ENGINEERING COMPONENTS FOR THE CONSTRUCTION OF AN ULTRACOLD LITHIUM MACHINE IN THE PURSUIT OF NOVEL LOW-TEMPERATURE PHYSICS

A Thesis

Presented to the Faculty of the Colleges of Engineering and Arts and Sciences

of Cornell University

in Partial Fulfillment of the Requirements for the Degrees of

B.A/B.S

by

John Lombard

Winter 2012

ABSTRACT

In order to extend the research of the Vengalattore group in exploring the nature of quantum phase transitions, non-equilibrium dynamics, quantum optomechanics, and the spin textures of ultracold quantum gases previously applied to bosonic rubidium, new experimental apparatuses for creating Bose-Einstein Condensates (BEC) and ultra cold Fermi gases with lithium were developed. Detailed are the engineering design, construction, implementation, and characterization of the mechanical components, including a heat pipe based laser locking setup, a spin-flip Zeeman slower, and a lithium oven.

BIOGRAPHICAL SKETCH

John Lombard grew up on Long Island Sound in Connecticut, where he gained an intuitive appreciation for the physical laws which govern nature and the engineering constructs that humanity has developed to better understand their inner workings. After attending the Taft School where he won awards for his independent research in dye-sensitized nano-crystalline electron injected solar cells and vehicular wind turbines, he attended Cornell University. John is actively pursuing a dual degree, with a double major in physics and mathematics from the College of Arts and Sciences, and third major in mechanical engineering with a minor in applied mathematics from the College of Engineering. After a period of working in the field of experimental particle physics as a researcher in the USuRP program, undertaking an efficiency analysis of the electromagnetic calorimeter of the CMS detector of the large hadron collider at CERN, he joined the Vengalattore group and has since been working toward probing cold lithium physics by applying his interdisciplinary skills acquired throughout his broad scope of study. In his spare time, John pursues additional research in analytic number theory, is a published photographer, and performs as a jazz pianist and vibraphonist.

To the universe and each of its ever-intriguing facets.

ACKNOWLEDGEMENTS

I would like to acknowledge my family for their relentless support of my ambitions and their dedication to my development as both a scientist and citizen of humanity, my friends for equally substituting as family, the Vengalattore group for equally substituting as friends and hence family, and Cornell University, without which I should think I may not have had any of the above.

CONTENTS

Biographical Sketch iii
Dedication iv
Acknowledgements v
Contents vi
List of Tables viii
List of Figures ix

1 Introduction **1**
1.1 The Briefest Overview of Cold Atom Physics 1
1.1.1 Bose Einstein Condensates 1
1.1.2 The Cooling of Atoms 2
1.1.3 The Challenges of Lithium 3

2 The Heat Pipe **7**
2.1 Functionality 7
2.2 Design Requirements 7
2.3 Construction 8
2.4 Contaminant Management 11
2.5 Temperature Management 12
2.5.1 Insulation Calculations 15
2.6 Implementation 16
2.7 Argon 18
2.7.1 Vacuum Testing 18
2.7.2 Lithium Transfer 19
2.8 Spectroscopy 23
2.8.1 Purpose 23
2.9 Laser Lock Circuits 24
2.10 Optical Layout 26
2.11 Optically Associated Electronics 27
2.11.1 EOM 27
2.12 Implementation 29
2.13 Characterization 32
2.13.1 First Attempt 32
2.13.2 Second Attempt 35
2.14 Error Signal Generation and Locking 37
2.14.1 Hybrid Error Signal 37
2.14.2 Li6-D2 Lines 43

3 The Zeeman Slower **45**
3.1 Functionality 45
3.2 Design Requirements 46
3.2.1 Dimensional Constraints 46

3.2.2 Velocity Constraints . . . 47
3.2.3 Magnetic Field Profile . . . 48
3.2.4 Power Budget . . . 51
3.3 Monte Carlo Simulations . . . 51
3.4 The Spin-Flip Slower . . . 57
3.5 Virtual Construction . . . 60
3.5.1 Initial Individual Coil Design . . . 60
3.5.2 Layered Design . . . 62
3.5.3 Virtual Data . . . 63
3.6 Construction . . . 65
3.6.1 The Lathe . . . 65
3.6.2 Binding the Coils . . . 67
3.6.3 Templates . . . 67
3.6.4 Set Up . . . 69
3.6.5 Winding . . . 72
3.7 Characterization . . . 74
3.7.1 Field Measurements . . . 74
3.7.2 Probe Drifts and Re-Measurement . . . 76
3.7.3 Current Adjustments . . . 78
3.7.4 Residual Analysis . . . 79

4 The Oven 80
4.1 Functionality . . . 80
4.2 Design Requirements . . . 80
4.3 Design . . . 81
4.4 Crucible Redesign . . . 84

A Heat Pipe Insulation Analysis 85

B Matlab MC Simulations 94

C Mathematicica Virtual Build and Characterization 105

Bibliography 148

LIST OF TABLES

1.1 Table of Lithium 7 Atomic Properties [7] [12] 3

3.1 Summary of Zeeman Slower Design 63
3.2 Summary of Virtual Analysis 65
3.3 Current Recipes for each Fitting 78
3.4 Final Field Residual Data (in Gauss) for Constructed and Analytically Optimized Zeeman Slower 79

LIST OF FIGURES

1.1 Lithium Energy Level Diagrams [13] 4
1.2 Lithium Vapor Pressure Curve[9] 5
1.3 Rubidium Vapor Pressure Curve[9] 6

2.1 Heat Pipe CAD Drawing 9
2.2 Heat Pipe CAD Figure 10
2.3 Heat Tape Wrapped Around the Pipe for Baking 12
2.4 Wet Moldable Insulation Sheets 14
2.5 Adhesive Backed Insulation Tape 14
2.6 Outside Temperature of Insulation (K) vs Insulation Thickness (m) 15
2.7 Variac Voltage Required (V) vs Insulation Thickness (m) 16
2.8 Brass Mounts Holding the T-branch 17
2.9 Heat Pipe Viewports and Mounts 17
2.10 Lithium Granules 20
2.11 Mesh Tube 22
2.12 Laser Lock Circuit 25
2.13 Optical Layout in the Breadboard Drawer: '>' = Laser Output; 'L' = Lens; 'M' = Mirror; 'WP' = Wave-Plate; 'BS' = Beam-Splitter; 'PD' = Photo-Detector 26
2.14 LRC Circuit 28
2.15 Lock Box Front Panel 30
2.16 Laser Circuit Box, Showing Contents 31
2.17 Laser Circuit Box, Sealed 31
2.18 Doppler Signals vs. Temp at the Cross 33
2.19 Heat Pipe with Adjusted Insulation and Fins 35
2.20 Saturated Absorption Signal with a Broad Sweep 36
2.21 The Li7-D2 Lines 36
2.22 Long Term Behavior of the DAVLL Error Signal 39
2.23 Long Term Oscillations Due to the 60 Hz Field 40
2.24 Final Stable DAVLL Error Signal 40
2.25 Schematic of the Absorption Splitting with Sidebands 42
2.26 Hybrid Error Signal 42

3.1 Ideal Magnetic Field Strength as a Function of Slower Length . . . 51
3.2 Capture Regions for the Ideal Field 53
3.3 Evidence of Near Resonant Atoms Being Slowed 54
3.4 Capture Histogram: X Axis = Final Velocity After Slower Given a Gaussian Input Beam; Y Axis = Number of Atoms 55
3.5 Beam Spread by the End of the Slower 56
3.6 New Spin Flip Magnetic Field Profile (in Gauss) as a Function of Distance Along the Slower (meters) 59

3.7 Ideal and Virtual Field Profiles Measured in Gauss vs Slower Length
Green and red hashes indicate the start and stop of individual coils, and the blue hashes indicate the boundaries of the cross 61
3.8 Individual Design Residuals of the Field Profile from the Ideal Along the Length of the Slower 61
3.9 Green and red hashes indicate the start and stop of layers or individual coils, and the blue hashes indicate the boundaries of the cross 62
3.10 Histogram of Virtual Currents Through Each Coil 64
3.11 Layered Design Residuals of the Field Profile from the Ideal Along the Length of the Slower 64
3.12 A Coil Mount on an Axel, Ready to be Wound Upon 66
3.13 Wire through the Tensioner 66
3.14 Examples of Used End-Plates 68
3.15 An Example of a Large Center Axle used in between Two End-Plates 69
3.16 Wire Spool Setup 71
3.17 The Start of a New Coil, Including the Initial Bend through the End-Plate Slot 71
3.18 One of the Independent Coils 73
3.19 The Finished Layered Portion of the Zeeman Slower 74
3.20 A Sketch of the Field Measurement Setup 76
3.21 Magnetic Field vs Axial Distance
Virtual Field = Green; Forward and Backward Data Sets = Red; Averaged Set = Black 77

4.1 Lithium Oven, Side View 82
4.2 Lithium Oven, Close Up on the Mounts and Feed-Through 83
4.3 Redesigned Drawing of the Crucible 84

CHAPTER 1

INTRODUCTION

1.1 The Briefest Overview of Cold Atom Physics

Ultra-cold atomic physics provides a window into the effects of quantum mechanics on macroscopic scales. Often, one may think of quantum physics being applicable to phenomena on the smaller scale of matter, that of the atomic and subatomic. However, statistical mechanics tells us that collections of bosons, particles or atoms with an effective integer spin, can occupy the same quantum state. Unlike fermions, particles with half-integer spin such as electrons which obey Fermi-Dirac statistics, a collection of bosons can, under the right conditions, form a macroscopic quantum body. Upon achieving a certain critical temperature, collections of over 100,000 atoms can collapse into a new state of matter known as a Bose-Einstein Condensate. This quantum object allows for experiments to be performed on a body much larger than the atomic scale while still probing the quantum nature associated with behaviors such as phase transitions or non-equilibrium dynamics, or to establish a highly refined instrument capable of measuring magnetic fields with immense accuracy and precision, to name a few practical implementations.

1.1.1 Bose Einstein Condensates

In order to form a BEC, the atoms must achieve a critical phase space density. This parameter is a function of atomic properties such as mass, as well as the number density of the collection and its temperature. By re-writing this criterion as a critical temperature, we have a clear expression for the temperature range at which we would expect to see a BEC.

$$T_c = (\frac{n}{\zeta(3/2)})^{2/3} \frac{2\pi\hbar^2}{mk_B} \tag{1.1}$$

Where

$$\begin{aligned} n &\equiv \text{Particle Density} \\ m &\equiv \text{Mass per Boson} \\ \hbar &\equiv \text{Reduced Planck Constant} \\ k_B &\equiv \text{Boltzmann Constant} \\ \zeta &\equiv \text{Riemann Zeta Function} \end{aligned}$$

Due to their well characterized electronic structures, most cold atom experiments involve elements from the Alkali group. For a typical rubidium experiment, one would expect temperatures in the nano-Kelvin range for condensation. This presents an engineering challenge of achieving and maintaining extremely cold temperatures in an ultra high vacuum environment on the order of tens of pico-Torr.

1.1.2 The Cooling of Atoms

The process of cooling atomic gases is a multi-phase series of techniques used in succession to step the atoms down past various thermal barriers. A standard BEC experiment might employ the following system of instruments and processes:

1. A heat pipe, used to vaporize the solid element sample
2. An oven, used to chamber the vapor and inject it into the next apparatus at high atom velocity through a needle nozzle

3. A Zeeman slower, the first stage in the cooling process in which the atoms are reduced from their post-nozzle velocities down to a trappable group velocity
4. A chamber in which the atoms will be held in a magneto-optical trap and slowed in an optical molasses by counter propagating lasers
5. A variety of additional deep cooling techniques such as Sisyphus[1] cooling or Raman sideband cooling[2], involving the application of electromagnetic fields and carefully coordinated optically excited transitions
6. A final stage of evaporative cooling, in which the atoms make the transition to a BEC

1.1.3 The Challenges of Lithium

Unlike rubidium, which the Vengalattore lab has been cooling since their first generation of BEC experiments, lithium presents a variety of new challenges to overcome in the construction of a machine due to its atomic properties. The following is an abbreviated table of some of the atomic properties of bosonic lithium 7 and fermionic lithium 6 which are relevant to this thesis, for reference.

Property	Symbol	Value	Li^6	Li^7
Abundance	-		7.59%	92.41%
Mass	m		6.015122794 amu	7.016004548 amu
Melting Point	T_M	453.69 K		
Spontaneous Radiative Lifetime	τ		27.102 ns	27.29 ns
D-1 Line	λ		670.992 nm	670.976 nm
D-2 Line	λ		670.977 nm	670.962 nm

Table 1.1: Table of Lithium 7 Atomic Properties [7] [12]

[1] See [5] for the original publication

[2] See [8] for the original publication

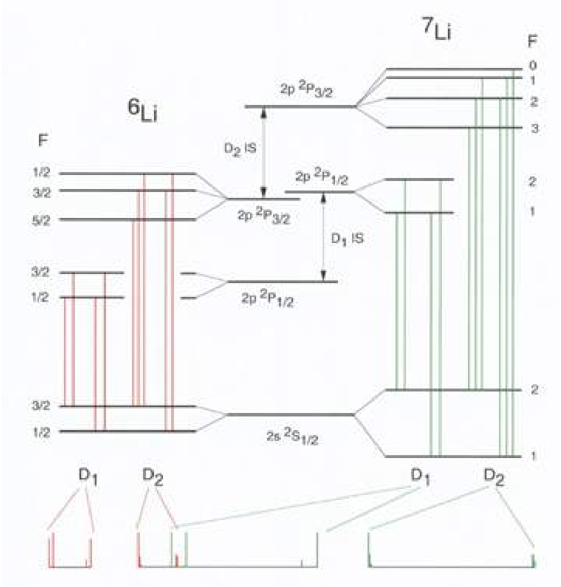


Figure 1.1: Lithium Energy Level Diagrams [13]

Corrosion

Lithium is a very corrosive element at high temperatures and vapor pressures. In terms of the vacuum equipment, often standard conflat flanges are used to create seals between components. These flanges utilize one-use soft metal gaskets which the knife edge of the flange bites into, creating an impermeable seal upon bolting. On the rubidium experiment, copper gaskets are sufficient for this purpose. Lithium, however, will eat away at the copper and weaken the seal, causing vacuum leaks.

Lithium also corrodes glass. The various optical access ports of the machine could be compromised by contact with hot lithium vapor.

Temperature

The temperature at which lithium achieves the necessary vapor pressure for spectroscopy is much higher than rubidium. As shown in the reference charts containing the lithium 1.2 and rubidium 1.3 vapor pressure curves, for an operational vapor pressure of roughly 10^{-6} ATM, lithium needs to be heated to a temperature of about 4 times that of rubidium, to approximately 450 °C. This presents engineering challenges in the realms of thermal stresses, gradients, insulation issues, and temperature stability.

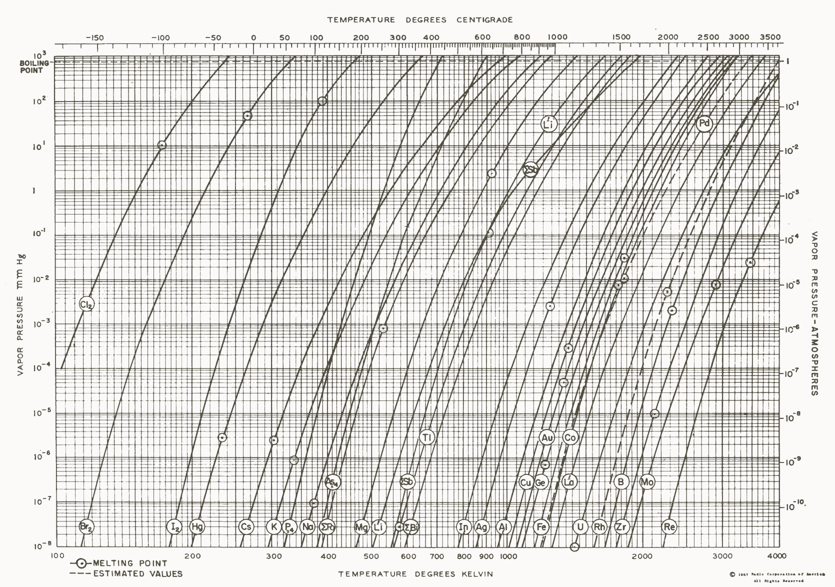


Figure A1(a). Vapor pressure curves for the more common elements. After Honig (Ref. 5:14). (*Courtesy RCA*

Figure 1.2: Lithium Vapor Pressure Curve[9]

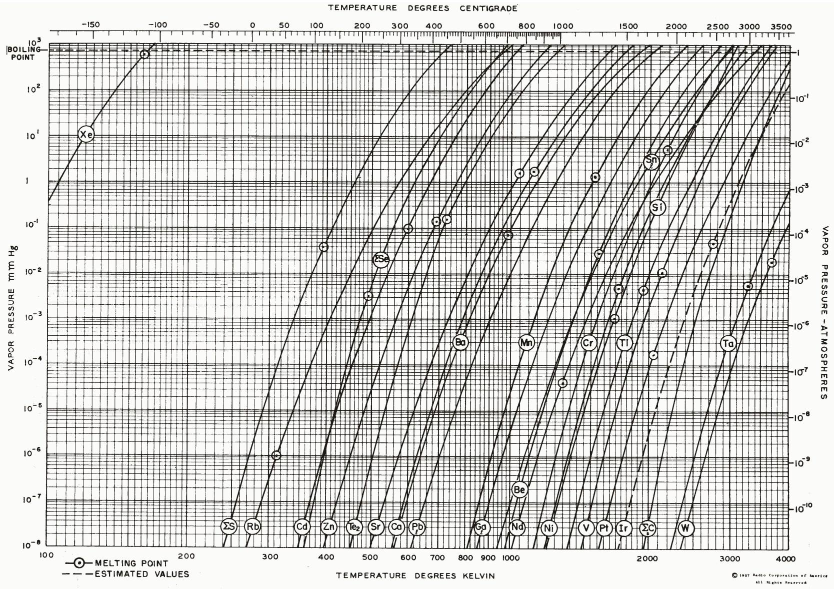


Figure A1(b). Vapor pressure curves for the more common elements (cont.). After Honig (Ref. 5:14). *(Courtesy RCA Laboratories.)*

Figure 1.3: Rubidium Vapor Pressure Curve[9]

Oxidation

Lithium, unlike rubidium, rapidly oxidizes in contact with the atmosphere. In order to achieve the appropriate vapor pressure, all impurities in the lithium sample must be minimized before insertion into the vacuum chamber. As a result, the process of insuring that the lithium remains unoxidized in transfer to the machine needed to be accounted for.

CHAPTER 2

THE HEAT PIPE

2.1 Functionality

Heat pipes are typically used in two respects for a cold atom machine. The first purpose would be a cell for spectroscopy. After creating a small, localized chamber in which the lithium can be heated to adequate vapor pressure and the atoms excited by laser, the resulting absorption spectral lines can then be used as a source to which the the rest of the experiment's lasers can lock to, fixing their frequency to the desired transition frequency. A second use for a heat pipe would be to act as the lithium source for an oven. Essentially a port through which lithium can be added and removed, this form of heat pipe would not require the optical access of the laser locking version.

2.2 Design Requirements

The material for the heat pipe would have to withstand vacuum pressures of 10^{-11} Torr, be resilient against lithium corrosion, and be able to be heated to temperatures as high as 450 °C. The cell would need to be compact, as to both minimize laboratory space and also surface area, limiting detrimental heat transfer to the surrounding experiment and the area which needs to be properly insulated. There would also need to be precautions against the lithium vapor attacking the viewport glass.

2.3 Construction

The heat pipe was constructed from stock 304L stainless steel ultra-high vacuum (UHV) components from Kurt Lesker. The lithium was housed in a $1\frac{3}{4}$" conflat (CF) nipple sealed with a blank flange, which was heated to the necessary temperature. That nipple connected to a 4 way cross. The cross itself connected to two arms, each nipples, with Kodial glass viewports on each side for spectroscopy. Directly across from the nipple containing the lithium, an additional nipple was installed. These components completed the main body of the heat pipe. The flanges that were in direct contact with the hottest regions of the heat pipe, (the four cross flanges and the blank flange), were secured with nickel gaskets instead of the standard copper. These gaskets are resistant to the lithium corrosion, but are much harder than the copper gaskets. As a result, the ultimate lifetime of the knife edge of the flange components is diminished with each use of one of these gaskets [1]. However, lithium corrosion in this region would be minimized. Standard copper gaskets were used elsewhere, along with standard silver-lubricated vacuum bolts, nuts, and washers. To seal off the heat pipe, a manual bellows sealed stainless steel conflat angle valve was installed off of the final open nipple to allow for manual pressure regulation. The total weight of this system was approximately 5 pounds, and required a working surface area of 10" at maximum width and 16" at maximum length.

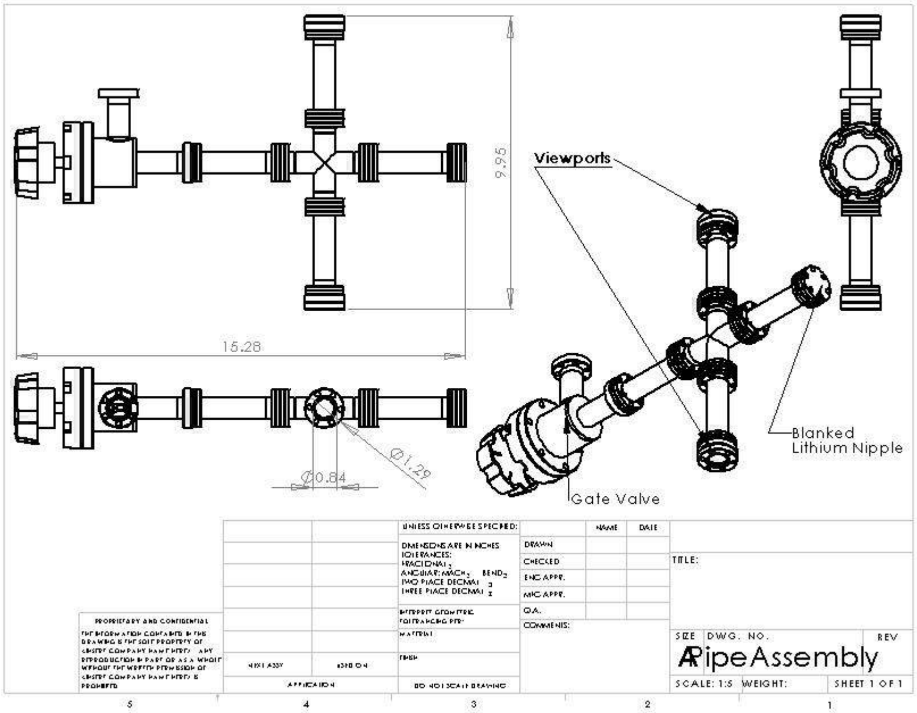


Figure 2.1: Heat Pipe CAD Drawing

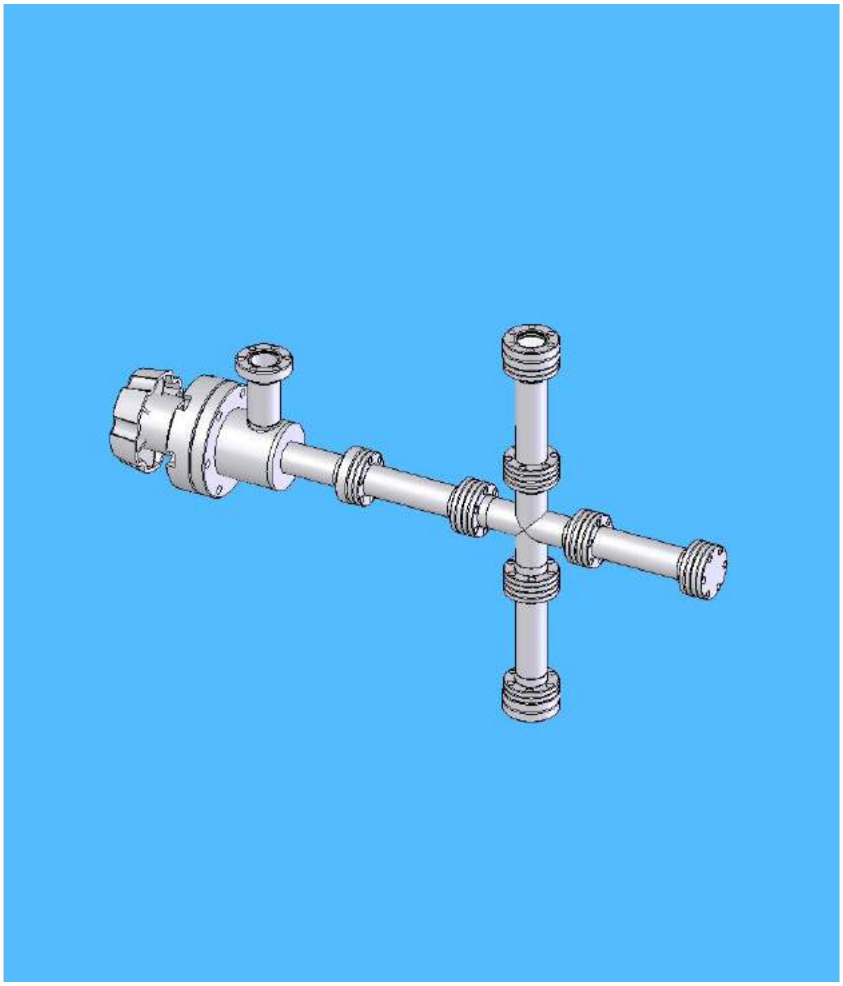


Figure 2.2: Heat Pipe CAD Figure

2.4 Contaminant Management

UHV components are extremely susceptible to any particulate matter, oils, or water. Any contaminants would prevent the system from achieving the necessary vacuum pressures in a reasonable time scale since the contaminants would continually out-gas. Therefore, extreme care was taken when handling the vacuum parts. Nitrile gloves were worn at all times, and the parts were kept in a dry environment sealed in aluminum foil. All parts underwent an ultra-sonic cleaning procedure before installation as well. This process is a 4 wash method for cleaning UHV components and is detailed below.

1. With the items placed in a clean glass container, fill the container with deionized water, cover the container carefully with foil to prevent splash contaminants, and place in an ultrasonic bath for between 5 and 10 minutes.
2. Remove the parts and clean the container with more deionized water. Refill and repeat.
3. Remove the parts, and dry carefully using lint-free cloth and a heat gun, ensuring that no water droplets remain.
4. Fill the container with acetone, and sonicate as before.
5. Remove the parts, and dry completely.
6. Fill the container with methanol, and repeat the sonication.
7. Dry the parts for the last time, and wrap carefully in aluminum foil for preservation.

The deionized water removes any particulate matter; the acetone removes oily residues; the methanol removes the acetone residue.

2.5 Temperature Management

The heat pipe was heated using heat tapes, braided resistive fabric tapes which generate local conductive heat upon the application of an current.

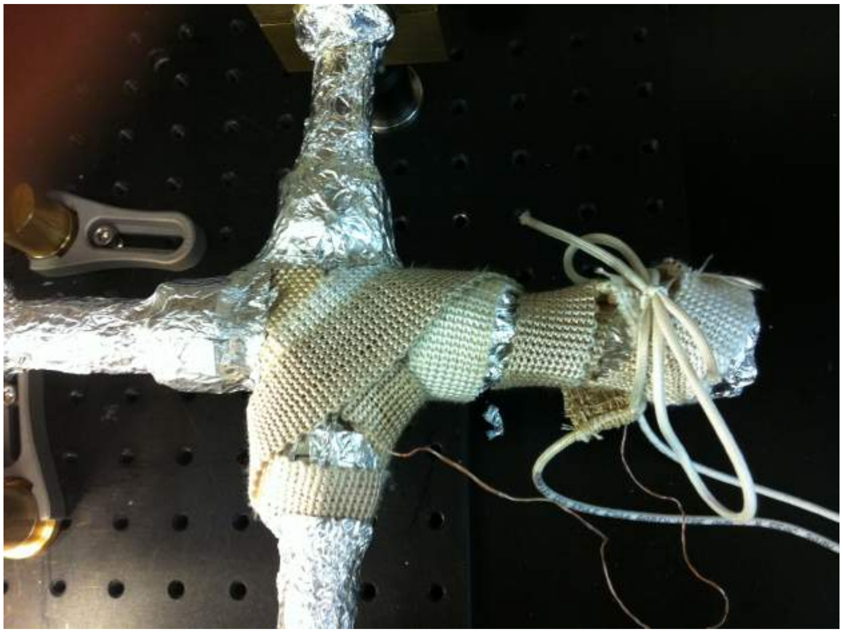


Figure 2.3: Heat Tape Wrapped Around the Pipe for Baking

These tapes were wrapped around the containment nipple of the heat pipe to create a region of locally high temperature. As the lithium vapor could still corrode the viewports, a large thermal gradient needed to be established to insure that although the vapor would be at the required density for spectroscopy in the region of the cross, by the time the atoms migrated to the outer flanges, they would condense on the walls of the nipples before reaching the viewports. The company Cotronics [4] specializes in high temperature insulation materials, and a combination of their products were implemented. First, their wet moldable ceramic sheets 2.4 were used to create an insulating 2-part mold around the central region of the heat pipe. Next, their high temperature adhesive insulating tape 2.5 was used to hold the cast together. This combination is rated to over 600 °C, well beyond the needs of the heat pipe. Its ease of removal also allowed for adjustments and accessibility to the flange components for replacing. The poor thermal transmittance of stainless steel was to create enough of a gradient on its own to avoid the need for cooling the viewports to introduce the condensation wall in the nipple.

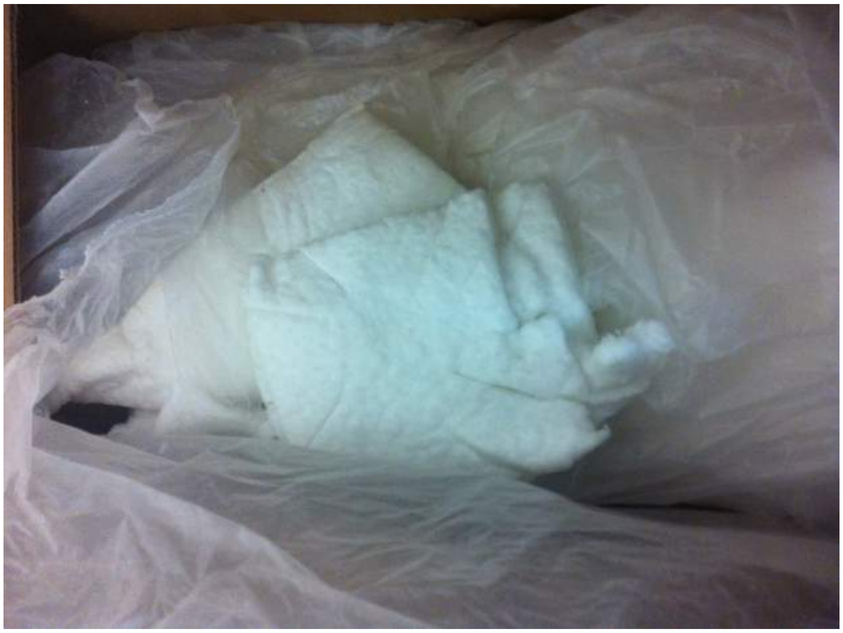


Figure 2.4: Wet Moldable Insulation Sheets

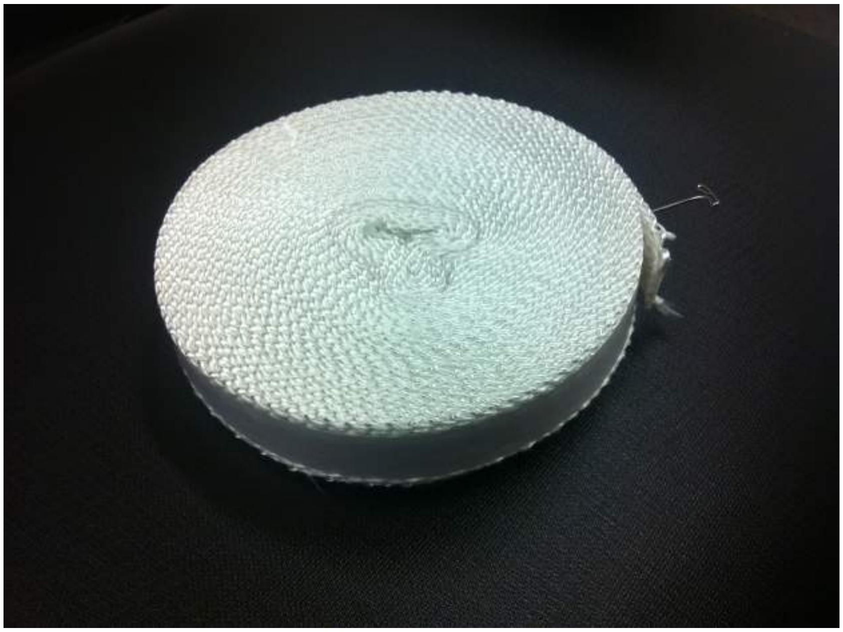


Figure 2.5: Adhesive Backed Insulation Tape

2.5.1 Insulation Calculations

In order to determine the required amount of insulation to minimize the optical space on the table, the required voltage input for the heating tape, and the external temperature of the insulation to avoid the heating of the optical space, a heat transfer analysis was undertaken. The results allowed us to consider a range of insulation thicknesses which would optimize the above parameters. We anticipated roughly 5 cm of insulation, which would yield a 30 °C outer temperature and would require only 50 V from the Variac to reach the desired steady state temperature. The experimental installation landed precisely within the calculated range, confirming the heat transfer model and curves shown below. The Mathematica code used to generate the results can be found in Appendix A.

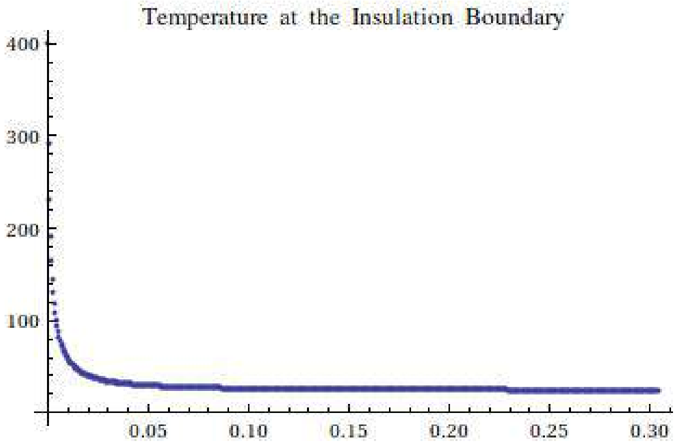


Figure 2.6: Outside Temperature of Insulation (K) vs Insulation Thickness (m)

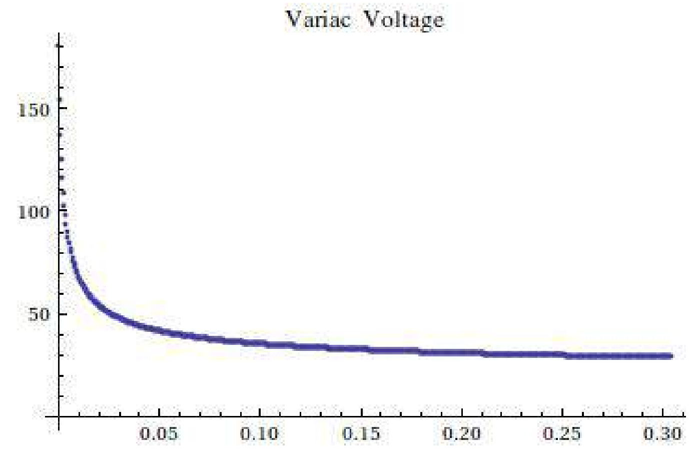


Figure 2.7: Variac Voltage Required (V) vs Insulation Thickness (m)

2.6 Implementation

The heat pipe was mounted on adjustable brass posts on an optical breadboard, maintaining a 10 ° inclination with the ground to insure that the lithium remained in the bottom nipple upon melting. The manual valve attached to a T-branch, one end of which heading toward a turbo pump, and the other end to an ion gauge for monitoring the more extreme heat pipe vacuum pressure, both secured by additional brass clamps.

Off of the turbo pump, a series of Klein flange (KF) vacuum tubes connected a thermocouple pressure gauge and a roughing pump. The roughing pump would need to achieve vacuum pressure of at least 100 mTorr, as measured by the course gauge thermocouple, before the turbo pump could be activated. From this pressure, the turbo has the capability to take the system to a vacuum pressure at least on the order of 10^{-8} Torr as measured by the ion gauge.

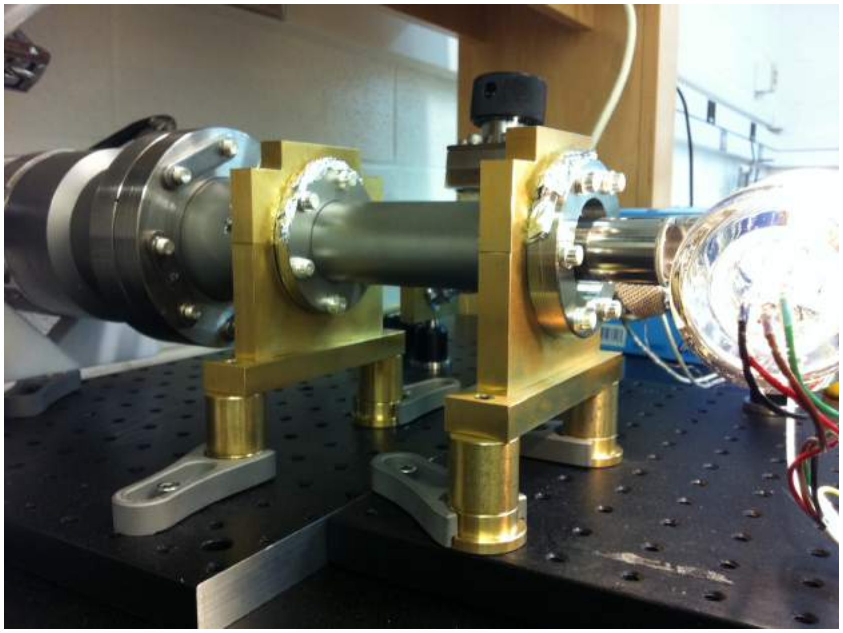


Figure 2.8: Brass Mounts Holding the T-branch

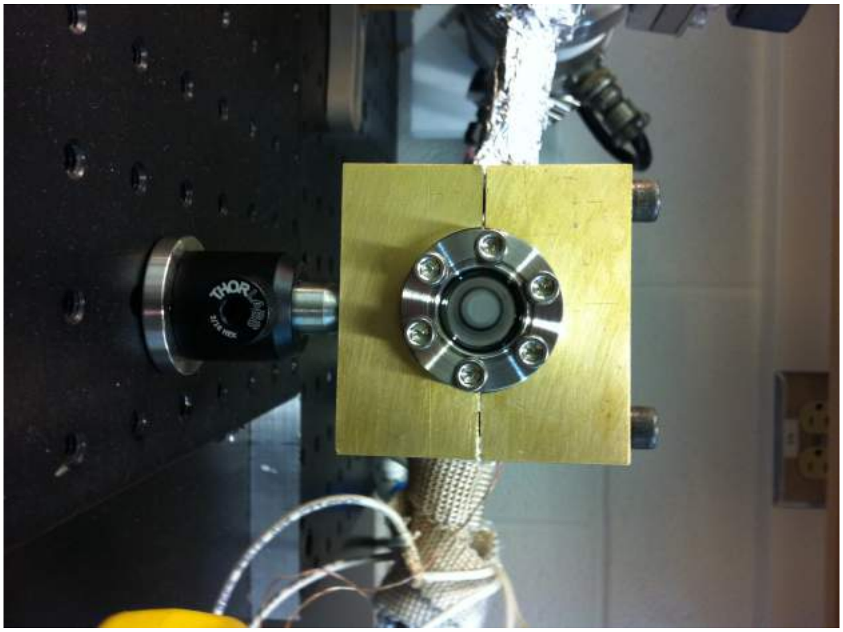


Figure 2.9: Heat Pipe Viewports and Mounts

2.7 Argon

In order to add an additional prevention measure for stopping the lithium from aggressively attacking the viewports, the mean free path of the lithium atoms was reduced by adding an inert buffer gas. By over-pressuring the heat pipe with argon, the hot lithium atoms can not accumulate enough velocity to travel all the way down the side nipples to the viewports without first colliding with a wall and condensing. The process of adding the argon without causing the vacuum environment to be exposed to atmosphere involved a few careful steps. First, the manual gate valve on the heat pipe was closed to secure the vacuum. Next, the turbo pump was turned off and a valved plastic gas hose was attached to an open port on the pump. The system was pumped down to mTorr by the roughing pump before the argon valve was opened to allow a steady stream to flow into the system. The heat pipe valve was then slowly opened and the vacuum was replaced with argon. After a pressure of roughly 300 mTorr was achieved, the heat pipe valve was closed, sealing the system. 300 mTorr of argon is too great a pressure to actually see the Doppler or fine structure signals for spectroscopy, however. The argon would have to be gradually leaked out until an empirical optimum could be achieved between the longevity of the heat pipe and the signal strength.

2.7.1 Vacuum Testing

The roughing section of the system settled to a vacuum pressure of 30 mTorr after 2 days of pumping. After fixing various hose and gasket leaks which were incurred during assembly due to poor clamp and gasket bindings, the turbo section of the system could immediately drop the heat pipe into the micro-Torr range, and would eventually settle in the 10's of nano-Torr after a few days.

To allow the heat pipe to out-gas any contaminants or other particulates of the inner metal surface, the heat pipe was preliminarily baked using the heat tapes. A slow bake of about 75 °C while being continually pumped with the roughing pump for about a week helped the heat pipe expel water vapor and particulates and achieve vacuum pressures as low as $2 * 10^{-8}$ Torr once the turbo was re-engaged.

2.7.2 Lithium Transfer

The initial order of lithium received contained a bottle of oxidized pellets which were approximately 2.5 mm long and of 1 mm diameter, as shown in figure 2.10. These pellets needed to be cleaned before transfer to the heat pipe to remove the oxidation. As per the procedure outlined in the Stan thesis [16], a mixture of methanol and dry ice created a bubbling cleanser with which the lithium, upon submersion for approximately 15 seconds, lost its oxide coating and could be removed as a shiny yellow-white sample of unoxidized lithium. Due to the small size of the lithium granules, however, it became apparent that a method for dipping and retrieving the lithium samples before they rapidly re-oxidized upon re-exposure to the atmosphere needed to be developed.

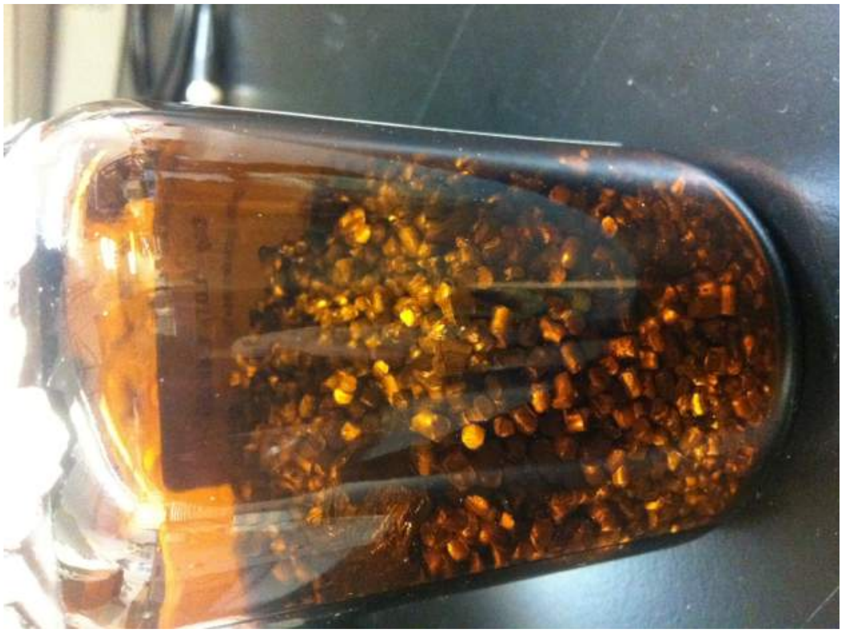


Figure 2.10: Lithium Granules

First Attempt

The first attempt at transfer simply involved a metal spoon, which we used to fish out the lithium pellets from the mixture, and a steady stream of argon, used to provide an inert atmosphere to prevent re-oxidation. The nipple of the heat pipe was removed and filled preliminarily with argon, in the hopes that since argon is heavier than the atmosphere, it would sit in the nipple and provide a temporary housing for the transfered lithium before the heat pipe was again pumped down to vacuum. The pellets were cleaned by the process described above, and after removal from the methanol solution, argon was blown onto the pieces to remove as much liquid methanol as possible before transfer to the nipple. Finally, after about 2 grams were transferred, the nipple was replaced on the heat pipe and the pump down was commenced. It was immediately noticeable through the viewports that although it was assumed that any excess liquid would just evaporate with time during the pump down, instead, the methanol created a dense foam which had bubbled up to the viewport level as it continued reacting with the lithium. This foam not only would not dissipate, but it also carried a gray powder with it which indicated that the lithium had re-oxidized. The heat pipe was subsequently disassembled, cleaned, and prepared for a second trial.

Second Attempt

Learning from the first trial, it was determined that the lithium needed to be completely dry before deposition in the nipple. The small size of the granules proved to be an issue, for although the granules had more surface area for drying, the larger surface area also made it difficult to completely insure that the samples were bathed in the stream of argon, and before adequate drying could be accomplished, the light samples would either re-oxidize or be blown away by the argon. A potential solution to this issue involved a tube of wire mesh with a removable closed bottom, fabricated to help house the lithium pellets while drying and facilitate their placement into the nipple, as shown in figure 2.11
Additionally, the tube through which the argon was being dispensed was bent into a circle, and holes were punctured along the inner circumference of the new geometry. This created an axially symmetric argon environment which could be placed over the containment tube and dry the lithium from all sides while preventing re-oxidation. A series of experimental tests confirmed that this system as well would not be sufficient for transfer into the nipple.

Third Attempt

The Cornell Center for Materials Research [6] recently converted their glove box into a sealed argon environment. This provided an ideal means to clean the lithium

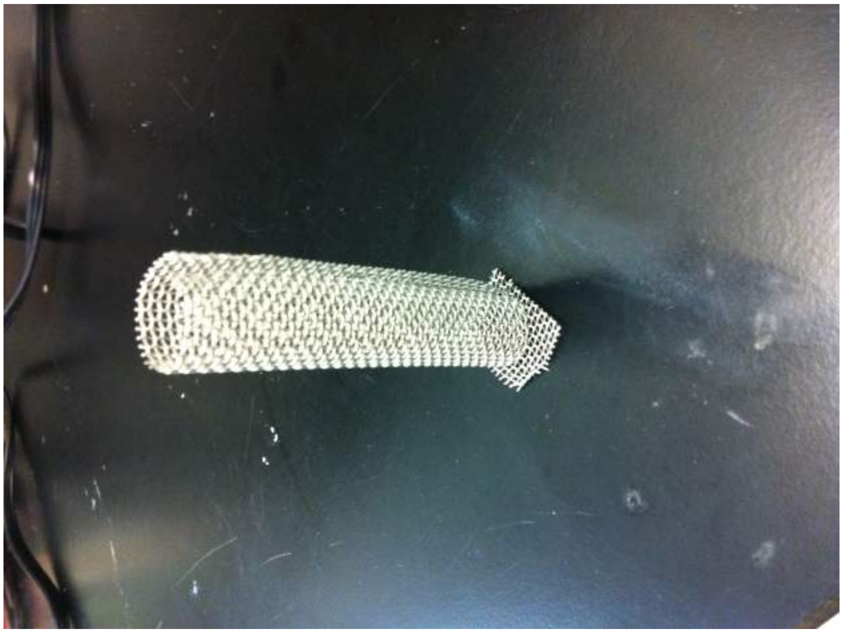


Figure 2.11: Mesh Tube

without fear of oxidation. Yet, upon the start of the methanol and dry-ice reaction in the glove box, the oxygen levels had risen from about 3 ppm to over 1000 ppm. Despite fact that the reaction itself should not have produced oxygen, the possibility for water condensation on the dry ice could have introduced the necessary elements for a secondary reaction to occur. This effectively contaminated the glove box, and we rapidly noticed re-oxidation of the lithium.

Final Success

It was concluded that the small lithium pellets were too difficult to both clean chemically, remain unoxidized, and transfer dry into the nipple. A larger sample of lithium was ordered instead. A 4” long, 2” diameter cylinder of lithium arrived, pre-packed in argon, and unoxidized. This sample was opened in the newly restored glove box. Although unoxidized, the lithium surface had some factory impurities which were scraped away with a razer blade. After caving off large samples, and using pliers to form the samples into shapes which could be easily inserted into the nipple, about 5 grams were transferred and sealed successfully. These 5 grams should be enough for over a year of operation, as based on the Stan experiment [16], and additional lithium will be added using this method in the future.

An unintended benefit of the glove box approach to adding lithium to the heat pipe was a natural pressuring of the heat pipe with atmospheric argon. This greatly reduced the complexity of the argon pressurization procedure outlined above: the pipe now only needed to be carefully pumped down to the empirical ideal argon pressure for a spectroscopy signal.

2.8 Spectroscopy

2.8.1 Purpose

In order to insure that the laser remains on resonance with the desired transition for the lithium atoms, the laser must be first tuned to that transition frequency within the natural linewidth of the transition, roughly 6 MHz, and then stabilized to avoid drifting away from resonance. One way that this can be accomplished is by using the re-emitted light of the lithium atoms themselves after absorbing the incident laser light. However, since the lithium atoms will be traveling at various speeds v_i due to thermal motion as given by the Maxwell-Boltzmann distribution,

$$f_v(v_i) = \sqrt{\frac{m}{2\pi k_B T}} e^{\frac{-mv_i}{2k_B T}} \tag{2.1}$$

the atoms will experience a Doppler shift with respect to the incident k-vector by

$$\Delta\nu = \frac{v}{c}\nu_0 \tag{2.2}$$

At the designed temperature of the heat pipe, and root-mean-square average speed of the atoms given by

$$v_{rms} = \sqrt{\frac{3k_BT}{m}} \tag{2.3}$$

This would correspond to a Doppler broadening of the absorption spectrum by 2.5 GHz. To avoid this, the method of saturated absorption spectroscopy was implemented to preselect atoms with no effective velocities for interaction with the probing laser. For an understanding of this technique, see the Widagdo undergraduate thesis [15].

2.9 Laser Lock Circuits

Locking the laser requires an active PID circuit to stabilize the feedback error signal around the desired frequency. The circuit constructed was the same design that the laboratory has previously implemented in its other experiments, and features a means for controlling the current modulation gain of the laser, the piezo voltage gain, an adjustable bias offset, and a functional sweep input with bias and gain.

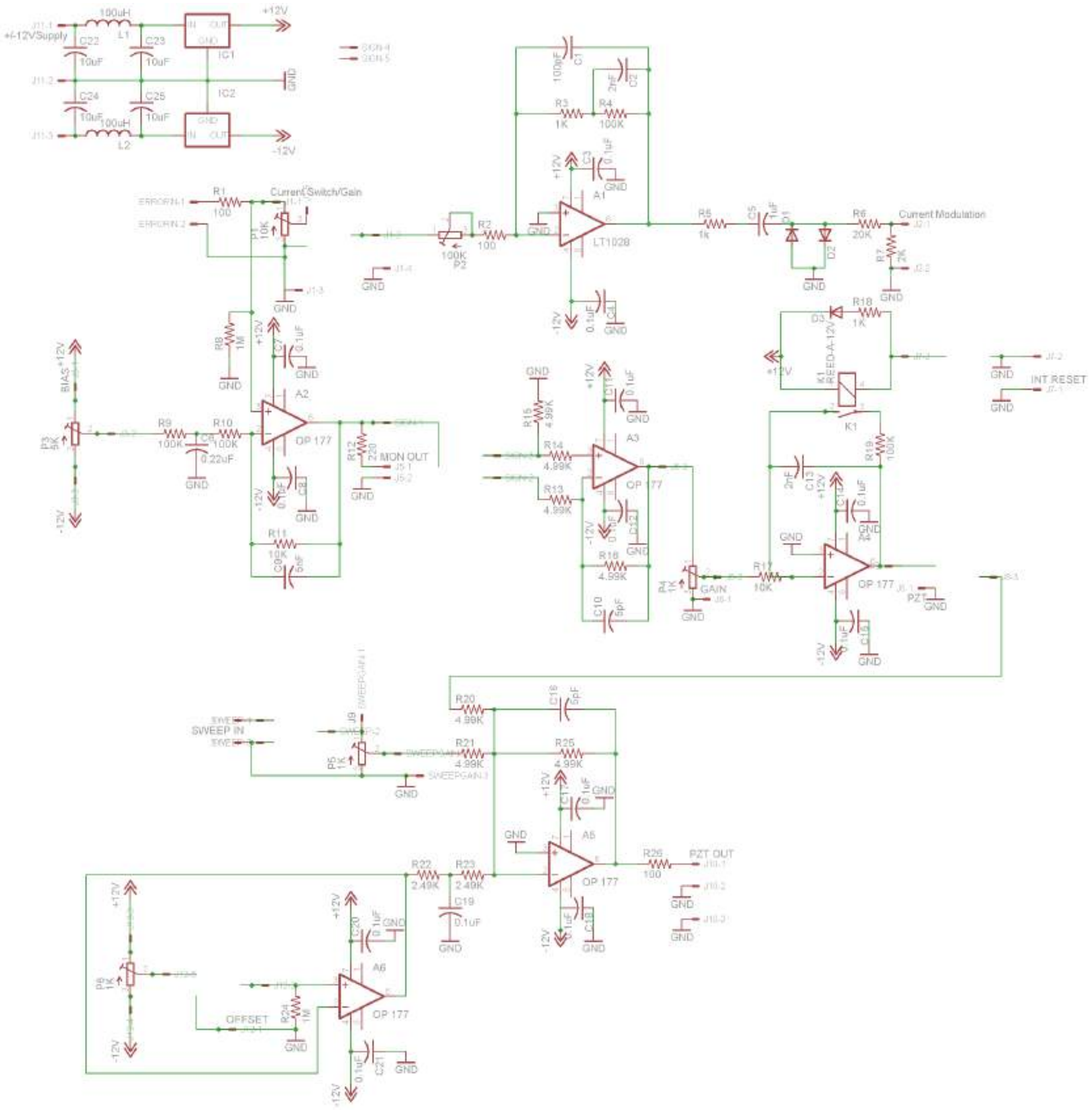


Figure 2.12: Laser Lock Circuit

2.10 Optical Layout

The optical layout follows a standard saturated absorption setup with a larger diameter and intensity pump beam and a smaller counter-propagating probe beam with frequency modulation before passing through the lithium cell. A schematic of this setup can be found in figure 2.13 below, and closely follows the functional block diagram of Widagdo [15] with alterations to the footprint to meet our geometries.

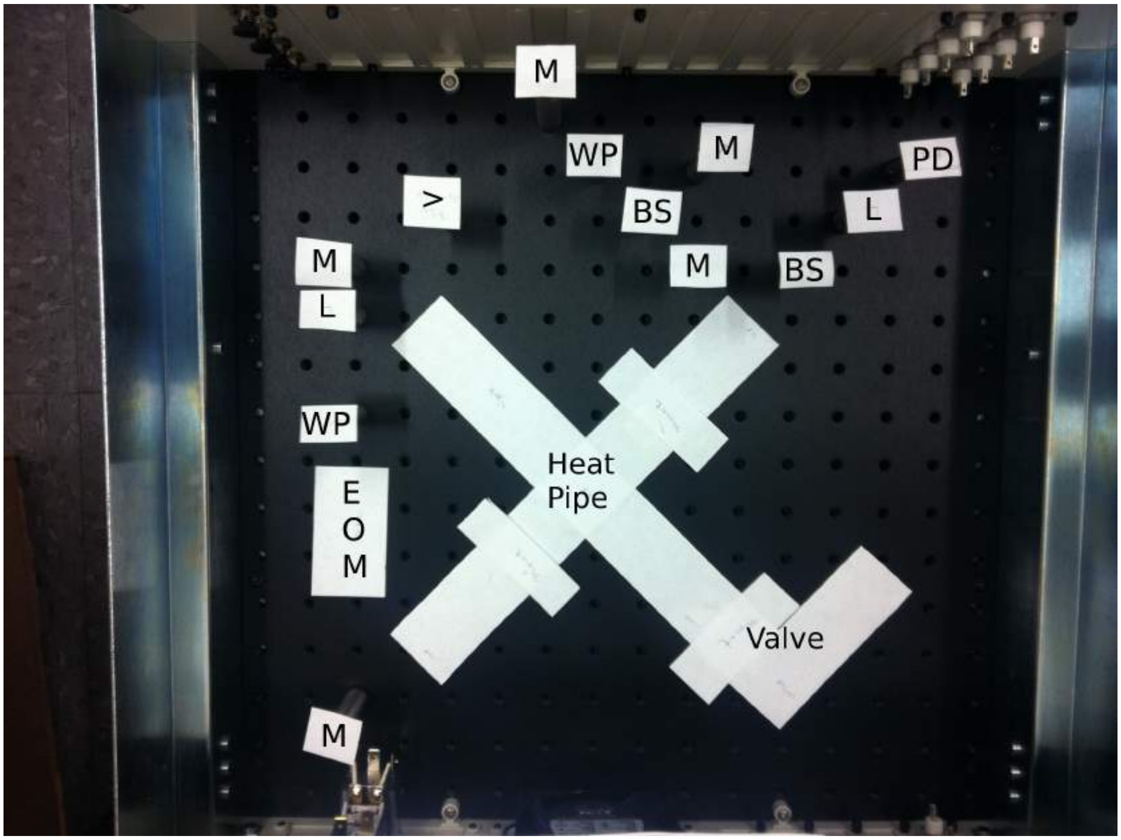


Figure 2.13: Optical Layout in the Breadboard Drawer: '>' = Laser Output; 'L' = Lens; 'M' = Mirror; 'WP' = Wave-Plate; 'BS' = Beam-Splitter; 'PD' = Photo-Detector

2.11 Optically Associated Electronics

2.11.1 EOM

The Electro-Optical Modulator used to add sidebands to the saturated absorption signal in order to produce an appropriate error signal which can be locked to required a nominal driving voltage of about 200V for our roughly 670 nm wavelength input light according to the product manual [18]. Additionally, the EOM was desired to be driven at a tunable 4-6 MHz range. This frequency range was selected in order to provide greater locking capabilities, as the system is most efficient when modulated around the natural linewidth of the atoms. Compared to a frequency of 35 MHz, a 5 MHz drive will allow for potentially large drifts away from the desired locking point while still being able to be forced back into lock, as opposed to a system which would have to be manually corrected if too great an error is accumulated. Generating this voltage drop at the desired frequencies was accomplished by a tunable 'tank circuit'.

The Tank Circuit

A standard series LRC resonant circuit as shown in figure 2.14 with high gain can build up substantial voltage drops from relatively small driving sources by having a large quality factor. In this implementation, the inductor was introduced by a surface mounted ferrite bead; the resistances considered were the 50Ω output impedance of the function generator source and the associated resistance of the inductor; and the capacitance considered was the fixed 12.1 pF of the EOM's lithium niobate crystal itself and additional parallel capacitors. By tuning the capacitance, a resonance in the desired frequency range was able to be achieved with a gain of roughly 20, allowing for the EOM to be driven by a function generator at only about 10 volts instead of a high voltage amplifier.

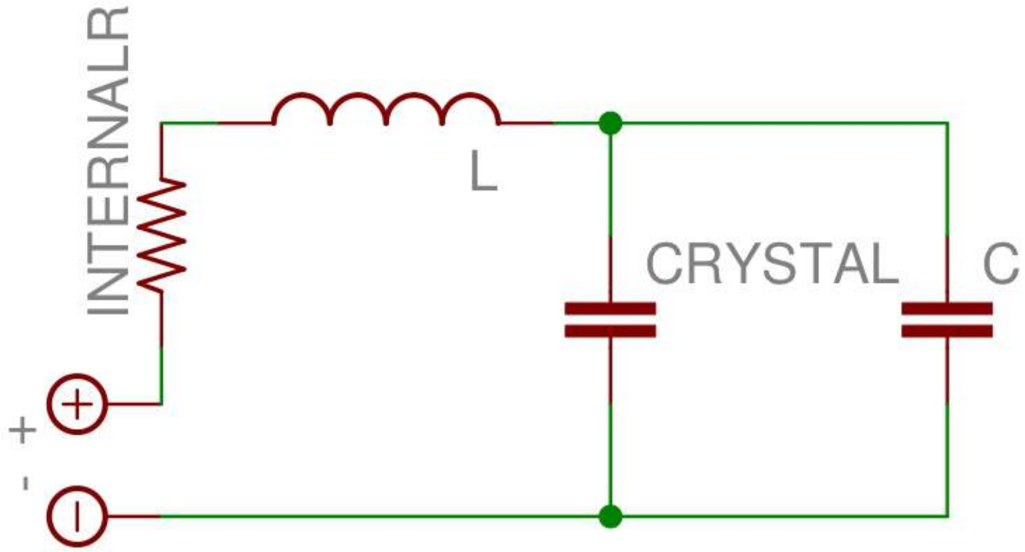


Figure 2.14: LRC Circuit

2.12 Implementation

The entire locking setup was installed in an optical breadboard drawer and inserted into a standard 19" rack from Thorlabs[17]. The drawer offers protection against dust or alignment shifting, while having the whole unit self-contained in a central and accessible location. The concern for overheating the drawer, since it is an enclosed space, was alleviated by a high-power 4" fan installed in the back of the drawer, providing adequate air exchange. The rear panel of the drawer also contained BNC power ports through which the fan and the lock circuits were powered, and a standard power jack which was used to provide power to the heat tapes. The front panel of the drawer was also fitted with BNC connectors for signal input and output, fiber optic ports for coupling the leaked light from the tapered amplifier to the inner optical setup, and the associated SPDT switches and turn-potentiometers for controlling the lock circuit. The front panel is shown in a figure 2.15 below.

The lock circuits themselves needed to be shielded from extreme temperature or field fluctuations to avoid drifts or behavioral changes. Therefore, the entire circuit was placed in an aluminum enclosure, and the leads for the switches, BNC signals, and potentiometers were taken through a port in the top of the box to their associated components. This setup is also pictured 2.17 below.

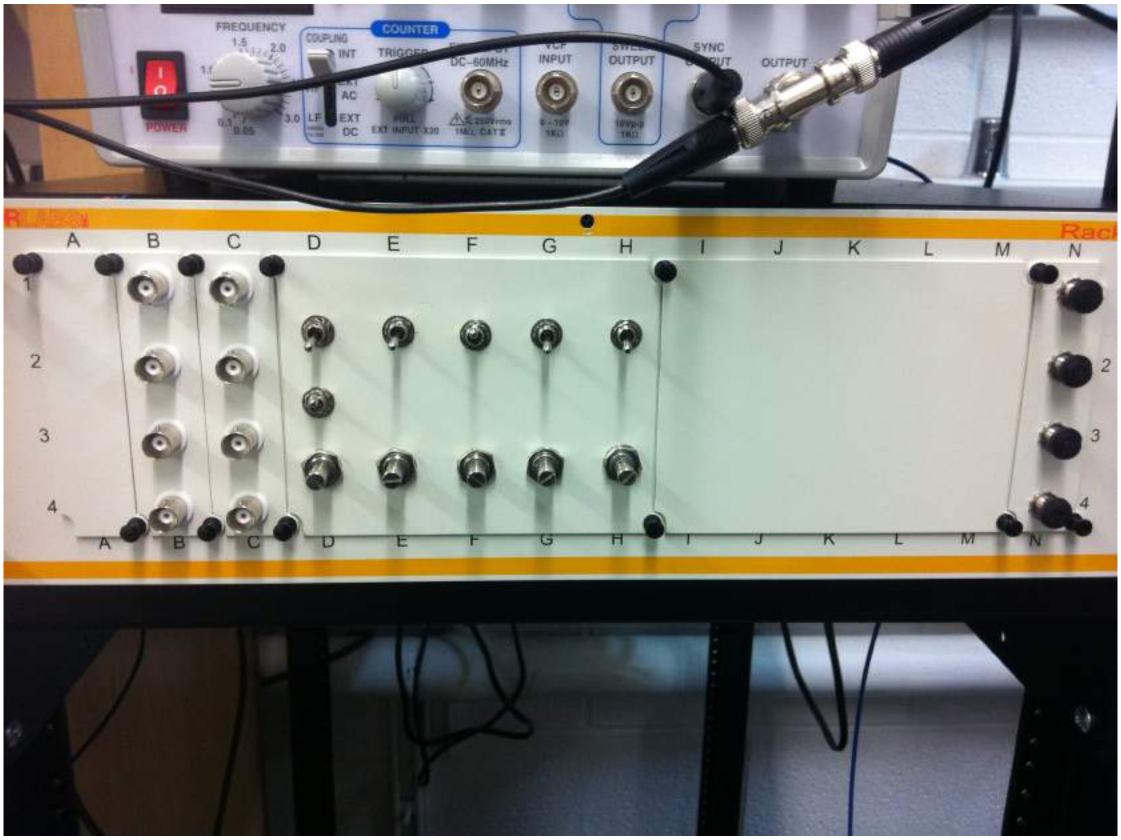


Figure 2.15: Lock Box Front Panel

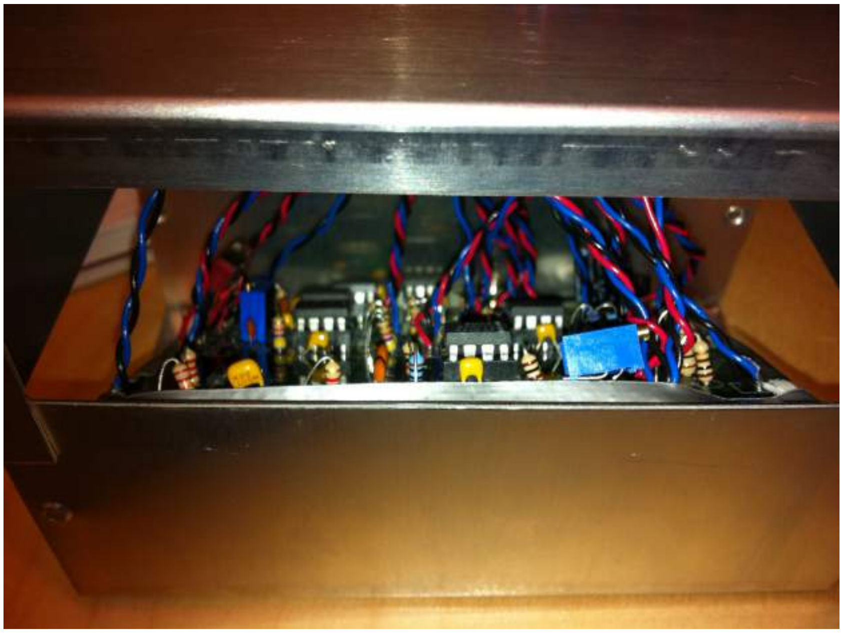


Figure 2.16: Laser Circuit Box, Showing Contents

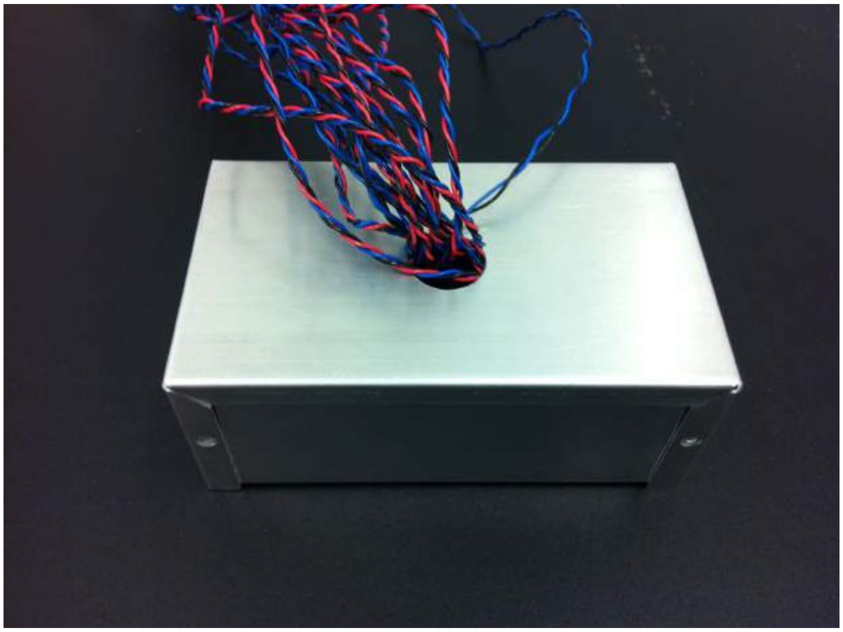


Figure 2.17: Laser Circuit Box, Sealed

2.13 Characterization

2.13.1 First Attempt

The initial trial at seeing a Doppler signal required a few adjustments to the heating and pressures of the heat pipe. Foremost, too much argon prevented the signals from appearing. After pumping out argon to roughly 10 mTorr, a weak signal was apparent. The signal strength could be increased by increasing the temperature of the heat tape. According to the Tory thesis [3], an optimum temperature should have been witnessed due to the fact that with too great an internal temperature, the optical density of lithium would be too great and the transmission of the laser through the heat pipe itself would be diminished despite the increased number of interacting atoms. This trend was not seen; instead, only an exponential growth was demonstrated as per figure 2.18. This fact, combined with a temperature difference between the center of the cross in the interaction region and the nipple containing the lithium itself, necessitated the re-examination of the temperatures along the heat pipe. Moreover, slight depositions of a gray substance began to appear on the viewports. At this point, the heat pipe was completely dismantled and re-evaluated.

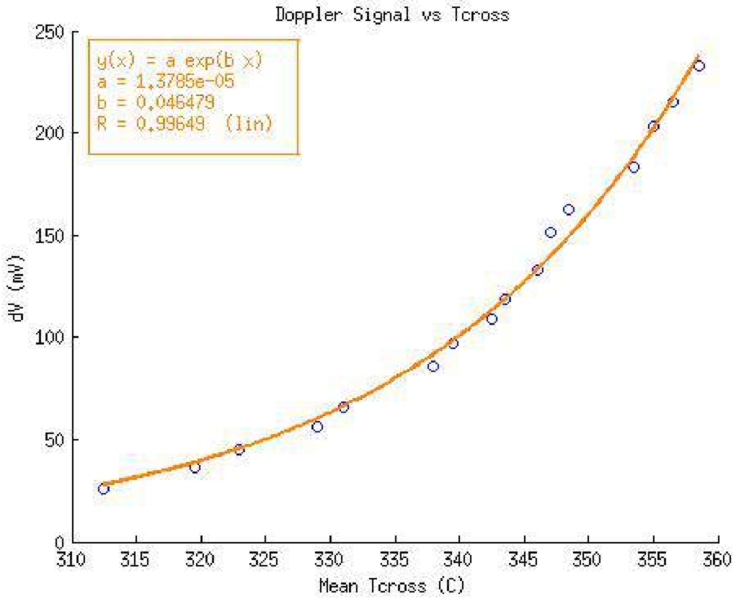


Figure 2.18: Doppler Signals vs. Temp at the Cross

Evaluation

The gray substance was determined to be lithium deposition on the viewports, assumed to have occurred during the argon reduction process, and had started to attack the Kodial glass. Also, the accidental use of a copper gasket instead of a nickel gasket in the joint between the lithium nipple and the cross reacted with the lithium and formed an alloy which blocked the lithium vapor in the nipple from reaching the cross. This provided an explanation for why the nipple needed to be heated to much higher temperatures to witness a signal which would normally correspond to a much more moderate heating. A final aspect which was noted from the first attempt was that the melted lithium appeared to have formed unusual pockets which seemed to suggest that the lithium was not flowing properly and settling into the base of the containment nipple. To correct for these issues, as series of changes were immediately implemented.

1. All nickel gaskets were installed and double-checked for placement.

2. Insulation and heat tape was placed more uniformly on the cross and nipple to reduce their temperature difference.

3. The Kodial glass viewports were exchanged with Sapphire viewports, which are nonreactive to lithium at temperatures through the desired heat pipe regime [10].

4. Heat exchanging fins identical to the brass mounts holding the viewports were machined and clamped just after the boundary of the insulation on the viewport nipples and the valve nipple. This created a steeper temperature gradient and encouraged earlier condensation of the lithium vapor before it could reach the viewports or before the high temperatures could damage the vacuum valve which was only rated to 200 °C.

5. The lithium slices themselves were compressed into smaller chunks and packed firmly into the base of the containment nipple, now filled only to 3/4 of the total available volume to prevent overflow.

The adjusted heat pipe is pictured below 2.19.

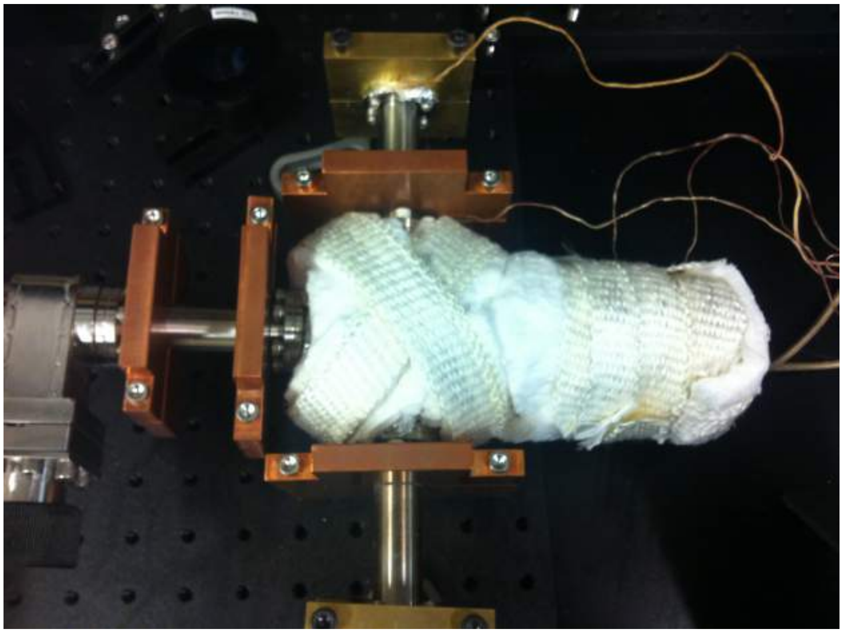


Figure 2.19: Heat Pipe with Adjusted Insulation and Fins

2.13.2 Second Attempt

With the aforementioned adjustments made, spectroscopy was again attempted. After first stabilizing the temperatures to a safe 375 °C-nipple, 340 °C-cross, 100 °C -viewports and valve, and then carefully pumping out argon again to roughly 10-20 mTorr, both the Doppler signals and the saturated absorption signals became strongly evident, as shown below.

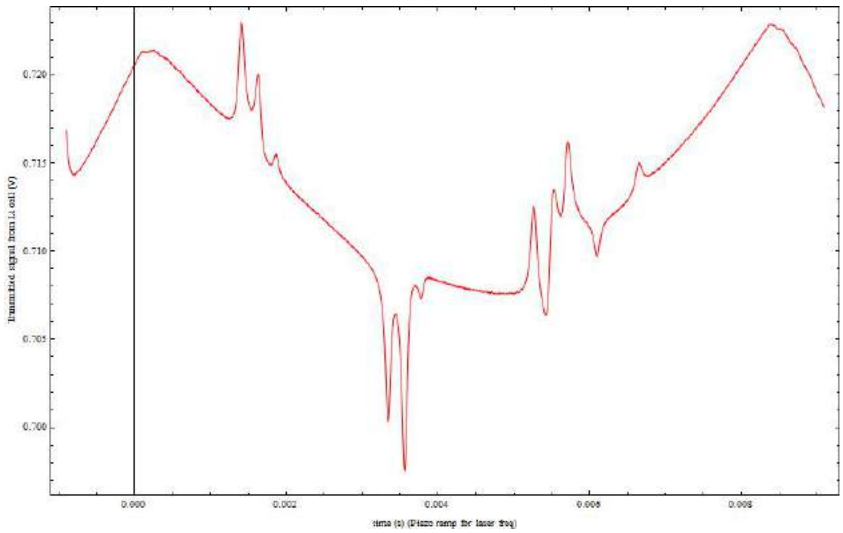


Figure 2.20: Saturated Absorption Signal with a Broad Sweep

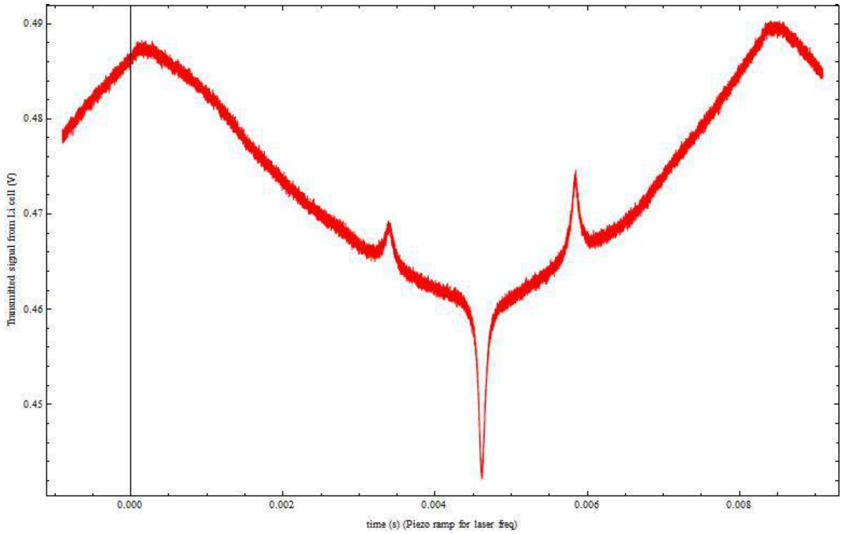


Figure 2.21: The Li7-D2 Lines

2.14 Error Signal Generation and Locking

The error signal which was produced from the frequency modulation was initially unlockable due to a peculiar oscillation pattern. The characteristic signal would appear to beat with the frequency of the sawtooth which was sweeping the master laser set-point. The beat frequencies appeared to match at 60 Hz, indicative of a standard AC interference issue. Tracing the root of the problem, the signal became stable when the variac controlling the heat tape wrapped around the pipe was turned off. This discovery resulted in the conclusion that the error signal was not purely frequency modulated, but was also sensitive to the slight magnetic fields which were being induced in the heat pipe due to the heat tape.

The tape was wound mostly cylindrically around the flanges containing the atoms. When the AC current was passed through tape, a solenoid with an effective AC magnetic field was created in the atom region. The foundations of this error signal were examined in more detail.

2.14.1 Hybrid Error Signal

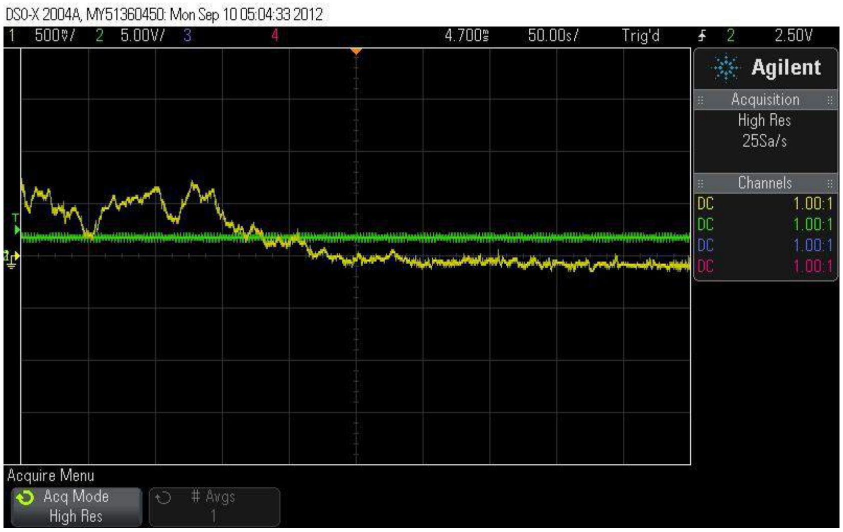
When the EOM was turned off, no error signal was evident. This directly indicated that frequency modulation was a necessary component to the generation of this error signal. Additionally, when the induced magnetic field was turned off briefly and measurements taken while the heat pipe was still at its designed temperature, the size of the pure frequency modulated error signal was at most 5% of the signal otherwise. This indicated that the EOM was providing insufficient sidebands for a large error signal to be generated amidst the background noise. It is hypothesized that the induced oscillations of the EOM were on the order of the signal noise and/or the sidebands were spaced too closely to be distinguished enough. Hence, the error signal witnessed was in fact a hybrid error signal which relied on both frequency modulation and elements of another error signal generating technique, DAVLL.

DAVLL

Dichroic Atomic Vapor Laser Locking is a system which relies on the Zeeman effect to generate an error signal. By using a constant magnetic field source of only a few gauss, such as a solenoid wrapped around the atoms, the hyperfine degeneracy is lifted and the atoms transition levels undergo a splitting. A linearly polarized laser incident on the atoms will be differentially absorbed, with right and left circularly polarized light being absorbed on different transition levels. In effect, for one circular polarization, the Doppler broadened curve is shifted up in frequency while the other polarization is shifted down. The resulting transmission can be separated by a wave plate and polarizing beam splitter. Two matched photodetectors measure the intensity of the two beams and take their difference. The difference signal is a directly generated error signal appropriate for locking. This technique relies on a very stable incoming polarization, or the resulting difference signal will be biased or oscillate.

For comparison to the hybrid system, a completely DAVLL system was also set up on the heat pipe. Two small solenoids were installed on each side of the cross to generate a tunable magnetic field independent of the induced field due to the heat tape. This system demonstrated excellent locking feasibility only after the polarization of the laser was cleaned up dramatically. Large fluctuations in the incoming light caused the error signal to have a large erratic behavior in its offset with additional long term drifts as shown in figure 2.22. By passing the beam through a series of polarizing beam splitters, however, polarization purity was obtained to a well enough order for the system to lock to the D2 line of Li7 for a few hours.

To note, an additional step that needed to be taken before a full DAVLL system could generate a lockable error signal was the use of a DC power supply for the heat tape. The beating effect of the oscillating error signal was present even after the larger solenoid DC fields were used to try and cancel out the AC stray heat tape fields. With the zero crossing always changing in time, the signal was unlockable, as shown in figure 2.23. The only way to rectify the signal was to remove the AC component of the field entirely.



DSO-X 2004A, MY51360450: Mon Sep 10 05:04:33 2012

1 500mV/ 2 5.00V/ 3 4 4.700ms 50.00s/ Trig'd 2 2.50V

Acquisition: High Res 25Sa/s

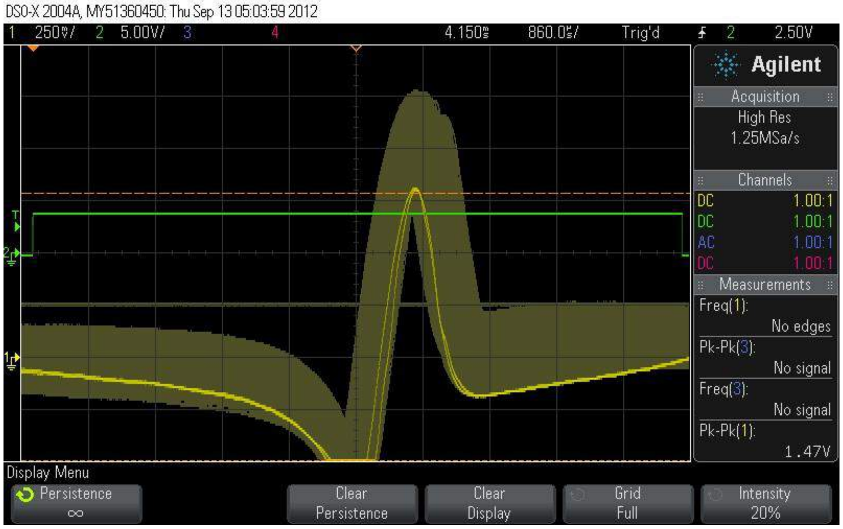
Channels:
DC 1.00:1
DC 1.00:1
DC 1.00:1
DC 1.00:1

Acquire Menu

Acq Mode: High Res

# Avgs: 1

Figure 2.22: Long Term Behavior of the DAVLL Error Signal



DSO-X 2004A, MY51360450: Thu Sep 13 05:03:59 2012

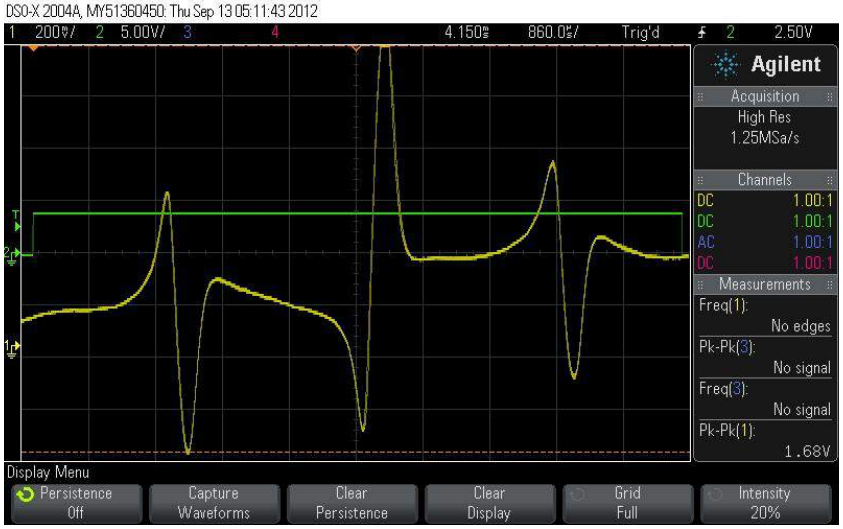
1 250mV/ 2 5.00V/ 3 4 4.150ms 860.0µs/ Trig'd 2 2.50V

Acquisition: High Res 1.25MSa/s

Channels: DC 1.00:1, DC 1.00:1, AC 1.00:1, DC 1.00:1

Measurements:
Freq(1): No edges
Pk-Pk(3): No signal
Freq(3): No signal
Pk-Pk(1): 1.47V

Figure 2.23: Long Term Oscillations Due to the 60 Hz Field



DSO-X 2004A, MY51360450: Thu Sep 13 05:11:43 2012

1 200mV/ 2 5.00V/ 3 4 4.150ms 860.0µs/ Trig'd 2 2.50V

Acquisition: High Res 1.25MSa/s

Channels: DC 1.00:1, DC 1.00:1, AC 1.00:1, DC 1.00:1

Measurements:
Freq(1): No edges
Pk-Pk(3): No signal
Freq(3): No signal
Pk-Pk(1): 1.68V

Figure 2.24: Final Stable DAVLL Error Signal

Hybrid Mechanism

In order for the hybrid system to be considered part DAVLL, the system would need to be implicitly seeing a differential absorption, as only a single photo-detector was used for obtaining this signal. It was confirmed that the atoms were experiencing a Zeeman splitting by using the same solenoids installed for the full DAVLL system.

By tuning the magnetic field strength, the size of the error signal would increase accordingly. However, after roughly 15-20 Gauss, the effect of the splitting seemed to saturate and no appreciable gain or loss was witnessed at higher magnetic fields. This indicates that after this critical field, an effect also seen in DAVLL systems, the limiting factor for a larger error signal would be the effect of the EOM.

Regarding the error signal mechanism itself, it is hypothesized that the EOM was inducing frequency side bands which were differentially absorbed with a greater magnitude due to the hyperfine splitting. For example, consider the absorption peak and splitting shown below in figure 2.25. For the natural $m_F = 0$ state without an applied field, the frequency sidebands are shown as the superimposed peaks. Assume that the differential absorption in its natural state occurred with a magnitude 3δ. After the Zeeman splitting, the population should be split into the $m_F = 0, \pm 1$ states equally, as shown. Define $\delta_{0,\pm 1}$ to be the three new differential magnitudes of the sidebands, with $\delta_0 = \delta$. Due to the direction of the shifting as shown, $\delta_{-1} > \delta_1$. Additionally, due to the nonlinear profile of the original spectrum, the differential strength may also split such that $\delta_{-1} + \delta_1 > 2\delta$. In this sense, the ultimate superimposed signal as seen by the photo-detector could speculatively witness a $\delta_{hybrid} > (\delta_{EOM} + \delta_{DAVLL})$ if the parameters of the DAVLL components and the frequency modulation components are optimized with respect to each other.

An image of the hybrid error signal is shown below in figure 2.26.

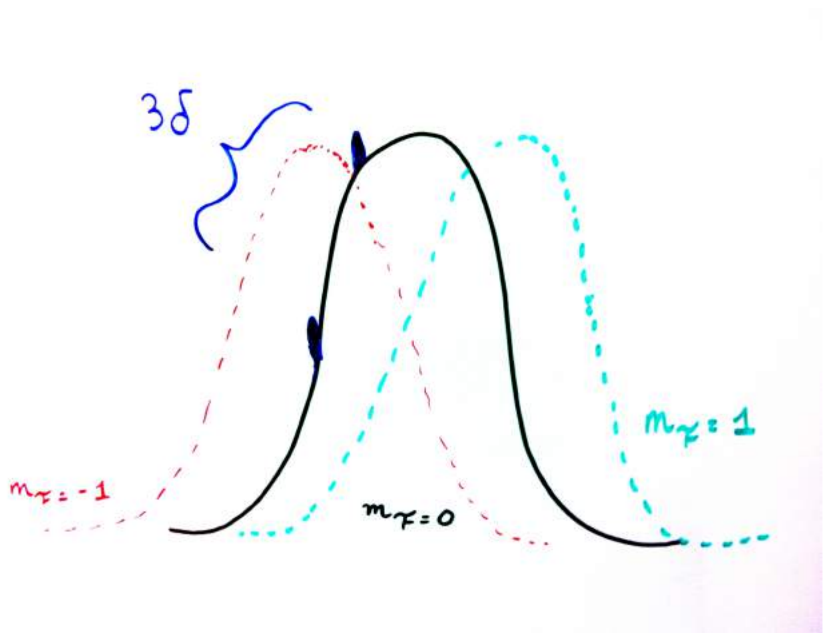
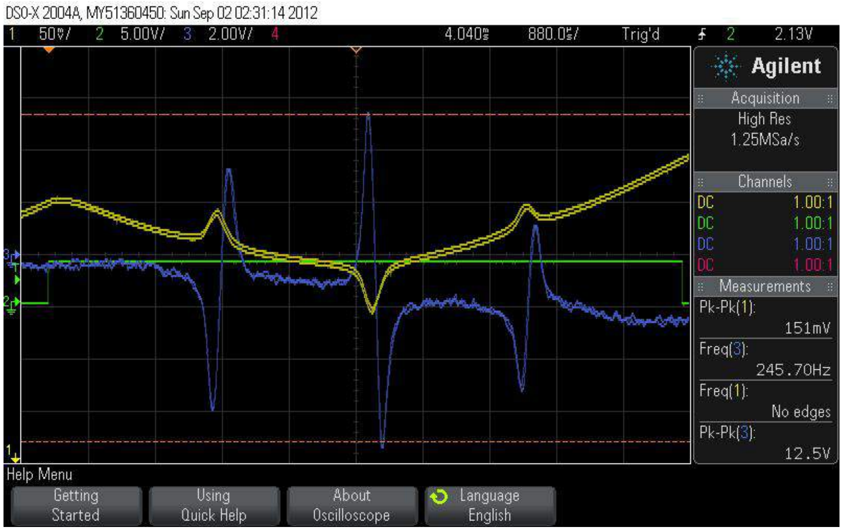


Figure 2.25: Schematic of the Absorption Splitting with Sidebands


DS0-X 2004A, MY51360450: Sun Sep 02 02:31:14 2012

1 50mV/ 2 5.00V/ 3 2.00V/ 4 4.040ms 880.0µs/ Trig'd ⨍ 2 2.13V

Acquisition
High Res
1.25MSa/s

Channels
DC 1.00:1
DC 1.00:1
DC 1.00:1
DC 1.00:1

Measurements
Pk-Pk(1): 151mV
Freq(3): 245.70Hz
Freq(1): No edges
Pk-Pk(3): 12.5V

Figure 2.26: Hybrid Error Signal

Unintended Benefits

Unlike the DAVLL system which required a DC source for the heat tape, the hybrid system was able to generate a lockable signal as long as the superimposed solenoid field was large enough to overtake the time varying component from the heat tape. Also, this system was seemingly polarization fluctuation insensitive (after the polarization axis was roughly aligned with an initial waveplate to maximize the signal) which allows for less concern for the polarization purity of the incoming laser light. The hybrid system also enables more tunability than the DAVLL system, with the error signal dependent on parameters present in both frequency modulation spectroscopy and DAVLL, at the expense of the necessary equipment for driving an EOM. The dual photo-detector system is not needed, however, which reduces the sources for error based on biased splitting and the need for a second beam.

A detailed exploration of this hybrid signal, its true causes and characterization, was not undertaken at this time due to the ultimate goals of the lab. In our experiment, laser locking is merely a tool used in the greater machine responsible for BEC physics. However, this accidental discovery provides a novel and potentially preferable approach to frequency locking a laser system. A more in depth look at the hybrid system may be conducted in the future.

2.14.2 Li6-D2 Lines

Ultimately, the system was designed to lock to the D2 lines of lithium-6 for fermionic experiments. However, the error signal in both methods for locking has a much smaller presence in the Li-6 transitions than the Li-7. This is primarily due to the natural abundance of the isotopes of lithium, as shown in table 1.1, which would naturally dictate a signal strength for Li-6 of only roughly 7% of the Li-7 peak-to-peak. At the time of writing, the fully DAVLL system is being implemented due to a greater understanding of its process when compared with the new hybrid method, and the system is locking to the Li-7 D2 line. Further experiments in frequency modulation are intended to improve the Li-6 lock, with an obvious immediate solution being the replacement of the lithium in the heat pipe with a sample of isotopically pure lithium-6.

CHAPTER 3

THE ZEEMAN SLOWER

3.1 Functionality

The Zeeman Slower is the first cooling stage in a cold atom machine, taking the steady jet of atoms from the oven at speeds exceeding 1000 m/s down to speeds which can be contained in a magneto-optical trap, less than 5 m/s. This process all occurs in the span of .5 ms. The atoms travel down the slower, essentially a long tube, where they are faced with a counter-propagating laser. If the atoms are on resonance with the laser, they will interact and absorb a photon, receiving a momentum impulse of

$$\Delta P = \hbar k \tag{3.1}$$

The atoms will soon spontaneously emit that photon, again receiving a new momentum boost. However, when the atoms absorb the photon, the change in momentum will be anti-parallel to their velocities, while the later spontaneous emission will be in any direction. As a result of this process happening at most roughly every 20 ns, the re-emission does not induce a large momentum change as it is canceled out over the solid angle. The atoms are subsequently slowed against the laser.

Yet, since the atoms will be continually changing velocity, atoms which started on resonance with the laser will soon drift off resonance due to the Doppler effect. Once off resonance, the atoms do not interact with the laser and maintain their longitudinal velocity. To insure that the atoms remain on resonance with the absorption transition, the Zeeman effect is employed. A consequence of the interaction of the magnetic moment of the atoms with an external magnetic field, the transitions undergo a splitting, and shift proportionally to the strength of the external field with a change in the Hamiltonian as given by

$$H_{weak\,zeeman} = \mu_B B m_j g_j \tag{3.2}$$

where

$$\begin{aligned} \mu_B &\equiv \text{Bohr Magneton} \\ B &\equiv \text{Magnitude of the External Magnetic Field} \\ m_j &\equiv \text{Spin Quantum Number} \\ g_j &\equiv \text{Landé g-factor} \end{aligned}$$

By designing a magnetic field profile over the extent of the slower pipe which precisely shifts the atomic transition to account for the Doppler shifting, the atoms can be continually on resonance with the laser.

3.2 Design Requirements

3.2.1 Dimensional Constraints

This particular slower was designed to be mounted on the existing rubidium chamber of the primary experiment which will be converted into a dual species machine. As a result, the slower had to fit into the pre-established dimensional constraints. The slower was to be mounted on the chamber at a 20 ° angle to accommodate the available port, and therefore needed to be both compact and lightweight to avoid a large torque on the chamber. Compared to the rubidium slower which had a length of over 1 meter, the lithium slower would have to accomplish a higher velocity differential roughly in half of that distance, resulting in a highly efficient design.

3.2.2 Velocity Constraints

Axial

Since the temperature of the atoms coming out of the oven is at least 670 °K, by the standard kinetic theory of gases, we can solve that the initial speed of the atoms will be at least

$$v_i = \sqrt{\frac{K_B T_i}{m}} \approx 890 m/s \tag{3.3}$$

However, since the atoms will actually be forced through a nozzle into the slower, the minimum nozzle velocity is given instead by a geometrically corrected expression:

$$v_{nozzle} = \sqrt{\frac{9\pi K_B T_i}{8m}} \approx 1700 m/s \tag{3.4}$$

Since slowing the atoms from this velocity would certainly require a length scale which exceeds the dimensional constraints, the slower was designed to capture atoms at a 1000 m/s threshold, giving a maximum 60% capture rate possible for any given incoming flow. Since the atom numbers of even the reduced capture range would be large enough to saturate the MOT on an acceptable timescale, the loss is not appreciable.

The MOT can only capture atoms moving less than 100 m/s. Therefore, the slower was designed for a 1000 → 100^- velocity drop.

Transverse

Since lithium is 12 times less massive than rubidium, the transverse velocities of the atoms must be taken into account. For rubidium, the slower initial velocity of the atoms insures that any initial transverse components won't typically cause the atom beam to broaden beyond the dense core of the flow, and the random re-emission of individual absorbed photons won't cause any appreciable momentum transfer in the transverse direction due to the timescale of the scattering rate and the significant mass of the atoms. The lithium beam, on the other hand, starts at a high velocity in a radial Gaussian distribution and can quickly bloom out of control by accumulating a net transverse momentum due to its light weight. Monte-Carlo simulations needed to be developed to monitor the blooming of the atom beam.

3.2.3 Magnetic Field Profile

In order to calculate the necessary magnetic field profile, three primary detuning factors for the incident laser beam needed to be accounted for.

1. Any intrinsic laser detuning to achieve a frequency closer to the transition lines. Since our laser has a large mode-hop-free range which can be controlled by both current modulation and voltage control of the piezo crystal, our laser was able to cover the frequency range we needed without considering any additional detuning.
2. The Doppler shift of the atom, given as the inner product with the incident

wave vector, k, and the atom velocity, v:

$$\delta_{doppler} \propto \vec{k} \cdot \vec{v} \tag{3.5}$$

3. The Zeeman shift of the atoms in the static magnetic field. The detuning is simply proportional to the strength of the magnetic field, and is given in the weak field limit by

$$\delta_{zeeman} \propto \frac{\mu_B \cdot B}{\hbar} \tag{3.6}$$

Simple Newtonian kinematics can now derive the ideal magnetic field profile that would appropriately slow the atoms to their intended velocity. The maximum deceleration which the atoms experience is directly given by the change in momentum from the photon interaction.

$$a_{max} = \Gamma \frac{\hbar k}{2m} \tag{3.7}$$

where, recalling the definition of τ as per 1.1,

$$\Gamma \equiv \frac{1}{\tau} \tag{3.8}$$

Using this, the velocity profile as a function of distance along the slower based on ideal deceleration would simply be given by

$$v(z) = \sqrt{v_i^2 - 2a_{max}z} \tag{3.9}$$

Two dimensionless parameters were used to optimize the design: the intensity ratio and the ‘f-number’.

1. The slower was designed with an intended intensity ratio of roughly 5. That is to say, that the incident intensity would be 5 times as great as the saturation

intensity, where

$$i_{sat} = \frac{3hc\Gamma}{2\pi\lambda^3} \tag{3.10}$$

$$i_{rat} \equiv i_{in}/i_{sat} \tag{3.11}$$

$$\approx 5$$

2. The slower was designed with an f-number of roughly .7, a value which essentially tunes the efficiency versus the build practicality. The functional form is given as follows

$$f = \frac{i_{rat}}{1 + i_{rat} + 4(\frac{\delta}{\Gamma})^2} \tag{3.12}$$

$$\approx .7 \tag{3.13}$$

Combining these equations, we can solve for the detuning δ and directly substitute into equation for the magnetic field profile as a function of distance along the slower

$$B(z) = \frac{\hbar}{\mu_B}(\delta + kv(z)) \tag{3.14}$$

$$k = \frac{2\pi}{\lambda} \tag{3.15}$$

A plot of this field is below.

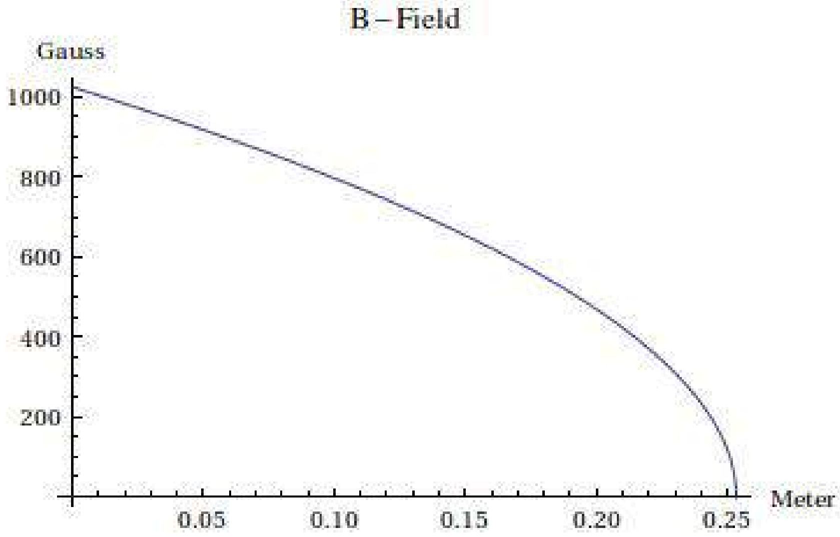


Figure 3.1: Ideal Magnetic Field Strength as a Function of Slower Length

3.2.4 Power Budget

The slower itself would be subject to a variety of power constraints.

1. The currents needed to generate the magnetic fields should not exceed roughly 25 A
2. The total power consumed should be on the order of 150 W
3. The voltage peak per coil should not exceed 5 V
4. The slower should be able to be powered by at most 2 power supplies

3.3 Monte Carlo Simulations

In order to first determine whether this field profile provided adequate slowing without losing too many atoms to the blooming effects along its length as discussed earlier, Monte Carlo simulations were developed in Matlab and can be found in

Appendix B.

First, the slowing effect itself was examined. Evidence of the proper slowing effects were immediately apparent. Atoms which began off resonance but within the slowing capability of the slower eventually fell into resonance with the laser and followed the resonance curve down to sub-capture velocities. Atoms with velocities too great for the slower passed directly through. Evidence of near-resonant slowing was also observed, which confirmed that the model was appropriate. The following figures summarize these first findings.

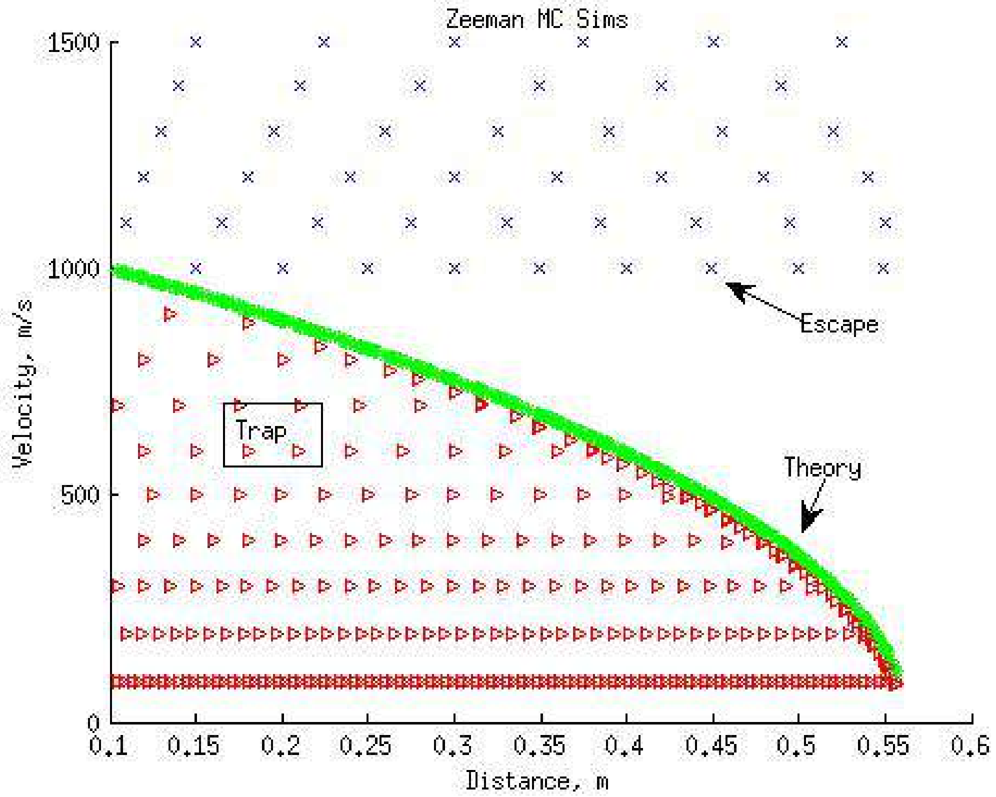


Figure 3.2: Capture Regions for the Ideal Field

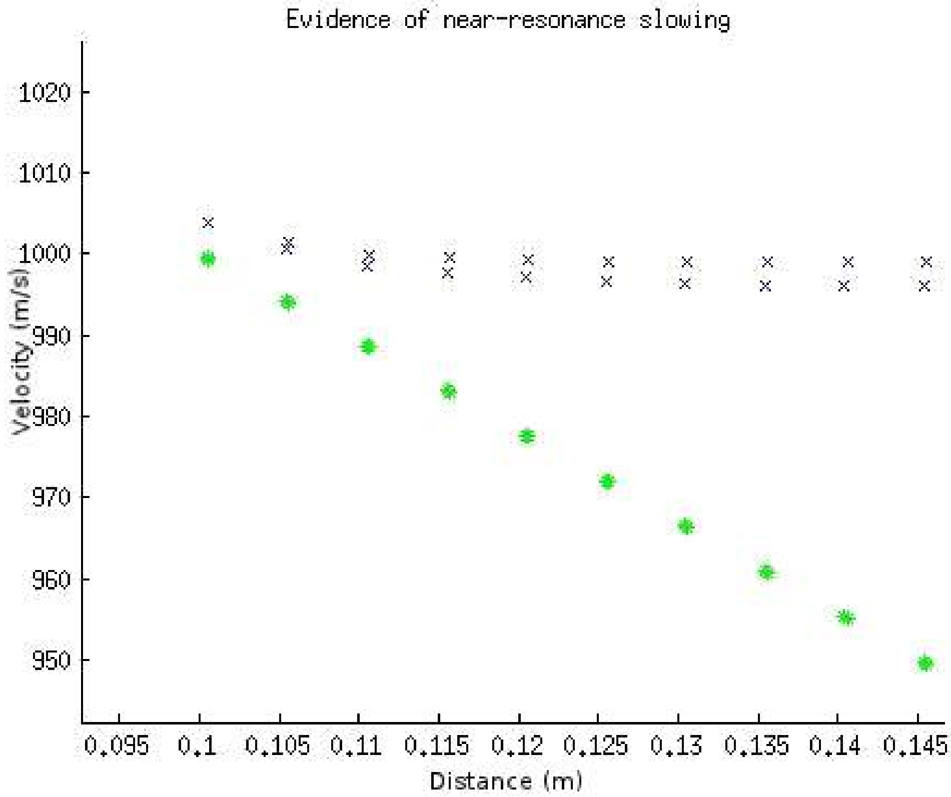


Figure 3.3: Evidence of Near Resonant Atoms Being Slowed

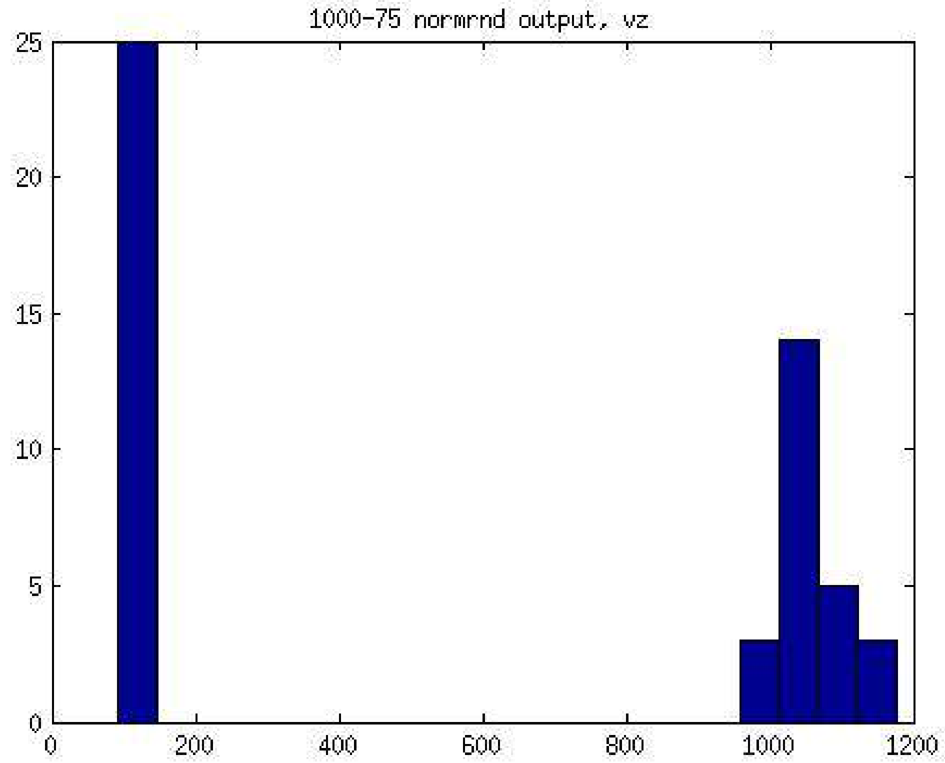
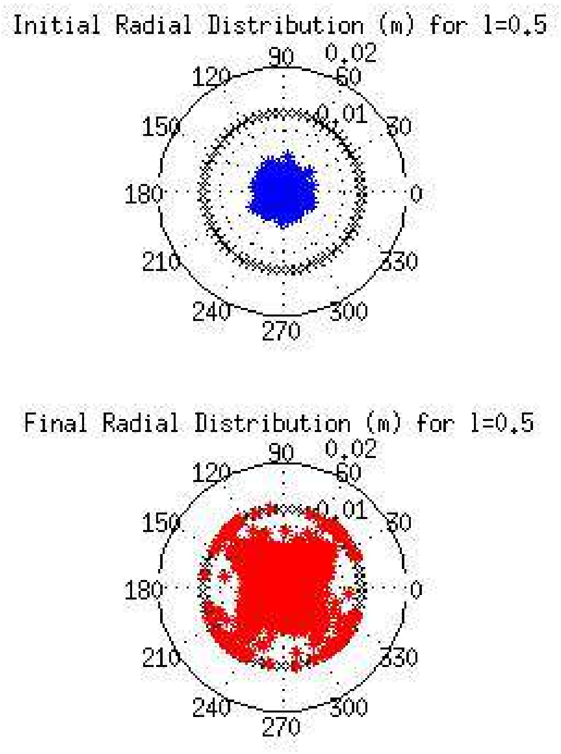


Figure 3.4: Capture Histogram: X Axis = Final Velocity After Slower Given a Gaussian Input Beam; Y Axis = Number of Atoms

However, upon examining the blooming effects, it was noted that for typical initial bloom parameters, by the end of the length of the slower, the atoms were hitting the outer walls as shown in the following figure. As a result, a new form of slower was designed.


Initial Radial Distribution (m) for l=0.5

Final Radial Distribution (m) for l=0.5

Figure 3.5: Beam Spread by the End of the Slower

3.4 The Spin-Flip Slower

In order to reign in the atoms and prevent blooming while still having a long enough slower to maximize the number of trappable atoms and their final slower velocities, a spin-flip slower was designed. In this iteration, the original magnetic field profile is simply shifted down by a few hundred Gauss in order to create a zero-crossing region. Here, the atoms will experience a change in their spin with the change in sign of the Zeeman shift and maintain resonance with the now negative field, continuing to slow until the field is ramped up again and brought to a null shift. This design has 2 primary advantages.

1. The zero crossing allows for the ability to eliminate a region where a magnetic field would have to be generated. Instead, a 6-way cross could be placed in the atom path, with the atoms traveling along the slower direction and a series of crossed laser beams passing through viewports on the other 4 cross ports. These lasers can be tuned to re-compress the atom beam, tightening its waist for a moment before it continues on in the slower toward the MOT, reducing the beam diameter and solving the blooming effect.
2. The overall lowering of the maximum magnitude of the magnetic field strength can result in a power reduction. The field strength is proportional to the currents used to create it, and for the same resistive solenoids generating the field, the power goes as the current squared. Therefore, the power consumption can in theory be reduced quadratically for any linear reduction in field magnitude.

This design does not come without its own difficulties, however. The design of the slower's field profile became more complicated with the addition of the cross due to mechanical space constraints on where solenoids can actually be placed. A design was first theorized and then virtually built and tested using Mathematica.

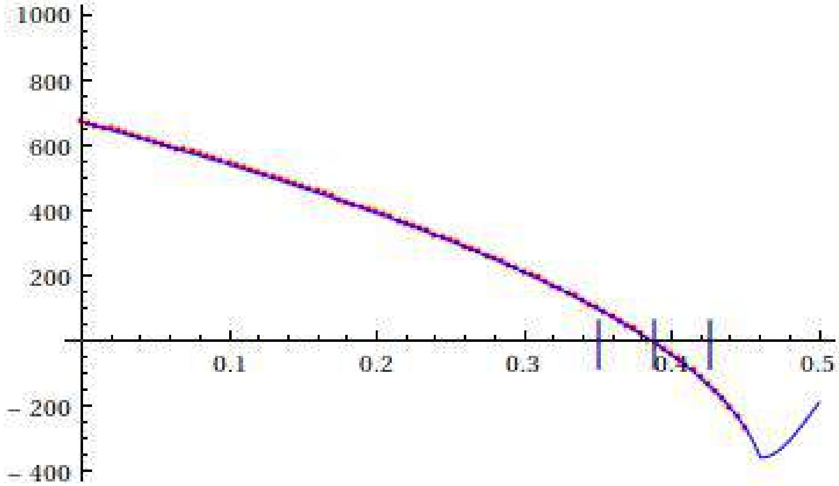


Figure 3.6: New Spin Flip Magnetic Field Profile (in Gauss) as a Function of Distance Along the Slower (meters)

3.5 Virtual Construction

3.5.1 Initial Individual Coil Design

Adapting the notebook created by Mike Gehm and Michael Stenner of Duke Physics attached in the first page of Appendix C used to generate magnetic field profiles based on customizable circular coils, the first implementation of the slower involved a series of individually wound coils of varying width and radius, placed adjacently along the slower length. While this method proved to be effective for continuous profiles where coils could be placed at all coordinates along the slower length, once the design developed into the spin-flip model, the consistency of the generated profile degenerated. The coils themselves would have to have mechanical supports to hold them, which take up space that could be used to gain a more uniform field profile by having a larger coil density in the same region. The ability to actually assemble the main body of the slower with the 6 way cross and a subsequent nipple required that enough space be maintained at the mechanical joints of the slower, again restricting coil placement. To compensate for all of these gaps and boundary constraints, a series of additional coils would have had to be wound and placed over various joints. At best, this method would have required a total of 21 coils, and still left field residuals from the ideal profile of as much as 28 Gauss. For the slower to be effective, the field profile can't vary from the ideal by more than residuals of at most 6 Gauss, the natural linewidth of the transition, before the atoms lose resonance. Hence, a new strategy needed to be implemented.

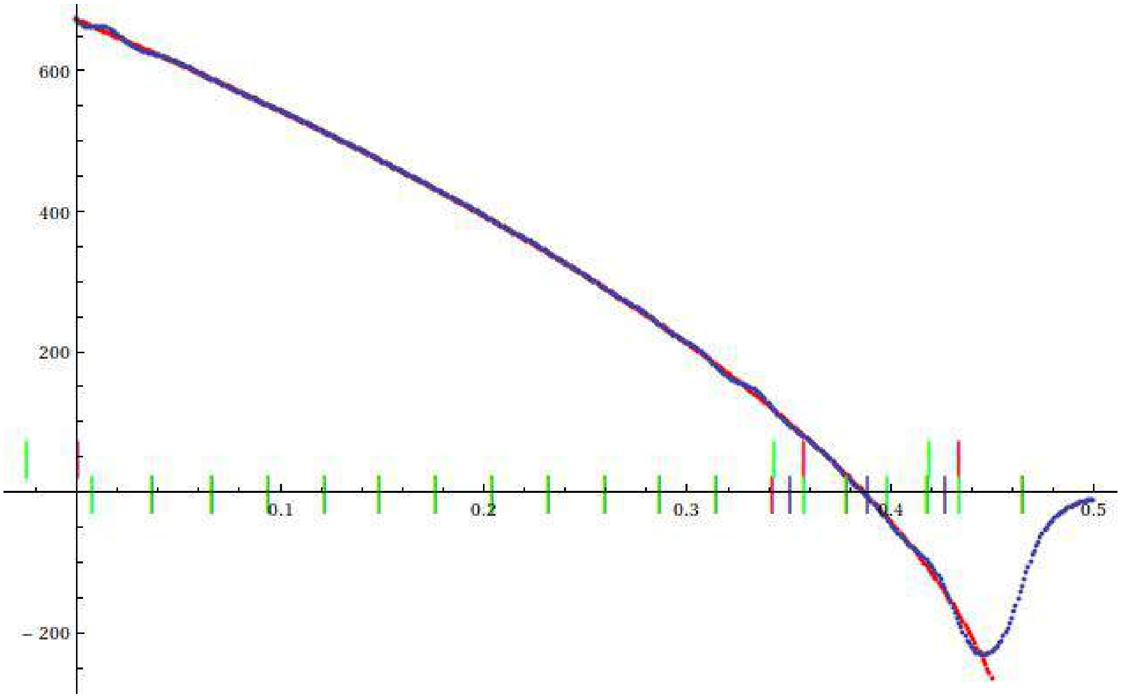


Figure 3.7: Ideal and Virtual Field Profiles Measured in Gauss vs Slower Length
Green and red hashes indicate the start and stop of individual coils, and the blue hashes indicate the boundaries of the cross

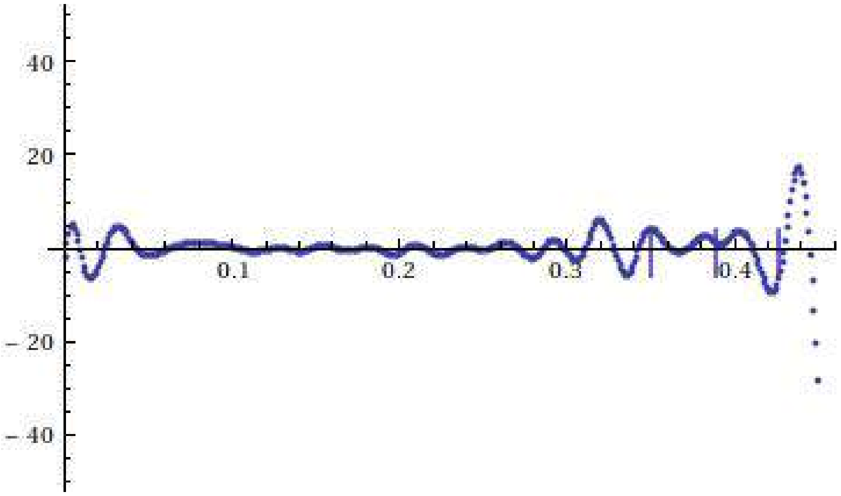


Figure 3.8: Individual Design Residuals of the Field Profile from the Ideal Along the Length of the Slower

3.5.2 Layered Design

Instead of a series of independent coils, the bulk of the slower was wound in layers. This allowed for the elimination of the need to position and hold 21 coils, with the added advantage of a larger coil density in a given region, resulting in a smoother field profile. Varying gauge wire was implemented, with the first 4 layers being water coolable hollow core 7 AWG square Kapton insulated copper wire to attain a certain base of broad field density which could also temperature regulate the slower from the inside out. The subsequent 3 layers were solid core 12 AWG of the same material wire, followed by a final 3 layers of 14 AWG. The layers all began at the end of the slower which would be closest to the oven, and varied in their stopping distance along the slower by adjusting the number of tightly packed turns. Finally, two coils were intended to be wound on the arms of the cross itself, and 4 independent coils would be implemented in key areas to attain the desired profile. The following table 3.1 summarizes the design.

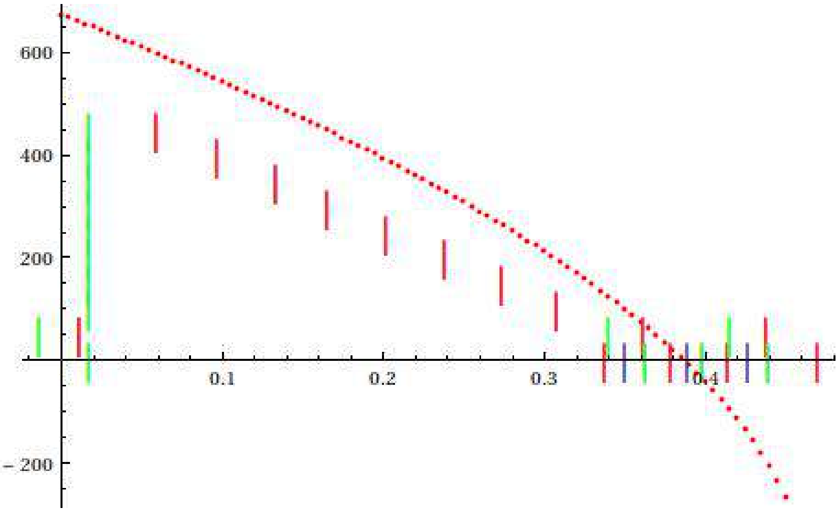


Figure 3.9: Green and red hashes indicate the start and stop of layers or individual coils, and the blue hashes indicate the boundaries of the cross

Layer #	Purpose	Turns	Radial Layers	Square Insulated AWG
1	Main gradient, water-cooled	89	2	7
2	Main gradient, water-cooled	84	1	7
3	Main gradient, water-cooled	74	1	7
4	Main gradient	104	1	12
5	Main gradient	87	1	12
6	Main gradient	70	1	12
7	Main gradient	68	1	14
8	Main gradient	47	1	14
9	Main gradient	25	1	14
NA	Independent ramp up coil	12	10	12
NA	Cross bridging coils	13	5	14
NA	Arms of cross	9	3	14
NA	Independent ramp down coil	18	3	14

Table 3.1: Summary of Zeeman Slower Design

3.5.3 Virtual Data

The variable current parameters were fit to the ideal theoretical profile by a non-linear regression. The results were within the constraints set forth from the initial design stage, and are detailed in the following graphs 3.10, 3.11 and table 3.2.

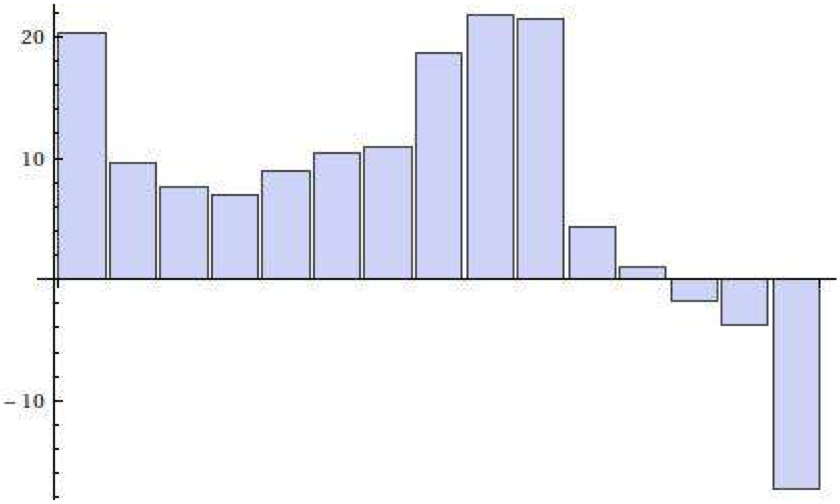


Figure 3.10: Histogram of Virtual Currents Through Each Coil

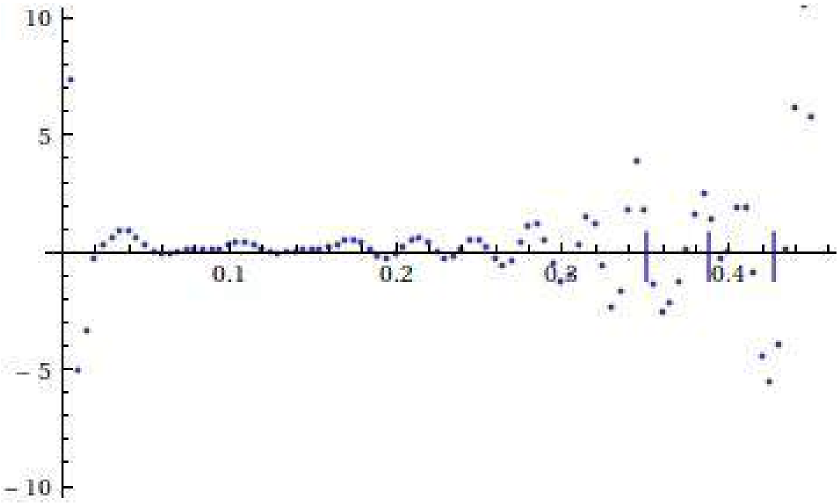


Figure 3.11: Layered Design Residuals of the Field Profile from the Ideal Along the Length of the Slower

Metric	Maximum	Total
Residuals	7.3	4.5 (avg)
Wire Lengths (ft)	67.8	538.1
Resistances (Ohm)	.12	.91
Currents (A)	21.8	164.8
Voltages (V)	2.2	9.8
Power (W)	44.6	136.3

Table 3.2: Summary of Virtual Analysis

The complete Mathematica notebook for the virtual design of the slower can be found in Appendix C.

3.6 Construction

The mathematical models of the individual coils and the layered segment of the slower relied on the tightest possible packing of the helicoids with respect to the gauge thickness. In order to build the solenoids as closely as possible to the ideal models, a variety of construction techniques were implemented.

3.6.1 The Lathe

Foremost, the coils were to be wound on a lathe, mounted as in figure 3.12. Although even the slowest lathe speed would be too fast to maintain the accuracy and packing necessary for the coils, the lathe provided a mechanical means to both control the advancement of a tensioner, pictured in 3.13 positioned between the winding coil and the wire spool, and maintain the tension in the spool by gear resistance.

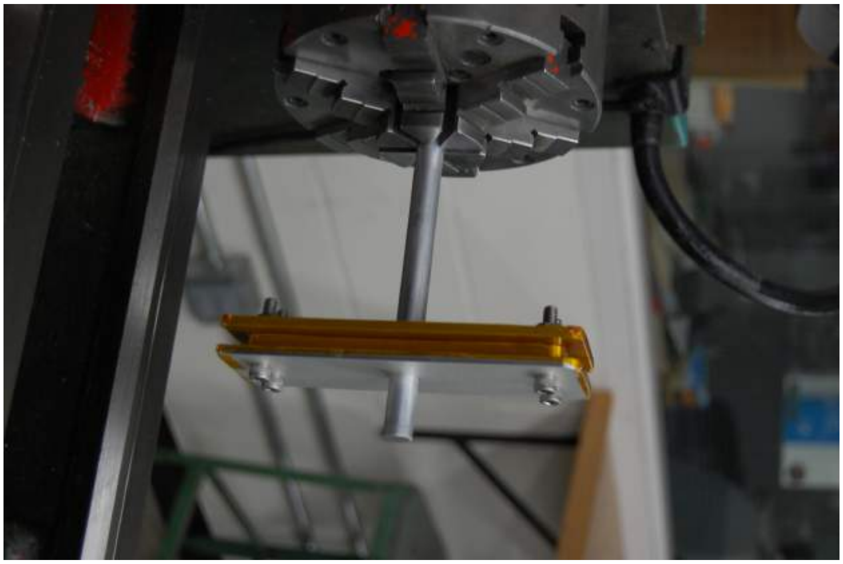


Figure 3.12: A Coil Mount on an Axel, Ready to be Wound Upon

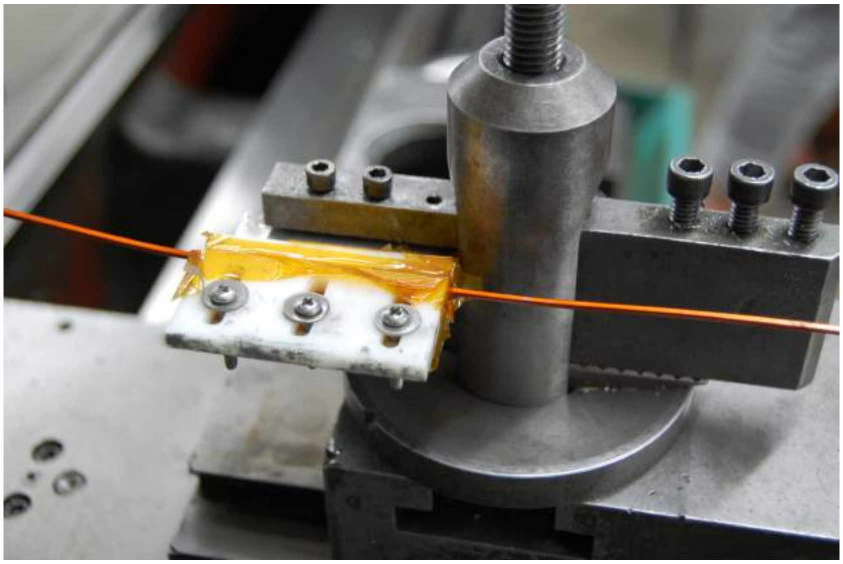


Figure 3.13: Wire through the Tensioner

3.6.2 Binding the Coils

In order to hold the layers together, the Cotronics Duralco 132-IP [4] high temperature quick-dry epoxy was used. This epoxy had a 24 hour curing time, but a tacky working period of roughly 30 minutes which was long enough to cover a layer and wind on top of it before solidifying too much. To minimize unaccounted for thicknesses not present in the mathematical model, the epoxy was applied in extremely thin layers and smoothed into the crevasses of the layers by means of Kapton covered Popsicle sticks or Q-tips. Care was taken to avoid getting epoxy on the end-plates, which would inhibit their later removal. A final feature of the epoxy was its ability to conduct heat extremely effectively. This allowed for increased thermal transfer between layers and enhanced the effectiveness of the water cooled coils in the core of the slower.

3.6.3 Templates

For the individual coils, a template onto which the coil was would was built out of acrylic, Delron, and ABS plastics. Two circular end-plates 3.14 with sufficient diameter to contain the radial measure of a particular coil were fastened via bolts to a Delron cylinder 3.15 of sufficient length and proper radial dimension to contain the axial measure. All contacting surfaces of the template were covered in Kapton tape to provide a smooth surface, facilitate coil removal, and prevent the insulation coating of the wire from being scraped off by any sharp edges and causing a short. A thin slot the width of the AWG in use was cut radially in one of the end-plates to provide a starting point for the winding lead. After winding, the center shaft was removed using a press, and the end-plates were pried off. For the main slower body, the same template process was implemented. However, for the shaft of the template, a hollow brass shaft just large enough to slide over the 1.33" outer diameter of the 35cm UHV rod was directly wound on. Brass was selected as a strong enough metal to withstand crushing or bending and also for its properties to inhibit eddy currents to form in the inner wall of the tube. Eddy currents that would have formed in a steel tube, for example, would alter the field profile and effect the atom beam.

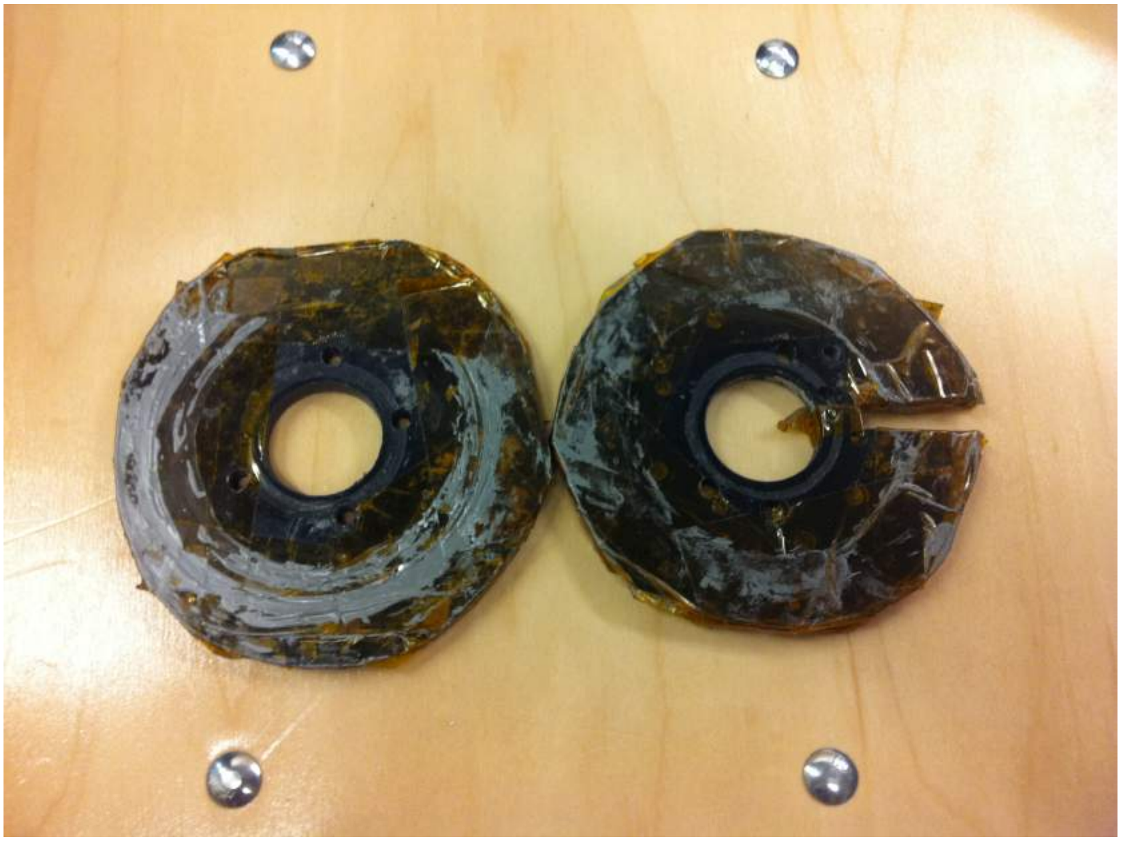


Figure 3.14: Examples of Used End-Plates

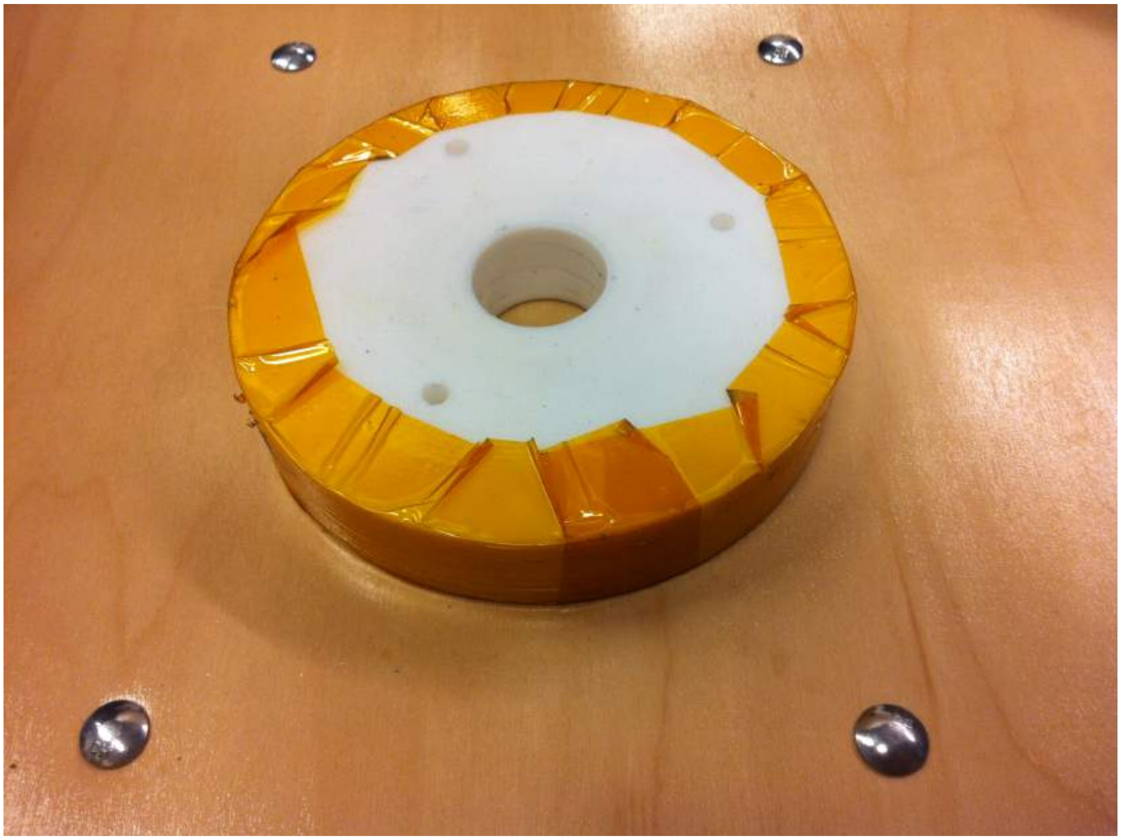


Figure 3.15: An Example of a Large Center Axle used in between Two End-Plates

3.6.4 Set Up

The wire spools were threaded onto a strong solid scrap rod and were thusly free to spin. The rod was strapped onto two inverted stools with the wire spool straddling the gap between them. The height of the spool was such that the angle between the wire spool and the solenoid was minimized. This setup is pictured below in figure 3.16. The wire from the spool was threaded through a tensioner attached to the tool arm of the lathe. Roughly 3 feet of initial lead was reserved for electrical connections and wrapped in tight circles outside of the template to keep it out of the way of the winding process. Finally, the inner wire was bent through the slot cut in the end-plate 3.17, and the winding then began.

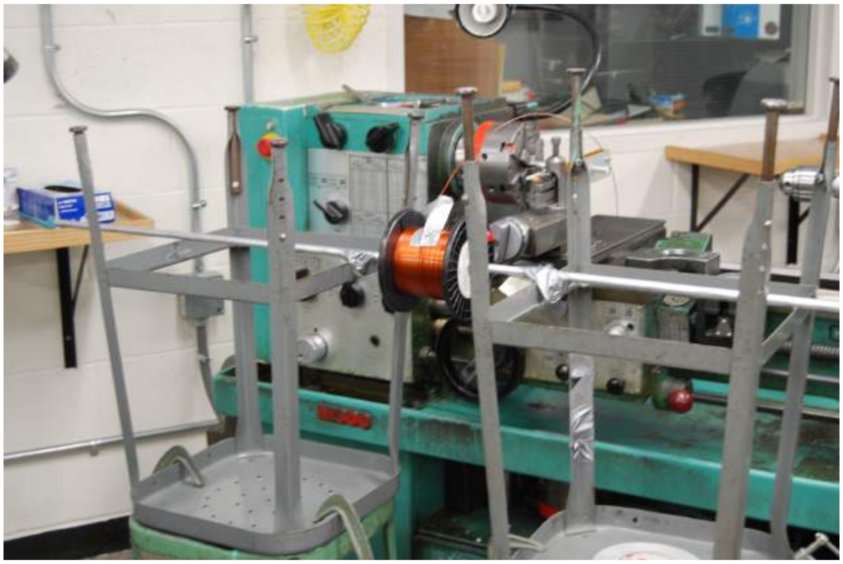


Figure 3.16: Wire Spool Setup

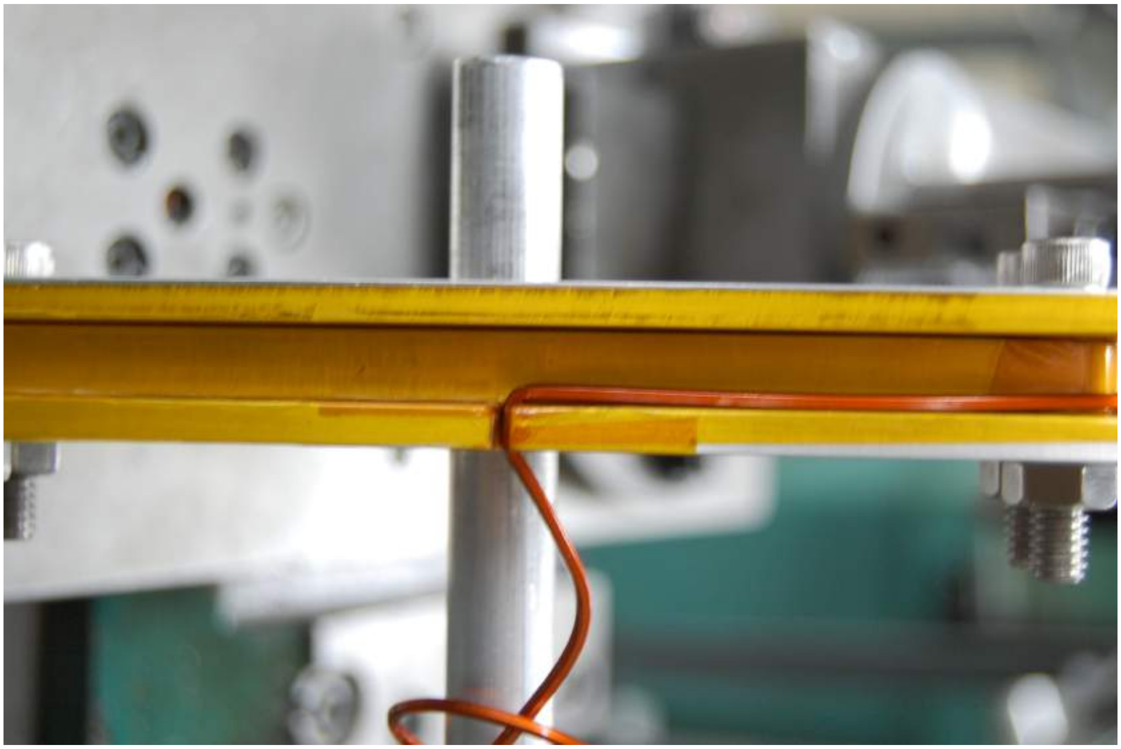


Figure 3.17: The Start of a New Coil, Including the Initial Bend through the End-Plate Slot

3.6.5 Winding

Winding the individual coils 3.18 was an entirely manual process with regard to the feed position of the tensioner. The position would have to be continually adjusted to insure that the square wire was falling snugly and perpendicularly into place, without twisting, which would disturb the packing and prevent subsequent layers from being wound neatly. For the layered segment 3.19, the auto-feed of the lathe was utilized. Calibrated to the wire gauge, the feed position would continually adjust as the center shaft, fed into the lathe chuck, was rotated. This allowed for a smoother winding process with less human error. In either case, the winds would be continually pressed against their neighbors to maintain the packing by means of wooden or Delron tools carved for this purpose. Occasionally, a rubber mallet and blunt Delron tool were used to hammer layers into place. Conversely, if too much space existed for any reason, Teflon spacers were temporarily introduced to fill the void. After each complete layer was wound, tension was secured and a layer of epoxy was applied. The next layer was promptly wound on top of the wet epoxy to secure the layers together. This process would continue until the coil met the designed dimensions, at which point a final layer of epoxy was applied to secure the coil. Clamps were used to hold the solenoid for curing over a few hours, before a final 3 foot end lead was cut from the spool and the coil could be removed from the lathe.

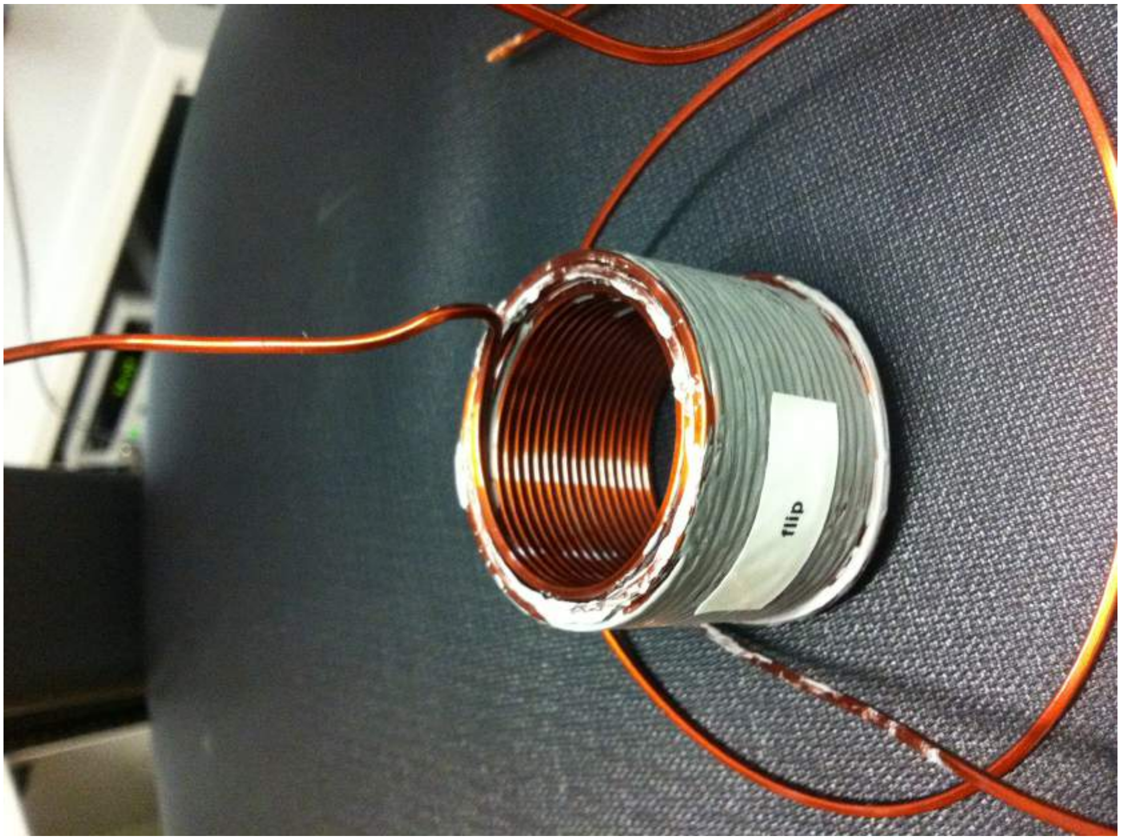


Figure 3.18: One of the Independent Coils

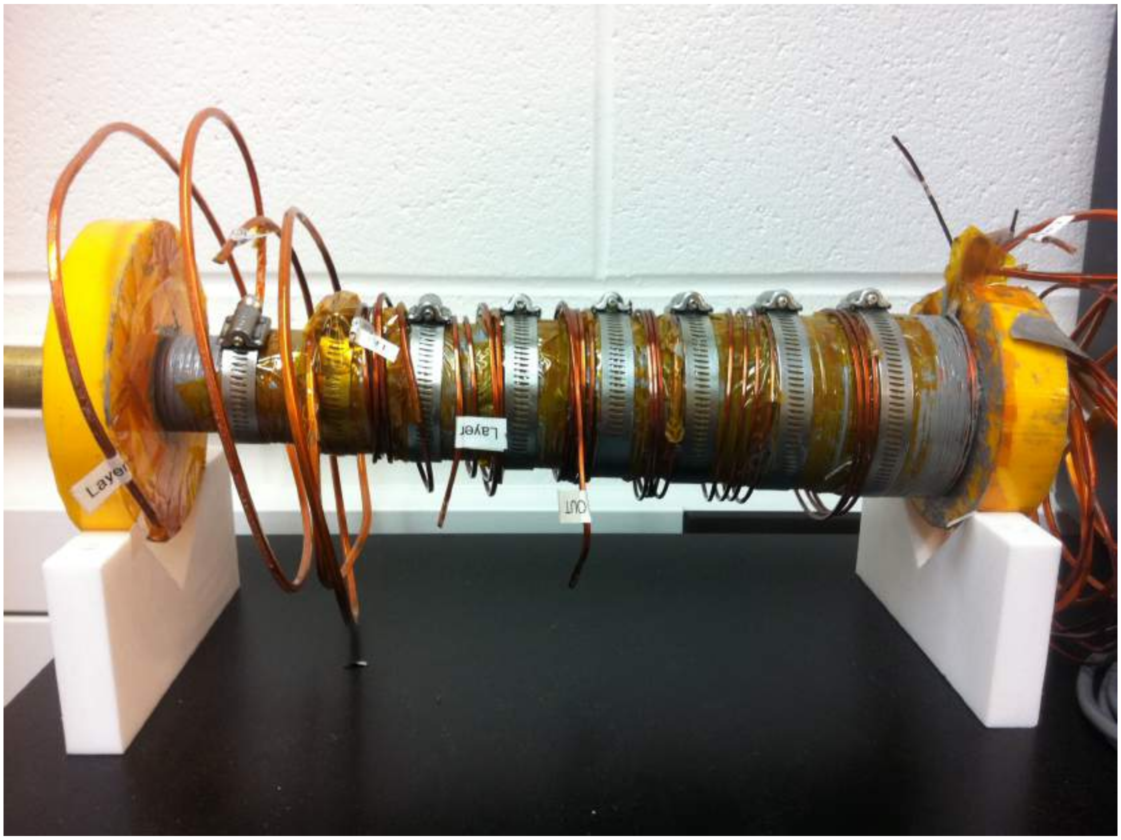


Figure 3.19: The Finished Layered Portion of the Zeeman Slower

3.7 Characterization

3.7.1 Field Measurements

To first characterize the slower, accurate field measurements had to be taken. An axial Hall probe connected to a Gauss meter was the primary instrument for measuring the field up to factory calibrated variations of 1 Gauss. The probe was attached to the end of a measuring tape, which acted both a reference for distances and as a translation stage for the probe. A diagram of this setup is pictured in figure 3.20. The solenoids were measured individually to determine the variations for each coil from the ideal. Each solenoid was firmly mounted on a table, and the tape-probe was secured such that the probe could be continually extended from one end of the solenoid to the other while remaining along the central axis. A current was passed through the solenoid such that the maximum reading of the Gauss meter provided a signal-to-noise ratio of roughly 20. Measurements of the magnetic field were taken at centimeter increments from each end of the coil, starting at a distance which represented a 0 field reading and ending equidistant from the other side. The data was then imported into Mathematica and scaled by the appropriate currents. The alignment of the profiles was undertaken by most closely matching the peak field positions and the half-width field positions, and then was compared to the predicted model.

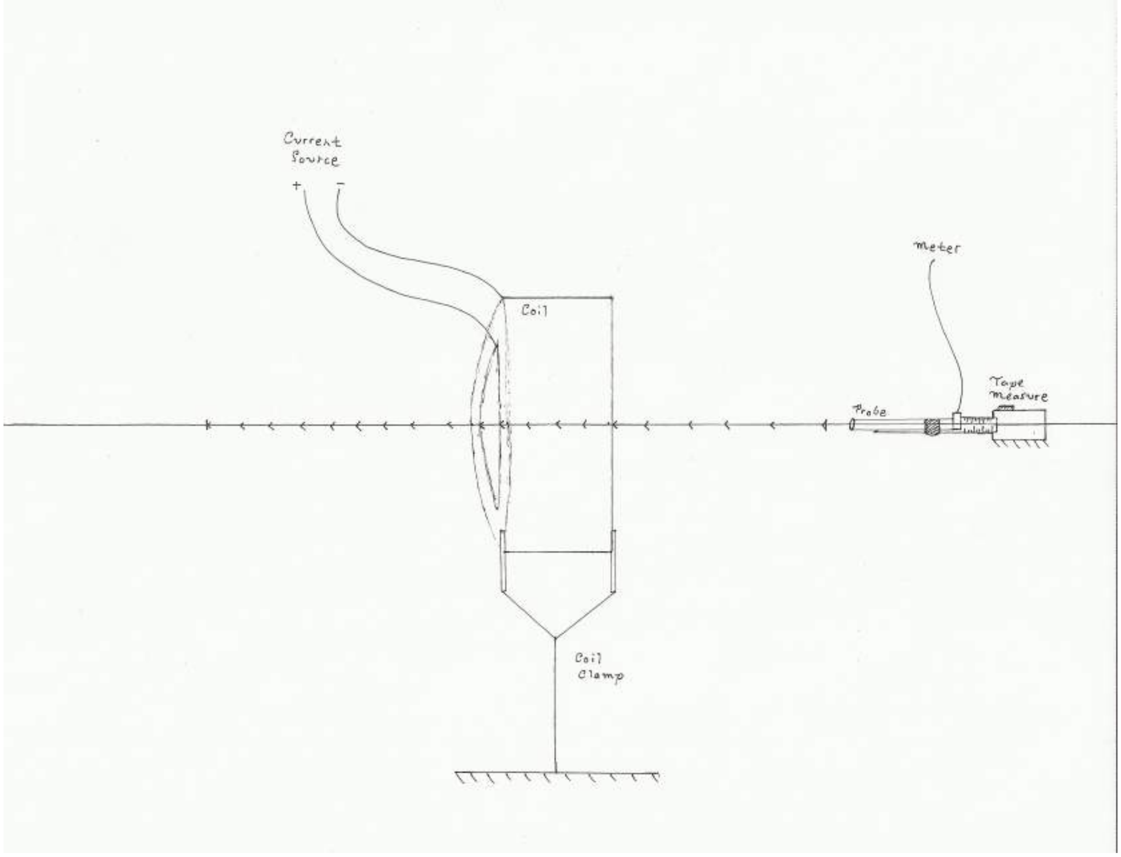


Figure 3.20: A Sketch of the Field Measurement Setup

3.7.2 Probe Drifts and Re-Measurement

Upon examining the data from the characterization, it was confirmed that each layer had the appropriate shape and no coil had any shorts, which would have necessitated the rebuilding of the entire body of the slower. However, upon closer inspection, it was determined that the Hall probe had an offset drift as a function of time. This caused the positioning of the virtual layers to be misdirected, and also resulted in incorrect current adjustments. To account for this, a new set of data was taken which started in the opposite end of the coil range and translated backwards. This set was also in centimeter increments, but offset from the first data by a half centimeter. The composition of these data sets resulted in a time averaged half-centimeter accurate measurement of field strengths as a function of position along the central axis of the solenoids. An example of this data averaging can be seen in the following figure.

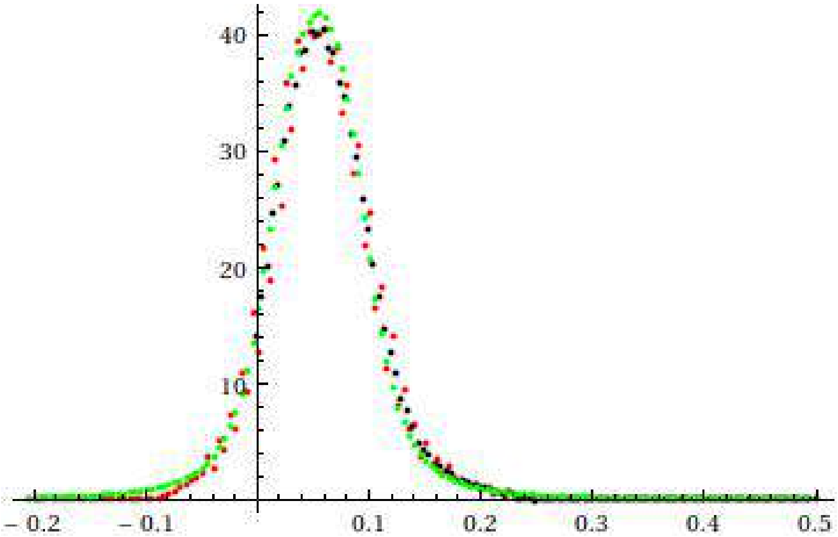


Figure 3.21: Magnetic Field vs Axial Distance
Virtual Field = Green; Forward and Backward Data Sets = Red;
Averaged Set = Black

3.7.3 Current Adjustments

The same non-linear regression approach to minimizing the residuals between the theoretical field profile and the new empirical data set as used in the virtual build was applied to tuning the currents of the constructed solenoids. In order to provide additional degrees of freedom to fine tune the current profile, the combined layered segments of the slower were fit to their own calculated field contributions; the individual coils were fit to their own profiles separately. This first-order fitting provided an approximation to the necessary currents needed for the combined profile. The total field profile as a whole including all wound solenoids was fit cohesively with restricting parameters on current deviations from the former field fit in order to avoid current configurations which were out of the constrained bounds or unphysical and hone in on the smoothest gradient. A comparison of the currents through each layer or coil is detailed in table 3.3 below.

Coil	Virtual Fit	Final Fit	Percent Change
Layer 1	21.5	20.71	-3.7
Layer 2	21.8	17.51	-19.7
Layer 3	18.7	16.58	-11.3
Layer 4	10.85	12.36	13.9
Layer 5	10.45	10.57	1.1
Layer 6	9	9.58	6.4
Layer 7	7	7.1	1.4
Layer 8	7.55	7.87	4.2
Layer 9	9.55	10.43	9.2
Ramp Up	20.35	21.12	3.8
Bridge 1	4.35	5.2	19.5
Cross 1	1	.99	-1
Cross 2	-1.75	-1.72	-1.7
Bridge 2	-3.7	-3.91	19.5
Ramp Down	-17.25	-17.72	2.7

Table 3.3: Current Recipes for each Fitting

With the final currents within 20% deviation from original theory, the power consumption remained within the established constraints.

3.7.4 Residual Analysis

In the region of interest to the atoms, that of the gradient after the ramp-up and prior to the ramp-down back to a null field, the residuals of the final field were closely examined. The field profile was able to meet the expectations set forth from the design with regard to its smoothness and variation from the ideal. The residual data is also tabulated below in 3.4

Max	5.86
Absolute Average	1.82
Standard Deviation $\Delta\sigma$	1.23
% of Field Beyond $2\Delta\sigma$	4.6

Table 3.4: Final Field Residual Data (in Gauss) for Constructed and Analytically Optimized Zeeman Slower

CHAPTER 4

THE OVEN

4.1 Functionality

The lithium oven acts as a constant source of lithium atoms for feeding the Zeeman slower. Essentially a specialized version of the containment nipple section of the heat pipe, the oven would house enough lithium to provide the machine with years of continuous operation, generating and channeling lithium vapor to be passed through the slower.

4.2 Design Requirements

Much like the heat pipe, the oven would need to withstand and tolerate various thermal, mechanical, and chemical stresses and constraints. Foremost, the lithium would again have to reach temperatures on the order of 400 °C while minimizing heat transfer to the environment. The system would need to be entirely corrosion resistant as well. Finally, the oven would have to be a compact and lightweight body to interface cleanly with the existing machine. The necessary components would have to include a containment module for the lithium, a heating element, a channel for directing the flow and a means to measure the internal temperature.

4.3 Design

The initial inspiration for the design came from a 2012 publication on a compact oven used for laser cooling experiments with strontium [14]. Using a model of that design as a starting point, improvements in heat transfer and lithium specific alterations were implemented.

The oven would be a nested system, all contained in an outer 2 3/4" stainless steel UHV nipple. As the lithium would be molten at the desired temperature, the outer nipple would be remain vertical with respect to ground to prevent any outflow of lithium, and the end of the nipple would attach to the machine via a Milter right angle elbow. Starting from the innermost part, the lithium would be housed in a boron-nitride crucible. The crucible volume was designed such that 6 grams of lithium could be housed, providing years of operation before replacement. Boron-nitride is an excellent thermal and electrical insulator, and as it is chemically inert even up to high temperatures, it is an ideal material to both house the corrosive lithium and act as the principle heating element. On one end, the tube-like crucible would make an abrupt right angle along with the outer elbow to direct the atoms toward the Zeeman slower. The length of the crucible would be threaded to accommodate the winding of tantalum wire. Tantalum wire is often used as a resistive heating element in vacuum systems, and by winding it within the threads on the crucible, a large surface contact is maintained while retaining electrical isolation. The base of the crucible has four posts for mounting, and a through hole which holds a 2-bore alumina cylinder. Alumina is another electrical and thermal insulator, and here would act as a via for the leads of the tantalum wire to be fed and secured separately. The posts in the base of the crucible are alumina as well, and connect to an aluminum disk. Aluminum has a low emissivity and will act as a radiation block, minimizing heat loss directly behind the crucible. The aluminum disk would be housed in a Kimbal Physics Groove Grabber [11] cylindrical mount, a device which interfaces with standard UHV equipment to provide unique mounting and instrumentation solutions. Finally, the outermost flange which the outer nipple attaches to would be a dual function thermocouple and electrical feed-through flange. This type of flange would provide both the electrical connections to which the tantalum wire would be attached, and a thermocouple system which would be measuring the temperature at the base end of the crucible directly, giving the most relevant data for the oven temperature where the molten lithium would be held. The inner surfaces of the outer nipple would be coated in a paint similar to the Cerablak [2] HTP high temperature paint. This paint is specifically designed for UHV use at high temperatures, and has an emissivity > .9, making it an efficient method of trapping the radiated heat from the crucible inside the oven. The outer surface would be similarly coated with a material similar to the Cerablak UTF to maintain the low emissivity of the stainless steel nipple. The figures below illustrate the oven design.

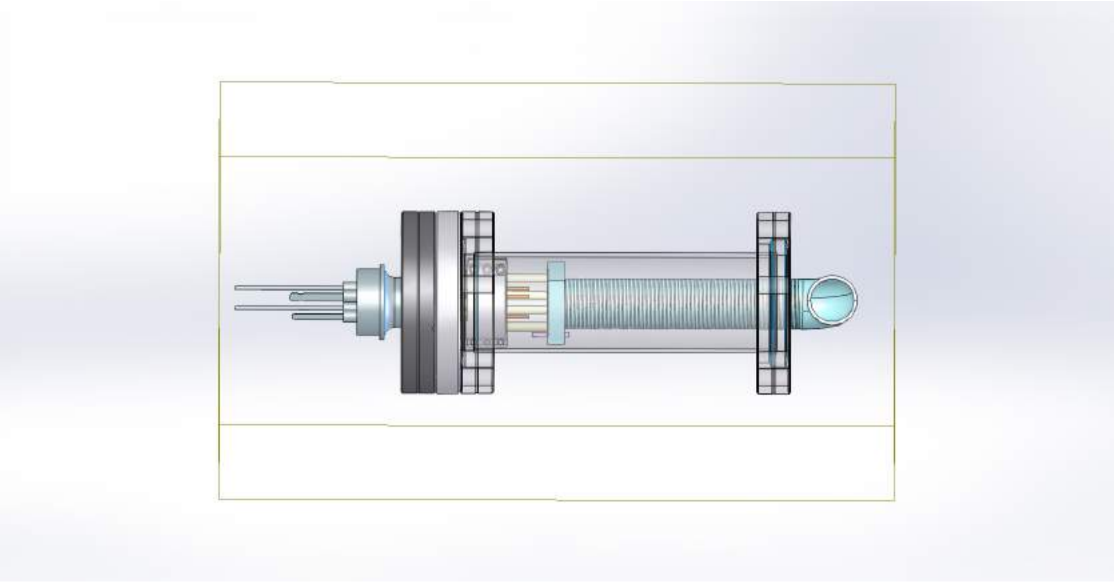


Figure 4.1: Lithium Oven, Side View

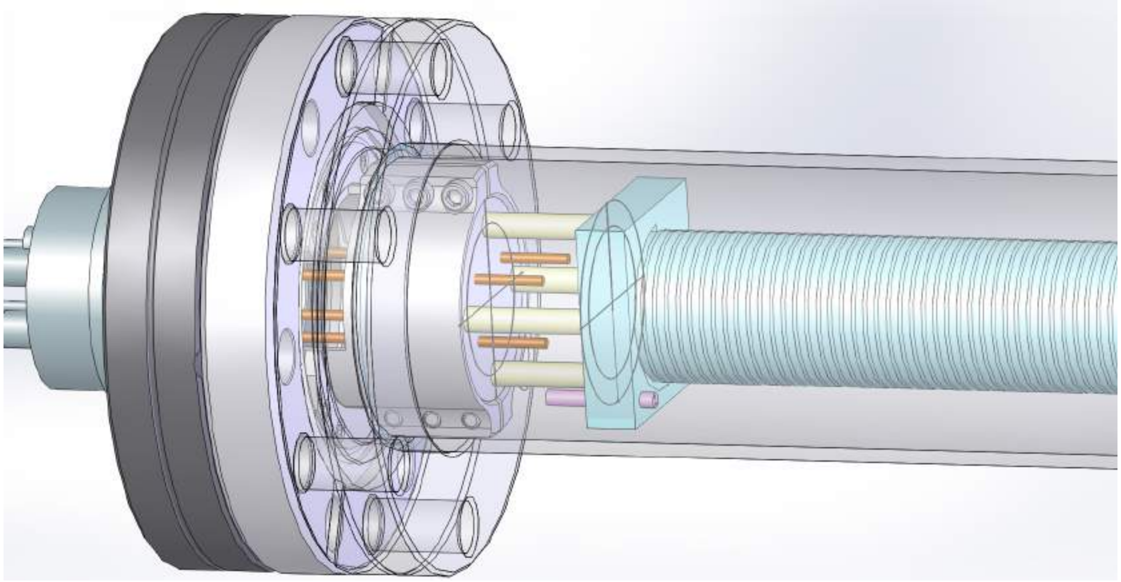
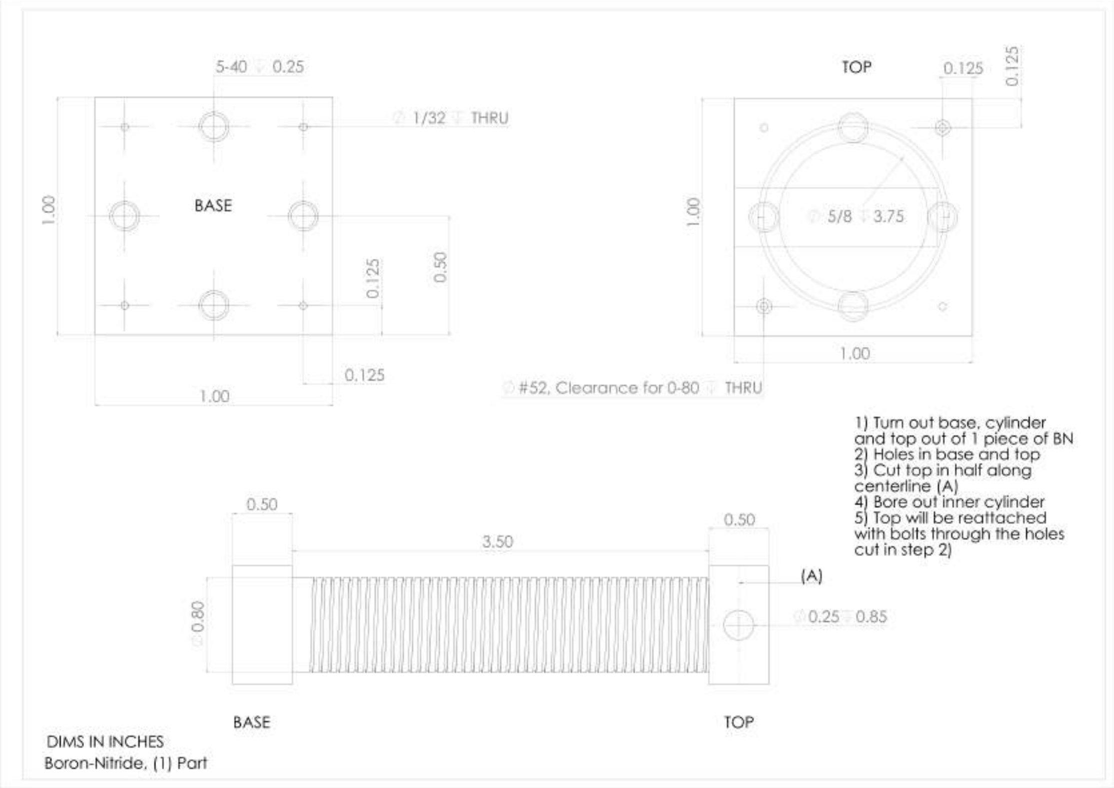


Figure 4.2: Lithium Oven, Close Up on the Mounts and Feed-Through

4.4 Crucible Redesign

In order for the boron-nitride crucible to be easily machined, a minor redesign was undertaken. In addition to removing the bend in place of a two-part system with orthogonal holes to direct the lithium atoms, additional holes were to be drilled in the base to accommodate two sets of tantalum wires, allowing for independent control of the base and crucible output temperatures. The alumina tube was also replaced by these holes for ease of construction, simplicity of the system, and limiting the variety of ceramic materials. The redesign of the crucible is pictured in the drawing below.



1) Turn out base, cylinder and top out of 1 piece of BN
2) Holes in base and top
3) Cut top in half along centerline (A)
4) Bore out inner cylinder
5) Top will be reattached with bolts through the holes cut in step 2)

DIMS IN INCHES
Boron-Nitride, (1) Part

Figure 4.3: Redesigned Drawing of the Crucible

APPENDIX A

HEAT PIPE INSULATION ANALYSIS

```
Needs["PlotLegends`"]
```

Author: John Lombard

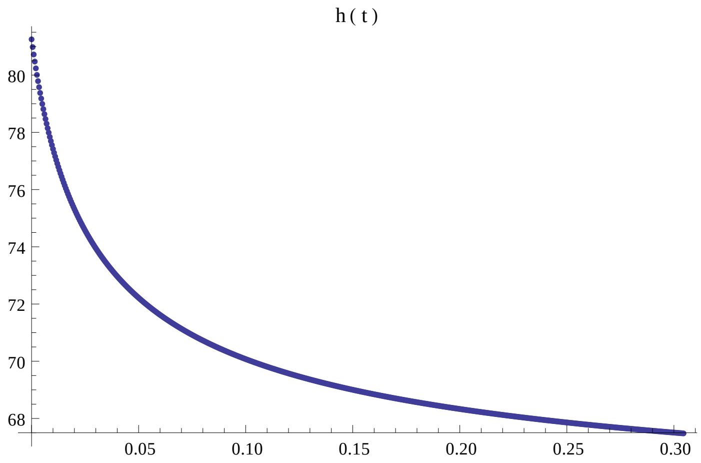
Insulation Analysis for the Heat Pipe using Empirically Correlated Free Convection Flow Over a Steady State Thermal Boundary Layer Grown over a Cylindrical Geometry

- **SI Units used, but temperature in degrees Celsius;**
  Plots are as a function of insulation thickness
  All measures are per unit length of cylinder (hence no dimensional/geometric input other than radius)

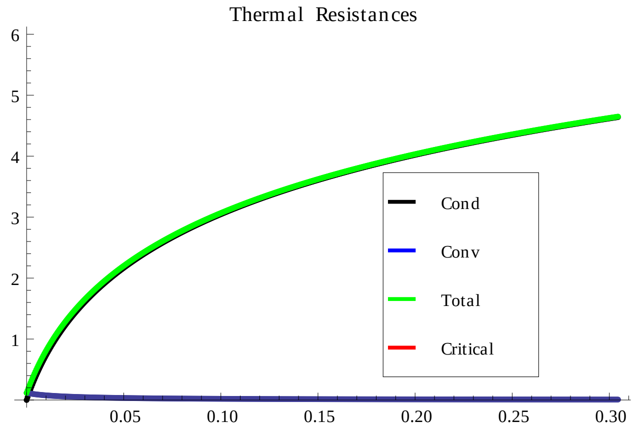
```
Clear[RCond, RConv, RTot, RaD, NuD, h]
steps = .0005; (* Plot refinement *)
rI = .0169; (* Radius of Nipple *)
k = .101; (* Conductivity of Insulation *)
tMax = .3047; (* Maximum feasible thickness for insulation, 1 foot *)
Ti = 400; (* Boundary temperature imposed on bake *)
Tinf = 23; (* Room Temp *)

(* Compute average convection coefficient, h,
for cylindrical free confection of air at the mean temperature *)

Tm = (Ti + Tinf) / 2;
α = 10.3 * 10 ^ -6; (* diffusion coeff at Tm *)
ν = 7.59 * 10 ^ -6;(* kinematic viscosity at Tm *)
g = 9.8; (* acceleration due to gravity *);
β = 1 / Tm; (* volumetrical thermal expansion coefficient *)
Pr = .737; (* Prandtl number at Tm *)
RaD[t_] := g * β * (Ti - Tinf) * (2 * (rI + t)) ^ 3 / ν / α; (*Rayleigh coeff *)
NuD[t_] := (.6 + .387 * RaD[t] ^ (1 / 6) / (1 + (.559 / Pr) ^ (9 / 16)) ^ (8 / 27)) ^ 2;
(* Nusselt Number *)

(* Empirical Nusselt good for RaD less than 10^12,
NB check rI and tMax limits to be sure that this condition holds *)
Print["Within Empirical Nusselt Bounds?"]
RaD[rI] < 10 ^ 12 && RaD[tMax] < 10 ^ 12

h[t_] := NuD[t] * k / (2 * (rI + t));
hData = Table[{t, h[t]}, {t, 0, tMax, steps}];
hPlot = ListPlot[hData, PlotRange → All, PlotLabel → "h(t)"]

(* Calculate Thermal Resistances, Conductive, Convective, and Total *)

RCond[t_] := Log[(t + rI) / rI] / 2 / Pi / k;
RConv[t_] := 1 / 2 / Pi / (t + rI) / h[t];
RTot[t_] := RCond[t] + RConv[t];
RCondData = Table[{t, RCond[t]}, {t, 0, tMax, steps}];
RCondPlot = ListPlot[RCondData, PlotStyle → Black];
```

```
RConvData = Table[{t, RConv[t]}, {t, 0, tMax, steps}];
RConvPlot = ListPlot[RConvData];
RTotData = Table[{t, RTot[t]}, {t, 0, tMax, steps}];
RTotPlot = ListPlot[RTotData, PlotStyle → Green];


(*Extrema = Sign[t/.Solve[D[RTot[t],{t,2}]==0,t]] *)
(* tells extrema is a minimum therefore the resistance has a minimum thickness
 needed for retarding heat transfer, instead of an ideal maximum for efficiency *)

(* Critical Minimum Thickness for positive benifit of insulation*)

TCr = t /. Flatten[NSolve[D[RTot[t], t] == 0, t]];
Print["Critical Insulation Bound:"]
If[TrueQ[TCr == t], Print["No Minimum Insulation Bound"], Print[TCr]];
TCrPlot = ListLinePlot[{{(TCr), 0}, {(TCr), 20}}, PlotStyle → {Dashed, Red}];

ComboPlot = Show[RCondPlot, RConvPlot, RTotPlot, TCrPlot,
   PlotRange → {{0, tMax}, {0, 6}}, PlotLabel → "Thermal Resistances"];
ShowLegend[ComboPlot, {{Graphics[{#1, Thick, Line[{{0, 0}, {1, 0}}]}], #2} & @@@
   Transpose[{{Black, Blue, Green, Red}, {Cond, Conv, Total, Critical}}],
  LegendPosition → {.1, -0.48}, LegendSpacing → 0, LegendShadow → None, LegendSize → 0.6}]

(* Convective and Conductive crossover point *)
CondConvTradeSolve = FindRoot[RConv[t] == RCond[t], {t, 0}];
Print["Conductive and Convective Crossover Point"]
CondConvTradePt = {t, RCond[t]} /. CondConvTradeSolve

(* Calculate temporal heat flux and Temperature at the outer wall of the insulation *)

QpT[t_] = (Ti - Tinf) / RTot[t];
QpTData = Table[{t, QpT[t]}, {t, 0, tMax, steps}];
QpTPlot = ListPlot[QpTData, PlotRange → All, PlotLabel → "Heat Flux per Unit Length"]
Twall[t_] = Ti - QpT[t] * RCond[t];
TwallData = Table[{t, Twall[t]}, {t, 0, tMax, steps}];
TwallPlot =
 ListPlot[TwallData, PlotRange → All, PlotLabel → "Temperature at the Insulation Boundary"]

(* Calculate Efficiency of heat retardation with respect to the
 resistive capabilities of the maximal feasible thickness of insulation *)

MaxR = Max[Flatten[RTotData][[2 ;; ;; 2]]];
EfficiencyData = Table[{t, RTot[t] / MaxR}, {t, 0, tMax, steps}];
EfficiencyPlot = ListPlot[EfficiencyData, PlotRange → All,
  PlotLabel → "Efficiency wrt Maximum Feasible Insulation"]

(*Calculate estimated voltage output required from variac *)

R = 10; (* Heat tape resistance per unit length *)
V[t_] = Sqrt[R * QpT[t]]; (* Voltage *)
VData = Table[{t, V[t]}, {t, 0, tMax, steps}];
VPlot = ListPlot[VData, PlotRange → All, PlotLabel → "Variac Voltage"]
```

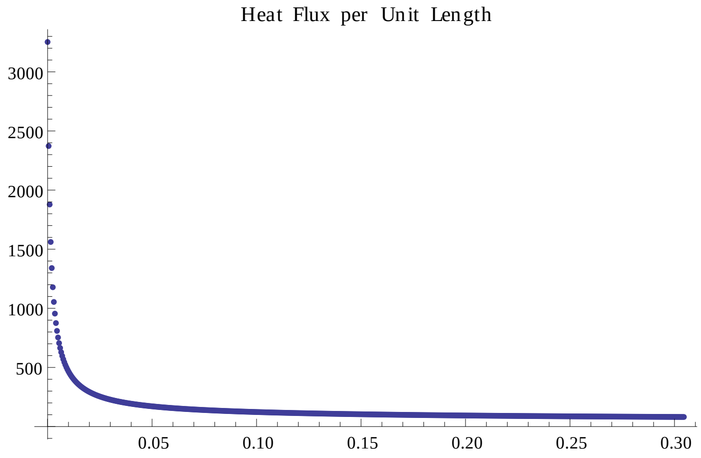
```
Within Empirical Nusselt Bounds?

True
```

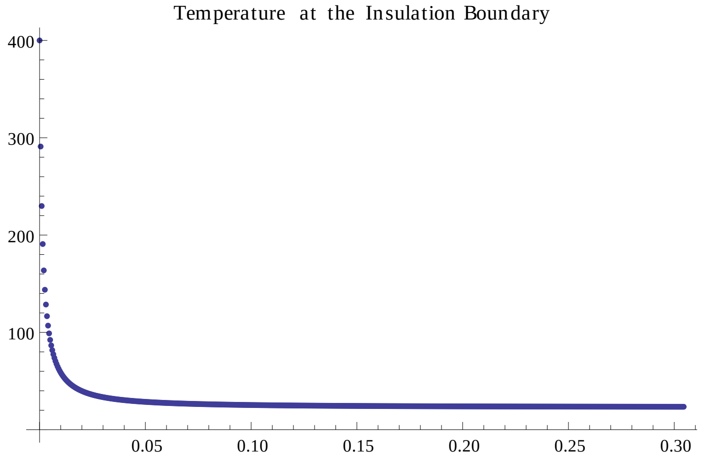


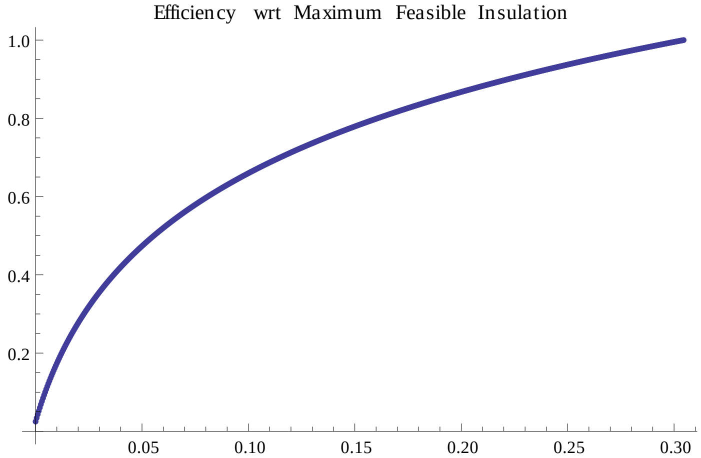
```
Critical Insulation Bound:

No Minimum Insulation Bound
```

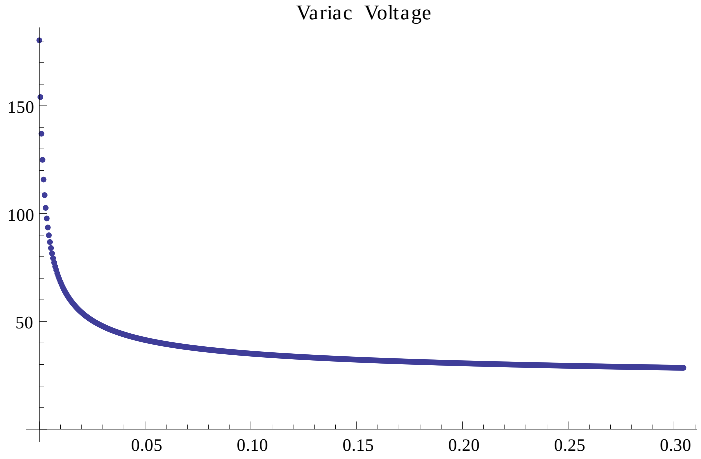


```
Conductive and Convective Crossover Point

{0.00121051, 0.109011}
```









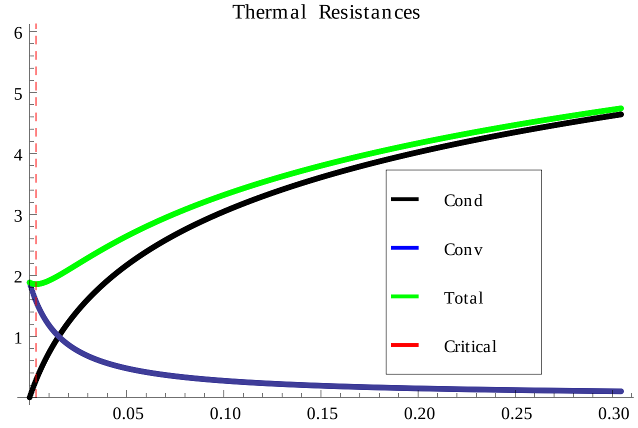
Insulation Analysis for the Heat Pipe using a Minimal Stagnent Air Convection Coefficient

- **SI Units used, but temperature in degrees Celsius;**
  Plots are as a function of insulation thickness
  All measures are per unit length of cylinder (hence no dimensional/geometric input other than radius)

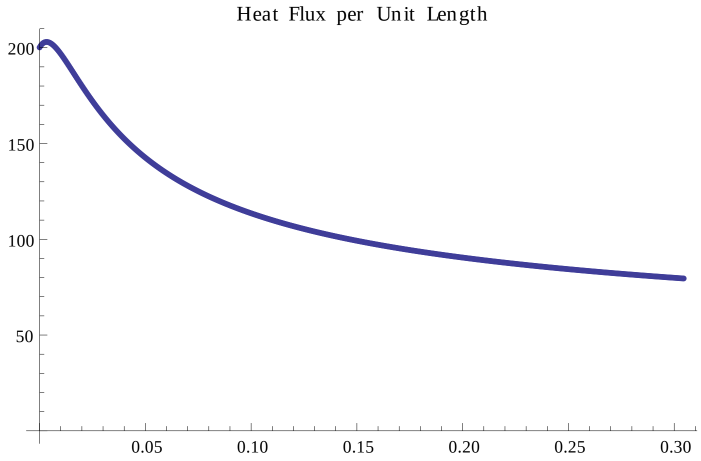
```
Clear[RCond, RConv, RTot, RaD, NuD, h]
```

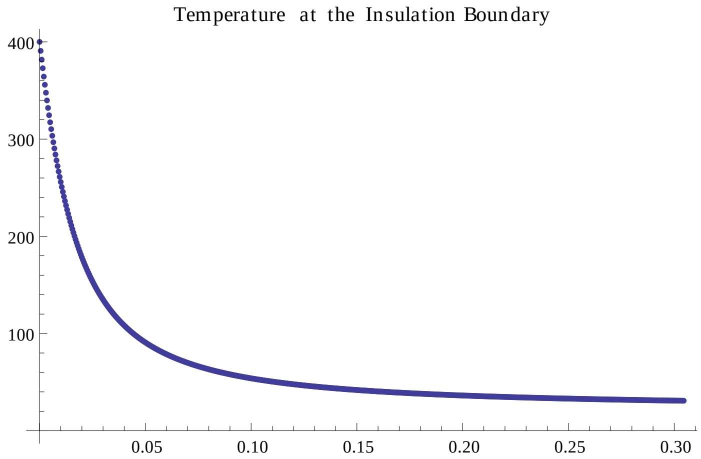
```
steps = .0005; (* Plot refinement *)
rI = .0169; (* Radius of Nipple *)
k = .101; (* Conductivity of Insulation *)
tMax = .3047; (* Maximum feasible thickness for insulation, 1 foot *)
Ti = 400; (* Boundary temperature imposed on bake *)
Tinf = 23; (* Room Temp *)


h = 5; (* Convection coefficient for free, but mainly stagnent air in a room,
no geometric considerations, plumage, boundary layer considerations, etc *)
(* Calculate Thermal Resistances, Conductive, Convective, and Total *)

RCond[t_] := Log[(t + rI) / rI] / 2 / Pi / k;
RConv[t_] := 1 / 2 / Pi / (t + rI) / h;
RTot[t_] := RCond[t] + RConv[t];
RCondData = Table[{t, RCond[t]}, {t, 0, tMax, steps}];
RCondPlot = ListPlot[RCondData, PlotStyle → Black];
RConvData = Table[{t, RConv[t]}, {t, 0, tMax, steps}];
RConvPlot = ListPlot[RConvData];
RTotData = Table[{t, RTot[t]}, {t, 0, tMax, steps}];
RTotPlot = ListPlot[RTotData, PlotStyle → Green];


(*Extrema = Sign[t/.Solve[D[RTot[t],{t,2}]==0,t]] *)
(* tells extrema is a minimum therefore the resistance has a minimum thickness
 needed for retarding heat transfer, instead of an ideal maximum for efficiency *)

(* Critical Minimum Thickness for positive benifit of insulation*)

TCr = t /. Flatten[NSolve[D[RTot[t], t] == 0, t]];
Print["Critical Insulation Bound:"]
If[TrueQ[TCr == t], Print["No Minimum Insulation Bound"], Print[TCr]];
TCrPlot = ListLinePlot[{{(TCr), 0}, {(TCr), 20}}, PlotStyle → {Dashed, Red}];

ComboPlot = Show[RCondPlot, RConvPlot, RTotPlot, TCrPlot,
   PlotRange → {{0, tMax}, {0, 6}}, PlotLabel → "Thermal Resistances"];
ShowLegend[ComboPlot, {{Graphics[{#1, Thick, Line[{{0, 0}, {1, 0}}]}], #2} & @@@
   Transpose[{{Black, Blue, Green, Red}, {Cond, Conv, Total, Critical}}],
  LegendPosition → {.1, -0.48}, LegendSpacing → 0, LegendShadow → None, LegendSize → 0.6}]

(* Convective and Conductive crossover point *)
CondConvTradeSolve = FindRoot[RConv[t] == RCond[t], {t, 0}];
Print["Conductive and Convective Crossover Point"]
CondConvTradePt = {t, RCond[t]} /. CondConvTradeSolve

(* Calculate temporal heat flux and Temperature at the outer wall of the insulation *)

QpT[t_] = (Ti - Tinf) / RTot[t];
QpTData = Table[{t, QpT[t]}, {t, 0, tMax, steps}];
QpTPlot = ListPlot[QpTData, PlotRange → All, PlotLabel → "Heat Flux per Unit Length"]
Twall[t_] = Ti - QpT[t] * RCond[t];
TwallData = Table[{t, Twall[t]}, {t, 0, tMax, steps}];
TwallPlot =
 ListPlot[TwallData, PlotRange → All, PlotLabel → "Temperature at the Insulation Boundary"]

(* Calculate Efficiency of heat retardation with respect to the
 resistive capabilities of the maximal feasible thickness of insulation *)

MaxR = Max[Flatten[RTotData][[2 ;; ;; 2]]];
EfficiencyData = Table[{t, RTot[t] / MaxR}, {t, 0, tMax, steps}];
EfficiencyPlot = ListPlot[EfficiencyData, PlotRange → All,
  PlotLabel → "Efficiency wrt Maximum Feasible Insulation"]

(*Calculate estimated voltage output required from variac *)

R = 10; (* Heat tape resistance per unit length, approx *)
V[t_] = Sqrt[R * QpT[t]]; (* Voltage *)
VData = Table[{t, V[t]}, {t, 0, tMax, steps}];
VPlot = ListPlot[VData, PlotRange → All, PlotLabel → "Variac Voltage"]
```

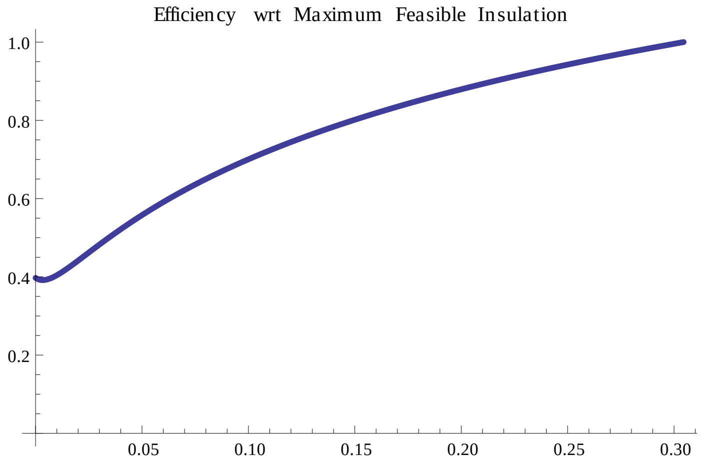
```
Critical Insulation Bound:

0.0033
```

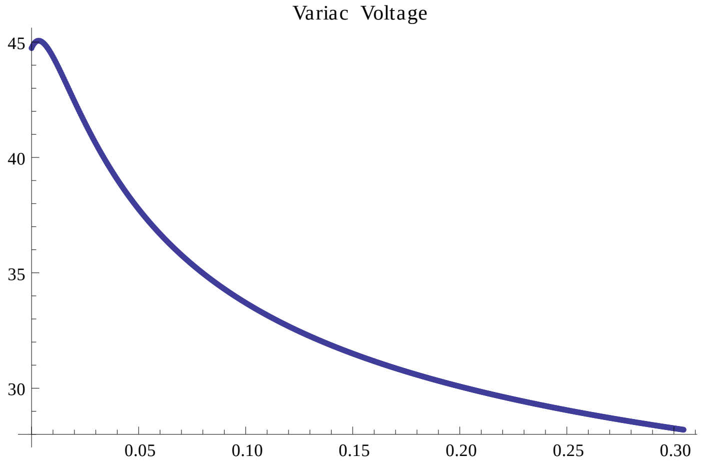


```
Conductive and Convective Crossover Point

{0.0149597, 0.999098}
```









Analysis:

- **Both models converge at large thickness in terms of relevant parameters for design (outer temp of insulation, and variac voltage)**

The trade-off in models is really how much you are heating up the room:

In the low thickness limit, where the models diverge, having a large convective resistance (model 2) implies the need for less voltage at the trade off of a more gradual temperature gradient. The bulk of the heat transfer gradient is occuring in the surrounding air, as it is acting more as an infinite conductive medium than a convective current. Therefore, less voltage is needed at the cost of back-action heating from warming up the surrounding environment significantly.

In model 1, the convection is extremely small compared to the thermal resistance of the insulation, due to the extreme heat transfer that is evident from the boundary layer analysis. Therefore, most of the thermal gradient is accounted for in the insulation, making the outer temperature decline at a much faster rate with respect to thickness, and giving us a more stable laboratory environment without a large temperature increase near the heat pipe. The trade off is a spike in voltage, since the heat flux has increased dramatically in the low thickness limit.

My Conclusion

- **Technically, model 1 is the most thorough and accounts for the most parameters and physical processes. Furthermore, due to the cooling system that we have in M4, which, with high fan settings and cold air, would actually force the convection current even beyond either the free convection or the stagnet model, I am inclined to trust model 1, the closest representation to the lab conditions. If anything, the currents may be higher due to bolstered heat flux, but we should expect that most of the thermal exchange is occuring in the insulation.**

  As shown by the critical insulation value, for insurance, we should make the insulation at least 3 mm thick. Due to geometric constraints, I think we can fit between 50-100 mm of insulation around the heat pipe.

  For this particular problem in insulative heat transfer, there is no law of diminishing returns, nor is there an ideal thickness due to a convective area trade-off. Thus, the rule of thumb will be: ‘The more we can fit, the better’.

  At 50 mm, we can expect minimums of about 50V and 32C outer temperature.

APPENDIX B
MATLAB MC SIMULATIONS

```
function zeemanTrendMC(l,natoms,plotflags)
%l, a vector of double precision input lengths in meters
%to condition slower build;
%natoms, an integer number of atoms to simulate for each length;
%plotflags, a vector of boolean values for plotting
%in the form of
%[Btheory fields, V Hists, cap vs l trends, radial dist].
%Example input: ([.3:.02:.5,100,[0,0,1,0])
%gives the capture trends for 100 atom simulations
%for lengths .3 to .5 meters in .02 m increments.
%Note: Only flag one plot per run, ie: [1,1,1,1] not viable
%Tip: for viewing the B and Vmax field graphs over a set of lengths,
%set natoms to 1, since the code will run faster and the
%fields are not a function of natoms.

%%Clear Screen
clc

%%Establish Global Variables
global kB Isat Int muB hbar vmin zstart Bnoise f lambda m G Tn k...
    vmean amax f0 delta0 plotflagFields plotflagHist numsamp...
    zmax sampleSize plotflagTrend capp ooBp Irat plotflagradDist

%%Sort Indicators for Plotting

plotflagFields = plotflags(1); % Field Profiles
plotflagHist = plotflags(2);    % Histograms
plotflagTrend = plotflags(3);  % Build Trends
plotflagradDist = plotflags(4);% Radial Distribution
```

```
%%Declare Variables
sampleSize = length(l);          % Num Slowers to Simulate
kB = 1.33*10^-23;                % Boltzman's Constant
Isat = 11.5043;                  % Saturation Intensity
Int = 57.5217;                   % Intensity
Irat = 5;                        % Intensity Ratio
muB = 9.27*10^-24;               % Bohr Magneton
hbar = 1.05*10^-34;              % Reduced Planck's Constant
f = 0.7;                         % F number
lambda = 670*10^-9;              % Wavelength
m = 1.165*10^-26;                % Mass
G = 3.66*10^7;                   % Transition rate
Tn = 670;                        % Nozzle Temp
vmin = 100;                      % Minimum capture velocity
zstart = 0.1;                    % Starting location of slowing field
Bnoise = 1.0e-4;                 % Gaussian noise in the field

%%Compute Maximum Decelleration


k = 2*pi/lambda;                 % Wavevector
amax = (G/2)*hbar*k/m;           % Max Decelleration
f0 = -1.95868e+07;               % f0 found by inversion of
                                 % f= Irate/(1+Irat+4f0^2/G^2)
delta0 = 2*pi*f0;                % Initial Detuning

numsamp=1;                       % Loop over all lengths

%% Begin Printing for Notification of Progress


    for i=1
```

```
fprintf('\n *** DATA SAMPLE %d/%d *** \n',numsamp,sampleSize)
disp('Percent Complete')

%% Calculate Maximum Capture Velocity

zmax = i;              % Maximum slower length and velocity
vmean = sqrt((zmax-zstart)*(2*f*amax)+vmin^2);


%% Plotting of Btheory and Vtheory Fields

if plotflagFields

    subplot(1,sampleSize,numsamp);
    q= 0:.0001:1;
    [Bt,vt] = Btheot(q);
    plot(q,Bt*10^4,q,vt)
    title(strcat('Length: ',num2str(i)))
    xlabel('Distance (m)')
    ylabel('B and v curves, (gauss, m/s)')

end

%% MC Sims for length i

beamsimult(natoms)    % Function call to beamsimult
numsamp = numsamp+1;

%% Plot the Trend Data

if plotflagTrend
```

```
            subplot(1,3,1)
            plot(i,vmean,'*b','MarkerSize',5) % Length vs max velocity
            xlabel('Length (m)')
            ylabel('Max Capture Velocity m/s')
            hold on
            subplot(1,3,2)
            plot(i,capp,'*b','MarkerSize',5)  % Length vs %capture
            xlabel('Length (m)')
            ylabel('%Captured')
            hold on
            subplot(1,3,3)
            plot(i,ooBp,'*b','MarkerSize',5)  % Length vs % OOB
            xlabel('Length (m)')
            ylabel('%ooB')
            hold on

        end
    end
end

%% B Field Calc Function
function[Bt,vt] = Btheot(z)

global muB hbar vmin zstart Bnoise f k vmean amax delta0 zmax

Bt = (hbar/muB)*(delta0 + k*sqrt(vmean^2 - 2*f*amax*(z-zstart)))...
    + Bnoise*sin(2*pi*z/0.003) -.035;
vt = sqrt(vmean^2-2*f*amax*(z-zstart));
Bmax =  (hbar/muB)*(delta0 + k*sqrt(vmean^2 -...
    2*f*amax*(zmax-zstart))) - .035;
Bmin =  (hbar/muB)*(delta0  + k*sqrt(vmean^2)) - .035;

```

```
%% Ramp Later Fifeld After Zmax
Bt(z(:)>zmax) = Bmax*exp(-(z(z(:)>zmax)-zmax).^2/0.05^2);
vt(z(:)>zmax) = vmin;

%% Ramp Early Field Before Zmin
Bt(z(:)<zstart) =  Bmin*exp(-(z(z(:)<zstart)-zstart).^2/0.025^2);
vt(z(:)<zstart) = 0;

end

%% Simulate Slowing Function
function [vmat,rmat] = singleatomSimult(vin,inputdx,inputdy)

global Irat muB hbar m G k delta0 zmax Bt vt inc

%% Establish Empty Storage Arrays
natoms = length(vin);
z0 = zeros(natoms,1);
numscatt=zeros(natoms,1);
stopp = zeros(natoms,1);
vmat = zeros(natoms,2);
rmat = zeros(natoms,3);

%% Establish Iterative Vectors
vzcurr = vin;
zcurr = z0;
xcurr = inputdx;
ycurr= inputdy;
rcurr = sqrt(xcurr.^2+ycurr.^2);

%% Establish Initial Transverse Jet
```

```
%Assumes jet with Sqrt(200) m/s initial radial velocity
%vxcurr = 10*ones(natoms,1); vycurr = vxcurr;

%Assumes jet with +- 0-15m/s initial rad velocity
%vxcurr = normrnd(0,4,natoms,1); vycurr = normrnd(0,4,natoms,1);

%Assumes jet with split gaussians centered at +-10
a = rand([natoms,1]); b = rand([natoms,1]); e = (a>b)+(a<b)*-1;
vxcurr = normrnd(10,3,natoms,1).*e;
a = rand([natoms,1]); b = rand([natoms,1]); e = (a>b)+(a<b)*-1;
vycurr = normrnd(10,3,natoms,1).*e;



dt = 50*10^-9;                % Establish time steps and
t=0;
vrec = hbar*k/m;              % Recoil kick

%% Calc BTheory
inc = .0001;
theoryRange = 0:inc:zmax;
[Bt,vt] = Btheot(theoryRange);

plusi = 0;

%% Simulate Until No Atoms Left
    while ~isempty(stopp)

        t = t + dt;           % Timestep

                              % Interpolate to nearest Btheory
        breal = interp1(theoryRange,Bt,zcurr,'nearest');

```

```
                      % Update current delta and Reff
currentdelta = delta0 + k*vzcurr -...
    muB*breal./hbar - muB*.035/hbar;
R = (Irat)./(1 + Irat + 4*currentdelta.^2./G^2);

%% Simulate scattering event
scats = unifrnd(0,1,length(stopp),1) < R;

    numscatt(scats) = numscatt(scats)+1;

                      % Update z velocity
    vzcurr(scats) = vzcurr(scats) - vrec;

                      % Generate angular recoil vector
    theta = rand(sum(scats),1)*pi;
    phi = rand(sum(scats),1)*2*pi;

    %% Test for no scattering and update transverse
    if ~isempty(theta)&& ~isempty(phi)
        vxcurr(scats) = vxcurr(scats) +...
            vrec.*sin(theta).*cos(phi);
        vycurr(scats) = vycurr(scats) +...
            vrec.*sin(theta).*sin(phi);
    end

%% Raise stop flag for exceeding max z or beam radius
stopp((zcurr>(zmax+zmax*.01))|(rcurr>=.0127)|(vzcurr<0)) = 1;

%% Update still simulating components


zcurr(~stopp) = zcurr(~stopp) + dt*vzcurr(~stopp);
xcurr(~stopp) = xcurr(~stopp) + dt*vxcurr(~stopp);
```

```
ycurr(~stopp) = ycurr(~stopp) + dt*vycurr(~stopp);
rcurr(~stopp) = sqrt(xcurr(~stopp).^2+ycurr(~stopp).^2);

%% Print progress for atoms reaching critical pts

if ~isempty(find(stopp))

    fprintf(' \t %4.2f \n',(1-length(stopp)/natoms)*100)

                      % Logical indexing for storage
    ind = find(stopp);
    plusi = max(plusi);
    plusi = plusi + (1:length(ind));



    %% Storage of key values at critical pts

    acurr = atan2(ycurr,xcurr);
    vmat(plusi,:) = [sqrt(vxcurr(ind).^2+...
        vycurr(ind).^2),vzcurr(ind)];
    rmat(plusi,:) = [rcurr(ind),zcurr(ind),acurr(ind)];

    %% Remove elements which simulate from crit atoms

    zcurr(ind) =  [];
    xcurr(ind) =  [];
    ycurr(ind) =  [];
    vzcurr(ind) = [];
    vxcurr(ind) = [];
    vycurr(ind) = [];
    rcurr(ind) =  [];
    stopp(ind) =  [];  % Clear and shrink stop sim flags
```

```
        end
    end
end

%% Beam Simulation Function

function beamsimult(natoms)

global kB m Tn vmin zmax vmean plotflagHist numsamp...
    sampleSize ooBp capp plotflagradDist

%% Nozzel Spread
vavg = sqrt(9*pi*kB*Tn/(8*m));
stand = sqrt(2*kB*Tn/m*(2-(9/16)*pi));
inputv = normrnd(vavg,stand,natoms,1);

    % Guarantee no negative velocities within stats
    while sum(inputv<0) ~= 0

        inputv(inputv(:)<0) = normrnd(vavg,stand,sum(inputv(:)<0),1);

    end

    %% Initial radial distributions for 0-.5cm gaussian rad

    inputdx = normrnd(0,.0015,natoms,1);
    inputdy = normrnd(0,.0015,natoms,1);
        inputdr = sqrt(inputdx.^2+inputdy.^2);
    inputda = atan2(inputdy,inputdx);

    %% Run simulation
```

```
[vmat,rmat] = singleatomSimult(inputv,inputdx,inputdy);


    outputv = vmat(:,2);      % Assign output Vz

    % All of above which fail radial crit get assigned -100
    outputv(((rmat(:,1)>=.0127) & (rmat(:,2) < zmax))...
        | vmat(:,2)<0) = -100;

%% Radial Distribution Plot
if plotflagradDist

    subplot(2,sampleSize,numsamp)
    fakeTheta = 0:.1:2*pi;
    polar(fakeTheta,.0127*ones(1,length(fakeTheta)),'xk')
    hold on
    polar(inputda,inputdr,'*b')
    title(strcat('Initial Radial Distribution (m) for l= ',...
        num2str(zmax)))
    subplot(2,sampleSize,numsamp+sampleSize)
    polar(fakeTheta,.0127*ones(1,length(fakeTheta)),'xk')
    hold on
    polar(rmat(:,3),rmat(:,1),'*r')
    title(strcat('Final Radial Distribution (m) for l= ',...
        num2str(zmax)))

end

%% Velocity Histogram Plots
if plotflagHist
```

```
        subplot(2,sampleSize,numsamp)
        hist(inputv,20)
        title(strcat('Gaussian Velocity Spread: ',num2str(zmax)))
        subplot(2,sampleSize,numsamp+sampleSize)
        hist(outputv,100)
        title(strcat('Output Velocity Spread: ',num2str(zmax)))

    end

%% Print Data per Run
fprintf('\n')
disp('Run Data')
fprintf('\t Vin Model %d m/s \n',vmean)
fprintf('\t Slower Length %5.2f cm \n\n',zmax*100)
disp('Slowing Data')

ooB = sum(outputv==-100);  % Calculate out of bounds
ooBp = 100*ooB/natoms;
capp = 100*(sum(outputv(:)<=vmin)-ooB)/natoms;
fprintf('\t Percent Cap %4.2f \n\t Percent ooB %4.2f \n\t Percent Lost %4.2f \n',..
    capp,ooBp,(100-capp-ooBp))

end
```

APPENDIX C

MATHEMATICICA VIRTUAL BUILD AND CHARACTERIZATION

The Lithium Zeeman Slower

Author: John Lombard, 6/1/11 - 2/17/12, Last Modified

Calculations for the magnetic fields from coaxial coils adapted from an original notebook written by Mike Gehm and modified by Michael Stenner, 5/18/04

Magnetic Fields from Coaxial Coils

- **Constants**

```
μ0 = 4 Pi * 10^-3;(* Gives answers in Gauss *)
```

- **Calculation Functions**

These functions calculate the radial and axial field for a single coil.

$$\mathtt{Bz[r\_, z\_, coil\_] := coil[[3]] * coil[[4]] *} \frac{\mathtt{\mu 0\ /\ (2\,Pi)}}{\sqrt{(\mathtt{coil[[1]]} + \mathtt{r})^2 + (\mathtt{z} - \mathtt{coil[[2]]})^2}} * \left(\mathtt{EllipticK}\left[\frac{4\ \mathtt{coil[[1]]\ r}}{(\mathtt{coil[[1]]} + \mathtt{r})^2 + (\mathtt{z} - \mathtt{coil[[2]]})^2}\right] + \left(\frac{\mathtt{coil[[1]]}^2 - \mathtt{r}^2 - (\mathtt{z} - \mathtt{coil[[2]]})^2}{(\mathtt{coil[[1]]} - \mathtt{r})^2 + (\mathtt{z} - \mathtt{coil[[2]]})^2}\right) * \mathtt{EllipticE}\left[\frac{4\ \mathtt{coil[[1]]\ r}}{(\mathtt{coil[[1]]} + \mathtt{r})^2 + (\mathtt{z} - \mathtt{coil[[2]]})^2}\right]\right)$$

$$\mathtt{Br[r\_, z\_, coil\_] :=}$$
$$\mathtt{coil[[3]] * coil[[4]] * (z - coil[[2]]) *} \frac{\mathtt{\mu 0\ /\ (2\,Pi\ r)}}{\sqrt{(\mathtt{coil[[1]]} + \mathtt{r})^2 + (\mathtt{z} - \mathtt{coil[[2]]})^2}} * \left(-\mathtt{EllipticK}\left[\frac{4\ \mathtt{coil[[1]]\ r}}{(\mathtt{coil[[1]]} + \mathtt{r})^2 + (\mathtt{z} - \mathtt{coil[[2]]})^2}\right] + \left(\frac{\mathtt{coil[[1]]}^2 + \mathtt{r}^2 + (\mathtt{z} - \mathtt{coil[[2]]})^2}{(\mathtt{coil[[1]]} - \mathtt{r})^2 + (\mathtt{z} - \mathtt{coil[[2]]})^2}\right) * \mathtt{EllipticE}\left[\frac{4\ \mathtt{coil[[1]]\ r}}{(\mathtt{coil[[1]]} + \mathtt{r})^2 + (\mathtt{z} - \mathtt{coil[[2]]})^2}\right]\right)$$

These functions calculate the field for all of the coils listed in "coils".

```
AxialField[r_, z_, coils_] := Module[{}, TempFunc[coil_] := Bz[r, z, coil];
  Plus @@ Map[TempFunc, coils, {1}]]
```

```
RadialField[r_, z_, coils_] := Module[{}, TempFunc[coil_] := Br[r, z, coil];
  Plus @@ Map[TempFunc, coils, {1}]]
```

This function calculates the field magnitude at any point using the previous functions.

```
FieldMag[r_, z_, coils_] := If[r == 0, AxialField[r, z, coils],
  √ (AxialField[r, z, coils]^2 + RadialField[r, z, coils]^2)]
```

Compute the Theoretical Ideal Magnetic Field Profile

Setting up the iterated coils:
InitR = Initial radius of coil, r0
RStep = Radial increase for each layer
RNum = Total number of radial winds
InitZ = Initial longitudinal position of the coil, z0
ZStep = Longitudinal increase for each turn
ZNum = Total number of longitudinal winds
Current = Current through the coil

```
RegularSpacedRealCoil[InitR_, RStep_, RNum_, InitZ_, ZStep_, ZNum_, Current_] :=
  Flatten[Table[{InitR + i * RStep, InitZ + j * ZStep, 1, Current}, {i, 0, RNum - 1},
    {j, 0, ZNum - 1}], 1];
```

Defining constants for velocity and B-field:

```
hb = 6.626 * 10 ^ -34 / 2 / Pi; (*hbar*)
delta = 2 * Pi * -480 * 10 ^ 6; (*detuning*)
lambda = 670 * 10 ^ -9; (*wavelength*)
MB = 9.27 * 10 ^ -28; (*bohr magneton*)
a = 1.61 * 10 ^ 6; (*acceleration*)
vf = 100; (*final velocity*)
l = .46; (*slower length*)
zstart = 0; (*slower position start*)
f = .76; (*f number*)
k = 2 * Pi / lambda; (*wave vector*)
offset = -120; (*spin flip offset*)
G = 37.7 * 10 ^ 6; (*gamma*)

grain = .005; (*discrete field profile granularity*)
fitShift = -.006; (*offset for shorter profile to fit*)
```

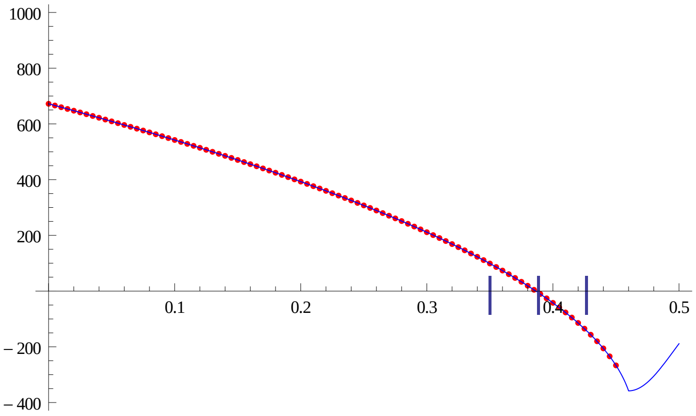
Compute Velocity and Bfield profiles

```
InitV[vf_, l_, a_] := Sqrt[vf ^ 2 + (l - zstart) * 2 * a * f]; (*initial velocity*)

Bmin = (hb / MB) * (delta + k * Sqrt[InitV[vf, l, a] ^ 2] - G / 2 * f) + offset;
(*Bfield mins and maxes*)
Bmax = (hb / MB) * (delta + k * Sqrt[InitV[vf, l, a] ^ 2 - 2 * a * (l - zstart) * f] - G / 2 * f) + offset;

bz[z_] :=
  If[z < zstart, Bmin * Exp[-(z - zstart) ^ 2 / (0.025 ^ 2)], If[z > l, Bmax * Exp[-(z - l) ^ 2 / .05 ^ 2],
    (hb / MB) * (delta + k * Sqrt[InitV[vf, l, a] ^ 2 - (2 * a * (z - zstart) * f)] - G / 2 * f) +
     offset]]; (*Bfield*)

vt[z_] := If[z < zstart, InitV[vf, l, a],
   If[z > l, vf, Sqrt[InitV[vf, l, a] ^ 2 - 2 * a * f * (z - zstart)]]] ;(*velocity profile*)
```

Setup and Display Plots

and Display Plots Setup

```
vtheoryPlot= Plot[vt[z], {z, 0, .5}, PlotStyle→ Green];
theoryPlot = Plot[bz[z], {z, 0, .5}, PlotStyle→ Blue];
theoryPoint= Table[{z, bz[z]}, {z, 0, .5, grain}];
theoryPointShort= Table[{z, bz[z]}, {z, 0, l + fitShift, grain}]; (*field to be fitted *)
theoryPointShortPlot= ListPlot[theoryPointShort, PlotStyle→ Red];
theoryPointPlot= ListPlot[theoryPoint, PlotStyle→ Red];
vtheoryPoint = Table[{z, vt[z]}, {z, 0, .5, grain}];
transverseBounds= ListPlot[{{0.35, 0}, {0.4262, 0}, {.3881, 0}}, PlotMarkers→ {"|", Large} ];
(*boundaries based on the cross position*)
Show[theoryPointShortPlot, theoryPlot, transverseBounds, PlotRange→ {{0, .5}, {-400, 1000}}]
```



Building the Virtual Slower:

- **Establish wire data and dimensions**

```
S1 = .0254/ 2;    (*First wind at what radius (brass tube)*)
SOverRad = .03556/ 2; (*" for the "over" independent hovering coils*)
SCrossRad = .01905/ 2; (*" for the coils on the cross*)
RInc = 0.0036 ;(*hollow wire radial increment by gauge*)
SZInc = RInc; (*vertical increment by gauge, hollow*)
SZIncEff2 = .2916/ 84; (*effective gauge determined after actual winding *)
CrossSlowSpace = .0762; (*total length of 6way *)
SZCrossInc = .001715; (*14AWG*)
RCrossInc = SZCrossInc;
RMedGauge = .00213; (*12AWG*)
SZMedGauge= RMedGauge;
flange = .00755+ .005; (*flange length*)
R10Gauge= .00259; (*10AWG*)
SZ10Gauge= R10Gauge;
```

■ Establish Turn Geometries

```
(*Change the SZTurn values in order to alter the field
 profile within the contraint of the geometry as outlined below,
change the RTurn value after the field profile is close to tollerance in order to
 optimize the current through each coil toward the fewest possible power sources*)

RLayer1= 2;
SZLayer1 = 89; (*actually 89 down, 92 back *)
RLayer2 = 1;
SZLayer2 = 84;
RLayer3 = 1;
SZLayer3 = 74;
RLayer4= 1;
SZLayer4= 104; (*actually 101,
but in same volume (not enought to warrent effective gauge change *)
RLayer5= 1;
SZLayer5= 87; (*actually 82, but.... ^ *)
RLayer6= 1;
SZLayer6= 70; (*actually 64, but ... ^  *)
RLayer7= 1;
SZLayer7= 68; (*57 *)
RLayer8= 1;
SZLayer8= 47;  (* 42 *)
RLayer9= 1;
SZLayer9= 25;

S2 = S1 + RLayer1* RInc;
S3 = S2 + RLayer2* RInc;
S4 = S3 + RLayer3* RInc;
S5 = S4 + RLayer4* RMedGauge;
S6 = S5 + RLayer5* RMedGauge;
S7 = S6 + RLayer6* RMedGauge;
S8 = S7 + RLayer7* RCrossInc;
S9 = S8 + RLayer8* RCrossInc;


RTurn2nd = 3; (*coil at the 2nd half of the slower, after the cross*)
SZTurn2nd = 18;

RTurnCross1= 3;
SZTurnCross1 = 9;
RTurnCross2= 3;
SZTurnCross2= 9;

RTurnOver1= 10; (*ramp up coil*)
SZTurnOver1= 12;(*Round[flange/SZCrossInc]; *)
RTurnOver2 = 5; (*before cross*)
SZTurnOver2= 13;
RTurnOver3 = 5; (*after cross*)
SZTurnOver3 = 13;

(*2*flange/SZCrossInc*)
```

■ Calculate Coil Widths

```
Width2nd = SZCrossInc * SZTurn2nd;

WidthCross1 = SZCrossInc * SZTurnCross1;
WidthCross2 = SZCrossInc * SZTurnCross2;

WidthOver1 = SZMedGauge * SZTurnOver1;
WidthOver2 = SZCrossInc * SZTurnOver2;
WidthOver3 = SZCrossInc * SZTurnOver3;

WidthLayer1 = SZLayer1 * SZInc;
(*.35 - 2 * flange
  WidthLayer1*)
WidthLayer2 = SZLayer2 * SZIncEff2;
WidthLayer3 = SZLayer3 * SZIncEff2;
WidthLayer4 = SZLayer4 * SZMedGauge;
WidthLayer5 = SZLayer5 * SZMedGauge;
WidthLayer6 = SZLayer6 * SZMedGauge;
WidthLayer7 = SZLayer7 * SZCrossInc;
WidthLayer8 = SZLayer8 * SZCrossInc;
WidthLayer9 = SZLayer9 * SZCrossInc;
```

■ Calculate Slower Build Positions

```
SlowerStart = flange + RInc;
StartofSecondPart = .35 + CrossSlowSpace + flange;
crossWindSpace = (CrossSlowSpace - 2 * flange - .01905) / 2;
(*crossWindSpace / SZCrossInc*)
```

■ Build Coils

```
SingleCoilLayer1 = RegularSpacedRealCoil[S1, RInc, RLayer1, SlowerStart, SZInc, SZLayer1, 1];
SingleCoilLayer2 = RegularSpacedRealCoil[S2, RInc, RLayer2, SlowerStart, SZIncEff2, SZLayer2, 1];
SingleCoilLayer3 = RegularSpacedRealCoil[S3, RInc, RLayer3, SlowerStart, SZIncEff2, SZLayer3, 1];
SingleCoilLayer4 =
  RegularSpacedRealCoil[S4, RMedGauge, RLayer4, SlowerStart, SZMedGauge, SZLayer4, 1];
SingleCoilLayer5 = RegularSpacedRealCoil[S5, RMedGauge,
   RLayer5, SlowerStart, SZMedGauge, SZLayer5, 1];
SingleCoilLayer6 = RegularSpacedRealCoil[S6, RMedGauge, RLayer6,
   SlowerStart, SZMedGauge, SZLayer6, 1];
SingleCoilLayer7 = RegularSpacedRealCoil[S7, RCrossInc, RLayer7,
   SlowerStart, SZCrossInc, SZLayer7, 1];
SingleCoilLayer8 = RegularSpacedRealCoil[S8, RCrossInc, RLayer8,
   SlowerStart, SZCrossInc, SZLayer8, 1];
SingleCoilLayer9 = RegularSpacedRealCoil[S9, RCrossInc, RLayer9,
   SlowerStart, SZCrossInc, SZLayer9, 1];

SingleCoil2nd =
  RegularSpacedRealCoil[SOverRad, RCrossInc, RTurn2nd, StartofSecondPart, SZCrossInc, SZTurn2nd, 1];

SingleCoilCross1 = RegularSpacedRealCoil[SCrossRad,
   RCrossInc, RTurnCross1, .35 + flange, SZCrossInc, SZTurnCross1, 1];
SingleCoilCross2 = RegularSpacedRealCoil[SCrossRad, RCrossInc, RTurnCross2,
   .35 + flange + .01905 + crossWindSpace, SZCrossInc, SZTurnCross2, 1];

SingleCoilOver1 = RegularSpacedRealCoil[SOverRad, RMedGauge,
   RTurnOver1, -WidthOver1 + flange - .005 + RInc, SZMedGauge, SZTurnOver1, 1];
SingleCoilOver2 = RegularSpacedRealCoil[SOverRad, RCrossInc, RTurnOver2,
   .35 - WidthOver2 / 2, SZCrossInc, SZTurnOver2, 1];
SingleCoilOver3 = RegularSpacedRealCoil[SOverRad, RCrossInc, RTurnOver3,
   .35 - WidthOver3 / 2 + CrossSlowSpace, SZCrossInc, SZTurnOver3, 1];

(*.075 - 2 * flange
  Width2nd + RCrossInc / 4

  .00755 + .004 + .004 + .001 + (.075 - 2 * flange - Width2nd)
  .35  -  2 * flange
  WidthLayer1
  4 * RInc + 3 * RMedGauge + 3 * RCrossInc
  (S9 + RCrossInc) * 2

  WidthLayer7 - WidthLayer8
  WidthLayer1 - WidthLayer8*)
```

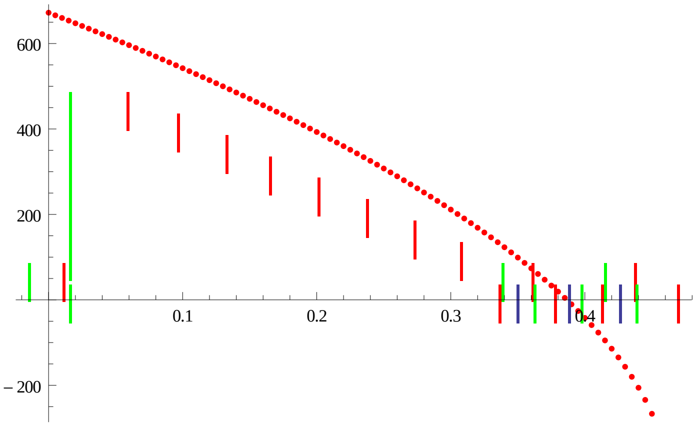
Create the boundary markers for the coil positions

```
CoilPositionsList = {{SlowerStart, 0}, {SlowerStart + WidthLayer1, 0}, {SlowerStart, 100},
   {SlowerStart + WidthLayer2, 100}, {SlowerStart, 150}, {SlowerStart + WidthLayer3, 150},
   {SlowerStart, 200}, {SlowerStart + WidthLayer4, 200}, {SlowerStart, 250},
   {SlowerStart + WidthLayer5, 250}, {SlowerStart, 300}, {SlowerStart + WidthLayer6, 300},
   {SlowerStart, 350}, {SlowerStart + WidthLayer7, 350}, {SlowerStart, 400},
   {SlowerStart + WidthLayer8, 400}, {SlowerStart, 450}, {SlowerStart + WidthLayer9, 450},
   {StartofSecondPart, 0}, {StartofSecondPart + Width2nd, 0}, {.35 + flange, 0},
   {.35 + flange + WidthCross1, 0}, {.35 + flange + .01905 + crossWindSpace, 0},
   {.35 + flange + .01905 + crossWindSpace + WidthCross2, 0},
   {.35 - WidthOver2 / 2, 50}, {.35 + WidthOver2 / 2, 50},
   {.35 - WidthOver3 / 2 + CrossSlowSpace, 50}, {.35 + WidthOver3 / 2 + CrossSlowSpace, 50},
   {-WidthOver1 + flange - .005 + RInc, 50}, {flange - .005 + RInc, 50}};
```

Build and Show Plots For the Coil Positions

```
and Build Coil For Plots Positions Show the
```

```
coilPositions1 = ListPlot[Table[CoilPositionsList[[2 * n]], {n, 1, Length[CoilPositionsList] / 2}],
   PlotMarkers → {"|", Large}, PlotStyle → Red];
coilPositions2 = ListPlot[Table[CoilPositionsList[[2 * n + 1]],
    {n, 0, Length[CoilPositionsList] / 2 - 1}], PlotMarkers → {"|", Large}, PlotStyle → Green];
(*Coil start in Green, Coil end in REd*)

Show[theoryPointShortPlot, coilPositions1, coilPositions2, transverseBounds, PlotRange → All]
(*{{-.05,.5},{-.01,100}}]*)           (* Shoot for ~.03m spacings between layers 1-9 *)
```



Building the Fields Up

```
BLayer1[z_] := AxialField[0, z, SingleCoilLayer1];
BLayer2[z_] := AxialField[0, z, SingleCoilLayer2];
BLayer3[z_] := AxialField[0, z, SingleCoilLayer3];
BLayer4[z_] := AxialField[0, z, SingleCoilLayer4];
BLayer5[z_] := AxialField[0, z, SingleCoilLayer5];
BLayer6[z_] := AxialField[0, z, SingleCoilLayer6];
BLayer7[z_] := AxialField[0, z, SingleCoilLayer7];
BLayer8[z_] := AxialField[0, z, SingleCoilLayer8];
BLayer9[z_] := AxialField[0, z, SingleCoilLayer9];
B2nd[z_] := AxialField[0, z, SingleCoil2nd];
BCross1[z_] := AxialField[0, z, SingleCoilCross1];
BCross2[z_] := AxialField[0, z, SingleCoilCross2];
BOver1[z_] := AxialField[0, z, SingleCoilOver1];
BOver2[z_] := AxialField[0, z, SingleCoilOver2];
BOver3[z_] := AxialField[0, z, SingleCoilOver3];
```

```
Bfield[iLayer1_, iLayer2_, iLayer3_, iLayer4_, iLayer5_, iLayer6_, iLayer7_,
   iLayer8_, iLayer9_, i2nd_, iCross1_, iCross2_, iOver1_, iOver2_, iOver3_] :=
  iLayer1* BLayer1[z] + iLayer2* BLayer2[z] + iLayer3* BLayer3[z] + iLayer4* BLayer4[z] +
   iLayer5* BLayer5[z] + iLayer6* BLayer6[z] + iLayer7* BLayer7[z] +
   iLayer8* BLayer8[z] + iLayer9* BLayer9[z] + i2nd * B2nd[z] + iCross1* BCross1[z] +
   iCross2* BCross2[z] + iOver1* BOver1[z] + iOver2* BOver2[z] + iOver3* BOver3[z];
Clear[iLayer1, iLayer2, iLayer3, iLayer4, iLayer5, iLayer6, iLayer7, iLayer8,
  iLayer9, i2nd, iCross1, iCross2, iOver1, iOver2, iOver3];
```

Perform A Nonlinear Regression for the Currents Fit to Theory:

```
currents = FindFit[theoryPointShort, Bfield[iLayer1, iLayer2, iLayer3, iLayer4, iLayer5,
    iLayer6, iLayer7, iLayer8, iLayer9, i2nd, iCross1, iCross2, iOver1, iOver2, iOver3],
   {iLayer1, iLayer2, iLayer3, iLayer4, iLayer5, iLayer6, iLayer7, iLayer8,
    iLayer9, i2nd, iCross1, iCross2, iOver1, iOver2, iOver3}, z];
```

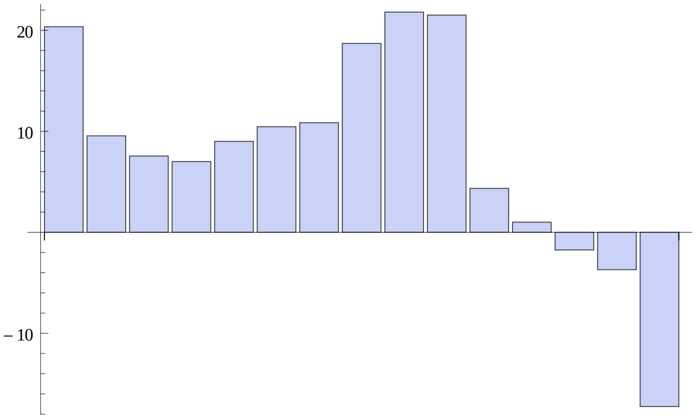
■ Round and Store the Fitted Currents

```
rounds = Round[Table[currents[[n]][[2]], {n, 1, Length[currents]}], .05];
currentsR = {iLayer1→ rounds[[1]], iLayer2→ rounds[[2]], iLayer3→ rounds[[3]],
  iLayer4→ rounds[[4]], iLayer5→ rounds[[5]], iLayer6→ rounds[[6]], iLayer7→ rounds[[7]],
  iLayer8→ rounds[[8]], iLayer9→ rounds[[9]], i2nd → rounds[[10]], iCross1→ rounds[[11]],
  iCross2→ rounds[[12]], iOver1→ rounds[[13]], iOver2→ rounds[[14]], iOver3→ rounds[[15]]}
```

```
{iLayer1 → 21.5, iLayer2 → 21.8, iLayer3 → 18.7, iLayer4 → 10.85, iLayer5 → 10.45,
 iLayer6 → 9., iLayer7 → 7., iLayer8 → 7.55, iLayer9 → 9.55, i2nd → -17.25,
 iCross1 → 1., iCross2 → -1.75, iOver1 → 20.35, iOver2 → 4.35, iOver3 → -3.7}
```

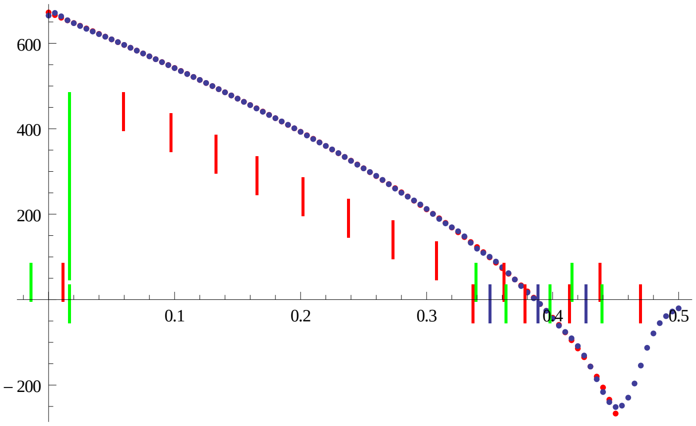
Graph the Currents

```
currentGraph= BarChart[{iOver1, iLayer9, iLayer8, iLayer7, iLayer6, iLayer5, iLayer4,
     iLayer3, iLayer2, iLayer1, iOver2, iCross1, iCross2, iOver3, i2nd} /. currentsR];
Show[currentGraph]
```



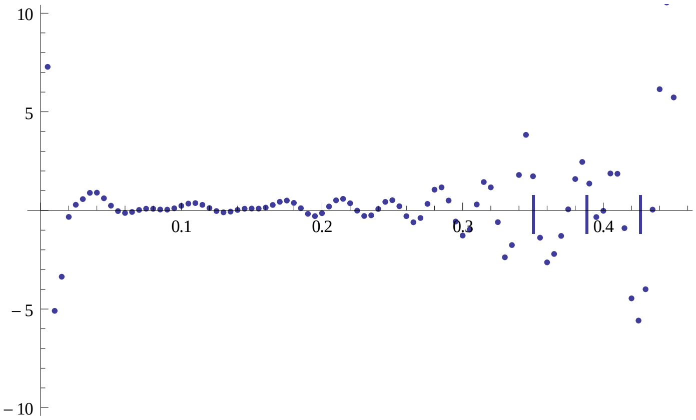
Plots with the Rounded Current Values

```
roundPoint=
  Table[{z, (iLayer1* BLayer1[z] + iLayer2* BLayer2[z] + iLayer3* BLayer3[z] + iLayer4* BLayer4[z] +
        iLayer5* BLayer5[z] + iLayer6* BLayer6[z] + iLayer7* BLayer7[z] + iLayer8* BLayer8[z] +
        iLayer9* BLayer9[z] + i2nd * B2nd [z] + iCross1* BCross1[z] + iCross2* BCross2[z] +
        iOver1* BOver1[z] + iOver2* BOver2[z] + iOver3* BOver3[z]) /. currentsR}, {z, 0, .5, grain}];
roundPointPlot = ListPlot[roundPoint];


Show[theoryPointShortPlot, roundPointPlot,
 coilPositions1, coilPositions2, transverseBounds, PlotRange→ All]
```



Calculate and Plot the Residuals Between Fitted Field and Theory

```
differenceRoundPoints= Flatten[theoryPoints[[2 ;; ;; 2]] - Flatten[roundPoints[[2 ;; ;; 2]];
differenceRoundPointPlot=
  ListPlot[Table[{z, differenceRoundPoints[[z / grain]]}, {z, 0, l + fitShift, grain}]];
Show[differenceRoundPointPlot, transverseBounds, PlotRange→ {{0, l + fitShift}, {-10, 10}}]
```



Calculate the Maximum Residuals in Particular Ranges

```
Max[Table[Abs[differenceRoundPoints[[n]]], {n,
   Round[.2 / .5 * Length[differenceRoundPoints]], Floor[(.42) / .5 * Length[differenceRoundPoints]]}]]
Max[Table[Abs[differenceRoundPoints[[n]]], {n, 1, Round[.2 / .5 * Length[differenceRoundPoints]]}]]
```

```
4.45864
```

```
7.27978
```

Export the data tables

```
zData = Table[Flatten[roundPoints][[2 * n + 1]], {n, 0, Length[Flatten[roundPoints]] / 2 - 1}];
bData = Table[Flatten[roundPoints][[2 * n]], {n, 1, Length[Flatten[roundPoints]] / 2}];
vData= Table[Flatten[vtheoryPoints][[2 * n]], {n, 1, Length[Flatten[vtheoryPoints]] / 2}];
dataMatrix = Flatten[Transpose[{zData, bData, vData}]];
Export["EffectiveGaugeAdj.txt", dataMatrix]
```

```
EffectiveGaugeAdj.txt
```

Data Regarding the Designed Slower

Length of Wire Calculations

```
LengthofOneWire[coil_] := Apply[Plus, Take[Flatten[coil], {1, (Length[coil] * 4), 4}] * 2 * Pi];
LengthofWires[coillist_] := Sum[LengthofOneWire[coillist[[n]]], {n, 1, Length[coillist]}];

totalLength = LengthofWires[
   {SingleCoilLayer1, SingleCoilLayer2, SingleCoilLayer3, SingleCoilLayer4, SingleCoilLayer5,
    SingleCoilLayer6, SingleCoilLayer7, SingleCoilLayer8, SingleCoilLayer9, SingleCoil2nd,
    SingleCoilCross1, SingleCoilCross2, SingleCoilOver1, SingleCoilOver2, SingleCoilOver3}];
lengthList = {LengthofOneWire[SingleCoilLayer1], LengthofOneWire[SingleCoilLayer2],
   LengthofOneWire[SingleCoilLayer3], LengthofOneWire[SingleCoilLayer4],
   LengthofOneWire[SingleCoilLayer5], LengthofOneWire[SingleCoilLayer6],
   LengthofOneWire[SingleCoilLayer7], LengthofOneWire[SingleCoilLayer8],
   LengthofOneWire[SingleCoilLayer9], LengthofOneWire[SingleCoil2nd], LengthofOneWire[
    SingleCoilCross1], LengthofOneWire[SingleCoilCross2], LengthofOneWire[SingleCoilOver1],
   LengthofOneWire[SingleCoilOver2], LengthofOneWire[SingleCoilOver3]} * 3.28399

lengthMax = Max[lengthList]
totalLengthf = totalLength * 3.2808399 (*convert to feet*)
```

```
{53.2561, 34.4917, 35.8824, 58.1546, 52.4723, 45.2956, 46.99,
 34.1416, 19.0451, 21.7219, 6.26198, 6.26198, 67.7577, 28.447, 28.447}

67.7577

538.11
```

Resistance Calculations

```
ResistivityHollow = .0006; (*ohms per foot for hollow core *)
Resistivity12 = .001588;
Resistivity14 = .002525;
Resistivity10 = .0009989;
(*Resistivity15 = .002761;*)

resistanceList = Flatten[{Table[ResistivityHollow * lengthList[[n]], {n, 1, 3}],
   Table[Resistivity12 * lengthList[[n]], {n, 4, 6}],
   Table[Resistivity14 * lengthList[[n]], {n, 7, 12}], Resistivity12 * lengthList[[13]],
   Table[Resistivity14 * lengthList[[n]], {n, 14, 15}]}]
resistanceMax = Max[resistanceList]
resistanceTotal = Sum[resistanceList[[n]], {n, 1, Length[resistanceList]}]
```

```
{0.0319537, 0.020695, 0.0215294, 0.0923496, 0.0833259, 0.0719294, 0.11865,
 0.0862075, 0.0480889, 0.0548479, 0.0158115, 0.0158115, 0.107599, 0.0718286, 0.0718286}

0.11865

0.912456
```

Current Calculations

```
currentList = {iLayer1, iLayer2, iLayer3, iLayer4, iLayer5, iLayer6, iLayer7,
   iLayer8, iLayer9, i2nd, iCross1, iCross2, iOver1, iOver2, iOver3} /. currentsR
currentMax = Max[Abs[currentList]]
currentTotal = Sum[Abs[currentList][[n]], {n, 1, Length[currentList]}]
```

```
{21.5, 21.8, 18.7, 10.85, 10.45, 9., 7., 7.55, 9.55, -17.25, 1., -1.75, 20.35, 4.35, -3.7}

21.8

164.8
```

Power Calculations

```
powerList = (currentList^2) * resistanceList
PowerMax = Max[powerList]
posPowerMax = Position[powerList, PowerMax]
PowerTotal = Sum[powerList[[n]], {n, 1, Length[powerList]}]
```

```
{14.7706, 9.83509, 7.52863, 10.8716, 9.0994, 5.82628, 5.81384,
 4.91405, 4.38583, 16.3207, 0.0158115, 0.0484227, 44.5592, 1.35918, 0.983333}

44.5592

{{13}}

136.332
```

Voltage Calculations

```
voltageList = currentList * resistanceList;
VoltageMax = Max[Abs[voltageList]]
VoltageTotal = Sum[Abs[voltageList][[n]], {n, 1, Length[voltageList]}]
```

```
2.18964

9.759
```

True Coil Readings Import

- ### Import Data Sheets and Normalize Currents

```
RealLayer1F = Flatten[Import["~/Research/MV Labs/SlowerData/LayerFa/layer1.txt",
     "Table", "FieldSeparators" → ";"], 1] / 3;

(*F for measurements taken in cm increments foward on the slower*)
RealLayer1B =
  Reverse[Flatten[Import["~/Research/MV Labs/SlowerData/LayerBa/layer1.txt", "Table",
      "FieldSeparators" → ";"], 1]] / 3;

(*B for measurements taken in cm increments offset from F backward on the slower,
reverse to flip orientation*)

RealLayer1 = Riffle[RealLayer1B, RealLayer1F];
(*combining the datasets for 1/2 cm increments and systematic error reduction *)
```

```
RealLayer2F = Flatten[Import["~/Research/MV Labs/SlowerData/LayerFa/layer2.txt",
     "Table", "FieldSeparators" → ";"], 1] / 5;
RealLayer2B = Reverse[Flatten[Import["~/Research/MV Labs/SlowerData/LayerBa/layer2.txt",
      "Table", "FieldSeparators" → ";"], 1]] / 5;
RealLayer2 = Riffle[RealLayer2B, RealLayer2F];

RealLayer3F = Flatten[Import["~/Research/MV Labs/SlowerData/LayerFa/layer3.txt",
     "Table", "FieldSeparators" → ";"], 1] / 5;
RealLayer3B = Reverse[Flatten[Import["~/Research/MV Labs/SlowerData/LayerBa/layer3.txt",
      "Table", "FieldSeparators" → ";"], 1]] / 5;
RealLayer3 = Riffle[RealLayer3B, RealLayer3F];

RealLayer4F = Flatten[Import["~/Research/MV Labs/SlowerData/LayerFa/layer4.txt",
     "Table", "FieldSeparators" → ";"], 1] / 4;
RealLayer4B = Reverse[Flatten[Import["~/Research/MV Labs/SlowerData/LayerBa/layer4.txt",
      "Table", "FieldSeparators" → ";"], 1]] / 4;
RealLayer4 = Riffle[RealLayer4B, RealLayer4F];

RealLayer5F = Flatten[Import["~/Research/MV Labs/SlowerData/LayerFa/layer5.txt",
     "Table", "FieldSeparators" → ";"], 1] / 4;
RealLayer5B = Reverse[Flatten[Import["~/Research/MV Labs/SlowerData/LayerBa/layer5.txt",
      "Table", "FieldSeparators" → ";"], 1]] / 4;
RealLayer5 = Riffle[RealLayer5B, RealLayer5F];

RealLayer6F = Flatten[Import["~/Research/MV Labs/SlowerData/LayerFa/layer6.txt",
     "Table", "FieldSeparators" → ";"], 1] / 4;
RealLayer6B = Reverse[Flatten[Import["~/Research/MV Labs/SlowerData/LayerBa/layer6.txt",
      "Table", "FieldSeparators" → ";"], 1]] / 4;
RealLayer6 = Riffle[RealLayer6B, RealLayer6F];

RealLayer7F = Flatten[Import["~/Research/MV Labs/SlowerData/LayerFa/layer7.txt",
     "Table", "FieldSeparators" → ";"], 1] / 4;
RealLayer7B = Reverse[Flatten[Import["~/Research/MV Labs/SlowerData/LayerBa/layer7.txt",
      "Table", "FieldSeparators" → ";"], 1]] / 4;
RealLayer7 = Riffle[RealLayer7B, RealLayer7F];

RealLayer8F = Flatten[Import["~/Research/MV Labs/SlowerData/LayerFa/layer8.txt",
     "Table", "FieldSeparators" → ";"], 1] / 4;
RealLayer8B = Reverse[Flatten[Import["~/Research/MV Labs/SlowerData/LayerBa/layer8.txt",
      "Table", "FieldSeparators" → ";"], 1]] / 4;
RealLayer8 = Riffle[RealLayer8B, RealLayer8F];

RealLayer9F = Flatten[Import["~/Research/MV Labs/SlowerData/LayerFa/layer9.txt",
     "Table", "FieldSeparators" → ";"], 1] / 5;
RealLayer9B = Reverse[Flatten[Import["~/Research/MV Labs/SlowerData/LayerBa/layer9.txt",
      "Table", "FieldSeparators" → ";"], 1]] / 5;
RealLayer9 = Riffle[RealLayer9B, RealLayer9F];

RealOver1 = Reverse[Flatten[Import["~/Research/MV Labs/SlowerData/IndependentCoils/ramp.txt",
     "Table", "FieldSeparators" → ";"], 1]];
(*flip coil orientation: for physical coil, smooth side faces slower,
not oven *)
RealOver2 = Flatten[Import["~/Research/MV Labs/SlowerData/IndependentCoils/beforecross.txt",
     "Table", "FieldSeparators" → ";"], 1] * (2 / 3);
RealOver3 = Flatten[Import["~/Research/MV Labs/SlowerData/IndependentCoils/aftercross.txt",
     "Table", "FieldSeparators" → ";"], 1] * (2 / 3);
Real2nd = Reverse[Flatten[Import["~/Research/MV Labs/SlowerData/IndependentCoils/flip.txt",
      "Table", "FieldSeparators" → ";"], 1]] * (2 / 3); (*also flip coil orientation *)

RealLayerList = {RealLayer1, RealLayer2, RealLayer3, RealLayer4, RealLayer5,
   RealLayer6, RealLayer7, RealLayer8, RealLayer9}; (*gather data into a list*)
```

■ Amp by Theoretical Currents

```
RealLayerListAdj = RealLayerList * Table[currentList[[n]], {n, 1, 9}];
RealOver1Adj = RealOver1 * currentList[[13]];
RealOver2Adj = RealOver2 * currentList[[14]];
RealOver3Adj = RealOver3 * currentList[[15]];
Real2ndAdj = Real2nd * currentList[[10]];
```

■ Setup Z axis

```
(*The coil positions were determined from matching the body of the plots,
more specifically, their average peak positions, as well as where in
 space the coils should have been located by the CoilPositions1/2 markings *)

PlotShiftLayers = SlowerStart - .2188;
PlotShiftOver1 = -WidthOver1 + flange - .005 + RInc - .205;
PlotShiftOver2 = .35 - WidthOver2 / 2 - .18;
PlotShiftOver3 = .35 + WidthOver2 / 2 - .126;
PlotShift2nd = StartofSecondPart - .204(*.182*);;
RealZLayers = .005 * Table[z, {z, 0, Length[RealLayer1] - 2}] + PlotShiftLayers;
RealZOver1 = .005 * Table[z, {z, 0, Length[RealOver1] - 2}] + PlotShiftOver1;
RealZOver2 = .005 * Table[z, {z, 0, Length[RealOver2] - 2}] + PlotShiftOver2;
RealZOver3 = .005 * Table[z, {z, 0, Length[RealOver3] - 2}] + PlotShiftOver3;
RealZ2nd = .005 * Table[z, {z, 0, Length[Real2nd] - 2}] + PlotShift2nd;
```

■ Sort B Data and Setup Plots

```
(*Interpolation was used to average out the
 systematic error in the forward and backward measurements*)

RealLayer1FieldS =
  Table[{RealZLayers[[z]], RealLayerListAdj[[1]][[z]]}, {z, 1, Length[RealZLayers]}];
RealLayer1FieldFctn = Interpolation[RealLayer1FieldS];
RealLayer1FieldPlotS = ListPlot[RealLayer1FieldS, PlotStyle -> Red]; (*split plot*)
RealLayer1Field = Table[{z, RealLayer1FieldFctn[z]}, {z, 0, .5, .005}];
RealLayer1FieldPlot = ListPlot[RealLayer1Field, PlotStyle → Black]; (*MVP *)

RealLayer2FieldS =
  Table[{RealZLayers[[z]], RealLayerListAdj[[2]][[z]]}, {z, 1, Length[RealZLayers]}];
RealLayer2FieldFctn = Interpolation[RealLayer2FieldS];
RealLayer2FieldPlotS = ListPlot[RealLayer2FieldS, PlotStyle -> Red]; (*split plot*)
RealLayer2Field = Table[{z, RealLayer2FieldFctn[z]}, {z, 0, .5, .005}];
RealLayer2FieldPlot = ListPlot[RealLayer2Field, PlotStyle → Black]; (*MVP *)

RealLayer3FieldS =
  Table[{RealZLayers[[z]], RealLayerListAdj[[3]][[z]]}, {z, 1, Length[RealZLayers]}];
```

```
RealLayer3FieldFctn = Interpolation[RealLayer3FieldS];
RealLayer3FieldPlotS = ListPlot[RealLayer3FieldS, PlotStyle -> Red]; (*split plot*)
RealLayer3Field = Table[{z, RealLayer3FieldFctn[z]}, {z, 0, .5, .005}];
RealLayer3FieldPlot = ListPlot[RealLayer3Field, PlotStyle → Black]; (*MVP*)

RealLayer4FieldS =
  Table[{RealZLayers[[z]], RealLayerListAdj[[4]][[z]]}, {z, 1, Length[RealZLayers]}];
RealLayer4FieldFctn = Interpolation[RealLayer4FieldS];
RealLayer4FieldPlotS = ListPlot[RealLayer4FieldS, PlotStyle -> Red]; (*split plot*)
RealLayer4Field = Table[{z, RealLayer4FieldFctn[z]}, {z, 0, .5, .005}];
RealLayer4FieldPlot = ListPlot[RealLayer4Field, PlotStyle → Black]; (*MVP*)

RealLayer5FieldS =
  Table[{RealZLayers[[z]], RealLayerListAdj[[5]][[z]]}, {z, 1, Length[RealZLayers]}];
RealLayer5FieldFctn = Interpolation[RealLayer5FieldS];
RealLayer5FieldPlotS = ListPlot[RealLayer5FieldS, PlotStyle -> Red]; (*split plot*)
RealLayer5Field = Table[{z, RealLayer5FieldFctn[z]}, {z, 0, .5, .005}];
RealLayer5FieldPlot = ListPlot[RealLayer5Field, PlotStyle → Black]; (*MVP*)

RealLayer6FieldS =
  Table[{RealZLayers[[z]], RealLayerListAdj[[6]][[z]]}, {z, 1, Length[RealZLayers]}];
RealLayer6FieldFctn = Interpolation[RealLayer6FieldS];
RealLayer6FieldPlotS = ListPlot[RealLayer6FieldS, PlotStyle → Red];
RealLayer6Field = Table[{z, RealLayer6FieldFctn[z]}, {z, 0, .5, .005}];
RealLayer6FieldPlot = ListPlot[RealLayer6Field, PlotStyle → Black]; (*MVP*)

RealLayer7FieldS =
  Table[{RealZLayers[[z]], RealLayerListAdj[[7]][[z]]}, {z, 1, Length[RealZLayers]}];
RealLayer7FieldFctn = Interpolation[RealLayer7FieldS];
RealLayer7FieldPlotS = ListPlot[RealLayer7FieldS, PlotStyle → Red];
RealLayer7Field = Table[{z, RealLayer7FieldFctn[z]}, {z, 0, .5, .005}];
RealLayer7FieldPlot = ListPlot[RealLayer7Field, PlotStyle → Black]; (*MVP*)

RealLayer8FieldS =
  Table[{RealZLayers[[z]], RealLayerListAdj[[8]][[z]]}, {z, 1, Length[RealZLayers]}];
RealLayer8FieldFctn = Interpolation[RealLayer8FieldS];
RealLayer8FieldPlotS = ListPlot[RealLayer8FieldS, PlotStyle → Red];
RealLayer8Field = Table[{z, RealLayer8FieldFctn[z]}, {z, 0, .5, .005}];
RealLayer8FieldPlot = ListPlot[RealLayer8Field, PlotStyle → Black]; (*MVP*)

RealLayer9FieldS =
  Table[{RealZLayers[[z]], RealLayerListAdj[[9]][[z]]}, {z, 1, Length[RealZLayers]}];
RealLayer9FieldFctn = Interpolation[RealLayer9FieldS];
RealLayer9FieldPlotS = ListPlot[RealLayer9FieldS, PlotStyle → Red];
RealLayer9Field = Table[{z, RealLayer9FieldFctn[z]}, {z, 0, .5, .005}];
RealLayer9FieldPlot = ListPlot[RealLayer9Field, PlotStyle → Black]; (*MVP*)

RealLayerFieldList = {RealLayer1Field, RealLayer2Field, RealLayer3Field, RealLayer4Field,
   RealLayer5Field, RealLayer6Field, RealLayer7Field, RealLayer8Field, RealLayer9Field};

RealLayerFieldFctnList = {RealLayer1FieldFctn, RealLayer2FieldFctn,
   RealLayer3FieldFctn, RealLayer4FieldFctn, RealLayer5FieldFctn, RealLayer6FieldFctn,
   RealLayer7FieldFctn, RealLayer8FieldFctn, RealLayer9FieldFctn};
```

```
RealLayerFieldS =
  Table[{RealZLayers[[z]], Sum[RealLayerListAdj[[n]][[z]], {n, 1, Length[RealLayerListAdj]}]},
   {z, 1, Length[RealZLayers]}];
RealLayerFieldPlotS = ListPlot[RealLayerFieldS, PlotStyle → Red];

RealLayerField =
  Table[{z, Sum[RealLayerFieldFctnList[[n]][z], {n, 1, Length[RealLayerFieldFctnList]}]},
   {z, 0, .5, .005}];
RealLayerFieldPlot = ListPlot[RealLayerField, PlotStyle → Black];

RealOver1Field = Table[{RealZOver1[[z]], RealOver1Adj[[z]]}, {z, 1, Length[RealZOver1]}];
RealOver1FieldPlot = ListPlot[RealOver1Field, PlotStyle → Red];

RealOver2Field = Table[{RealZOver2[[z]], RealOver2Adj[[z]]}, {z, 1, Length[RealZOver2]}];
RealOver2FieldPlot = ListPlot[RealOver2Field, PlotStyle → Red];

RealOver3Field = Table[{RealZOver3[[z]], RealOver3Adj[[z]]}, {z, 1, Length[RealZOver3]}];
RealOver3FieldPlot = ListPlot[RealOver3Field, PlotStyle → Red];

Real2ndField = Table[{RealZ2nd[[z]], Real2ndAdj[[z]]}, {z, 1, Length[RealZ2nd]}];
Real2ndFieldPlot = ListPlot[Real2ndField, PlotStyle → Red];

(* NB IGNORE INTERPOLATION ERRORS.  Extrapolation is fine,
since at the beginning and end of every data set,
I put a series of 10 zeros to insure that the extrapolation won't take the field to +
 - infinity due to an initial small offset slope that happened to grow massively during
  extrapolation  This effectively forces the field to die to 0 beyond data taking range,
which as confirmed by the graphs, is not a bad approximation *)
```

```
InterpolatingFunction::dmval :
  Input value {0.495} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫

InterpolatingFunction::dmval :
  Input value {0.5} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫

InterpolatingFunction::dmval :
  Input value {0.495} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫

InterpolatingFunction::dmval :
  Input value {0.5} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫

InterpolatingFunction::dmval :
  Input value {0.495} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫

InterpolatingFunction::dmval :
  Input value {0.5} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫

InterpolatingFunction::dmval :
  Input value {0.495} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫

InterpolatingFunction::dmval :
  Input value {0.5} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫

InterpolatingFunction::dmval :
  Input value {0.495} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫

InterpolatingFunction::dmval :
  Input value {0.5} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫
```

```
InterpolatingFunction::dmval :
  Input value {0.495} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫
InterpolatingFunction::dmval :
  Input value {0.5} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫
InterpolatingFunction::dmval :
  Input value {0.495} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫
InterpolatingFunction::dmval :
  Input value {0.5} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫
InterpolatingFunction::dmval :
  Input value {0.495} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫
InterpolatingFunction::dmval :
  Input value {0.5} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫
InterpolatingFunction::dmval :
  Input value {0.495} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫
InterpolatingFunction::dmval :
  Input value {0.5} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫
InterpolatingFunction::dmval :
  Input value {0.495} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫
InterpolatingFunction::dmval :
  Input value {0.495} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫
InterpolatingFunction::dmval :
  Input value {0.495} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫
General::stop : Further output of InterpolatingFunction::dmval will be suppressed during this calculation. ≫
```

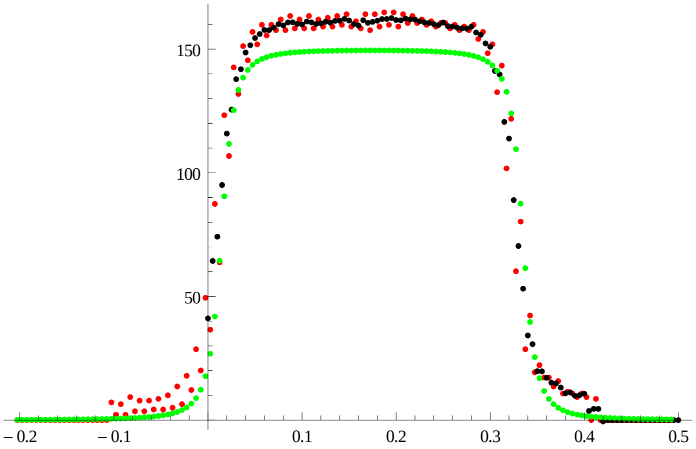
■ Setup Theory Plots

```
TheoryLayer1Field = Table[{RealZLayers[[z]], (iLayer1 * BLayer1[RealZLayers[[z]]]) /. currentsR},
   {z, 1, Length[RealZLayers]}];
TheoryLayer1FieldPlot = ListPlot[TheoryLayer1Field, PlotStyle → Green];

TheoryLayer2Field = Table[{RealZLayers[[z]], (iLayer2 * BLayer2[RealZLayers[[z]]]) /. currentsR},
   {z, 1, Length[RealZLayers]}];
TheoryLayer2FieldPlot = ListPlot[TheoryLayer2Field, PlotStyle → Green];

TheoryLayer3Field = Table[{RealZLayers[[z]], (iLayer3 * BLayer3[RealZLayers[[z]]]) /. currentsR},
   {z, 1, Length[RealZLayers]}];
TheoryLayer3FieldPlot = ListPlot[TheoryLayer3Field, PlotStyle → Green];

TheoryLayer4Field = Table[{RealZLayers[[z]], (iLayer4 * BLayer4[RealZLayers[[z]]]) /. currentsR},
   {z, 1, Length[RealZLayers]}];
TheoryLayer4FieldPlot = ListPlot[TheoryLayer4Field, PlotStyle → Green];

TheoryLayer5Field = Table[{RealZLayers[[z]], (iLayer5 * BLayer5[RealZLayers[[z]]]) /. currentsR},
   {z, 1, Length[RealZLayers]}];
TheoryLayer5FieldPlot = ListPlot[TheoryLayer5Field, PlotStyle → Green];

TheoryLayer6Field = Table[{RealZLayers[[z]], (iLayer6 * BLayer6[RealZLayers[[z]]]) /. currentsR},
   {z, 1, Length[RealZLayers]}];
TheoryLayer6FieldPlot = ListPlot[TheoryLayer6Field, PlotStyle → Green];

TheoryLayer7Field = Table[{RealZLayers[[z]], (iLayer7 * BLayer7[RealZLayers[[z]]]) /. currentsR},
   {z, 1, Length[RealZLayers]}];
TheoryLayer7FieldPlot= ListPlot[TheoryLayer7Field, PlotStyle→ Green];

TheoryLayer8Field= Table[{RealZLayers[[z]], (iLayer8* BLayer8[RealZLayers[[z]]]) /. currentsR},
   {z, 1, Length[RealZLayers]}];
TheoryLayer8FieldPlot= ListPlot[TheoryLayer8Field, PlotStyle→ Green];

TheoryLayer9Field= Table[{RealZLayers[[z]], (iLayer9* BLayer9[RealZLayers[[z]]]) /. currentsR},
   {z, 1, Length[RealZLayers]}];
TheoryLayer9FieldPlot= ListPlot[TheoryLayer9Field, PlotStyle→ Green];

TheoryLayerField=
  Table[{z, (iLayer1* BLayer1[z] + iLayer2* BLayer2[z] + iLayer3* BLayer3[z] + iLayer4* BLayer4[z] +
       iLayer5* BLayer5[z] + iLayer6* BLayer6[z] + iLayer7* BLayer7[z] +
       iLayer8* BLayer8[z] + iLayer9* BLayer9[z]) /. currentsR}, {z, 0, .5, .005}];
TheoryLayerFieldPlot= ListPlot[TheoryLayerField, PlotStyle→ Green];

TheoryOver1Field= Table[
   {RealZOver1[[z]], iOver1* BOver1[RealZOver1[[z]]] /. currentsR}, {z, 1, Length[RealZOver1]}];
TheoryOver1FieldPlot= ListPlot[TheoryOver1Field, PlotStyle→ Green];

TheoryOver2Field= Table[
   {RealZOver2[[z]], iOver2* BOver2[RealZOver2[[z]]] /. currentsR}, {z, 1, Length[RealZOver2]}];
TheoryOver2FieldPlot= ListPlot[TheoryOver2Field, PlotStyle→ Green];

TheoryOver3Field= Table[
   {RealZOver3[[z]], iOver3* BOver3[RealZOver3[[z]]] /. currentsR}, {z, 1, Length[RealZOver3]}];
TheoryOver3FieldPlot= ListPlot[TheoryOver3Field, PlotStyle→ Green];

Theory2ndField=
  Table[{RealZ2nd[[z]], i2nd * B2nd[RealZ2nd[[z]]] /. currentsR}, {z, 1, Length[RealZ2nd]}];
Theory2ndFieldPlot= ListPlot[Theory2ndField, PlotStyle→ Green];
```

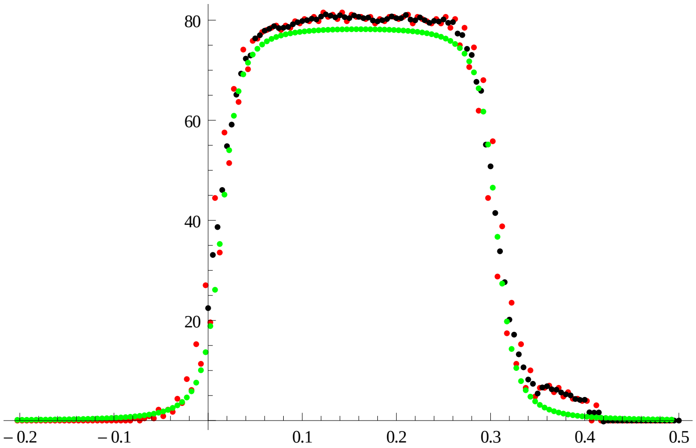
- ### Display Plot Comparison

```
Layer1
Show[RealLayer1FieldPlotS, RealLayer1FieldPlot, TheoryLayer1FieldPlot, PlotRange → All]
Layer2
Show[RealLayer2FieldPlotS, RealLayer2FieldPlot, TheoryLayer2FieldPlot, PlotRange → All]
Layer3
Show[RealLayer3FieldPlotS, RealLayer3FieldPlot, TheoryLayer3FieldPlot, PlotRange → All]
Layer4
Show[RealLayer4FieldPlotS, RealLayer4FieldPlot, TheoryLayer4FieldPlot, PlotRange → All]
Layer5
Show[RealLayer5FieldPlotS, RealLayer5FieldPlot, TheoryLayer5FieldPlot, PlotRange → All]
Layer6
Show[RealLayer6FieldPlotS, RealLayer6FieldPlot, TheoryLayer6FieldPlot, PlotRange → All]
Layer7
Show[RealLayer7FieldPlotS, RealLayer7FieldPlot, TheoryLayer7FieldPlot, PlotRange → All]
Layer8
Show[RealLayer8FieldPlotS, RealLayer8FieldPlot, TheoryLayer8FieldPlot, PlotRange → All]
Layer9
Show[RealLayer9FieldPlotS, RealLayer9FieldPlot, TheoryLayer9FieldPlot, PlotRange → All]
Layers
Show[RealLayerFieldPlotS, RealLayerFieldPlot, TheoryLayerFieldPlot, PlotRange → All]
Over1
Show[RealOver1FieldPlot, TheoryOver1FieldPlot,
 coilPositions1, coilPositions2, PlotRange → {{-.2, .2}, {0, 600}}]
Over2
Show[RealOver2FieldPlot, TheoryOver2FieldPlot,
 coilPositions1, coilPositions2, PlotRange → {{.2, .5}, {0, 500}}]
Over3
Show[RealOver3FieldPlot, TheoryOver3FieldPlot,
 coilPositions1, coilPositions2, PlotRange → {{.2, .5}, {50, -100}}]
Flip
Show[Real2ndFieldPlot, Theory2ndFieldPlot,
 coilPositions1, coilPositions2, PlotRange → {{.3, .6}, {0, -300}}]
```

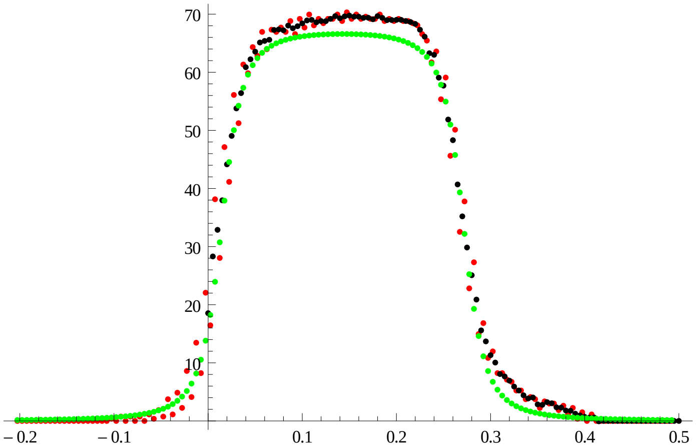
```
Layer1
```



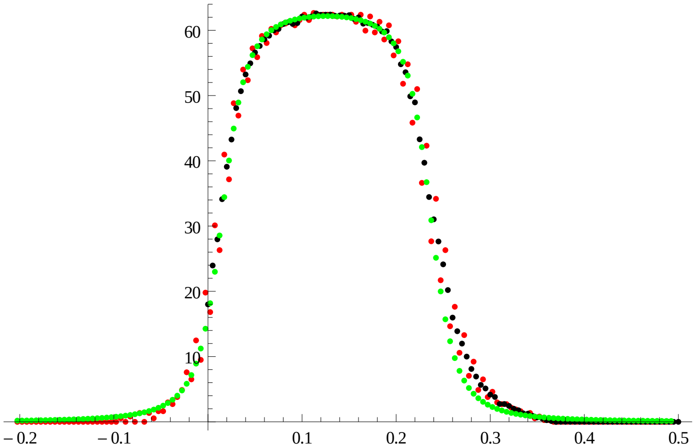
```
Layer2
```



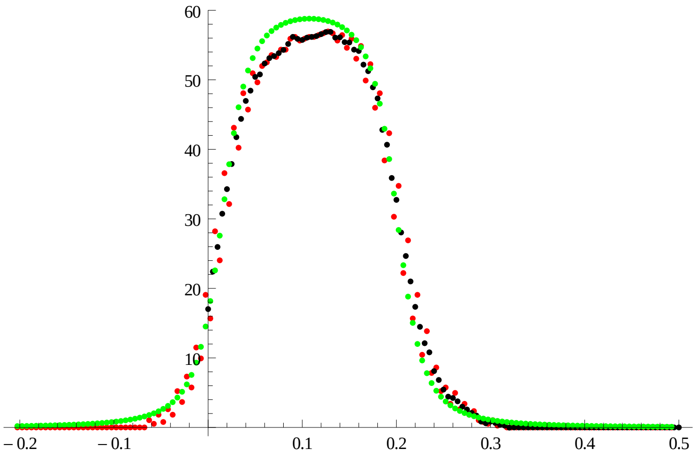
Layer3



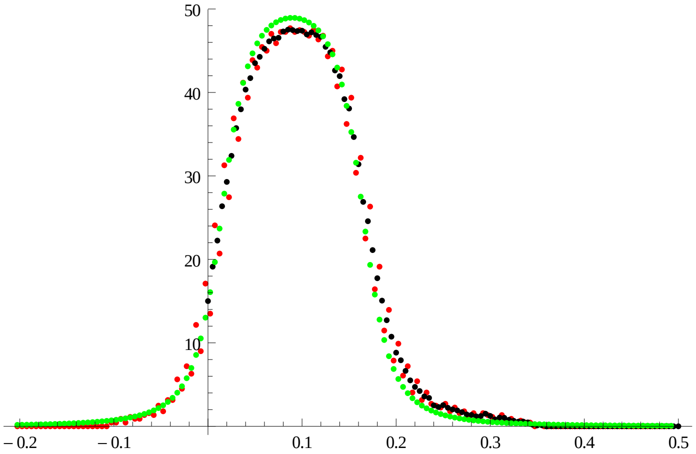
Layer4



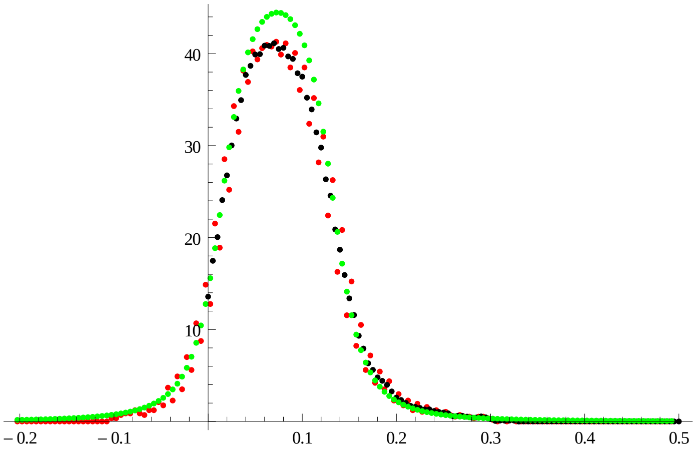
Layer5



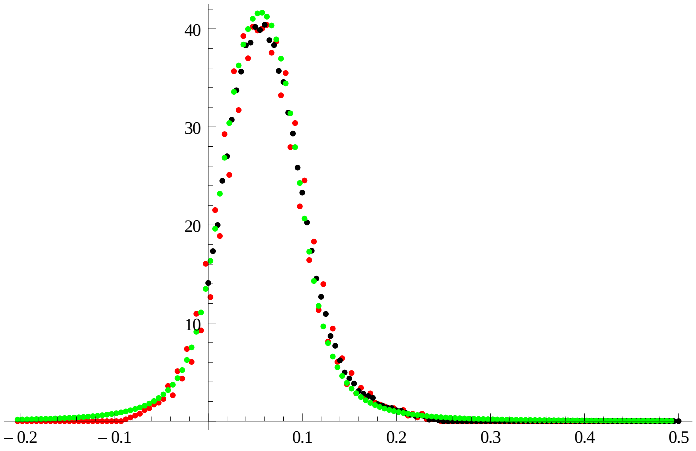
Layer6



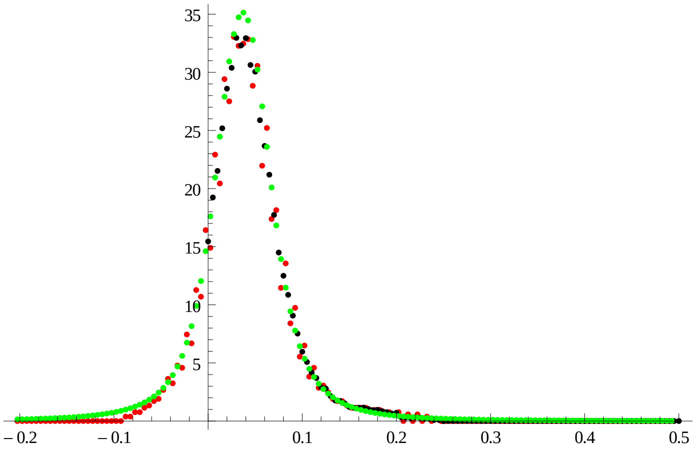
Layer7



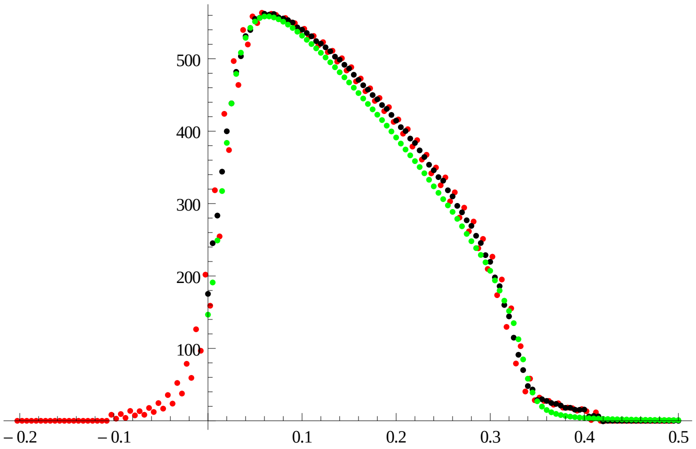
Layer8



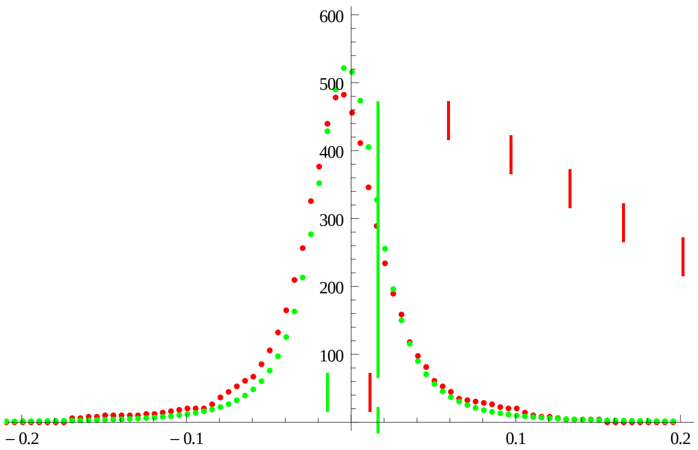
Layer9



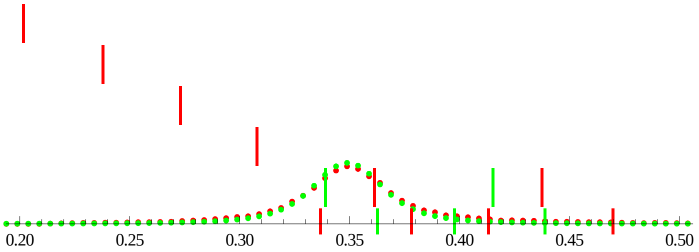
Layers



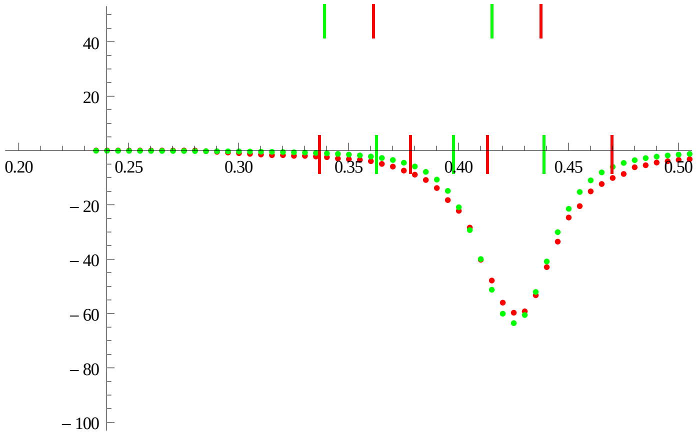
Over1



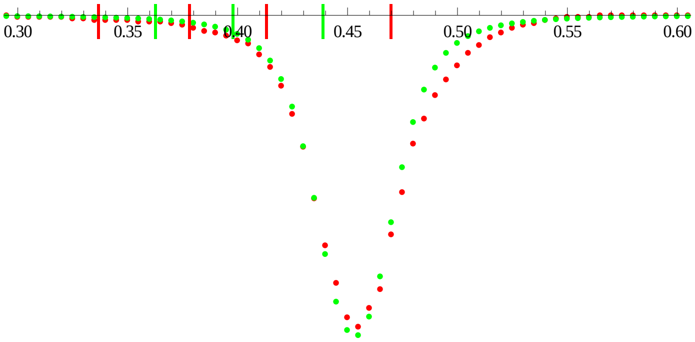
Over2



Over3

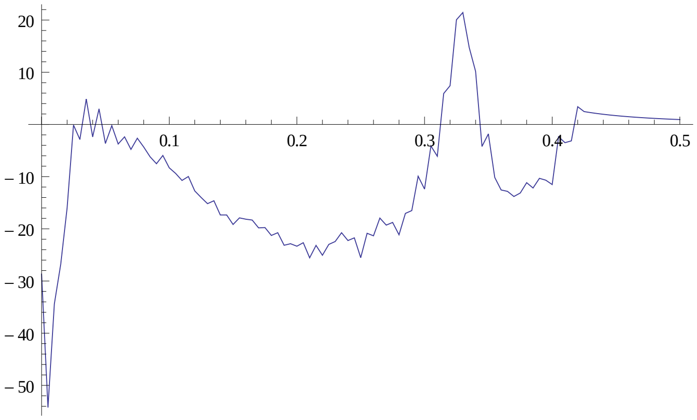


Flip



■ Residuals For Layers

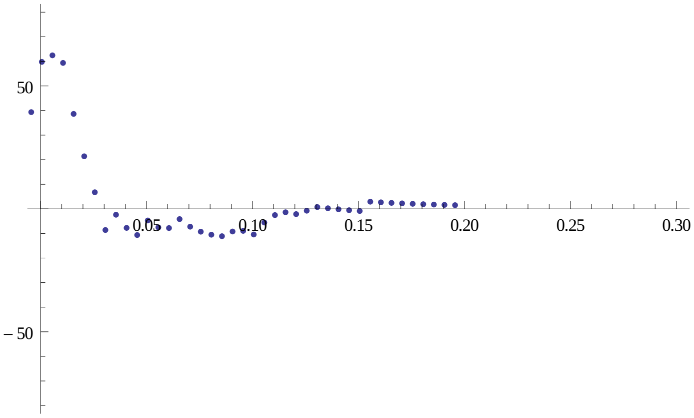
```
RealLayerResiduals = Table[{z * .005 - .005,
    (Flatten[TheoryLayerField][[2 ;; ;; 2]] - Flatten[RealLayerField][[2 ;; ;; 2]])[[z]]},
   {z, 1, Length[RealLayerField]}];
RealLayerResidualsPlot = ListPlot[RealLayerResiduals, Joined → True];
Show[RealLayerResidualsPlot, PlotRange → All]
Max[Abs[RealLayerResiduals]]
```



```
54.236
```

Residuals for Over1

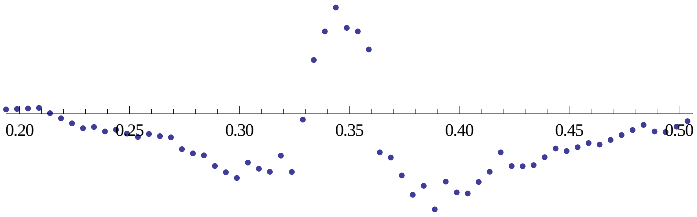
```
RealOver1Residuals = Table[{RealZOver1[[z]],
    (Flatten[TheoryOver1Field][[2 ;; ;; 2]] - Flatten[RealOver1Field][[2 ;; ;; 2]])[[z]]},
   {z, 1, Length[RealZOver1]}];
RealOver1ResidualsPlot = ListPlot[RealOver1Residuals];
Show[RealOver1ResidualsPlot, PlotRange → {{0, .3}, {-80, 80}}]
Max[Abs[RealOver1Residuals[[Flatten[Position[Flatten[RealZOver1], _ ? Positive]]]]]]
```



```
62.4417
```

Residuals for Over2

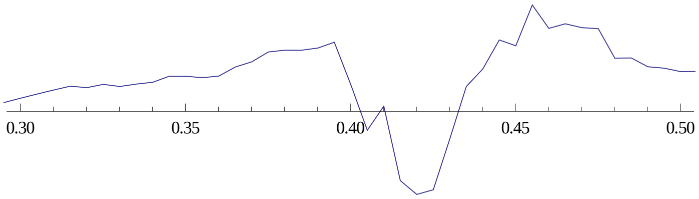
```
RealOver2Residuals = Table[{RealZOver2[[z]],
    (Flatten[TheoryOver2Field][[2 ;; ;; 2]] - Flatten[RealOver2Field][[2 ;; ;; 2]])[[z]]},
   {z, 1, Length[RealZOver2]}];
RealOver2ResidualsPlot = ListPlot[RealOver2Residuals];
Show[RealOver2ResidualsPlot, PlotRange → {{.2, .5}, {-10, 10}}]
Max[Abs[RealOver2Residuals]]
```



```
5.21262
```

Residuals for Over3

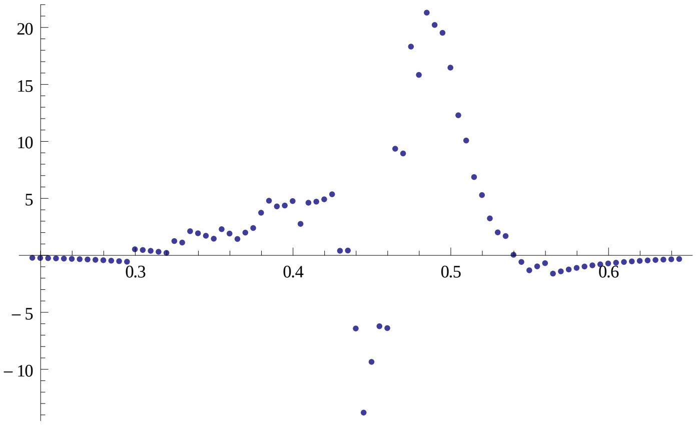
```
RealOver3Residuals = Table[{RealZOver3[[z]],
    (Flatten[TheoryOver3Field][[2 ;; ;; 2]] - Flatten[RealOver3Field][[2 ;; ;; 2]])[[z]]},
   {z, 1, Length[RealZOver3]}];
RealOver3ResidualsPlot = ListPlot[RealOver3Residuals, Joined → True];
Show[RealOver3ResidualsPlot, PlotRange → {{.3, .5}, {-10, 10}}]
Max[Abs[RealOver3Residuals]]
```



```
5.2071
```

Residuals for 2nd

```
Real2ndResiduals = Table[
   {RealZ2nd[[z]], (Flatten[Theory2ndField][[2 ;; ;; 2]] - Flatten[Real2ndField][[2 ;; ;; 2]])[[z]]},
   {z, 1, Length[RealZ2nd]}];
Real2ndResidualsPlot = ListPlot[Real2ndResiduals];
Show[Real2ndResidualsPlot, PlotRange → All]
Max[Abs[Drop[Real2ndResiduals, -35]]]
```



```
13.7973
```

New Fitting for Currents

■ Layers

```
BaseLayerList = Table[Flatten[RealLayerFieldList[[n]]][[2 ;; ;; 2]] / currentList[[n]], {n, 1, 9}];
BaseLayerFieldList = Table[Table[
    {.005 * z - .005, BaseLayerList[[n]][[z]]}, {z, 1, Length[BaseLayerList[[1]]]}], {n, 1, 9}];

RealLayerFieldFit[{iAdjLayer1_, iAdjLayer2_, iAdjLayer3_, iAdjLayer4_,
    iAdjLayer5_, iAdjLayer6_, iAdjLayer7_, iAdjLayer8_, iAdjLayer9_}] := Interpolation[
   Table[{.005 * z - .005, BaseLayerList[[1]][[z]] * iAdjLayer1 + BaseLayerList[[2]][[z]] * iAdjLayer2 +
      BaseLayerList[[3]][[z]] * iAdjLayer3 + BaseLayerList[[4]][[z]] * iAdjLayer4 +
      BaseLayerList[[5]][[z]] * iAdjLayer5 + BaseLayerList[[6]][[z]] * iAdjLayer6 +
      BaseLayerList[[7]][[z]] * iAdjLayer7 + BaseLayerList[[8]][[z]] * iAdjLayer8 +
      BaseLayerList[[9]][[z]] * iAdjLayer9}, {z, 1, Length[BaseLayerList[[1]]]}]];

currentListH = currentList + (Abs[currentList] * 1.3 - Abs[currentList]);
(*30% bounds on the original current recepie for the fitting*)
currentListL = currentList + (Abs[currentList] * .7 - Abs[currentList]);

(*Drop signifies a smaller region in which to fit to theory*)
(*Norm[#,9]& signifies the 9th root norm,
which happened to be the est fit in this case.  Could have used
 any number from 1 to infinity, and 9 was found by guess and check *)
LayerCurrentsAdj = FindFit[Drop[Drop[TheoryLayerField, 12], -40],
   {RealLayerFieldFit[{iAdjLayer1, iAdjLayer2, iAdjLayer3, iAdjLayer4,
       iAdjLayer5, iAdjLayer6, iAdjLayer7, iAdjLayer8, iAdjLayer9}][x],
    {(currentListH[[1]]) > iAdjLayer1 > (currentListL[[1]]), (currentListH[[2]]) >
      iAdjLayer2 > (currentListL[[2]]), (currentListH[[3]]) > iAdjLayer3 > (currentListL[[3]]),
     (currentListH[[4]]) > iAdjLayer4 > (currentListL[[4]]), (currentListH[[5]]) >
      iAdjLayer5 > (currentListL[[5]]), (currentListH[[6]]) > iAdjLayer6 > (currentListL[[6]]),
     (currentListH[[7]]) > iAdjLayer7 > (currentListL[[7]]), (currentListH[[8]]) > iAdjLayer8 >
      (currentListL[[8]]), (currentListH[[9]]) > iAdjLayer9 > (currentListL[[9]])}},
   {iAdjLayer1, iAdjLayer2, iAdjLayer3, iAdjLayer4, iAdjLayer5, iAdjLayer6, iAdjLayer7,
    iAdjLayer8, iAdjLayer9}, x, NormFunction → (Norm[#, 9] &)];
LayerRoundsAdj = Round[Table[LayerCurrentsAdj[[n]][[2]], {n, 1, Length[LayerCurrentsAdj]}], .01];
LayerCurrentsAdjR = {iAdjLayer1 → LayerRoundsAdj[[1]],
   iAdjLayer2 → LayerRoundsAdj[[2]], iAdjLayer3 → LayerRoundsAdj[[3]],
   iAdjLayer4 → LayerRoundsAdj[[4]], iAdjLayer5 → LayerRoundsAdj[[5]],
   iAdjLayer6 → LayerRoundsAdj[[6]], iAdjLayer7 → LayerRoundsAdj[[7]],
   iAdjLayer8 → LayerRoundsAdj[[8]], iAdjLayer9 → LayerRoundsAdj[[9]]};
LayerCurrentListAdj = {iAdjLayer1, iAdjLayer2, iAdjLayer3, iAdjLayer4, iAdjLayer5,
   iAdjLayer6, iAdjLayer7, iAdjLayer8, iAdjLayer9} /. LayerCurrentsAdjR
currentList[[
 1
  ;;
  9]]
```

```
{22.21, 15.26, 17.64, 10.11, 11.35, 9.08, 8.87, 8.42, 10.83}
```

```
{21.5, 21.8, 18.7, 10.85, 10.45, 9., 7., 7.55, 9.55}
```

Over1

```
RealOver1FieldFit[iAdjOver1_] := Interpolation[Table[{RealZOver1[[z]],
     (iAdjOver1 * RealOver1[[z]])}, {z, 1, Length[RealZOver1]}], x];
Clear[iAdjOver1];
Over1CurrentsAdj = FindFit[TheoryOver1Field[[
    Drop[Drop[Flatten[Position[Flatten[TheoryOver1Field[[1 ;; ;; 2]]], _? Positive]], -19], 2]]],
   {RealOver1FieldFit[iAdjOver1]}, {iAdjOver1}, x, NormFunction → (Norm[#, 1] &)];
Over1RoundsAdj = Round[Table[Over1CurrentsAdj[[n]][[2]], {n, 1, Length[Over1CurrentsAdj]}], .01];
Over1CurrentsAdjR = {iAdjOver1 → Over1RoundsAdj[[1]]};
Over1CurrentListAdj = {iAdjOver1} /. Over1CurrentsAdjR
```

```
{21.08}
```

Over2

```
RealOver2FieldFit[iAdjOver2_] := Interpolation[
   Table[{RealZOver2[[z]], (iAdjOver2 * RealOver2[[z]])}, {z, 1, Length[RealZOver2]}], x];
Clear[iAdjOver2];
Over2CurrentsAdj = FindFit[TheoryOver2Field,
   {RealOver2FieldFit[iAdjOver2]}, {iAdjOver2}, x, NormFunction → (Norm[#, 3] &)];
Over2RoundsAdj = Round[Table[Over2CurrentsAdj[[n]][[2]], {n, 1, Length[Over2CurrentsAdj]}], .01];
Over2CurrentsAdjR = {iAdjOver2 → Over2RoundsAdj[[1]]};
Over2CurrentListAdj = {iAdjOver2} /. Over2CurrentsAdjR
```

```
{4.41}
```

Over3

```
RealOver3FieldFit[iAdjOver3_] := Interpolation[
   Table[{RealZOver3[[z]], (iAdjOver3 * RealOver3[[z]])}, {z, 1, Length[RealZOver3]}], x];
Clear[iAdjOver3];
Over3CurrentsAdj = FindFit[Drop[TheoryOver3Field, -12],
   {RealOver3FieldFit[iAdjOver3]}, {iAdjOver3}, x, NormFunction → (Norm[#, 33] &)];
Over3RoundsAdj = Round[Table[Over3CurrentsAdj[[n]][[2]], {n, 1, Length[Over3CurrentsAdj]}], .01];
Over3CurrentsAdjR = {iAdjOver3 → Over3RoundsAdj[[1]]};
Over3CurrentListAdj = {iAdjOver3} /. Over3CurrentsAdjR
```

```
{-3.65}
```

2nd

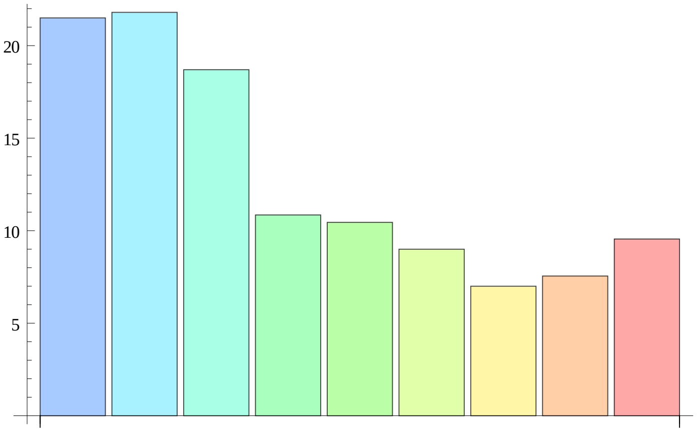
```
Real2ndFieldFit[iAdj2nd_] :=
  Interpolation[Table[{RealZ2nd[[z]], (iAdj2nd * Real2nd[[z]])}, {z, 1, Length[RealZ2nd]}], x];
Clear[iAdj2nd];
SndCurrentsAdj = FindFit[Drop[Theory2ndField, -22],
   {Real2ndFieldFit[iAdj2nd]}, {iAdj2nd}, x, NormFunction → (Norm[#, 1] &)];
SndRoundsAdj = Round[Table[SndCurrentsAdj[[n]][[2]], {n, 1, Length[SndCurrentsAdj]}], .05];
SndCurrentsAdjR = {iAdj2nd → SndRoundsAdj[[1]]};
SndCurrentListAdj = {iAdj2nd} /. SndCurrentsAdjR
```

```
{-17.2}
```

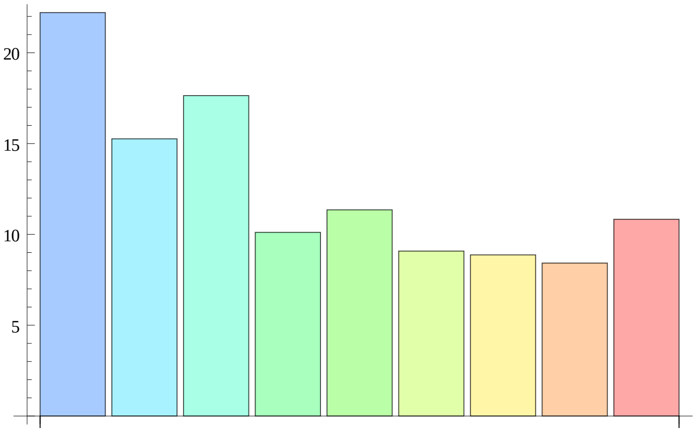
Layer Current Comparisons

```
LayerCurrentGraph = BarChart[{currentList[[1 ;; 9]]}];
LayerCurrentAdjGraph = BarChart[{LayerCurrentListAdj}];
LayerCurrentCompGraph = BarChart[{LayerCurrentListAdj - currentList[[1 ;; 9]]}];
Original Layer Currents
Show[LayerCurrentGraph]
Modified Layer Currents
Show[LayerCurrentAdjGraph]
Effective Difference
Show[LayerCurrentCompGraph]
```

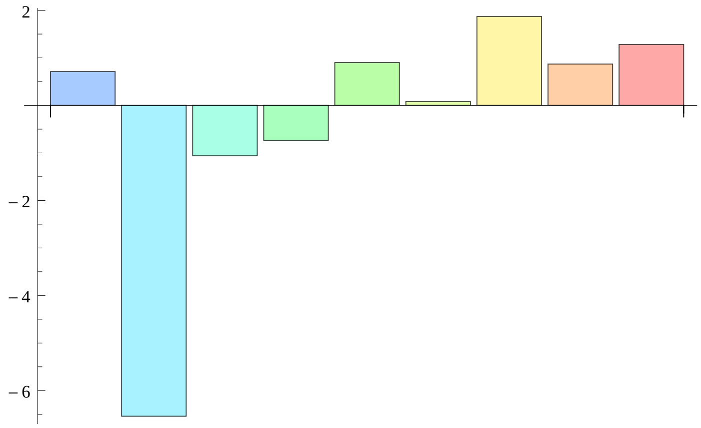
```
Currents Layer Original
```



```
Currents Layer Modified
```



```
Difference Effective
```



Adjusted Current Profile Comparisons

- ## Amp Fields by Adjusted Currents

```
RealLayerListReAdj= BaseLayerList* LayerCurrentListAdj;
RealOver1ReAdj= RealOver1* Over1CurrentListAdj[[1]];
RealOver2ReAdj= RealOver2* Over2CurrentListAdj[[1]];
RealOver3ReAdj= RealOver3* Over3CurrentListAdj[[1]];
Real2ndReAdj= Real2nd* SndCurrentListAdj[[1]];
```

- ## Sort Adjusted B Data

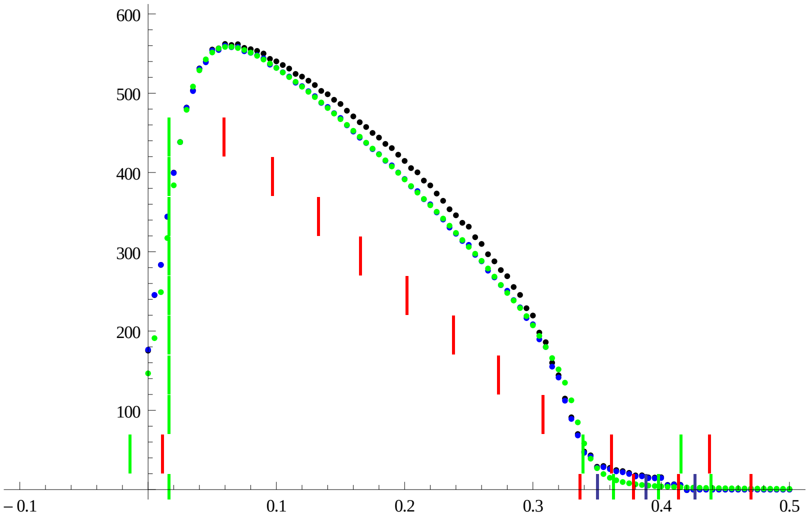
```
RealLayerFieldReAdjList= Table[Table[
    {.005 * z - .005, RealLayerListReAdj[[n]][[z]]}, {z, 1, Length[BaseLayerList[[1]]]}], {n, 1, 9}];
RealLayerFieldReAdjFctnList= Table[Interpolation[RealLayerFieldReAdjList[[n]]], {n, 1, 9}];

RealLayerFieldReAdj=
  Table[{z, Sum[RealLayerFieldReAdjFctnList[[n]][z], {n, 1, Length[RealLayerFieldReAdjFctnList]}]},
   {z, 0, .5, .005}];

RealOver1FieldReAdj= Table[{RealZOver1[[z]], RealOver1ReAdj[[z]]}, {z, 1, Length[RealZOver1]}];
RealOver2FieldReAdj= Table[{RealZOver2[[z]], RealOver2ReAdj[[z]]}, {z, 1, Length[RealZOver2]}];
RealOver3FieldReAdj= Table[{RealZOver3[[z]], RealOver3ReAdj[[z]]}, {z, 1, Length[RealZOver3]}];
Real2ndFieldReAdj= Table[{RealZ2nd[[z]], Real2ndReAdj[[z]]}, {z, 1, Length[RealZ2nd]}];
```

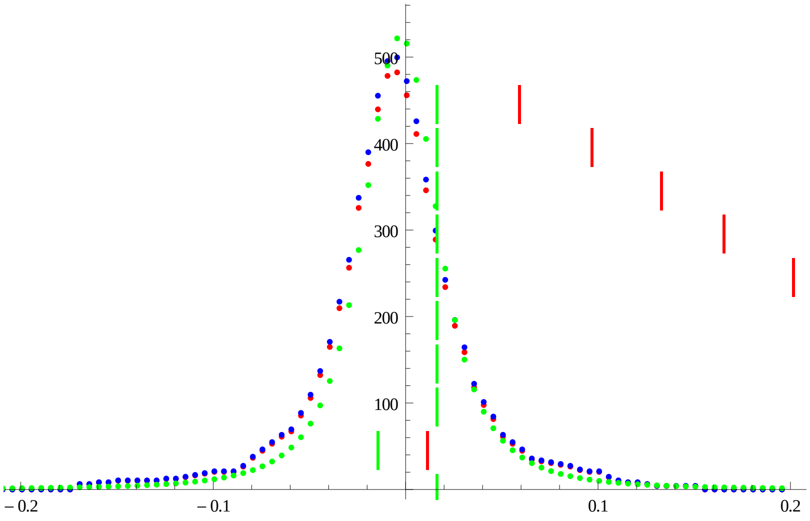
- ## Setup Adjusted Layer Plots

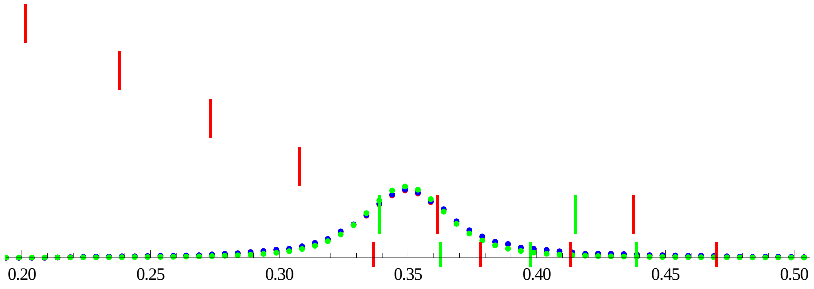
```
RealLayerFieldPlotReAdj= ListPlot[RealLayerFieldReAdj, PlotStyle→ Blue];

RealOver1FieldPlotReAdj= ListPlot[RealOver1FieldReAdj, PlotStyle→ Blue];
RealOver2FieldPlotReAdj= ListPlot[RealOver2FieldReAdj, PlotStyle→ Blue];
RealOver3FieldPlotReAdj= ListPlot[RealOver3FieldReAdj, PlotStyle→ Blue];
Real2ndFieldPlotReAdj= ListPlot[Real2ndFieldReAdj, PlotStyle→ Blue];
```

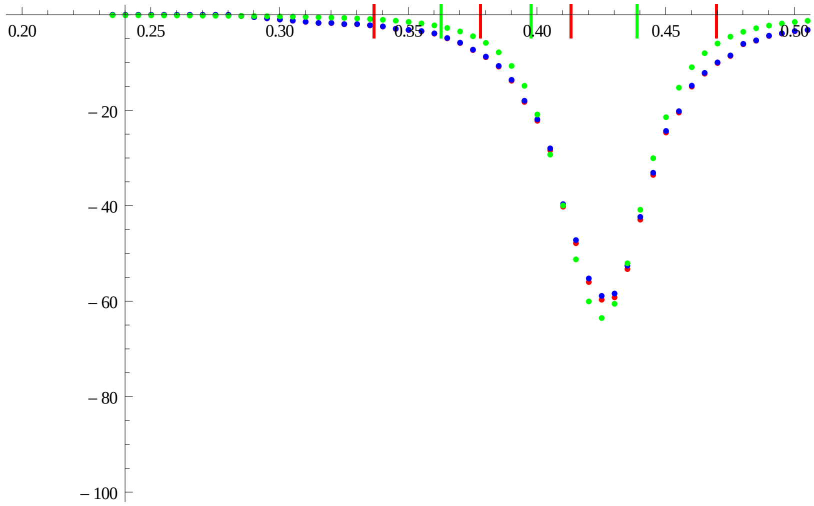
■ Display New Plot Comparison

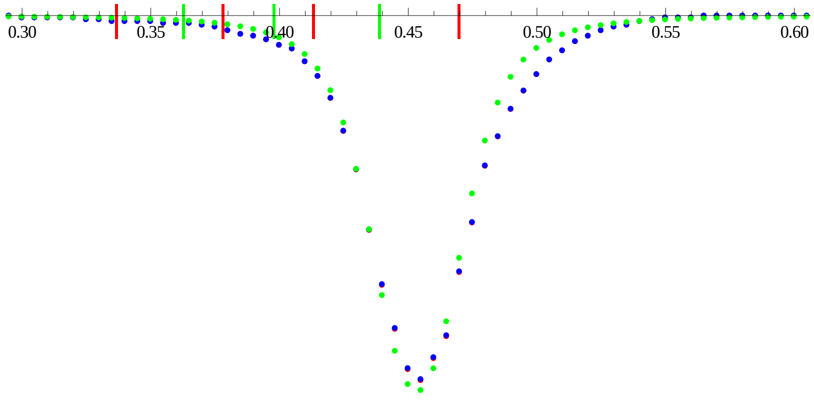
```
Show[RealLayerFieldPlot, RealLayerFieldPlotReAdj, TheoryLayerFieldPlot,
 coilPositions1, coilPositions2, transverseBounds, PlotRange → {{-.1, .5}, {0, 600}}]
Show[RealOver1FieldPlot, RealOver1FieldPlotReAdj, TheoryOver1FieldPlot,
 coilPositions1, coilPositions2, PlotRange → {{-.2, .2}, {0, 550}}]
Show[RealOver2FieldPlot, RealOver2FieldPlotReAdj, TheoryOver2FieldPlot,
 coilPositions1, coilPositions2, PlotRange → {{.2, .5}, {0, 500}}]
Show[RealOver3FieldPlot, RealOver3FieldPlotReAdj, TheoryOver3FieldPlot,
 coilPositions1, coilPositions2, PlotRange → {{.2, .5}, {0, -100}}]
Show[Real2ndFieldPlot, Real2ndFieldPlotReAdj, Theory2ndFieldPlot,
 coilPositions1, coilPositions2, PlotRange → {{.3, .6}, {0, -300}}]
```





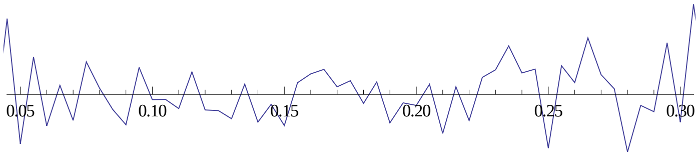

0.20
0.25
0.30
0.35
0.40
0.45
0.50


0.20
0.25
0.30
0.35
0.40
0.45
0.50
– 20
– 40
– 60
– 80
– 100


0.30
0.35
0.40
0.45
0.50
0.55
0.60

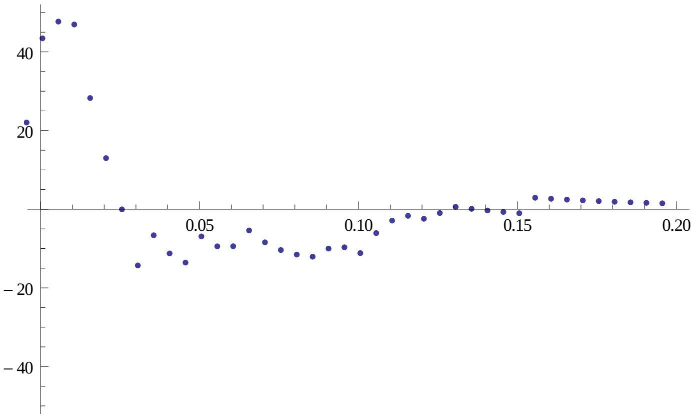
■ New Residual Comparison

```
RealLayerResidualsReAdj = Table[{z * .005 - .005,
    (Flatten[TheoryLayerField][[2 ;; ;; 2]] - Flatten[RealLayerFieldReAdj][[2 ;; ;; 2]])[[z]]},
   {z, 1, Length[RealLayerFieldReAdj]}];
RealLayerResidualsReAdjPlot = ListPlot[RealLayerResidualsReAdj, Joined → True];
Show[RealLayerResidualsReAdjPlot, PlotRange → {{.05, .3}, {-10, 10}}]
Max[Abs[Drop[Drop[RealLayerResidualsReAdj, 10], -40]]]
```



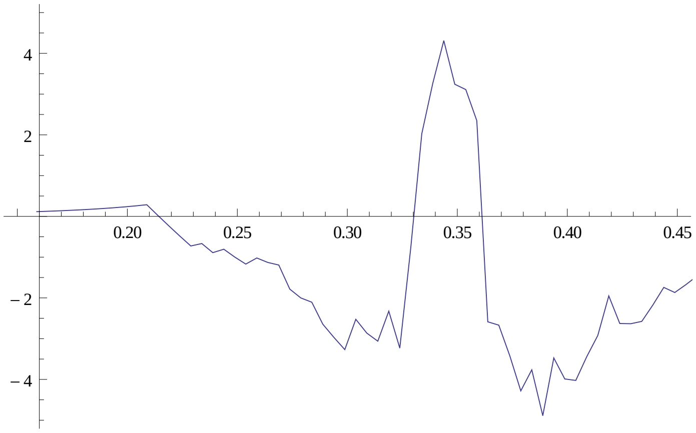
```
2.81385
```

```
RealOver1ResidualsReAdj = Table[{RealZOver1[[z]],
    (Flatten[TheoryOver1Field][[2 ;; ;; 2]] - Flatten[RealOver1FieldReAdj][[2 ;; ;; 2]])[[z]]},
   {z, 1, Length[RealZOver1]}];
RealOver1ResidualsReAdjPlot = ListPlot[RealOver1ResidualsReAdj];
Show[RealOver1ResidualsReAdjPlot, PlotRange → {{0, .2}, {-50, 50}}]
Max[Abs[RealOver1ResidualsReAdj[[Flatten[Position[Flatten[RealZOver1], _ ? Positive]]]]]]
```



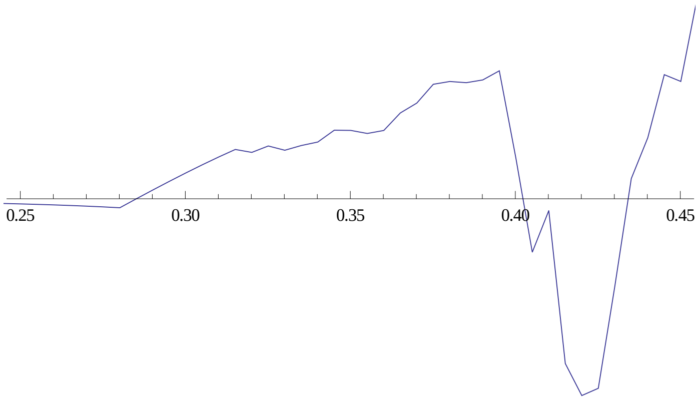
```
47.6957
```

```
RealOver2ResidualsReAdj = Table[{RealZOver2[[z]],
    (Flatten[TheoryOver2Field][[2 ;; ;; 2]] - Flatten[RealOver2FieldReAdj][[2 ;; ;; 2]])[[z]]},
   {z, 1, Length[RealZOver2]}];
RealOver2ResidualsReAdjPlot = ListPlot[RealOver2ResidualsReAdj, Joined → True];
Show[RealOver2ResidualsReAdjPlot, PlotRange → {{.15, .45}, {-5, 5}}]
Max[Abs[Drop[RealOver2ResidualsReAdj, -24]]]
```



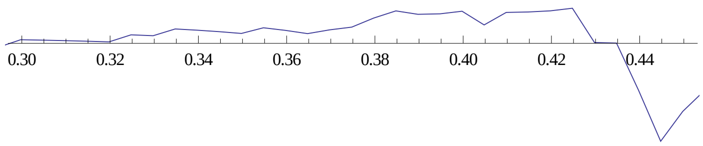
```
4.88938
```

```
RealOver3ResidualsReAdj = Table[{RealZOver3[[z]],
    (Flatten[TheoryOver3Field][[2 ;; ;; 2]] - Flatten[RealOver3FieldReAdj][[2 ;; ;; 2]])[[z]]},
   {z, 1, Length[RealZOver3]}];
RealOver3ResidualsReAdjPlot = ListPlot[RealOver3ResidualsReAdj, Joined → True];
Show[RealOver3ResidualsReAdjPlot, PlotRange → {{.25, .45}, {-5, 5}}]
Max[Abs[Drop[RealOver3ResidualsReAdj, -41]]]
```



```
4.82153
```

```
Real2ndResidualsReAdj = Table[
   {RealZ2nd[[z]], (Flatten[Theory2ndField][[2 ;; ;; 2]] - Flatten[Real2ndFieldReAdj][[2 ;; ;; 2]])[[
     z]]}, {z, 1, Length[RealZ2nd]}];
Real2ndResidualsReAdjPlot = ListPlot[Real2ndResidualsReAdj, Joined → True];
Show[Real2ndResidualsReAdjPlot, PlotRange → {{.3, .45}, {-30, 30}}]
Max[Abs[Drop[Real2ndResidualsReAdj, -41]]]
```



```
6.90569
```

Total Field Profile

■ Theory Profile

```
FullTheoryProfile =
  Table[{z, (iLayer1 * BLayer1[z] + iLayer2 * BLayer2[z] + iLayer3 * BLayer3[z] + iLayer4 * BLayer4[z] +
       iLayer5 * BLayer5[z] + iLayer6 * BLayer6[z] + iLayer7 * BLayer7[z] + iLayer8 * BLayer8[z] +
       iLayer9 * BLayer9[z] + i2nd * B2nd[z] + iCross1 * BCross1[z] + iCross2 * BCross2[z] +
       iOver1 * BOver1[z] + iOver2 * BOver2[z] + iOver3 * BOver3[z]) /. currentsR}, {z, 0, .5, .005}];
FullTheoryProfilePlot = ListPlot[FullTheoryProfile, PlotStyle → Green];


theoryPointShort = Table[{z, bz[z]}, {z, 0, .45, .001}]; (* field to be fitted *)
theoryPointShortPlot = ListPlot[theoryPointShort, PlotStyle → Green];
theoryPoint = Table[{z, bz[z]}, {z, 0, .5, .001}];
theoryPointPlot = ListPlot[theoryPoint, PlotStyle → Green];
```

■ Setup Interpolation Fctns

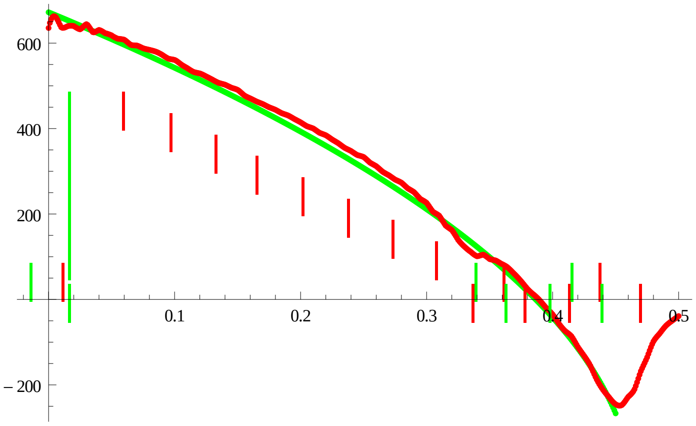
```
RealLayerFieldFctn = Interpolation[RealLayerField];
RealOver1FieldFctn = Interpolation[RealOver1Field];
RealOver2FieldFctn = Interpolation[RealOver2Field];
RealOver3FieldFctn = Interpolation[RealOver3Field];
Real2ndFieldFctn = Interpolation[Real2ndField];
```

▪ Plot of the Raw Profile

```
RawProfile = Table[{z, RealOver1FieldFct[z] + RealLayerFieldFct[z] +
     RealOver2FieldFct[z] + RealOver3FieldFct[z] + Real2ndFieldFct[z] +
     currentList[[11]] * BCross1[z] + currentList[[12]] * BCross2[z]}, {z, 0, .5, .001}];
RawProfilePlot = ListPlot[RawProfile, PlotStyle → Red];
```

```
InterpolatingFunction::dmval:
  Input value {0.} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫

InterpolatingFunction::dmval:
  Input value {0.} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫

InterpolatingFunction::dmval:
  Input value {0.} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫

General::stop : Further output of InterpolatingFunction::dmval will be suppressed during this calculation. ≫
```

```
Show[theoryPointShortPlot, RawProfilePlot, coilPositions1, coilPositions2, PlotRange → All]
```



▪ Adjust Currents With Refittings

```
PlotShiftLayers = SlowerStart - .219;
(*for tuning the coil positions where possible, but weary of physical constraints*)
PlotShiftOver1 = -WidthOver1 + flange - .005 + RInc - .2046;
PlotShiftOver2 = .35 - WidthOver2 / 2 - .1875;
PlotShiftOver3 = .35 + WidthOver2 / 2 - .1275;
PlotShift2nd = StartofSecondPart - .203;

(*PlotShiftLayers = SlowerStart - .2188;
(*measuring distance was ~12 centimeters from layer 1 start*)
PlotShiftOver1 = -WidthOver1+flange-.005+RInc - .205;
PlotShiftOver2 = .35-WidthOver2/2 -.18;
PlotShiftOver3 = .35+WidthOver2/2 - .126;
PlotShift2nd = StartofSecondPart- .204(*.182*);*)


RealZLayers = .005 * Table[z, {z, 0, Length[RealLayer1] - 2}] + PlotShiftLayers;
RealZOver1 = .005 * Table[z, {z, 0, Length[RealOver1] - 2}] + PlotShiftOver1;
RealZOver2 = .005 * Table[z, {z, 0, Length[RealOver2] - 2}] + PlotShiftOver2;
RealZOver3 = .005 * Table[z, {z, 0, Length[RealOver3] - 2}] + PlotShiftOver3;
```

```
RealZ2nd = .005 * Table[z, {z, 0, Length[Real2nd] - 2}] + PlotShift2nd;

RealFieldFit[{iAdjLayer1_, iAdjLayer2_, iAdjLayer3_,
    iAdjLayer4_, iAdjLayer5_, iAdjLayer6_, iAdjLayer7_, iAdjLayer8_, iAdjLayer9_,
    iAdj2nd_, iAdjCross1_, iAdjCross2_, iAdjOver1_, iAdjOver2_, iAdjOver3_}] :=
  Interpolation[Table[{.005 * z - .005, BaseLayerList[[1]][[z]] * iAdjLayer1 +
       BaseLayerList[[2]][[z]] * iAdjLayer2 + BaseLayerList[[3]][[z]] * iAdjLayer3 +
       BaseLayerList[[4]][[z]] * iAdjLayer4 + BaseLayerList[[5]][[z]] * iAdjLayer5 +
       BaseLayerList[[6]][[z]] * iAdjLayer6 + BaseLayerList[[7]][[z]] * iAdjLayer7 +
       BaseLayerList[[8]][[z]] * iAdjLayer8 + BaseLayerList[[9]][[z]] * iAdjLayer9},
     {z, 1, Length[BaseLayerList[[1]]]}], x] + Interpolation[
    Table[{RealZOver1[[z]], iAdjOver1 * RealOver1[[z]]}, {z, 1, Length[RealZOver1]}], x] +
   Interpolation[Table[{RealZOver2[[z]], iAdjOver2 * RealOver2[[z]]},
     {z, 1, Length[RealZOver2]}], x] + Interpolation[
    Table[{RealZOver3[[z]], iAdjOver3 * RealOver3[[z]]}, {z, 1, Length[RealZOver3]}], x] +
   Interpolation[Table[{RealZ2nd[[z]], iAdj2nd * Real2nd[[z]]}, {z, 1, Length[RealZ2nd]}], x] +
   iAdjCross1 * BCross1[x] + iAdjCross2 * BCross2[x]
   (* Interpolation[Table[{z, iAdjCross1 * BCross1[z] + iAdjCross2 * BCross2[z]}, {z, 0, .5, .005}], x] *);

currentListAdj = Flatten[{LayerCurrentListAdj, SndCurrentListAdj, currentList[[11]],
    currentList[[12]], Over1CurrentListAdj, Over2CurrentListAdj, Over3CurrentListAdj}];
currentListH = currentListAdj + (Abs[currentListAdj] * 1.24 - Abs[currentListAdj]);
currentListL = currentListAdj + (Abs[currentListAdj] * .8 - Abs[currentListAdj]);

Clear[iFullAdjLayer1, iFullAdjLayer2, iFullAdjLayer3, iFullAdjLayer4, iFullAdjLayer5,
 iFullAdjLayer6, iFullAdjLayer7, iFullAdjLayer8, iFullAdjLayer9, iFullAdj2nd,
 iFullAdjCross1, iFullAdjCross2, iFullAdjOver1, iFullAdjOver2, iFullAdjOver3]
FullCurrentsAdj = FindFit[Drop[Drop[theoryPointShort, 5], -5],
   {RealFieldFit[{iFullAdjLayer1, iFullAdjLayer2, iFullAdjLayer3, iFullAdjLayer4, iFullAdjLayer5,
      iFullAdjLayer6, iFullAdjLayer7, iFullAdjLayer8, iFullAdjLayer9, iFullAdj2nd,
      iFullAdjCross1, iFullAdjCross2, iFullAdjOver1, iFullAdjOver2, iFullAdjOver3}],
    {(currentListH[[1]]) > iFullAdjLayer1 > (currentListL[[1]]),
     (currentListH[[2]]) > iFullAdjLayer2 > (currentListL[[2]] + 5.3),
     (currentListH[[3]]) > iFullAdjLayer3 > (currentListL[[3]]),
     (currentListH[[4]]) > iFullAdjLayer4 > (currentListL[[4]]),
     (currentListH[[5]] - 3.5) > iFullAdjLayer5 > (currentListL[[5]]),
     (currentListH[[6]]) > iFullAdjLayer6 > (currentListL[[6]]),
     (currentListH[[7]]) > iFullAdjLayer7 > (currentListL[[7]]),
     (currentListH[[8]]) > iFullAdjLayer8 > (currentListL[[8]]),
     (currentListH[[9]] - 3) > iFullAdjLayer9 > (currentListL[[9]] ),
     (currentListH[[13]]) > iFullAdjOver1 > (currentListL[[13]] ),
     (currentListH[[14]]) > iFullAdjOver2 > (currentListL[[14]]),
     (currentListH[[15]]) > iFullAdjOver3 > (currentListL[[15]]),
     (currentListH[[10]]) > iFullAdj2nd > (currentListL[[10]]),
     (currentListH[[11]]) > iFullAdjCross1 > (currentListL[[11]]),
     (currentListH[[12]]) > iFullAdjCross2 > (currentListL[[12]])}},
   {iFullAdjLayer1, iFullAdjLayer2, iFullAdjLayer3, iFullAdjLayer4, iFullAdjLayer5, iFullAdjLayer6,
    iFullAdjLayer7, iFullAdjLayer8, iFullAdjLayer9, iFullAdj2nd, iFullAdjCross1, iFullAdjCross2,
    iFullAdjOver1, iFullAdjOver2, iFullAdjOver3}, x, NormFunction → (Norm[#, 4] &)];
FullRoundsAdj = Round[Table[FullCurrentsAdj[[n]][[2]], {n, 1, Length[FullCurrentsAdj]}], .01];
FullCurrentsAdjR = {iFullAdjLayer1 → FullRoundsAdj[[1]],
   iFullAdjLayer2 → FullRoundsAdj[[2]], iFullAdjLayer3 → FullRoundsAdj[[3]],
   iFullAdjLayer4 → FullRoundsAdj[[4]], iFullAdjLayer5 → FullRoundsAdj[[5]],
   iFullAdjLayer6→ FullRoundsAdj[[6]], iFullAdjLayer7→ FullRoundsAdj[[7]],
   iFullAdjLayer8→ FullRoundsAdj[[8]], iFullAdjLayer9→ FullRoundsAdj[[9]],
   iFullAdj2nd→ FullRoundsAdj[[10]], iFullAdjCross1→ FullRoundsAdj[[11]],
   iFullAdjCross2→ FullRoundsAdj[[12]], iFullAdjOver1→ FullRoundsAdj[[13]],
   iFullAdjOver2→ FullRoundsAdj[[14]], iFullAdjOver3→ FullRoundsAdj[[15]]};
NEW CURRENTS
FullCurrentListAdj=
 {iFullAdjLayer1, iFullAdjLayer2, iFullAdjLayer3, iFullAdjLayer4, iFullAdjLayer5,
   iFullAdjLayer6, iFullAdjLayer7, iFullAdjLayer8, iFullAdjLayer9, iFullAdj2nd, iFullAdjCross1,
   iFullAdjCross2, iFullAdjOver1, iFullAdjOver2, iFullAdjOver3} /. FullCurrentsAdjR
OLD ADJUSTED CURRENTS
currentListAdj
ORIGINAL CURRENTS
currentList
PERCENT CHANGE FROM ORIGINAL
(FullCurrentListAdj - currentList) / currentList * 100 (*percentage change from original recepie*)
```

```
InterpolatingFunction::dmval:
  Input value {0.005} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫

InterpolatingFunction::dmval:
  Input value {0.005} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫

InterpolatingFunction::dmval:
  Input value {0.005} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫

General::stop : Further output of InterpolatingFunction::dmval will be suppressed during this calculation. ≫
```

```
CURRENTS NEW

{20.71, 17.51, 16.58, 12.36, 10.57, 9.58, 7.1, 7.87, 10.43, -17.72, 0.99, -1.72, 21.12, 5.2, -3.91}

ADJUSTED CURRENTS OLD

{22.21, 15.26, 17.64, 10.11, 11.35, 9.08, 8.87, 8.42, 10.83, -17.2, 1., -1.75, 21.08, 4.41, -3.65}

CURRENTS ORIGINAL

{21.5, 21.8, 18.7, 10.85, 10.45, 9., 7., 7.55, 9.55, -17.25, 1., -1.75, 20.35, 4.35, -3.7}

CHANGE FROM ORIGINAL PERCENT

{-3.67442, -19.6789, -11.3369, 13.9171, 1.14833, 6.44444, 1.42857,
 4.23841, 9.21466, 2.72464, -1., -1.71429, 3.78378, 19.5402, 5.67568}
```

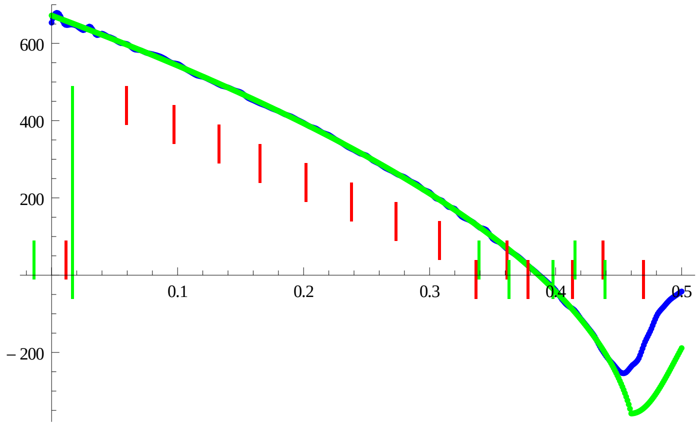
■ Plot Adj Full Profile and Residuals, as well as Final Statistics

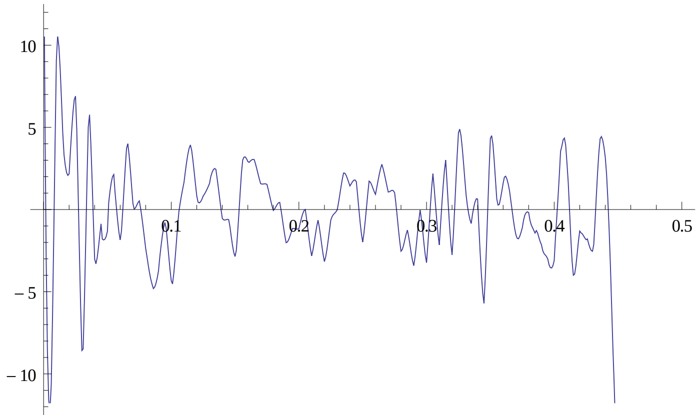
```
RealLayerListAdjF= BaseLayerList* FullCurrentListAdj[[1 ;; 9]];
RealOver1AdjF = RealOver1* FullCurrentListAdj[[13]];
RealOver2AdjF = RealOver2* FullCurrentListAdj[[14]];
RealOver3AdjF = RealOver3* FullCurrentListAdj[[15]];
Real2ndAdjF = Real2nd * FullCurrentListAdj[[10]];

RealLayerFieldAdjFList= Table[Table[
    {.005 * z - .005, RealLayerListAdjF[[n]][[z]]}, {z, 1, Length[BaseLayerList[[1]]]}], {n, 1, 9}];
RealLayerFieldAdjFFctnList= Table[Interpolation[RealLayerFieldAdjFList[[n]]], {n, 1, 9}];

RealLayerFieldAdjF=
  Table[{z, Sum[RealLayerFieldAdjFFctnList[[n]][z], {n, 1, Length[RealLayerFieldAdjFFctnList]}],
   {z, 0, .5, .001}];
```

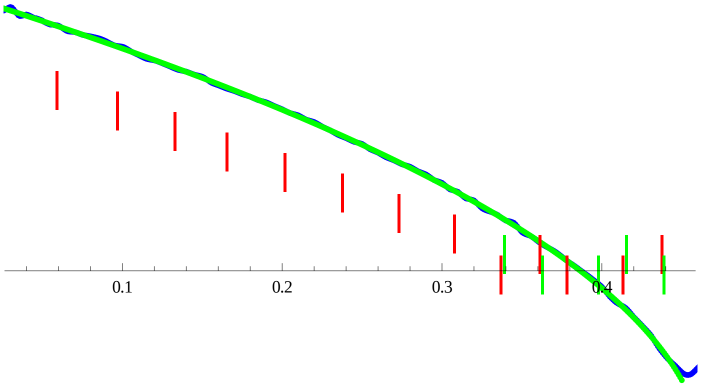
```
RealOver1FieldAdjF= Table[{RealZOver1[[z]], RealOver1AdjF[[z]]}, {z, 1, Length[RealZOver1]}];
RealOver2FieldAdjF= Table[{RealZOver2[[z]], RealOver2AdjF[[z]]}, {z, 1, Length[RealZOver2]}];
RealOver3FieldAdjF= Table[{RealZOver3[[z]], RealOver3AdjF[[z]]}, {z, 1, Length[RealZOver3]}];
Real2ndFieldAdjF= Table[{RealZ2nd[[z]], Real2ndAdjF[[z]]}, {z, 1, Length[RealZ2nd]}];
RealLayerFieldAdjFFctn= Interpolation[RealLayerFieldAdjF];
RealOver1FieldAdjFFctn= Interpolation[RealOver1FieldAdjF];
RealOver2FieldAdjFFctn= Interpolation[RealOver2FieldAdjF];
RealOver3FieldAdjFFctn= Interpolation[RealOver3FieldAdjF];
Real2ndFieldAdjFFctn= Interpolation[Real2ndFieldAdjF];
RealOver1FieldAdjFPlot= ListPlot[Table[{z, RealOver1FieldAdjFFctn[z]}, {z, -.3, .3, .001}]];
RealOver2FieldAdjFPlot= ListPlot[Table[{z, RealOver2FieldAdjFFctn[z]}, {z, .2, .5, .001}]];
RealOver3FieldAdjFPlot= ListPlot[Table[{z, RealOver3FieldAdjFFctn[z]}, {z, .2, .5, .001}]];
Real2ndFieldAdjFPlot= ListPlot[Table[{z, Real2ndFieldAdjFFctn[z]}, {z, .3, .5, .001}]];
RawProfileAdjF=
  Table[{z, RealOver1FieldAdjFFctn[z] + RealLayerFieldAdjFFctn[z] + RealOver2FieldAdjFFctn[z] +
     RealOver3FieldAdjFFctn[z] + Real2ndFieldAdjFFctn[z] + FullCurrentListAdj[[11]] * BCross1[z] +
     FullCurrentListAdj[[12]] * BCross2[z]}, {z, 0, .5, .001}];
RawProfileAdjFPlot= ListPlot[RawProfileAdjF, PlotStyle→ Blue];
RawResidualsAdjF= Flatten[theoryPoint][[2 ;; ;; 2]] - Flatten[RawProfileAdjF][[2 ;; ;; 2]];
RawResidualsAdjFPlot=
  ListPlot[Table[{Flatten[theoryPoint][[1 ;; ;; 2]][[z]], RawResidualsAdjF[[z]]},
    {z, 1, Length[theoryPoint]}], Joined → True];
Show[RawProfileAdjFPlot, theoryPointPlot, coilPositions1, coilPositions2, PlotRange→ All]
Show[RawResidualsAdjFPlot, PlotRange→ {{0, .5}, {-12, 12}}]
Show[RawProfileAdjFPlot, theoryPointShortPlot,
 coilPositions1, coilPositions2, PlotRange→ {{.035, .45}, {-300, 700}}]
Show[RawResidualsAdjFPlot, PlotRange→ {{.035, .45}, {-12, 12}}]

Print[MAX RESIDUAL]
Max[Abs[Drop[Drop[RawResidualsAdjF, -11 * 5], 7 * 5]]]
Print[AVG RESIDUAL]
Avg = Mean[Abs[Drop[Drop[RawResidualsAdjF, -11 * 5], 7 * 5]]]
Print[STANDARD, DEVIATION]
Std = StandardDeviation[Abs[Drop[Drop[RawResidualsAdjF, -11 * 5], 7 * 5]]]
numOut=
  LengthWhile[Sort[Abs[Drop[Drop[RawResidualsAdjF, -11 * 5], 7 * 5]], Greater], # > Avg + 2 * Std &];
Print[PERCENTAGE, OF, POINTS, BEYOND, TWO, SIGMA, DEVIATION]
Avg + 2 * Std;
N[numOut/ Length[Drop[Drop[RawResidualsAdjF, -11 * 5], 7 * 5]] * 100]
```

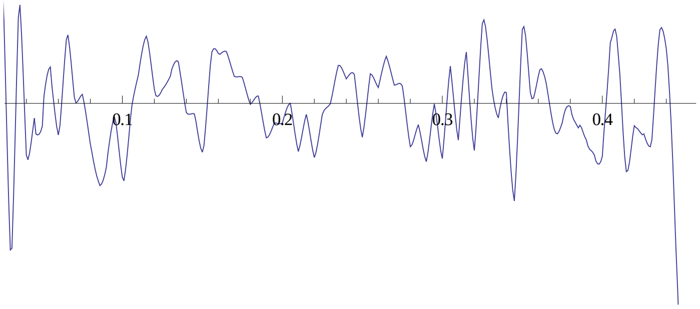
```
InterpolatingFunction::dmval:
  Input value {-0.3} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫

InterpolatingFunction::dmval:
  Input value {-0.299} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫

InterpolatingFunction::dmval:
  Input value {-0.298} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫

General::stop : Further output of InterpolatingFunction::dmval will be suppressed during this calculation. ≫

InterpolatingFunction::dmval:
  Input value {0.2} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫
```

```
InterpolatingFunction::dmval:
  Input value {0.201} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫

InterpolatingFunction::dmval:
  Input value {0.202} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫

General::stop : Further output of InterpolatingFunction::dmval will be suppressed during this calculation. ≫

InterpolatingFunction::dmval:
  Input value {0.} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫

InterpolatingFunction::dmval:
  Input value {0.} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫

InterpolatingFunction::dmval:
  Input value {0.} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫

General::stop : Further output of InterpolatingFunction::dmval will be suppressed during this calculation. ≫
```









```
MAX RESIDUAL

5.86168

AVG RESIDUAL

1.82381

STANDARD DEVIATION

1.23078

PERCENTAGE OF POINTS BEYOND TWO SIGMA DEVIATION

4.62287
```

BIBLIOGRAPHY

[1] Annealed nickel cf gaskets. http://www.lesker.com/newweb/flanges/hardware_cf_gaskets.cfm?pgid=nickel.

[2] ATFI. High temperature paint: Cerablak. http://www.atfinet.com/index.php/products/high-temperature-paint.

[3] Adra Victoria Carr. *Hyperfine Studies of Lithium Vapor Using Saturated Absorption Spectroscopy*. University of Arizona, May 2007. Undergraduate Thesis.

[4] Cotronics Corp. http://cotronics.com/vo/cotr/, 2008.

[5] J. Dalibard and C. Cohen-Tannoudji. Laser cooling below the doppler limit by polarization gradients: Simple theoretical models. *Optical Society of America B*, 6(2023), 1989.

[6] Cornell Center for Materials Research. http://www.ccmr.cornell.edu/.

[7] Michael E. Gehm. *Preparation of an Optically-Trapped Degenerate Fermi Gas of 6Li: Finding the Route to Degeneracy.* PhD thesis, Duke University, 2003.

[8] Mark Kasevich and Steven Chu. Laser cooling below photon recoil with three-level atoms. *Phys. Rev. Lett.*, 69(1741), 1992.

[9] Mark W. Lund. Vapor pressure of the chemical elements. http://www.powerstream.com/vapor-pressure.htm, 2012.

[10] P. Minguzzi, F. Strumia, and P. Violino. Optical pumping of lithium vapour. *Optics Communications*, 1(1):1 – 2, 1969.

[11] Kimball Physics. Internal mounting. http://www.kimballphysics.com/multicf-hardware/mcf-mounting.

[12] Craig J. Sansonetti, C. E. Simien, J. D. Gillaspy, Joseph N. Tan, Samuel M. Brewer, Roger C. Brown, Saijun Wu, and J. V. Porto. Absolute transition frequencies and quantum interference in a frequency comb based measurement of the ^{6,7}Li d lines. *Phys. Rev. Lett.*, 107:023001, Jul 2011.

[13] Craig J. Sansonetti1, C. E. Simien, J. D. Gillaspy, Joseph N. Tan, Samuel M. Brewer, Roger C. Brown, Saijun Wu, and J. V. Porto. Absolute transition

frequencies and quantum interference in a frequency comb based measurement of the 6,7li d lines. *Phys. Rev. Lett.*, 107(023001), 2011.

[14] Marco Schioppo, Nicola Poli, Marco Prevedelli, Stephan Falke, Christian Lisdat, Uwe Sterr, and Guglielmo Maria Tino. A compact and efficient strontium oven for laser-cooling experiments. *Rev. Sci. Instrum.*, 83(103101), 2012.

[15] Widagdo Setiawan. *Engineering Aspects of Ultracold Fermions Machine*. Massachusetts Institute of Technology, May 2007. Undergraduate Thesis.

[16] Claudiu Andrei Stan. *Experiments with Interacting Bose and Fermi Gases*. PhD thesis, Massachusetts Institute of Technology, July 2005.

[17] Thorlabs. Rackbox system. `http://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=1455`.

[18] Thorlabs. *Electro-Optic Amplitude Modulator: Operating Manual*, 15956-d02 rev.b edition, January 2012.