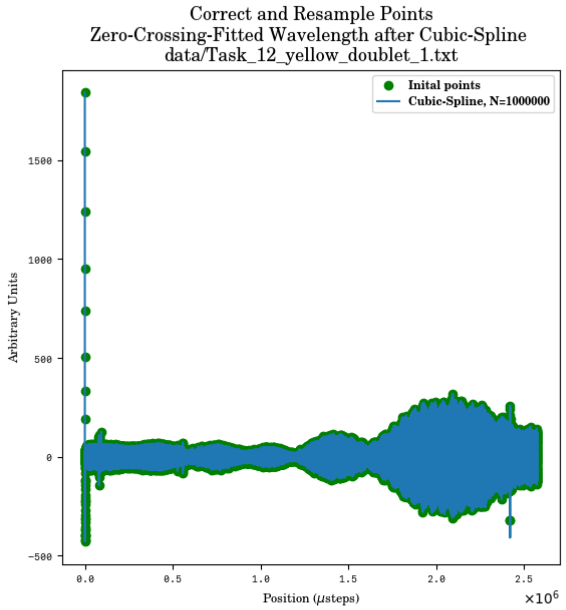


Figure 1: Pre

Task 12b Using the Hg green line to correct the entire Hg spectrum Repeat what you did for Task 12a but with the yellow filter removed. This should allow you to measure the entire Hg spectrum (or at least that part of it that is within the sensitivity range of the photodiode).

Task_12_green_singlet.txt

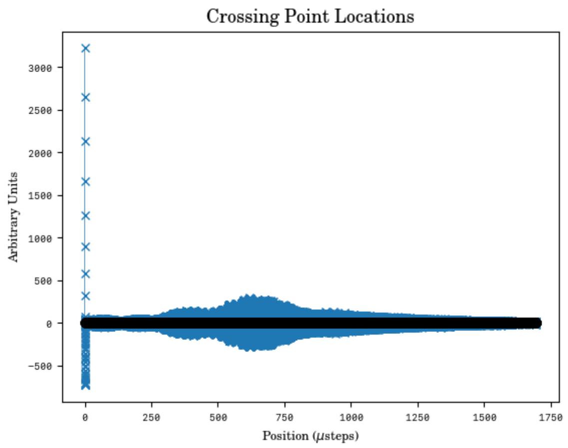


Figure 1: Pre

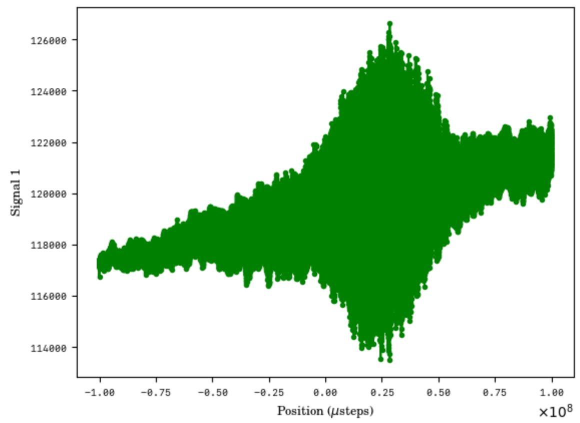


Figure 1: Pre

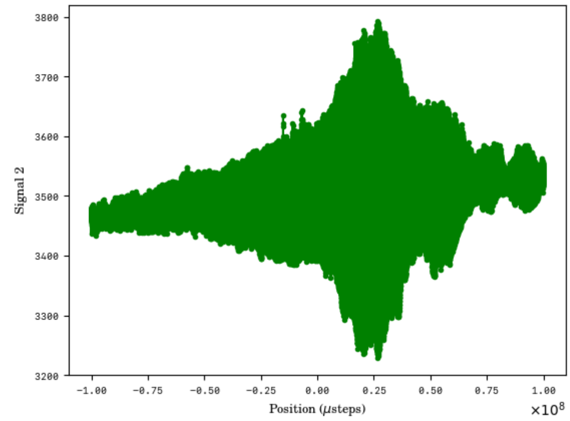


Figure 1: Pre

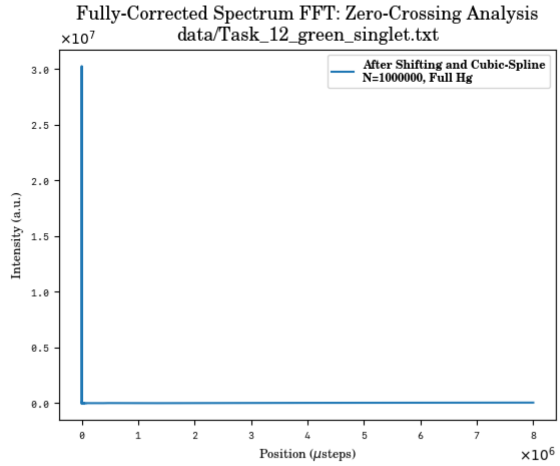


Figure 1: Pre

Task_12_green_singlet_1.txt

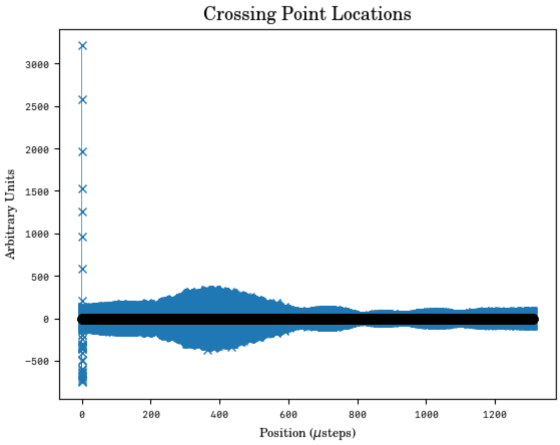


Figure 1: Pre

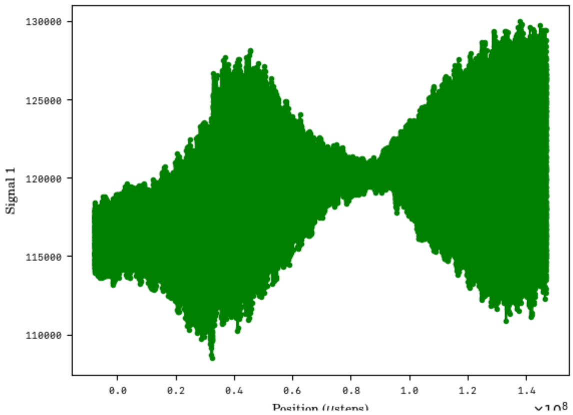


Figure 1: Pre

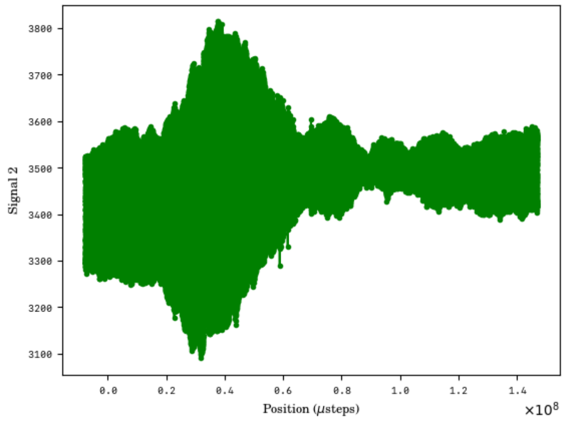


Figure 1: Pre

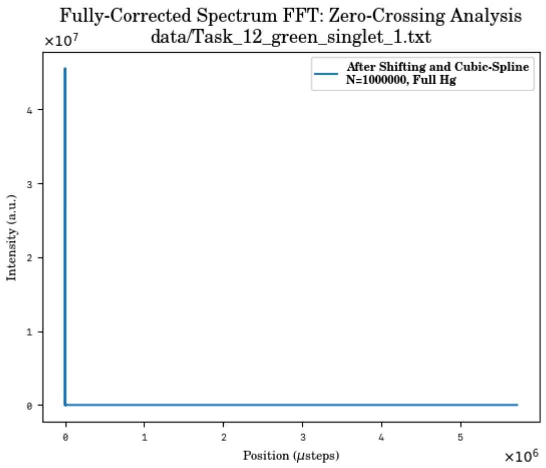


Figure 1: Pre

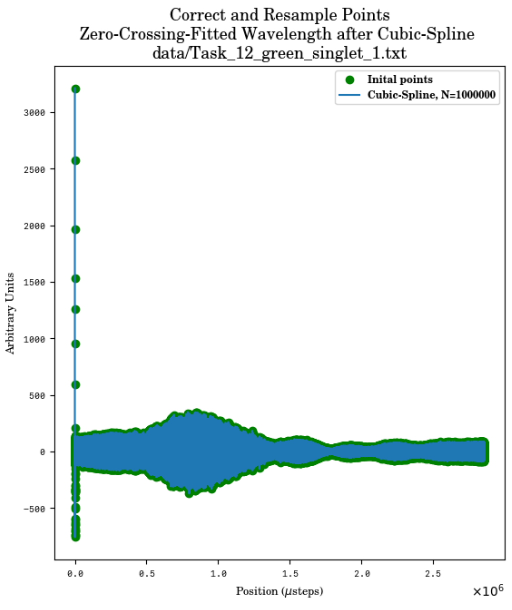
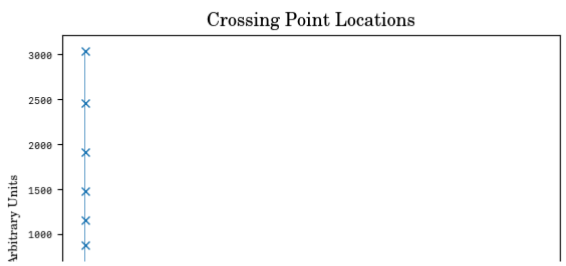


Figure 1: Pre

Task_12_green_singlet_2.txt



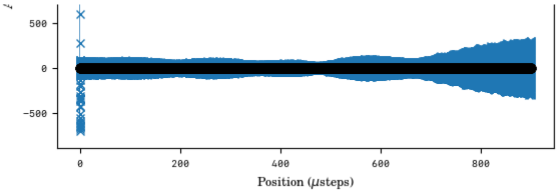


Figure 1: Pre

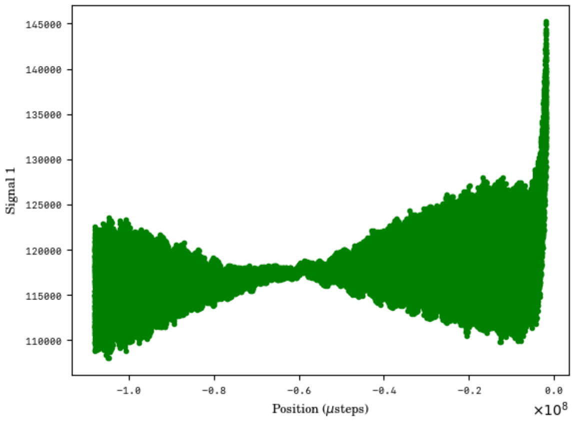


Figure 1: Pre

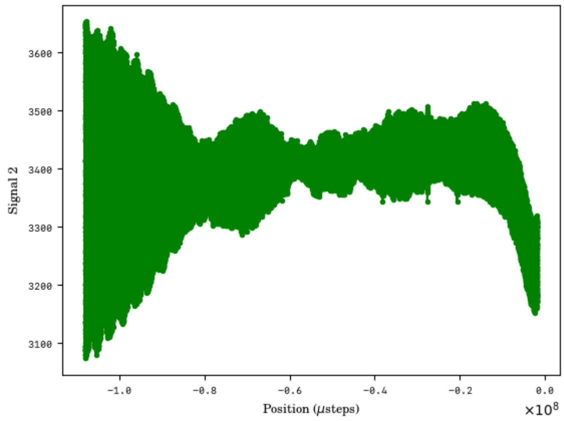


Figure 1: Pre

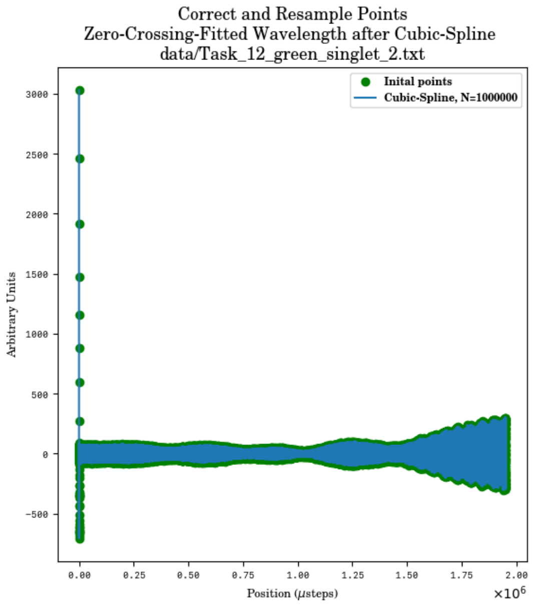
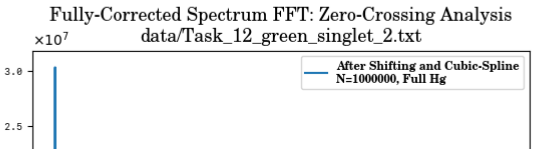


Figure 1: Pre



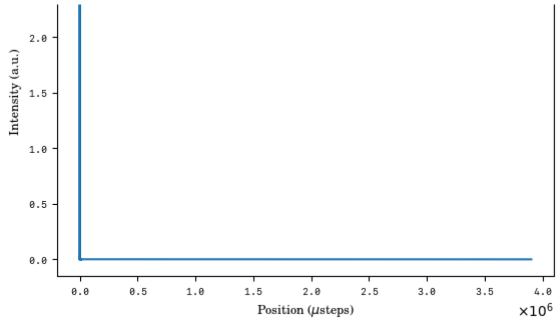


Figure 1: Pre

Aside: Double Gaussian Curve Fitting
For the purpose of performing the pioneering curve fitting of two superimposed Gaussian curves, the data from *Task_12_green_singlet_2.txt* was chosen for this purpose. The relative *overkill* of this plot was in part motivated by the need to determine above how many standard deviations (σ) away we need to discard the data for before deciding whether 2σ or 3σ is used in order to ascertain a suitable mean value for the *"background noise"* that exists.

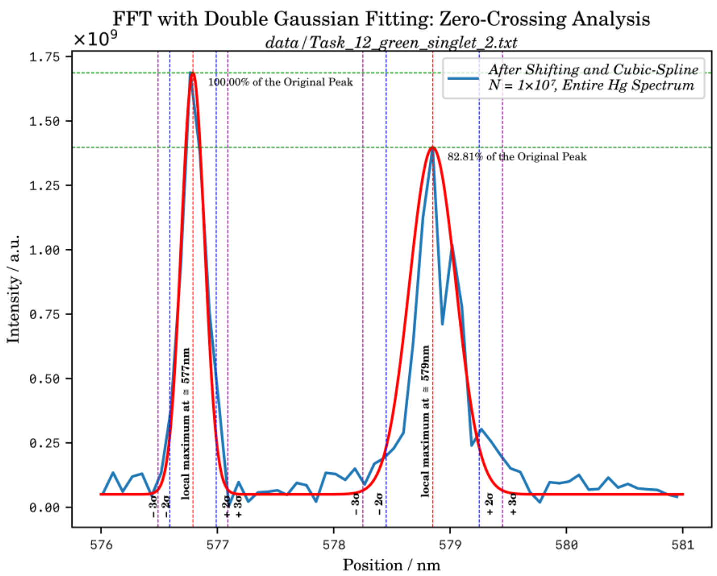


Figure 37: The expanded *double Gaussian* plot for the FFT'd data. As a result of this graph, we can clearly see that 3σ is the suitable value if we wish to eliminate any systematic errors in the data, as opposed to 2σ. This documents the positions of the yellow doublet spectral lines (at approximately 577nm and 579nm). This results in a near-perfect match, as cross-referenced with the NIST dataset. This is includes as an inset in Figure 38.

50	576.9598 nm	Hg I	BAL50	576.7671 nm	0.033% difference
60	579.0663 nm	Hg I	BAL50	579.0185 nm	0.008% difference

Table 19:

Next, we shall consider

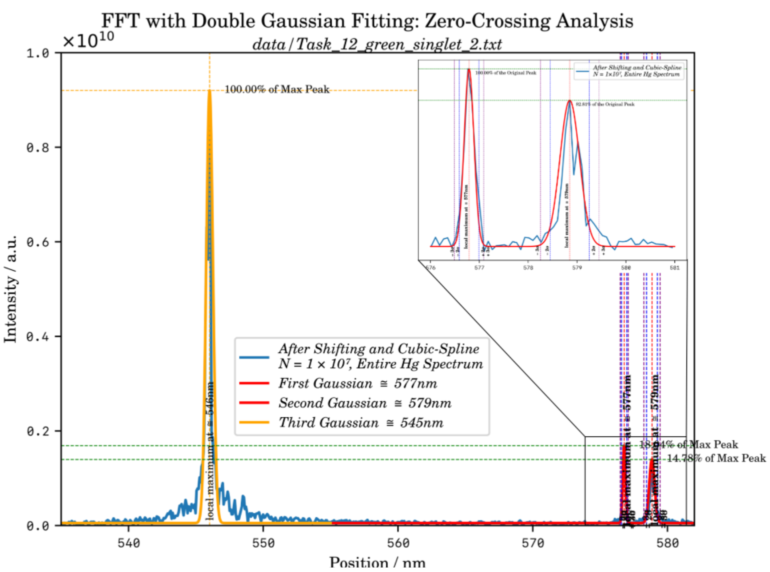


Figure 38: An overview of the yellow doublet, as compared with the largest peak which occurs at approximately 546nm, which is almost perfectly corroborated by the NIST database, as shown below.

500 P	546.0735 nm	Hg I	BAL50	546.0220 nm	0.009% difference

Finally, we expand to consider the entire range of the mercury spectrum, including a treatment of wavelengths over 1 micron. These are expanded across Figures **39** and **40**.

Task_12_green_singlet_3.txt

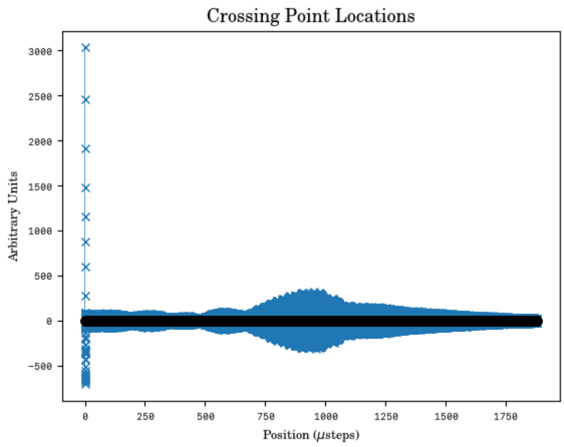


Figure 1: Pre

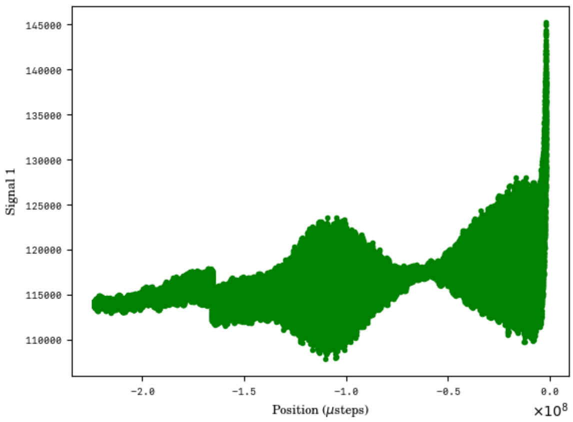


Figure 1: Pre

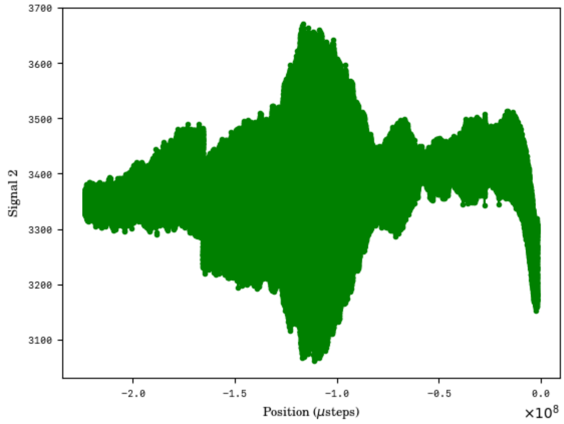
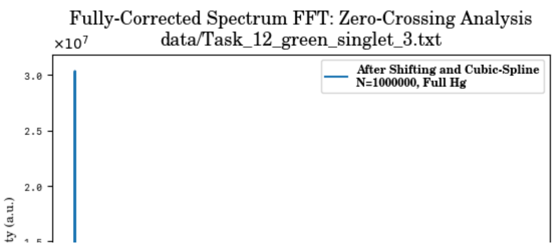


Figure 1: Pre



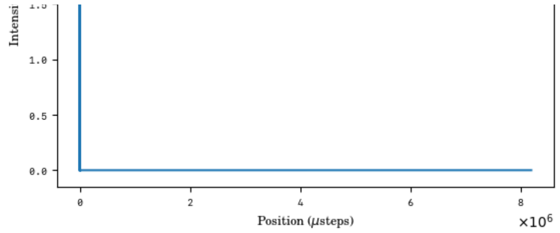


Figure 1: Pre

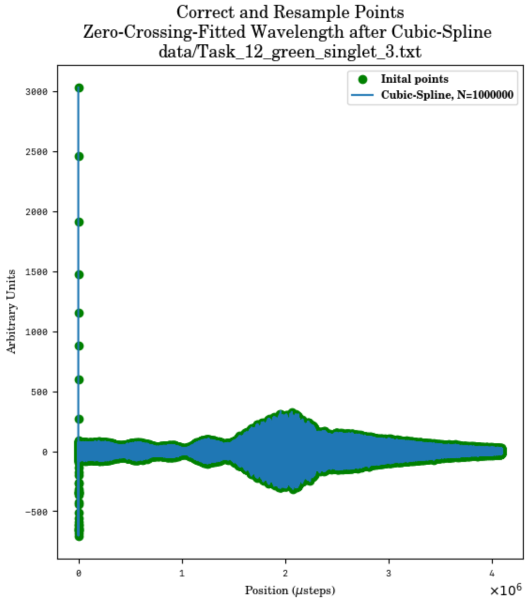


Figure 1: Pre

Green filter

Task_12_green_singlet(_1,2,3).txt

Starts at [variable]	[unknown]
Ends at [variable]	[unknown]

Additional Figures and Reference Section

Source:
https://pe2bz.philpem.me.uk/Lights/-%20Laser/Info-999-LaserCourse/C10-M04-MichelsonInterferometers/Module10-4.htm

Source:
https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwiaqM3F3PH8AhVRXcAKHTOLCvcQFnoECA0QAQ&url=https%3A%2F%2Fwww.ld-didactic.de%2Fdocuments%2Fen-US%2FEXP%2FPHO%2F4747112EN.pdf&usg=AOvVaw1bwzoBoQqfSb8DzLHSjLtu

Source:
https://www.repairfaq.org/sam/laserhen.htm

Source:
https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwi29d673fH8AhUWglwKHSewCo0QFnoECA0QAQ&url=https%3A%2F%2Fdigital.wpi.edu%2Fdownloads%2F6d5700218%3Flocale%3Den&usg=AOvVaw048Hb0stmDj6xWN2tLJOEe