Optical Pumping of 37 K Ground State for Population Distribution Modelling

by

Alex Tofini
University of Victoria

Term Report for TRIUMF co-op
Under the Supervision of John Behr, the Head of

TRINAT

Contents

Table of Contents **ii**

List of Figures **iv**

Acknowledgements **vi**

1 Introduction **1**

2 Camera Characterization **2**

2.1 Trigger Delay 2

2.2 MOT Image Analysis 4

2.3 Luminosity Tracker 5

2.4 Scalar & T7 Board 5

3 405 nm Laser **7**

3.1 Introduction 7

3.2 Alignment and Rough Calibration 7

3.3 Saturated Spectroscopy 8

3.4 Laser Power Adjustment 8

3.5 Hyperfine Structure 10

3.5.1 Zeeman Splitting 10

3.5.2 Spectra Fitting 10

4 Conclusions **14**

A Script Explanations **15**

A.1 Troubleshooting 15

A.2 Scripts 15

B Apparatus **21**

Bibliography **22**

List of Figures

Figure 2.1 The camera's delay with respect to a software trigger was quantified by means of forcefully varying the delay between LED fire and camera fire. The Peak corresponds to the maximum light captured and hence the amount of time in which the camera needs in advance prior to a signal arriving in order to capture it in its entirety 3

Figure 2.2 Camera delay for short exposure time shows similar behavior as long exposure time with the addition of a plateau. Additionally the delay time of the camera is reported to be approximately 55 μs 4

Figure 3.1 The linear relation between temperature and current is demonstrated here. The units are based off the fine adjustment potentiometer readings, while the coarse ranges were set to 3 for the temperature and 4 for current. This corresponds plotted temperature range of 19.3-20.1C and a plotted current range of 46-50 mA 9

Figure 3.2 Example of fitted peaks. The reported line width is based on a relative scale and not an absolute scale. X in this represents the separation distance between each peak since they were forced to be identical. The reported errors illogical since there was no weighting given to the fit. 12

Figure 3.3 The three data sets are fitted to Equation 3.4 with weight values governed by $1/\sigma^2$. The reported line width range can be used as an upper limit on our achieved laser resolution. 13

Figure A.1 Transition matrix with example corresponding to F=2 m_F=-2 to F$'$=1 m'_F=0 19

Figure B.1 Diagram explaining the implemented optics 21

ACKNOWLEDGEMENTS

I would like to thank:

John Behr for providing me with this unique opportunity and teaching me a vast amount of things that will be crucial in my future career path.

Chapter 1

Introduction

TRINAT is measuring spin polarized observables such as A$_\beta$. This is done by utilizing the following expression as well as a precise measurement of the nuclear polarization.

$$\frac{d^3\Gamma_{angular}}{dE_p d\omega_\beta} \propto 1 + b\frac{m_e}{E_p} + \mathbf{P}_{nucl} \cdot (A_\beta \frac{\mathbf{p}_\beta}{E_\beta}) \tag{1.1}$$

In the past TRINAT has maintained nuclear polarization of 99.13 ±0.09% by means of optically pumping ultra-cold 37 K atoms and spin polarizing them. They then quantified the polarization by means of monitoring the fluorescence of the atoms as they are being pumped. A large restricting factor on the precision of this experiment was the fact that there were no direct measurements of the initial population levels of the atoms. Instead they were modelled and the standard optical Bloch equations were implemented. In order to improve these results we have set out to create a way to probe the initial population levels, thus allowing for more accurate measurements of the final population level and hence the polarization. This is done using a 405 nm external cavity diode laser in order to photo-ionize a specific ground state level and monitoring the resulting fluorescence, as well as the resulting ions which are transported to a micro-channel plate via an electric field.

Chapter 2

Camera Characterization

2.1 Trigger Delay

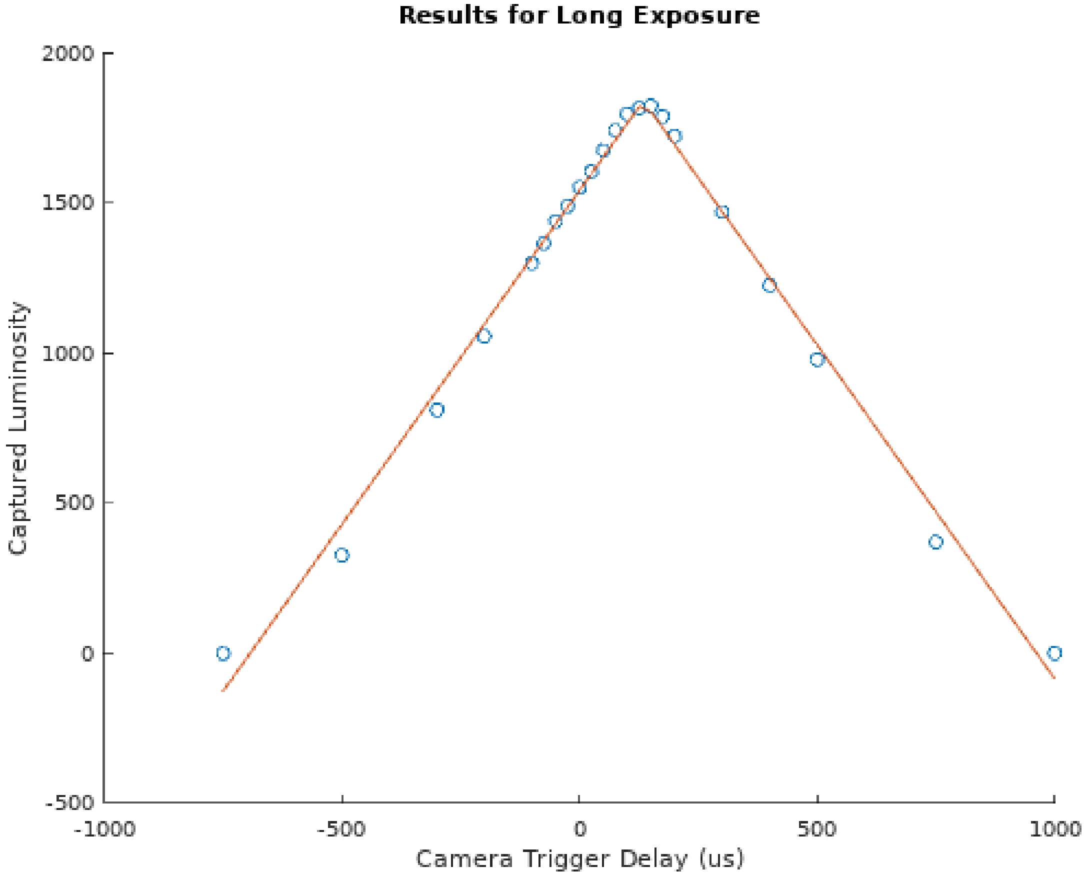
In order to quantify the trigger delay for the Firefly MV FMVU-03MTM camera, the supplier provided code known as AsyncTrigger.cpp was modified. The changes done to the default code were as follows: Hardware trigger enabled instead of software trigger, exposure time readout was implemented, as well as custom image names for each capture. The new code was saved as AsyncTriggerAT.cpp which can be compiled using the associated MakeFile.

It was noted that the camera's registry will save information even after being powered down. This results in unexpected camera behavior if registries are not reset to default after manipulating. Therefore it is good practice to load the current state of the registries upon powering up the camera, and forcing the registries back into their original states at the end of any code or task. If it is unknown which registry is not set to default, it is possible to run the supplier provider GUI known as Flycap. By doing this, all registries will be set back to a functioning default state.

In order to determine the camera's response delay with respect to the trigger signal, an LED was used to simulate the atom cloud. The LED and camera trigger were controlled via a function generator that was set to manually triggered pulse mode. The LED required a pulse of amplitude 2 V, while the camera trigger required a pulse of 3.5 V. The relative delay between the LED pulse and the trigger pulse was used to quantify the camera's delay.

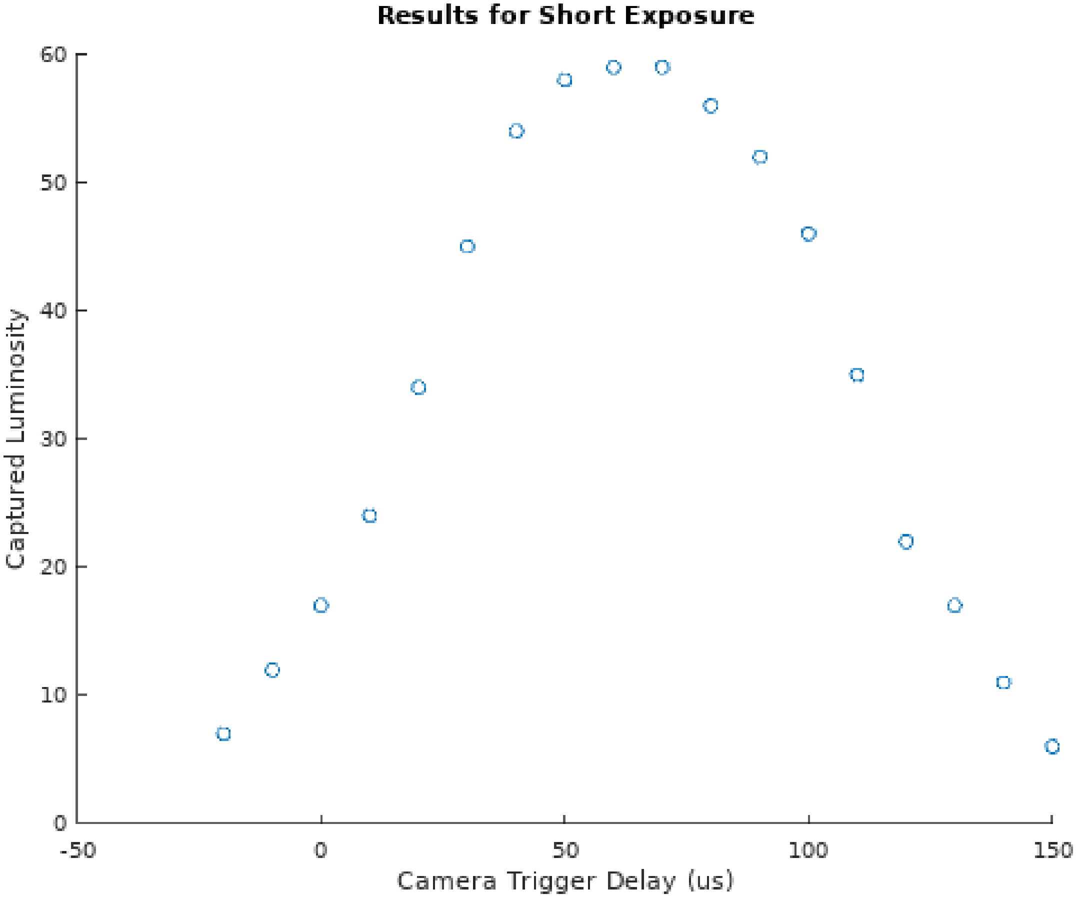
The first test run consisted of setting the camera's exposure time to 1 ms and the LED pulse width to 1 ms. The pulse width of the camera trigger is irrelevant due to the fact that the camera's software simply looks for a rising edge in voltage and not a falling edge. By varying the relative delays, a plot of captured luminosity versus trigger delay was created, Fig 2.1. The response time of the LED was also checked by analyzing its rise time. It was found that the LED had a 2 μs rise time which was then accounted for when computing the camera's delay time.

Figure 2.1: The camera's delay with respect to a software trigger was quantified by means of forcefully varying the delay between LED fire and camera fire. The Peak corresponds to the maximum light captured and hence the amount of time in which the camera needs in advance prior to a signal arriving in order to capture it in its entirety



This plot clearly demonstrates that the camera has a distinct delay time of 134 μs, but when compared to previous co-op reports this value was larger than expected and the curve did not demonstrate the plateau as previously observed. This was further investigated by repeating the test with a short exposure time (cant remember what it was.... figure this out). The results of this test run demonstrated the plateau as previously expected and reported a different delay time which implies that the camera delay is dependent on the shutter time, Fig 2.2

Figure 2.2: Camera delay for short exposure time shows similar behavior as long exposure time with the addition of a plateau. Additionally the delay time of the camera is reported to be approximately 55 μs



2.2 MOT Image Analysis

This analysis algorithm is a function named AnalyzeROI.m which can be called easily by passing it the desired MOT image to be analyzed as well as the location of the top left pixel. This simply means in a non-cropped image from the camera it would be (1,1), but if the image has been cropped it needs the new location in order to maintain awareness of the cloud location. The notation is (y location, x location) or (row,col). It is advised to subtract a background from the desired image prior to sending it to the analysis algorithm. This is done in the script SinglePictureAnalysis.m as an example.

The algorithm was built off the frame work of an old code called sidwp.m, but now shares very little similarities. For a more detailed explanation of how the algorithm works see Appendix A. It's important to note that this algorithm has had minimal implementation with real data and instead used simulated data from an LED. Therefore when actual data is being collected it may be necessary to make changes to the algorithm.

2.3 Luminosity Tracker

Since the fluorescence of the atoms within the MOT is needed to quantify population levels, it is vital to be able to track how the cavities luminosity is changing over time. This can be achieve by implementing the LightCurve.m script which calls a simpilied version of the main analysis script. This simplified script known as Luminosity.m simply fits a large spline fit to the data or Gaussian, and returns the area under the curve after subtracting the background. The calculated area is then interpreted as a luminosity, with no units associated with it yet therefore it is a relative scale. The design of the luminosity tracker is to continuously search a specified directory for a new file which follows a given file name format. This format has been coded into the firefly camera and can be seen by checking the code AsynchTriggerAT.cpp. Once the script locates a new file that it has not yet processed, it will extract the time stamp out of the name and the it will send the image for analysis. The returned results and the time stamp are then plotted on a rolling plot that has a fixed x axis size. The scrolling plot is achieved by calling a custom made function called AddData.m, see Appendix A for more details.

2.4 Scalar & T7 Board

The AsynchTriggerAT.cpp code was improved for the trinatblack computer and now has the name AsynchTriggerATCounter.cpp. This script can now change a DAC voltage from 5-7 V which will sweep the AOM frequency from 260-280 MHz with some extra range as a cushion. The conversion between DAC voltage and resulting frequency was determined to be approximately 17 MHz/V. At each incremental change to the voltage, a scalar is also read out which will allow for the atom cloud's fluorescence to be recorded at different laser frequencies. Once fully implemented this will allow for the MOT to be swept over resonance.

Chapter 3

405 nm Laser

3.1 Introduction

A 405 nm diode laser was implemented in order to probe the excitation levels of the potassium atoms trapped in the MOT. This will be done by photo-ionizing a specific ground-state level and observing the resulting fluorescence to quantify population levels. At any point during the following explanation, refer to Appendix B for an apparatus diagram explaining the optics used in this project.

3.2 Alignment and Rough Calibration

The first step in setting up the 405 nm laser consisted of adjusting the grating and collimation lens until lasing was achieved. This was done by following the DL 100 user manual which has step by step instructions on how to recreate this. One can also find additional information in the electronic log book pertaining to how this process went. Once initial lasing was achieved, the lasing threshold was recorded and the calibration steps were repeated in hopes of lowering the lasing threshold. This was repeated until we were within a couple mA of the reported minimum lasing threshold. At this point it is necessary to determine the frequency of the laser or its energy level in cm^{-1}. This is done by coupling the laser through a fiber optic cable and sending it to be analyzed by a wavemeter junior. Upon doing this the initial laser frequency was drastically off from the desired wavenumber of 24701.382 cm^{-1}. The next step consists of altering the coarse adjustment screw of the grating and the current supplied to the diode while monitoring the resulting change on the wavemeter junior. This was done until the laser was off by 1-2 cm^{-1}, which then requires fine adjustment via a voltage sent to the piezo. At this point the wavemeter junior does not have high enough resolution, therefore to properly quantify the fine adjustments it is necessary to use a new more precise tool known as saturated spectroscopy.

3.3 Saturated Spectroscopy

In saturated spectroscopy or ”sat spec” a Potassium vapour cell is used to tune the laser to the desired wavelength corresponding to the desired atomic excitation level. This is done by passing the beam through a vapour cell and adjusting the laser frequency via the piezo's voltage and the diode current until fluorescence is observed. Once this was achieved, the optics layout was modified to allow for a probe and pump beam to counter-propagate within the cell. This allows for both beam to *talk* to the same atoms simultaneously. The result of this is three distinct peaks in the absorption curve, two reductions in absorption and one increase in absorption. The increase in absorption is a result of a specific group of atoms that are travelling perpendicular to the beam paths, i.e they are part of the zero-velocity group with respect to the beam path. These atoms experience no Doppler shift along the beam path thus allowing for both beams to excite the same group of atoms. This phenomena is only present in the atomic vapour cell and not the MOT because the atoms are trapped and not allowed to move freely, hence there are no velocity groups.

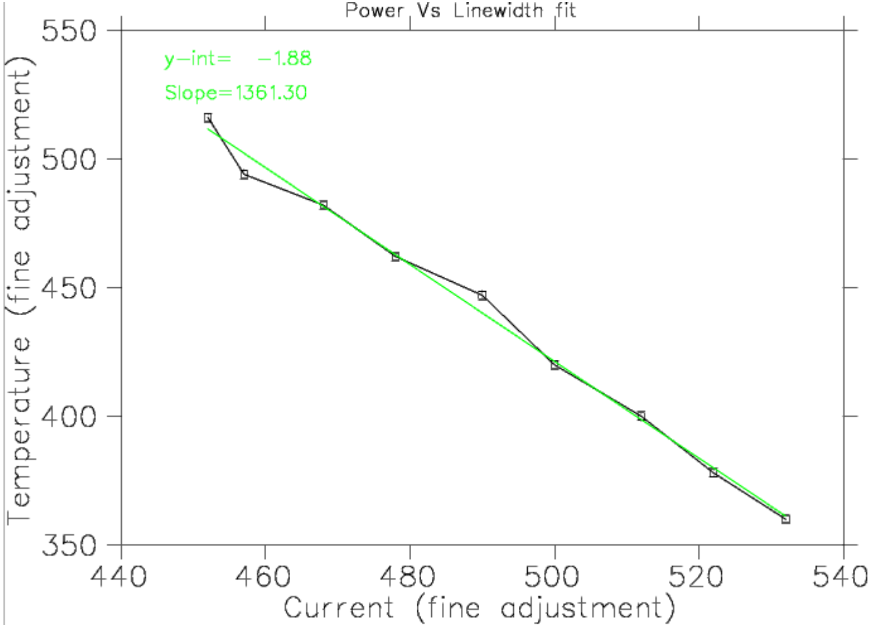
3.4 Laser Power Adjustment

Once the laser was fine tuned to the desired frequency, it became necessary to attempt to increase the laser power. The first thing considered was the spacial mode of the laser since it directly effects how well the light can be coupled into the fiber. It was concluded that the light was clipping at the optical isolater therefore the beam needed to be reduced in size prior to entering it. This was achieved by inverting and rotating the anamorphic prism by 90 degrees such that the output port was used as the input port and the prism was on its side. This results in a horizontal *crunch* of the beam as opposed to a vertical stretch which was the original setup.

This is done by changing the temperature of the laser and compensating with a current change in order to reestablish the absorption. It is also necessary to change the piezo voltage to relocate the absorption since the required voltages will drift. Typically one can find two to three different voltage ranges that will product absorption. Unfortunately attempting to reestablish the absorption after any significant change in the temperature proved too difficult.

Instead in order to predict the required current and temperature settings, a set of data consisting of small temperature changes and their respective current required for absorption were recorded. This can then be used to extrapolate the linear trend to predict where absorption will occur at lower/higher temperatures, Figure 3.1

Figure 3.1: The linear relation between temperature and current is demonstrated here. The units are based off the fine adjustment potentiometer readings, while the coarse ranges were set to 3 for the temperature and 4 for current. This corresponds plotted temperature range of 19.3-20.1C and a plotted current range of 46-50 mA



While constructing this curve, it was not possible to obtained absorption at a temperature lower than 19.3 C. This could possible be explained by the fact that the grating profile needs to be adjusted in order to reestablish absorption at this new current range but this was not tested due to the amount of time required. Another theory was that we were pushing the laser into a sort of forbidden region that we could in theory *jump* over by doing a large change in temperature following the extrapolated curves results. Unfortunately even after doing this it was not possible to find the absorption.

3.5 Hyperfine Structure

3.5.1 Zeeman Splitting

When atoms are in the presence of a static magnetic field they experience an effect known as Zeeman splitting. This refers to the splitting of the spectral lines into several previously degenerate states. These states are differentiated by the quantum number m_F, where m_F can take the values F, F-1, ..., F+1,F. The relative splitting that each m_F level experiences is governed by the expression

$$\Delta E_F = g_F \mu_B B m_F \tag{3.1}$$

Where μ_B is the magnetic permeability of free space, B is the magnetic field experienced by the atom and the coupling constant g_F is calculated with the following expression.

$$g_F = g_J \frac{(F(F+1) + J(J+1) - I(I+1))}{2F(F+1)} \tag{3.2}$$

and lastly the coupling constant g_J is calculated with

$$g_J = \frac{3}{2} + \frac{(S(S+1) - L(L+1))}{2J(J+1)} \tag{3.3}$$

In order to simulate the behavior of 39K and 41K, a simulation called ZeemanSim.m was created. This program floats the magnetic field value as well as the line width of the peaks and allows the user to control them via two slider bars. Additionally one can control which peak to observe with the use of a set of radio buttons. As of now the code is based off the assumption that the shape of the peaks takes on the form of a Lorentzian, but has the capability of being switched to a Gaussian by slight alteration of the code.

3.5.2 Spectra Fitting

Upon analysis of the Zeeman Simulation, it was concluded that it was not necessary to simulated all the Zeeman splitting transitions since the magnetic field inside our cell is only in the range of 1-3 Gauss. Where the field drops by a factor of 2 from the center of the cell to the face of the cell. Additionally since our laser line width is fixed to be greater than 1 MHz by the natural line width, we will not be able to differentiate the different Zeeman levels in our spectra. Taking this into consideration, three peaks corresponding to the merging of these Zeeman levels were used to create a function to fit our data to. Originally it was attempted to simulate a lock in signal by differentiating the theoretical peaks with respect to frequency, but this was dropped due to the fact that the experimental lock in signal executes the derivative with non infinitesimal steps in frequency which results in an averaging error. Instead it was concluded that it would be simpler to fit the saturated spectroscopy peaks themselves. It's important to note that this was always considered the easier method computationally but unfortunately there were hardware limitations that prevented sufficient resolution of the peaks for a significant amount of time. Once it was possible to acquire a high enough resolution trace of the peaks, the left peak corresponding to F=2 to F'=2 and F=2 to F'=1 and their respected crossover were maximized by fine tuning the alignment. These peaks were then captured at three different pump beam intensity levels. Due to the difficulty to exactly quantify the pump beam's intensity, resulting from multiple passes inside the cell causing absorption at each interface, a relative intensity scale was used. This was done by capturing the spectrum when it was at its maximum that we could achieve then it was reduced by a factor of two and then another factor of two after that by rotating a $\lambda/2$ thus changing the reflection vs transmission in the PBS. At each intensity level the peaks were captured and fitted, Figure 3.2, in order to determine the line width at that given pump beam intensity.

In order to convert between the relative frequency scale and the absolute frequency scale, the magnetic field inside the cell was measured and implemented in the Zeeman simulation code to determine the Zeeman splitting distance. Then the ratio Γ'/X' acquired from each fitted trace is compared to Γ/X where X is from the Zeeman simulation in order to get Γ our desired result.

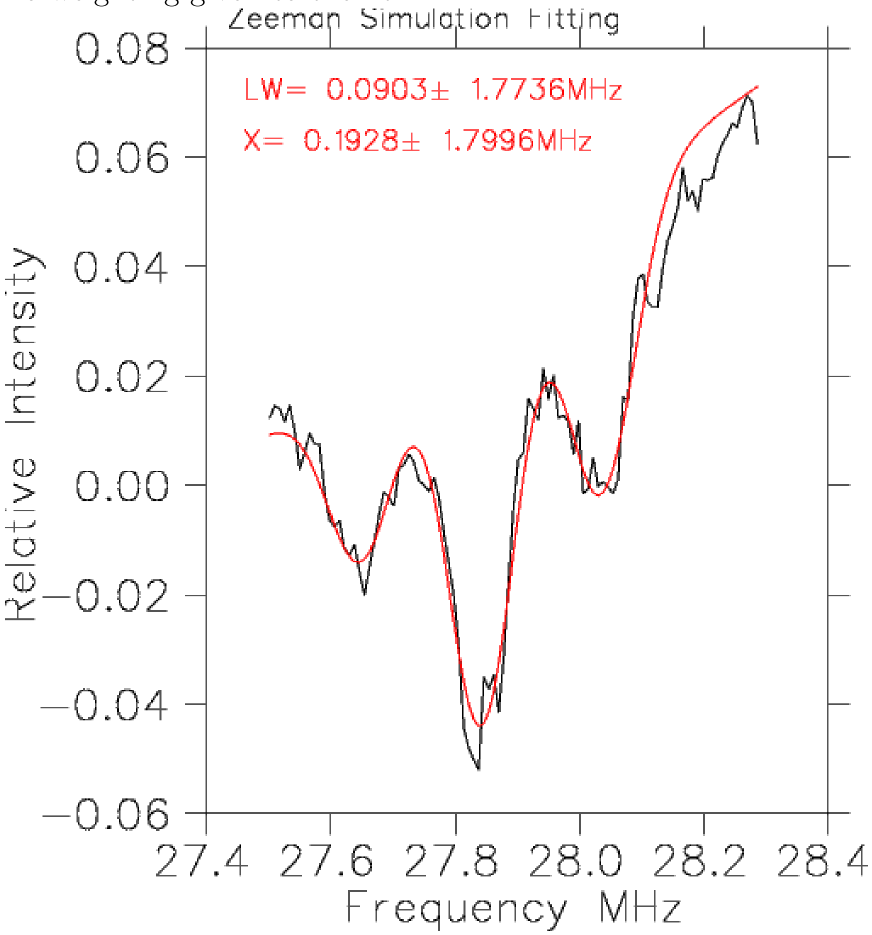
The errors were estimated by taking three different data captures at all three intensity level and taking the average of the fitted line width. Then the standard deviation was used as an estimate of the errors. This estimate is most likely an overestimate but there is another source of error that was not accounted for and that is the accuracy in which the relative signal strength was halved.

The resulting mean value and standard deviation of the three sample sets were then plotted and fitted, Figure 3.3, to the following function:

$$\Gamma' = \Gamma\sqrt{1 + AS} \tag{3.4}$$

where A is a floated scalar values used to account for the relative scale used for the saturation intensity, Γ is the natural line width of the laser and Γ' is the broadened line width.

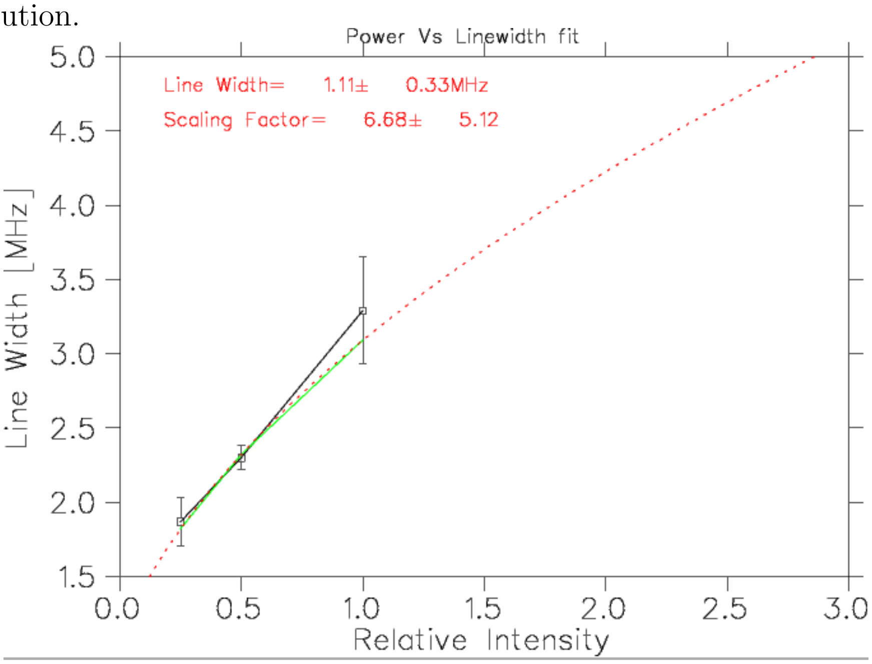
Figure 3.2: Example of fitted peaks. The reported line width is based on a relative scale and not an absolute scale. X in this represents the separation distance between each peak since they were forced to be identical. The reported errors illogical since there was no weighting given to the fit.



Since we know that our 405 nm laser has a natural line width of approximately 1 MHz, the reported result of 1.11 ± 0.33 MHz can be used as an upper bound but since a value lower than the natural line width does not make sense it can not be used as a lower bound. The scaling factor directly corresponds to how many times saturation intensity the apparatus was currently operating at. In order to determine if this result is accurate, the theoretical saturation intensity was calculated using

$$I_s = \frac{1}{1.23}\frac{\kappa\gamma_0^2}{4\pi^2 d^2} \tag{3.5}$$

Figure 3.3: The three data sets are fitted to Equation 3.4 with weight values governed by $1/\sigma^2$. The reported line width range can be used as an upper limit on our achieved laser resolution.



where κ is the saturation parameter which corresponds to the ration between rate of excitation and rate of emission which was set to be equal to 1 for our case, γ_0 is the homogeneous width of the transition and d is the electric dipole transition strength. Using the known values for the $4s_{1/2} - 4p_{1/2}$ transition, and using values of d acquired from David McKay [1] which are acquired based off the values for A$_{ki}$, a ratio of saturation intensity between $4s_{1/2} - 4p_{1/2}$ and $4s_{1/2} - 45_{1/2}$ was done. His calculations were quickly verified by confirming the ratio of the dipole elements were as he reported. This was done using the following expression:

$$A_{JJ'} = \frac{\omega_0^3}{3\pi\hbar c^3} \frac{|\langle nJ||er|n'J'\rangle|^2}{2J+1} \tag{3.6}$$

This suggested that the saturation intensity needed was on the order of 6 times larger than the D1 transition of $4s_{1/2} - 4p_{1/2}$. Since the pump beam is roughly 2 mW with a size of 2 mm^2, or equivalently about 50 mW/cm^2. The reported scaling factor of 6.68 is of the correct order of magnitude.

Chapter 4

Conclusions

The 405 nm laser designed for optically pumping the ground state energy levels of the $4s_{1/2}$ to $5p_{1/2}$ energy levels of 39K and 41K was successfully locked to the enhanced cross-over lines of the saturated spectroscopy spectrum. The stability of the locking scheme was tested by monitoring the 2f signal over a large period of time. As long as this signal remains constants, ignoring minor noise, it corresponds to a locked laser. This signal was monitored for for over 1200 seconds and showed no sign of destabilizing. This showed that the lock is able to compensate for thermal drifts with minor corrections to the current supplied to the diode. Unfortunately the lock is extremely sensitive to acoustic noise and will instantly destabilize if any moderate noise is made in its vicinity. This implies that the PI gain needs to be further optimized. Once this is accomplished the lock should be robust enough to be utilize in future experiments. As for the MOT image analysis scripts, they have not been fully tested on real data and will require further adjustment once the MOT is back in operation.

Appendix A

Script Explanations

A.1 Troubleshooting

When first booting up octave it may be necessary to load some software packages in order to execute the following scripts. If a script requires a package it will notify the user in the command window as a error when executing the script. If one wants to be thorough and avoid this, type in these commands into the octave command window:
pkg load optim
pkg load io
pkg load statistics
pkg load splines
pkg load general

Additionally if one wants to see all the currently downloaded/loaded package, one can type in the command window "pkg list", where the package names that are loaded will have an asterisk next to their names.

A.2 Scripts

DataFileEditor.m:

This script has the option of generating a new .csv file in which image analysis results can be saved. After this, data can be acquired from a single picture and appended to the .csv file specified by the user. Once this is done they are prompted if they want to append more data or exit note: the error catching in this is poor so try not to mess up the responses.

CSVAppend.m (function):

This script can be called to append data to a .csv file. Input 1 acts as the desired .csv file and input 2 is the data to be appended. note: the data in which you append must in the form of a row vector with the same length as the header(generated by DataFileEditor), or else it will throw an error.

SinglePictureAnalysis.m: NEEDS TO BE MODIFIED WITH REAL DATA

This script is designed as a test environment and was used to build up the analysis process step by step. All it does is load an image, then call the AnalyzeROI.m script to get the desired information of the atom cloud. The main use of this script is to manual isolate a region of interest (ROI), and crop the orginal image. The cropped image is then sent to AnalyzeROI.m

ProcessImage.m (function): NEEDS TO BE MODIFIED WITH REAL DATA

This is a functional form of the SinglePictureAnalysis.m script in the sense that it does not run on its own and is required to be called by a main script. It has a single input which is the filename of the picture to be analyzed. It also returns a value called "results" which is a row vector containing all the desired information which is returned by AnalyzeROI.m

AnalyzeROI.m (function):

This is the main function that does all the image analysis, it was based off the old code called sidwp.m, but now contains very little similarities. This script takes in the desired ROI image as input 1 and the coordinates used to crop the original image as input 2. The returned results are a row vector with all the desired information. For a more detailed explanation, you can read the following paragraph:

The code starts off by doing a vertical and horizontal projection of the data and fitting a spline fit to each projection. The parameters of the spline fit were arbitrarily set to 11 cuts which corresponds to 10 parametric equations. This number could be changed if needed, but as of now it returns an accurate fit with only small computational time. The code has an optional thing of outputting the coefficients of the fit if desired but is currently commented out. Once the spline fits are done,the maximum value is found which corresponds to the pixel location of the centroid, it is also used as a reference to locate the nearest local minimum. For the vertical projection is was necessary to cut off some of the outer values due to high noise (an automated version of this will be needed at some point I believe). Once the local minimums are found, they are used to once again crop the data into a cleaner form with minimal noise. This clean data set is then used to do a Gaussian fit to get the FWHM, where the equation used accounts for background have a non constant offset. The location of the minimums are also used as a reference to create the baseline (drawn in green) used to calculate the area under the spline fit. This area corresponds to the luminosity of the atom cloud. Lastly the conversion between pixels and mm is done using the results from Anya Forestell's work and the default reference point (top left corner of image) is shifted to the center of the image, once again using her results. The final converted results are then put into a row vector and returned to the calling script or function.

LightCurve.m:

This is a testing environment for the real time display of atom cloud centroid location as well as luminosity. It utilizes a broken down version of AnalyzeROI.m which simply calculates the luminosity of the atom cloud, this function is called Luminosity.m. Once it has acquired new data it will call the AddData.m function in order to add it to the currently open figure in a scrolling fashion,

DelayCurve.m:

This simply plots the data collected on May 21/24 which dealt with the camera trigger delayed response time.

Luminosity.m:

Simplified version of AnalyzeROI.m which simply returns the area under the horizontal data curve.

AddData.m (function):

This function is used to add data to currently opened figures without erasing the old data.

HyperfineStuff.m:

This is a test environment which was used to calculate the hyperfine structure for 39k and 41k. It has a built in function that calculates the hyperfine splitting of an F level. (Was later improved into ZeemanSim.m which has all this plus extra)

ZeemanSim.m:

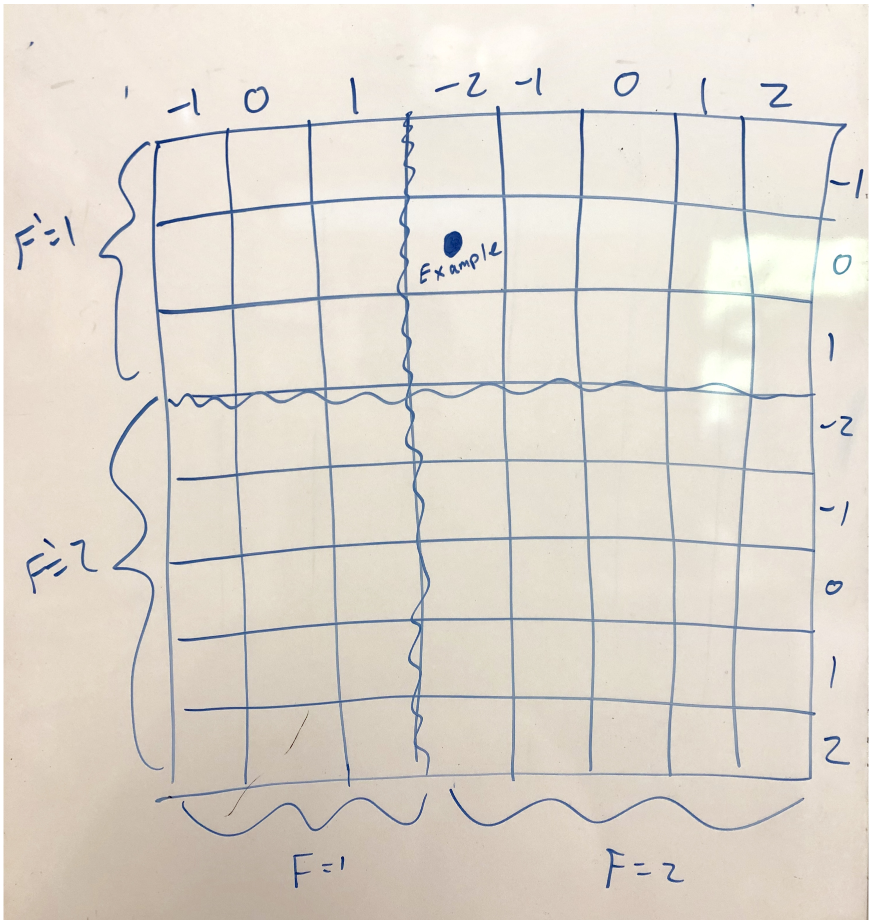
This script allows for an atoms Zeeman splitting dependence on magnetic field as line-width to be demonstrated via simulation. In order to change from 39K /41K one would need to change the initial parameters such as: Quantum Numbers,Hyperfine Coefficient A, Isotope shift. The fundamental basis of the code is the creation of a Lorentzian (or Gaussian which is commented out) to represent a single given transition. Then once each transition has an associated function to it, they are all summed up.

PopulationSim.m:

This code allows for a given atoms energy level population to be simulated depending on which state is being pumped via Pi Light only!. In order to accomplish this each ground state is given a population of 1 and then throughout the emissions and adsorptions, fractions of these states move around. In order to modify this one needs to consider all the possible transitions of an atomic state (see Metcalf), then it is necessary to modify the transition matrices (see footnote). In all honesty the code is pretty brute force and works well with this case but would need sizeable modification to the for loop indexing to account for a different state

Note: the transition matrices are simply a method I came up with of coding the allowed transitions with their relative strengths, see figure A.1 for example of the indexing. The resulting matrices are always symmetric by nature.

Figure A.1: Transition matrix with example corresponding to F=2 m_F=-2 to F$'$=1 m_F'=0



ImageSum.m:

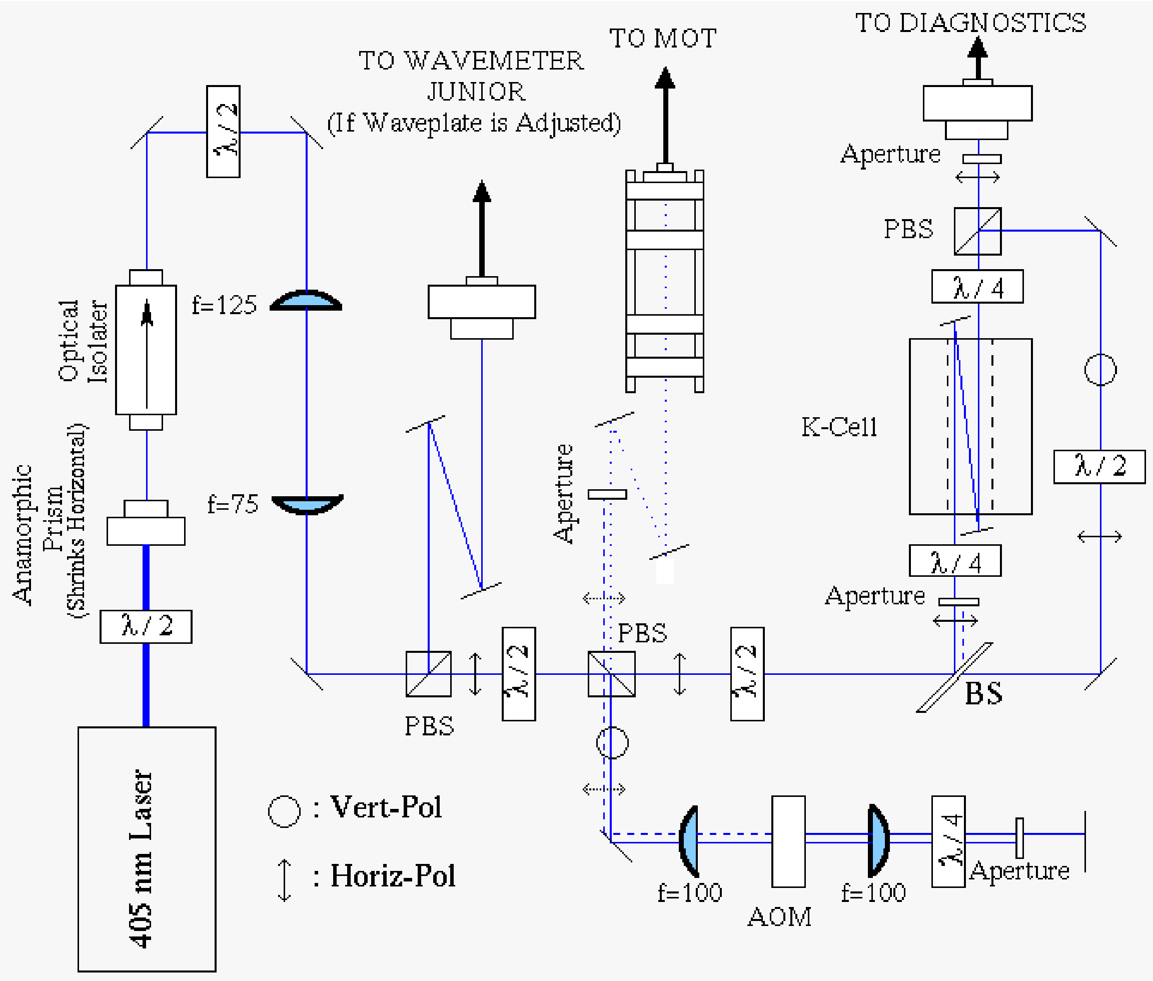
This code is a test algorithm to see if octave can easily sum multiple images without causing any pixel saturation. I add Gaussian noise to each image to simulate the behavior of the MOT then I smooth out the noise to see if the original faint image will get stronger.

*** There are other scripts not listed here but they were purely work spaces to test ideas or to plot some data quickly. I have not removed them in case someone wants to work through my thought process. Full disclosure, these scripts are not commented nor well structured.

Appendix B

Apparatus

Figure B.1: Diagram explaining the implemented optics



Bibliography

[1] David McKay. Potassium 5p line data, Jun 2009.