NORTHWESTERN UNIVERSITY

Quantum State Control of Trapped Atomic and Molecular Ions

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Physics

By

Christopher M. Seck

EVANSTON, ILLINOIS

August 2016

ABSTRACT

Quantum State Control of Trapped Atomic and Molecular Ions

Christopher M. Seck

Full quantum control of a molecule would have a significant impact in molecular coherent control (alignment and orientation) and ultracold and quantum chemistry [1, 2, 3, 4, 5], quantum computing and simulation as well as hybrid quantum devices [6, 7, 8, 9, 10, 11], and precision spectroscopy of importance to fundamental physics research [12, 13, 14, 15, 16, 17, 10, 11, 18, 19, 20]. Precision spectroscopy of even simple diatomic molecules offers the possibility of uncovering physics beyond the standard model, specifically time variation of the proton-to-electron mass ratio μ, which is currently constrained by astronomical molecular observations at $\frac{\dot{\mu}}{\mu} = (-7.7 \pm 6.1) \times 10^{-16}$ yr^{-1} [19] and laboratory atomic measurements at $\frac{\dot{\mu}}{\mu} = (0.2 \pm 1.1) \times 10^{-16}$ yr^{-1} [21]. To achieve this level of measurement and to avoid the complications of diatomic structure on traditional spectroscopy methods, molecular quantum logic spectroscopy (mQLS) will be the spectroscopy technique of choice, which demands quantum control of both the spectroscopy and logic ions. Efficient mQLS requires (1) tunable, narrow-linewidth laser

sources; (2) molecular ion state preparation; and (3) motional ground state preparation of the atomic logic and molecular spectroscopy ion pair in at least one motional mode.

We discuss development of in-house external-cavity diode laser (ECDL) systems and improvements to the Libbrecht-Hall circuit, which is a well-known, low-noise current driver for narrow-linewidth diode lasers. An important feature of the circuit is a current limit to protect the laser diode. However, as the current approaches the maximum limit, the noise in the laser current increases dramatically. This behavior is documented and simple circuit modifications to alleviate this issue are explored.

We cool trapped AlH^+ molecules to their ground rotational-vibrational quantum state using an electronically-exciting broadband laser to simultaneously drive cooling resonances from many different rotational levels. Undesired vibrational excitation is avoided because of vibrational-electronic decoupling in AlH^+. We demonstrate rotational cooling on the 140(20) ms timescale from room temperature to $3.8^{+0.9}_{-0.3}$ K, with the ground state population increasing from ∼3% to $95.4^{+1.3}_{-2.1}$%. This cooling technique could be applied to several other neutral and charged molecular species useful for quantum information processing, ultracold chemistry applications, and precision tests of fundamental symmetries.

Since QLS does not require the high gate fidelities usually associated with quantum computation and quantum simulation, it is possible to make simplifying choices in ion species and quantum protocols at the expense of some fidelity. We demonstrate sideband cooling and motional state detection protocols for $^{138}Ba^+$ of sufficient fidelity for QLS without an extremely narrowband laser or the use of a species with hyperfine structure. We use the two $S_{1/2}$ Zeeman sublevels of $^{138}Ba^+$ to Raman sideband cool a single ion to the motional ground state. Because of the small Zeeman splitting, continuous near-resonant

Raman sideband cooling of ${}^{138}\mathrm{Ba}^{+}$ requires only the Doppler cooling lasers and two additional AOMs. Observing the near-resonant Raman optical pumping fluorescence, we extract relevant experimental parameters and estimate a final average motional quantum number $\bar{n} \approx 0.17$. We additionally employ a second, far-off-resonant laser driving Raman π pulses between the two Zeeman sublevels to provide motional state detection for QLS and to confirm the sideband cooling efficiency, measuring a final $\bar{n} = 0.15(6)$.

Acknowledgements

No dissertation of graduate study can be written without acknowledging the years of collaboration and assistance from colleagues. I would like to thank Brian, the current, and the previous members of the group for all of their assistance and discussion over the seven years of work this dissertation discusses. Edward Hohenstein, thank you for your calculations of AlH^+ potential energy surfaces and dipole moments, and Pat Stollenwerk, thank you for your following REMPD cross section calculations to enable the AlH^+ rotational cooling experiment. Chien-Yu Lien, thank you for your help conducting the molecular rotational cooling experiment. Michael Schmitt, thank you for your assistance in binomial confidence intervals outside of the central limit approximation. Yen-Wei Lin, thank you for gracefully gifting me your single-ion apparatus. Matt Dietrich, thank you for guiding me to Python for apparatus control and for being an inspiration during times of dwindling hope. Mark Kokish, thank you for giving me hope that the foundation I have built will be left in good, capable hands. The Dan Steck group, thank you for your collaboration in improving the diode laser current driver. The National Science Foundation, the David and Lucile Packard Foundation, the Air Force Office of Scientific Research, and Northwestern University, thank you for the funding for which this work could not have been accomplished. And Matt Brayley-Berger, Michael Ko, Michael Turchin, and so many others, thank you for your unwavering support and guidance for which I am eternally grateful.

Preface

This dissertation is not a grand tour of some fantastic, new physics. This will be a technical discussion that will describe the building blocks and expertise developed over seven years of dedicated work that will provide the next generation of graduate students the opportunity to make those discoveries upon the building blocks that I have laid with the help of so many others. It is my hope that this will be a reference for them and future students as an example of which to hope for, to strive for, and with hard work and determination to achieve.

I will assume that the reader will have a graduate-level understanding of basic quantum mechanics, ion trap technology including single-ion dynamics, optics and optical systems, analog and digital electronics, molecular structure, and AMO physics experimental techniques. Additionally, some chapters or sections may be inconsistent with terminology, tone, and/or general grammatical structure as this work interleaves and builds upon the published work listed in the vita (F.4.4).

List of abbreviations

ADC: Analog to Digital Converter

AI: Analog Input

AM: Amplitude Modulation

AMO: Atomic, Molecular, and Optical

AO: Analog Output

AOM: Acousto-Optic Modulator

AR: Anti-Reflection

ASE: Amplified Spontaneous Emission

BROC: Broadband Rotational Optical Cooling

BSB: Blue Sideband

CCD: Charge-Coupled Device

CMOS: Complementary Metal-Oxide Semiconductor

CP: Charge Parity

CPM: Critical Phase Matching

CPT: Coherent Population Trapping

CW: Continuous-Wave

DAC: Digital to Analog Converter

DC: Direct Current

DDS: Direct Digital Synthesizer

ECDL: External Cavity Diode Laser

EDM: Electric Dipole Moment

EIT: Electromagnetically Induced Transparency

EMCCD: Electron Multiplying Charge Coupled Device

EMI: Electromagnetic Interference

EOM: Electro-Optic Modulator

ESD: Electro-Static Discharge

FC/APC: Ferrule Connector / Angled Physical Contact

FC/PC: Ferrule Connector / Physical Contact

FCF: Franck-Condon Factor

FHG: Fourth Harmonic Generation

FM: Frequency Modulation

FOR: Far-Off-Resonant

FSK: Frequency Shift Keying

FSR: Free Spectral Range

FWHM: Full Width at Half Maximum

GPIB: General Purpose Interface Bus

GUI: Graphical User Interface

HWP: Half Wave Plate

HV: High Voltage

HVAC: Heating, Ventilation, and Air Conditioning

I/O: Input / Output

IF: Intermediate Frequency

INA: Instrumentation Amplifier

IR: Infrared

LED: Light Emitting Diode

LO: Local Oscillator

LPF: Low Pass Filter

MCP: Micro-Channel Plate

MFD: Mode Field Diameter

MIT: Massachusetts Institute of Technology

MOPA: Master Oscillator Power Amplifier

MOSFET: Metal-Oxide-Semiconductor Field-Effect Transistor

mQLS: Molecular Quantum Logic Spectroscopy

NCPM: Non-Critical Phase Matching

NI: National Instruments

NIST: National Institute of Standards and Technology

NPBS: Non Polarizing Beamsplitter (50:50)

NR: Near Resonant

OPO: Optical Parametric Oscillator

PBS: Polarizing Beamsplitter

PC: Personal Computer

PD: Photodiode

PDH: Pound-Drever-Hall

PDM: Permanent Dipole Moment

PES: Potential Energy Surface

PI: Proportional Integral

PID: Proportional Integral Derivative

PLL: Phase Locked Loop

PM: Polarization Maintaining

PMF: Polarization Maintaining Single Mode Fiber

PMT: Photo Multiplier Tube

PP: Peak to Peak

PZT: Lead Zirconate Titanate, or piezo

QLS: Quantum Logic Spectroscopy

QWP: Quarter Wave Plate

RAP: Rapid Adiabatic Passage

REMPD: Resonance Enhanced Multi Photon Dissociation

REMPI: Resonance Enhanced Multi Photon Ionization

RF: Radio Frequency

RGA: Residual Gas Analyzer

RMS: Root Mean Square

ROI: Region Of Interest

RSB: Red Sideband

RSC: Raman Sideband Cooling

SHCS: Socket Head Cap Screw

SHG: Second Harmonic Generation

SM: Single Mode

SRS: Stanford Research Systems

STIRAP: Stimulated Raman Adiabatic Passage

TCP/IP: Transmission Control Protocol / Internet Protocol

TDM: Transition Dipole Moment

TEC: Thermo Electric Cooler

THG: Third Harmonic Generation

TIA: Trans-Impedance Amplifier

TOF: Time-Of-Flight

TOFMS: Time-Of-Flight Mass Spectrometer, -metry

TTL: Transistor-Transistor Logic

USB: Universal Serial Bus

UV: Ultraviolet

VA-BROC: Vibrationally-Assisted Broadband Rotational Optical Cooling

VCO: Voltage Controlled Oscillator

Authors:

CMS: Christopher M. Seck (author)

CYL: Chien-Yu Lien

EGH: Edward G. Hohenstein

JCM: Joseph Cordero-Mercado

JHVN: Jason H. V. Nguyen

JTM: Jonathan Trossman

MGK: Mark G. Kokish

MRD: Matthew R. Dietrich

MZI: Marcin Ziolkowski

PRS: Pinrui Shen

VNR: Vaishnavi Rajagopal

YWL: Yen-Wei Lin

Dedication

To my family, old and new.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

Quantum control of atomic species revolutionized the field of atomic, molecular, and optical physics. Doppler cooling [22, 23, 24] enabled the invention of the magneto-optical trap [25] and ion traps [26] capable of cooling to millikelvin temperatures with ions forming so-called Coulomb crystals [27, 28, 29, 30, 31]. Further work in the following decades resulted in Bose-Einstein condensation [32, 33, 34, 35] via sub-Doppler cooling, widely used along with ultracold neutral atoms in sensing [36, 37, 35, 38] and fundamental science applications [34, 39, 35]; precision magnetometers with subfemtotesla sensitivity [40]; and atomic clocks with fractional frequency uncertainties on the order of 1 part in 10^{-17} [41, 42, 43]. The field of quantum optics is similarly advanced [44, 45].

Nearly all of these results have been performed with atoms and quantum optics; notably missing from the work above are *molecular* results. If the quantum control toolbox developed for atomics could be extended to include molecules, the implications for the field could be profound. However, state manipulation of molecules requires control of the rotational degree of freedom, which has historically been a significant challenge. If full molecular quantum control could be realized, it would have a significant impact in quantum computing and simulation as well as hybrid quantum devices [6, 7, 8, 9], molecular coherent control (alignment and orientation) and ultracold and quantum chemistry [1, 2, 3, 4, 5], and precision spectroscopy of importance to fundamental physics research [12, 13, 14, 15, 16, 17, 18].

Although the Standard Model of physics [46] has demonstrated and continues to demonstrate significant success in predicting experimental results [47], it cannot account for some experimental results, i.e. neutrino oscillations and mass in the minimal Standard Model [48, 49], the strong CP problem [50], the matter-antimatter asymmetry of the universe as well as the dark matter and dark energy problems [51], and the 3.2σ deviation between experimental results and the Standard model prediction of the magnetic dipole moment of the muon [52]. Many extensions to and theories beyond the Standard Model have been proposed with some theories making predictions of measurable values that are either non-zero or orders of magnitude larger than the Standard Model prediction, which can be experimentally investigated in atomic and molecular systems, i.e. parity violation [53, 13], an electron EDM [54, 55, 17, 18], and time-variation of fundamental constants such as the fine-structure constant α and the proton-to-electron mass ratio $\mu = \frac{m_p}{m_e}$ [56, 57, 58].

These experimental studies, even with null results and an experimental limit set on a particular value, can begin to exclude some extensions to and theories beyond the Standard Model. For example, many of these theories predict an electron EDM value d_e of 10^{-26} to 10^{-30} $e$cm [55, 17, 18, 59]. However, substantial experimental efforts by the ACME collaboration have experimentally constrained the value of $|d_e| < 8.7 \times 10^{-29}$ $e$cm to the 90% confidence limit [18, 59], which excludes some theories (Fig. 1.1).

To experimentally investigate time variation of fundamental constants using molecules, promising due to their sensitivity to α and μ time-variation [14, 60, 16, 61, 62, 63], two approaches can be used: either (1) astrophysical spectroscopic observations where the look-back time can be $\sim 10^9$ years, or (2) current-epoch laboratory precision spectroscopy

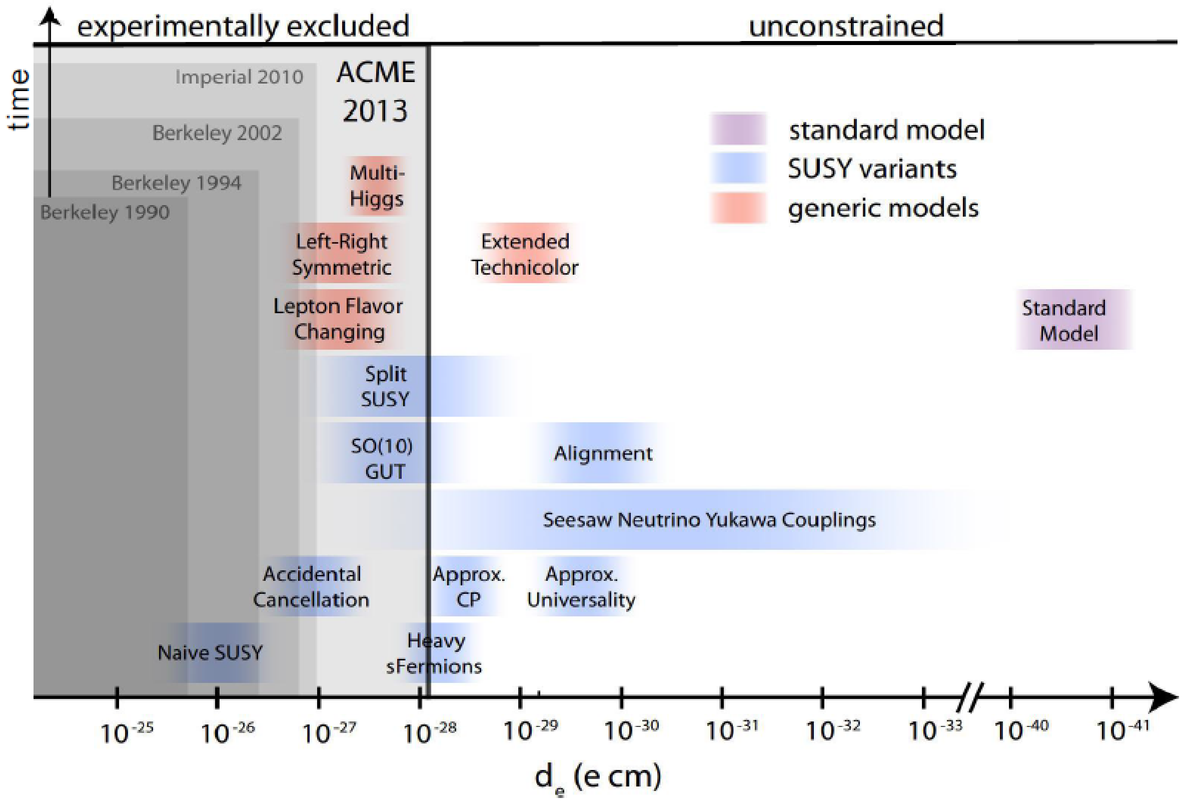


Figure 1.1. Predictions of electron EDM value of several Standard Model extensions and experimental limits set by several experiments including molecular results [55, 17, 18, 59], which exclude several theories. Figure from Ref. [59].

experiments. Multiple astrophysical investigations have been performed [64, 65, 66, 67, 19, 20] with the current best time variation value of $\frac{\dot{\mu}}{\mu} = (-7.7 \pm 6.1) \times 10^{-16}$ yr^{-1} set by H_2 lines [19] .

The second approach, present-epoch laboratory investigations, are promising to investigate time-variation in a well-controlled laboratory setting [60, 68]. The best laboratory limit on the time variation of μ using molecules is $\frac{\dot{\mu}}{\mu} = (-3.8 \pm 5.6) \times 10^{-14}$ yr^{-1} in SF_6 [15]. Additionally, both time-variation of α and μ can be constrained by comparing the ratios of atomic clock frequencies of different species and clock transitions [69, 21]. These

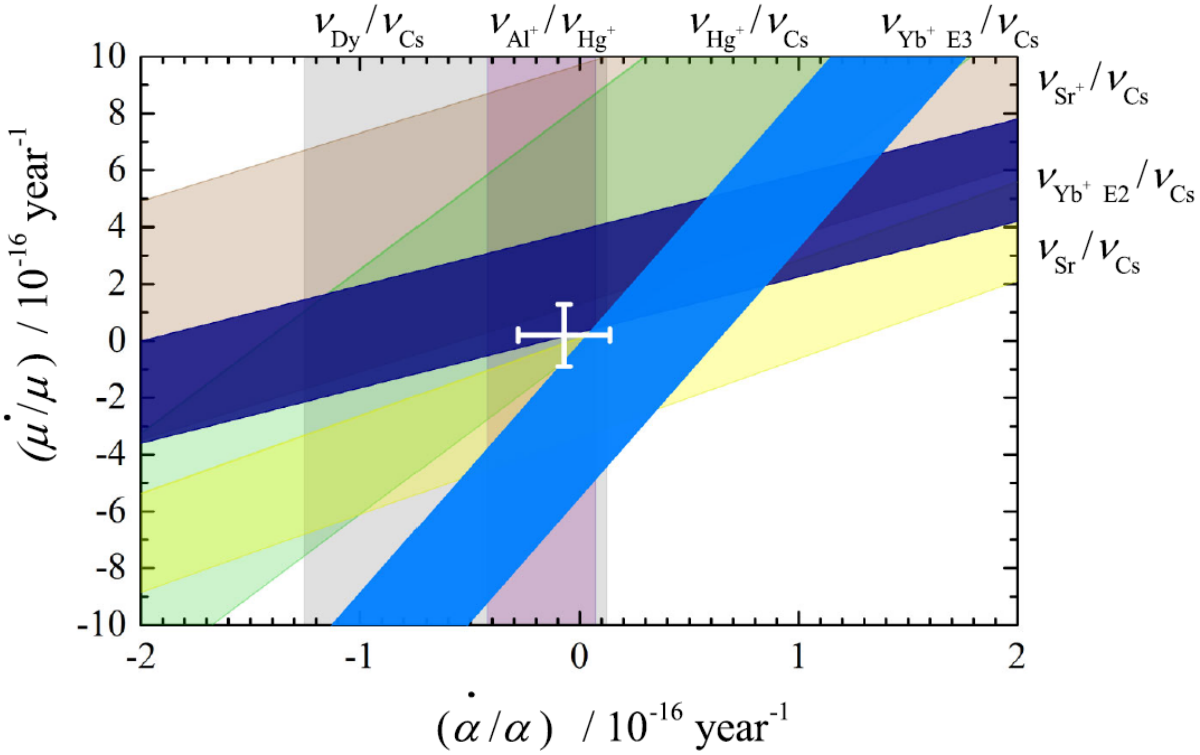


Figure 1.2. Constraints placed on the time-variation of α and μ using atomic clock frequency measurement ratios (shaded regions). The white error bars represent the constrained time variations of α and μ given all data presented. Figure from Ref. [21].

measurements constrain $\frac{\dot{\mu}}{\mu} = (0.2 \pm 1.1) \times 10^{-16}$ yr^{-1} (Fig. 1.2) [21], but extracting the time-variation of μ from experimental atomic optical transition frequencies is non-trivial [69, 21]; the μ dependence arises from the hyperfine structure transition frequency having a functional dependence on $\frac{\mu_N}{\mu_B} \propto \frac{1}{\mu}$, where μ_N (μ_B) is the nuclear (Bohr) magneton, together with a dependence on the strong-interaction parameter X_q denoting the ratio between the average quark mass and the quantum chromodynamic scale Λ_{QCD} [70, 71, 69].

However, moleculear vibrational energy scales at $\mu^{-\frac{1}{2}}$ and rotational energy scales at μ^{-1}. Therefore, if rovibrational spectroscopy can be performed with accuracy better than the $\sim 10^{-16}$ level, new limits of μ time-variation using a molecular transition can be set. While molecular vibrations and rotations offer higher sensitivities to measure μ time-variation, this additional structure is problematic because rovibrational state control and readout are a challenge. This is caused by a lack of closed optical cycling transitions in molecules since electronic spontaneous emission generally excites the vibrational degree of freedom. Although in a special class of molecules, vibrational excitation is minimized because of an unusual degree of electronic-vibrational decoupling, a vibrationally-exciting electronic spontaneous emission will eventually occur, and the resulting vibrational level(s) must be repumped or allowed to spontaneously decay back into the desired vibrational state(s), which can take $\gtrsim$100 ms to $\sim$10 minutes. Since molecules *generally* cannot be directly Doppler cooled due to electronic-vibrational coupling and vibrational spontaneous relaxation times are typically $\gtrsim$100 ms, both problematic for low experimental systematic shifts and uncertainties, using molecular ions co-trapped with a Doppler-cooled atomic species is a promising system for an experimental search for time-variation of μ.

A similar example of this in an atomic ion, in which the spectroscopy ion of interest lacks desirable features to permit Doppler cooling, is the Al^+ ion clock at NIST [**72**, **41**, **73**]. This dual-species ion clock uses a pair of ions, an Al^+ spectroscopy ion co-trapped with either a Be^+ or Mg^+ logic ion, with the logic ion being Doppler and sub-Doppler cooled, which sympathetically cools the Al^+ ion via the Coulomb interaction [**27**, **28**, **29**, **74**, **30**, **75**, **76**, **77**, **31**, **78**, **79**].

In order to map a spectroscopy excitation of the Al^+ ion to the logic ion, a Cirac-Zoller CNOT gate is used in which the final state of the logic ion is dependent only on if the spectroscopy excitation occurs [80, 81] with the implementation referred to as quantum logic spectroscopy (QLS). The QLS state-readout technique (Fig. 1.3) proceeds as follows:

(a) The spectroscopy and logic ions are prepared in the motional ground state of the ion trap of at least one motional mode, denoted by $|0\rangle_n$, using motional sideband cooling techniques on the logic ion.

(b) The spectroscopy ${}^1\mathrm{S}_0 \rightarrow {}^3\mathrm{P}_0$ transition is probed such that if the transition occurs, the spectroscopy ion is left in the metastable ${}^3\mathrm{P}_0$ state.

(c) A blue sideband π pulse on the spectroscopy ion ${}^1\mathrm{S}_0 \rightarrow {}^3\mathrm{P}_1$ transition is applied. This adds one motional quantam to the system only if the spectroscopy transition in (b) is unsuccessful.

(d) A red sideband π pulse on the logic ion is applied. If the spectroscopy transition in (b) occurred, the ion pair remains in $|0\rangle_n$ and the pulse is unsuccessful because no lower-energy motional state exists to transition to. Thus, the logic ion remains in its initial electronic state. If the spectroscopy transition in (b) did not occur and the system contains one motional quantum, the red sideband π pulse occurs, which shelves the logic ion.

(e) Lasers resonant with the logic ion initial state are then applied. If the logic ion is in the initial state, which occurs when the spectroscopy transition in (b) is successful, the ion fluoresces. If the logic ion is in the shelved state, which occurs when the spectroscopy transition in (b) is not successful, the ion is dark to the applied laser light.

The net result of this procedure is the logic ion fluoresces if the spectroscopy transition occurs and does not fluoresce is the spectroscopy transition does not occur, hence an

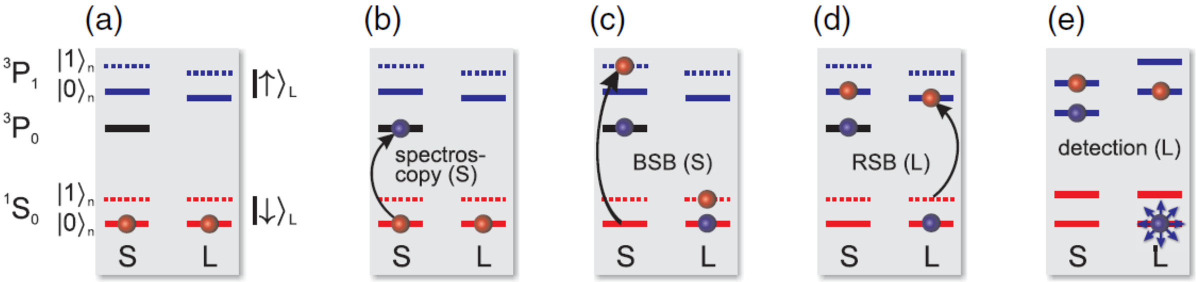


Figure 1.3. QLS state-readout technique used in the NIST Al$^+$ clock with relevant electronic levels shown and labeled as well as motional mode of the ion pair with ground (first-excited) state labeled as $|0\rangle_n$ ($|1\rangle_n$). (a) motional ground state preparation; (b) spectroscopy transition attempt; (c) spectroscopy ion blue sideband transition, dependent on if (b) is successful; (d) logic ion red sideband transition, dependent on if (c) is successful; (e) fluorescence detection of logic ion, dependent on if (d) is successful. The net result is (no) logic ion fluorescence if the spectroscopy transition (does not) occurs. Figure from Ref. [72].

implementation of a CNOT gate; the state of the logic ion changes depending on the state of the Al$^+$ ion.

We wish to extend this QLS technique, demonstrated to fractional frequency measurement uncertainties at the 10^{-17} level [73], to include molecules. We will call this molecular QLS or mQLS, which has been partly demonstrated by the Schmidt group [82] without molecular ion state preparation. The experiment relies on blackbody-radiation-driven transitions between rotational levels to place the molecule in the desired initial rotational state, which is both slow and uncontrolled; the state preparation technique is *inefficient*. If the rotational degree of freedom could be controlled, mQLS could be performed *efficiency*.

This dissertation is organized with relatively-independent discussions on each of the requirements that must be fulfilled in order to perform mQLS *efficiently*:

1. Laser systems required for Ba^+, our atomic logic ion of choice (Sec. 1.1, Part 1);
2. Demonstration of rotational ground state cooling as a spectroscopy ion state preparation technique (Sec. 1.2, Part 2);
3. Motional ground state preparation of a single $^{138}\mathrm{Ba}^+$ as a demonstration of the technique in our laboratory (Sec. 1.3, Part 3); and
4. A summary and future outlook (Sec. 1.4, Ch. 17)

A wide array of tools is required to enable trapping and subsequent optical laser cooling of trapped ions. The most basic of these tools is the ion trap itself, specifically the linear RF ion trap [26, 83, 84, 85, 86, 87, 88]. Additionally, light-matter interactions of two- and three-level systems play a crucial role for both Doppler and sub-Doppler cooling applications in ion traps [22, 23, 24, 89, 90, 91, 92, 93, 85, 94, 95, 96].

1.1. External Cavity Diode Lasers, Part 1

As their wavelength-coverage range continues to expand, diode lasers play an increasingly important role in experimental physics and related fields. For many ultracold atomic and molecular experiments, the linewidth of the diode laser system must be significantly less than the natural linewidth of the addressed transition. This requires the diode laser linewidth to be narrowed via an external cavity, forming an external-cavity diode laser (ECDL). In this configuration, the linewidth of the ECDL is generally dominated by drive current noise.

For example, a typical AlGaAs laser diode undergoes $\sim$3 MHz/μA of frequency change with injection current [97, 98, 99]. To achieve a $\leq$1 MHz linewidth, the integrated current noise must be $\leq$300 nA. In more precise applications, such as optical frequency standards, linewidths of $\ll$1 kHz are desired, necessitating a current drive noise $\ll$1 nA. This level of current noise is probably unrealistic, but a low-noise current source lessens the demands on the servo system that is needed to narrow the laser's linewidth and stabilize it to an atomic or optical frequency reference.

Here, we discuss the in-house ECDL systems including principles of operation, optomechanical design, current-generation of driving electronics, step-by-step assembly instructions, and details of select ECDL systems in the laboratory. Additionally, a Rb reference laser is constructed and characterized including investigations of acoustic resonances in the ECDL optomechanics and resultant requirements placed on the negative-feedback servo loop. Further current driver noise reduction and future ECDL development is also discussed.

1.2. Molecular Ion Quantum State Preparation, Part 2

Before the full potential of molecules can be harnessed, methods are needed to prepare and reset their rotational and vibrational quantum states with laser control offering a promising option.

Translational cooling of molecules initially in a particular rotational-vibrational state has been achieved using narrowband lasers, with unwanted vibrational excitation minimized because of an unusual degree of electronic-vibrational decoupling in the selected species [**100**, **101**, **102**]. Additionally, optoelectrical Sisyphus cooling has been employed to translationally cool samples of polyatomic molecules [**103**, **104**].

Vibrational cooling of translationally cold molecules has been performed using broadband lasers [**105**] and incoherent light sources [**106**] with spectral filtering removing light that would lead to vibrational heating. Spectrally tuned multimode diode lasers [107] have also been used for this vibrational cooling.

Rotational cooling is particularly critical for many applications because of the relatively small energy of rotational excitations; for instance, while many species are already in their ground vibrational state at room temperature, several to many rotational states are populated for all molecules. Cryogenic methods have been used to cool the rotations of MgH^+ with cryogenic helium buffer gas [108]. Optical rotational cooling has previously only been accomplished using narrowband lasers optically pumping population from a small number of rotational states at one time. A static narrowband laser was used to transfer population from two excited rotational states to the ground state in a beam experiment [18]. In an experiment on molecules formed by photoassociation from an ultracold atomic gas, cooling from the few populated rotational states to a single state was achieved by pumping from each level serially using a swept narrowband laser, in conjunction with a broadband laser providing vibrational cooling [109]. Finally, rotational cooling of trapped molecular ions from room temperature to the ground state was achieved on relatively slow timescales (seconds) by waiting for spontaneous and blackbody radiation-driven rotational transitions to redistribute population to levels pumped by narrowband lasers [110, 111].

Broad-spectrum rotational cooling has previously been proposed [106, 112, 113, 114] using a broadband laser tuned to an electronic resonance to cool rotations of trapped molecules from room temperature to their ground rotational state. More recent efforts at NIST and the Georgia Institute of Technology have focused on using Raman transitions between *rotational* states with a broadband femtosecond laser for deterministic rotational state detection by using a QLS-type readout technique. Once the molecular rotational state is known, rotational cooling could then be performed [115]. Broadband rotational cooling will allow rapid quantum state preparation both in molecular beam experiments, in which the time available for laser cooling is limited, and in trapped-molecule experiments where the initial rotational temperature is too high for cooling by a swept narrowband laser. In non-destructive trapped-molecule work such as spectroscopy or quantum information processing, particularly in ion traps where hold times can extend to many hours, broadband rotational cooling could also provide rapid state resets allowing the same molecules to be reused repeatedly, rather than requiring the trap to be dumped and reloaded in each experimental iteration.

Here, we discuss the implementation of broad-spectrum rotational cooling of AlH^+. As in the previous molecular translational cooling demonstrations [100, 101, 102], undesired vibrational excitation during electronic-resonance cycling is avoided because of vibrational-electronic decoupling in our chosen molecule. Owing largely to astrophysical interest, some spectroscopic properties of the AlH^+ molecular ion are already well understood with good agreement between experiment [116, 117, 118, 119] and theory [120, 113].

We perform experiments on Coulomb crystals of AlH^+ held in a linear RF trap and sympathetically cooled via laser cooling of co-trapped Ba^+. Previous calculations [113] are extended to include states relevant for an experimentally-convenient pathway for REMPD rotational state readout, and we predict an experimentally accessible dissociation cross section. Using time-of-flight mass spectrometry, we demonstrate the first REMPD of AlH^+ and use REMPD/time-of-flight mass spectrometry for rotational state analysis of our trapped sample, which has previously been used for MgH$^+$ [121] and HD$^+$ [122], where the molecular fraction dissociated was analyzed by observing deformation of a fluorescing Coulomb crystal of co-trapped atomic ions, as well as non-crystallized trapped HfF$^+$ using REMPD/time-of-flight mass spectrometry [123]. We then use this destructive rotational state readout technique to quantify our broad-spectrum rotational cooling of AlH$^+$.

1.3. Motional Quantum State Preparation, Part 3

The qubits in atomic ion quantum computation and digital quantum simulation, subclasses of quantum information processing, require high-fidelity coherent operations between long-lived qubit states with error correction protocols typically requiring gate fidelities of >99% [124, 125, 126, 127]. The atomic ion(s) are typically cooled to the motional ground states of the harmonic ion trap in order to facilitate quantum gate protocols and to minimize decoherence effects from residual thermal motion [80, 128, 93, 129]. Quantum logic spectroscopy (QLS), a different subclass of quantum information processing, does not require the high gate fidelities needed for quantum computation; a lower gate fidelity only increases the experimental integration time required to achieve a given spectroscopic statistical uncertainty. Most QLS-type protocols do, however, require or benefit from cooling at least one normal mode of the logic and spectroscopy ion pair to the motional ground state of the ion trap [72, 41, 130, 131, 132, 82].

Barium is one of the many Group II atomic ions that can be used as a coolant and logic ion for QLS. Barium has a large mass, convenient for sympathetic cooling of heavy atomic or molecular ions of interest to precision measurement. It also has the reddest

Doppler cooling transition wavelength of the Group II ions, useful for avoiding unwanted laser coupling between molecular electronic states and also for integration with fiber optic technology [133].

For high-fidelity coherent operations or sideband cooling of Ba$^+$, a 1.76 μm laser is sometimes used to drive the quadrupole $S_{1/2} \rightarrow D_{5/2}$ transition ($D_{5/2}$ lifetime $\sim$ 35 s [134]) [135, 136, 137, 138, 139]. With the $D_{5/2}$ lifetime of $\sim$ 35 s compared with the same state lifetime of $\sim$ 1.1 s in Ca$^+$ [140], the requirements for the 1.76 μm laser source are then demanding.

Rather than using a narrow quadrupole transition, one can also sideband cool using a Raman transition, tailored to be narrow by choices of detuning and Rabi frequencies [141], between two quantum states, e.g., electronic [136, 142], hyperfine [143, 144, 145, 146], or Zeeman levels [147, 148, 149, 150, 151, 31, 152]. For Ba$^+$, both electronic [136, 142] and hyperfine levels [146] have been used for Raman sideband cooling. Choice of a hyperfine interval for Raman sideband cooling has the advantage that the states can be chosen to be magnetically insensitive, allowing the same lasers to be used for cooling and high-fidelity coherent operations. On the other hand, Raman sideband cooling on a Zeeman splitting, previously demonstrated [147, 148, 150, 151, 31], allows use of even isotopes, which are simpler to cool and manipulate due to their lack of hyperfine structure. Additionally, ground state hyperfine intervals for some ions are too large for easy access by AOM (8 GHz for ^{137}Ba$^+$), and would require e.g. either phase-locking of separate lasers or EOM modulation with optical filtering for sideband cooling. For these reasons, the Zeeman splitting of ^{138}Ba$^+$ is an attractive system for Raman sideband cooling for QLS, in which high quantum gate fidelities are not required. Here, each of the two Raman beams is red-detuned from the single-photon resonance. We make this choice to simplify the experimental apparatus and prevent ion heating during beam alignment. The advantages of using blue-detuning will additionally be discussed.

1.4. Summary and Outlook, Ch. 17

Once solutions to the above challenges are realized in our laboratory, precision spectroscopy of molecular ions will provide a new tool to investigate time-variation of fundamental constants and probe at the many theories beyond and extensions to the Standard Model [12, 13, 14, 15, 16, 17, 10, 11, 18, 19, 20]. Additionally, the molecular state preparation techniques developed will have significant impact on molecular coherent control (alignment and orientation) and ultracold and quantum chemistry [1, 2, 3, 4, 5], all-optical cooling of all internal and external molecular degrees of freedom, single-molecule fluorescence detection, and quantum computing and simulation as well as hybrid quantum devices [6, 7, 8, 9, 10, 11].

Part 1

External Cavity Diode Laser Systems

An ever-increasing workhorse in experimental atomic, molecular, and optical physics and beyond for tunable, CW laser applications, diode laser systems continue to cover a broader wavelength range. For Ba^+, the atomic ion of choice in our laboratory, ECDLs are used for isotope-selective photoionization of Ba^+, Ba^+ Doppler and sub-Doppler cooling, and electron deshelving. Ba^+ electron shelving is achieved via a laser system without an external cavity, but the system still uses the drive electronics and optomechanics necessary for an ECDL system. Additional applications are neutral Rb trapping and laser cooling, possible fluorescence detection of SiO^+ in work by JHVN, YWL, and others, and possible future fluorescence detection of AlH^+ (Sec. 11.8).

This part discusses the principles of operation of ECDLs, the construction of an ECDL system, discussion of the latest in-house control and drive electronics, specific EDCL system details, and SHG modeling of the output of an ECDL with a resonant cavity that is necessary to generate the laser light at some of our wavelengths. Construction of a reference laser, locked to Rb using polarization spectroscopy, is then discussed along with further current driver noise reduction, which was in collaboration with the Dan Steck group. This part concludes with a brief discussion on future improvements that can be made to the EDCL optomechanics, enclosure, and current driver modulation electronics.

CHAPTER 2

External Cavity Diode Laser Systems

This chapter discusses ECDL laser systems including principles of operation, drive and control electronics, in-house ECDL system construction, and details of specific laser systems employed in this work and beyond.

2.1. Principles of Operation

Given a stable optical cavity of length L comprised of two mirrors with reflectivities R_1 and R_2, the free spectral range FSR, defined as the frequency between cavity resonances, is given by

$$FSR = \frac{c}{2nL} \tag{2.1}$$

where c is the speed of light and n is the index of refraction of the optical medium [153, 154, 155]. The cavity linewidth at FWHM $\delta\nu$ is then

$$\delta\nu = \frac{FSR}{\mathcal{F}} \tag{2.2}$$

where $\mathcal{F}$ is the finesse of the cavity, which is given by

$$\mathcal{F} = \frac{\pi}{2\arcsin\left(\frac{1}{\sqrt{F}}\right)} \tag{2.3}$$

and F is the coefficient of finesse, given by

$$F = \frac{4\sqrt{R_1 R_2}}{\left(1 - R_1 R_2\right)^2}. \tag{2.4}$$

So for an optical cavity of length ∼1 mm of material with index $n = 3.0$ surrounded by air with mirror reflectivities of $R_1 = 85\%$ and $R_2 = 25\%$, the FSR is ∼50 GHz with a linewidth of ∼20 GHz.

When the cavity contains a gain medium, the Schawlow-Townes linewidth is the fundamental quantum limit for the linewidth of a laser [156, 157, 158, 159, 160, 161] with the linewidth FWHM for a semiconductor laser given by

$$\Delta\nu_{\text{laser}} = \frac{h\nu \left(1 + \alpha^2\right) l_{\text{tot}} T_{\text{oc}}}{4\pi T_{\text{rt}}^2 P_{\text{out}}} \tag{2.5}$$

where $h\nu$ is the photon energy, α is a linewidth enhancement factor that takes into account some degree of amplitude-to-phase coupling, l_{tot} is the total laser resonator losses, T_{oc} is the output coupler transmission, T_{rt} is the resonator round-trip time, and P_{out} is the output power of the laser. For a typical laser diode the Schawlow-Townes linewidth is typically $\gtrsim$100 MHz due to multiple amplitude-to-phase coupling effects and fluctuations of the charge carriers in the semiconductor [157, 97]. However, if a laser diode is placed within an external optical cavity such that the cavity length increases by a factor of ∼10, the Schawlow-Townes linewidth will decrease by a factor of ∼100. Furthermore, much of the additional broadening effects are suppressed such that the Schawlow-Townes linewidth of an ECDL is typically $\ll$100 kHz, where residual drive current and acoustic noise of the external cavity dominate the output laser linewidth.

The optical length of the two cavities, the one formed by the laser diode front and rear facets and the one formed by the external cavity and rear laser diode facet, and reflectivities of the surfaces involved couple the cavities in a complex way [162]. The diode laser system tends to lase at the frequency with greatest gain and lowest loss with the external cavity providing an additional loss mechanism [163, 98, 158, 164, 165, 99]. So the grating profile selects the diode-chip mode in which to lase by suppressing the other diode-chip modes. The selected external-cavity mode frequency is dependent upon the optical length of the external cavity, providing a frequency tuning mechanism. Approximating, the ECDL will lase on the external-cavity mode that is closest to the center of the fundamental diode laser chip mode selected by the external-cavity feedback (grating) profile (Fig. 2.1).

Two common geometries used for ECDLs are the Littrow [167, 168, 169, 163, 170, 158, 160] and Littman [171, 168, 172] configurations (Fig. 2.2). Interference filters [173] and an ECDL with additional external cavity feedback [174, 175, 176] have also been used. Care must be used to not make the external cavity too long or islands of stable operation shrink dramatically as the phase of the light reflected by the external cavity is no longer suitably phase coherent with laser gain medium [177].

ECDL acoustic resonances are typically at frequencies $\lesssim$2 kHz. Although with sufficient engineering, the acoustic resonances can be designed to be at $\gtrsim$10 kHz and not couple strongly to environmental noise [160]. On sufficiently-short timescales, the residual drive current noise dominates the output laser linewidth. For example, a typical AlGaAs laser diode undergoes $\sim$3 MHz/μA of frequency change with injection current [97, 98, 99]. To achieve a $\leq$1 MHz linewidth, the integrated current noise must be $\leq$300 nA$_{\mathrm{RMS}}$.

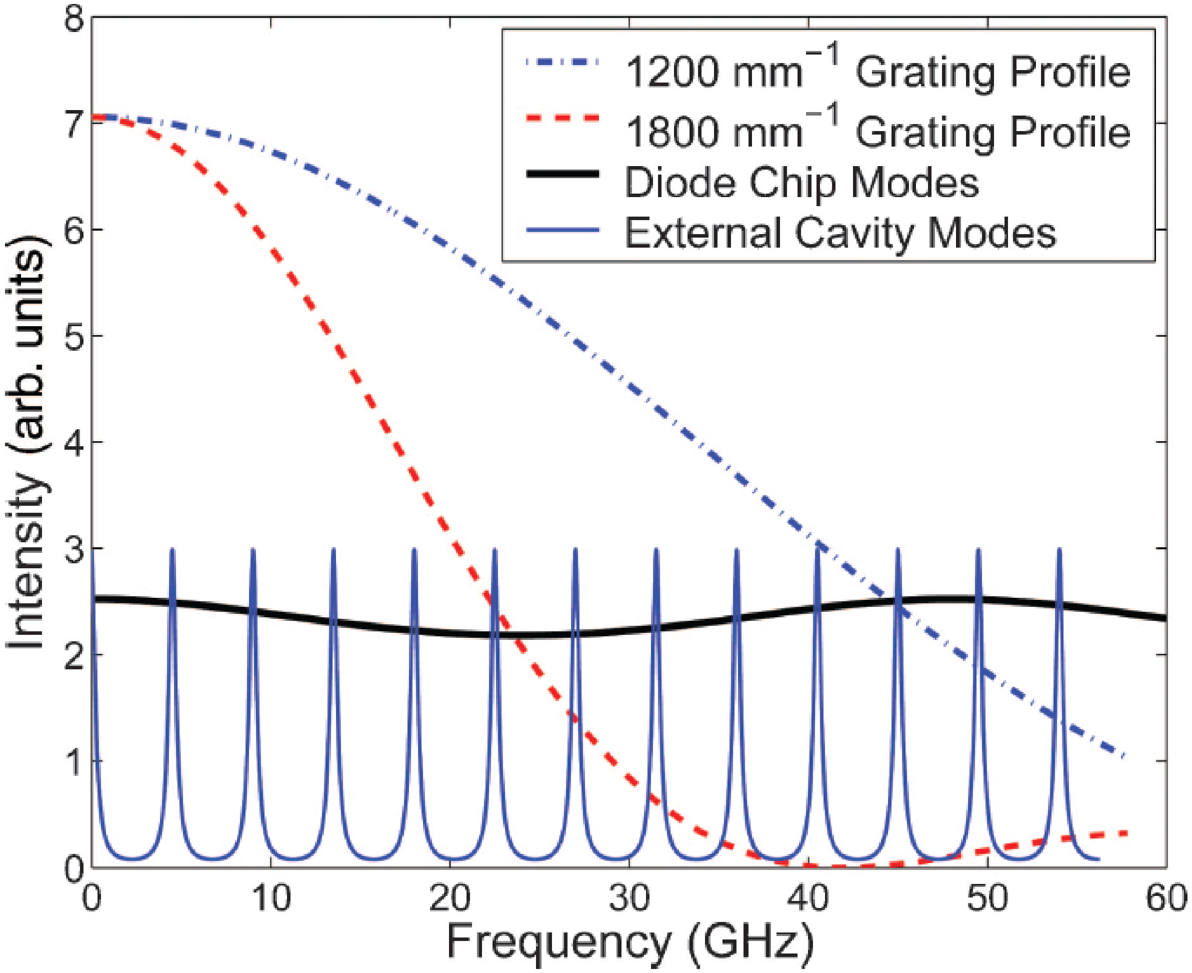


Figure 2.1. Calculated profiles of Littrow-cavity grating feedback (dashed) overlapped on diode-chip modes (solid black) and external-cavity modes (solid blue) for a typical ∼780 nm ECDL using an AR-coated laser diode, which suppresses the diode-chip modes. In this case, the ECDL will lase on the external-cavity mode at 0 GHz. Figure taken from Ref. [166].

The diode laser frequency changes with temperature due to a change in (1) the optical path length of the cavity, and (2) the frequency dependence of the gain profile. For (1), typical AlGaAs devices tune ∼0.06 nm/K while for (2), this change is ∼0.25 nm/K. These two effects result in the staircase-step wavelength output of a device as the temperature

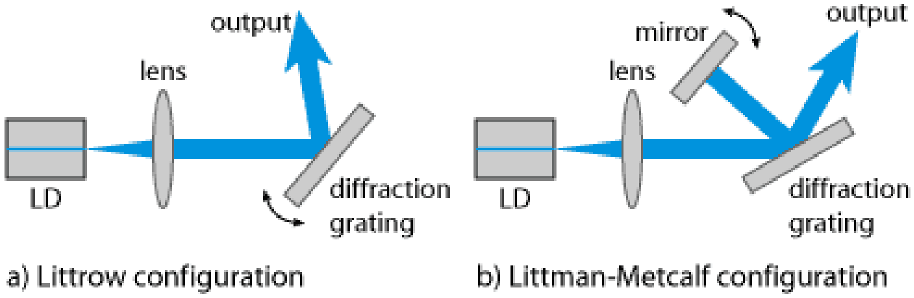


Figure 2.2. The common (a) Littrow and (b) Littman ECDL configurations. For wavelength tuning, the angle of the (a) diffraction grating or (b) mirror is rotated. Figure from RP Photonics GMBH.

is increased. Additionally, changes in the drive current affect both diode temperature and electron/hole carrier density. At drive current modulation frequencies $\lesssim$100 kHz, the frequency modulation is dominated by thermal changes in the diode chip due to Joule heating. The carrier density dominates at higher frequencies with a typical frequency variation of ∼-3 GHz/mA for frequencies <1 MHz, ∼-300 MHz/mA for frequencies between 1 MHz to 3 GHz, and ∼-1 GHz/mA for frequencies ∼3 GHz. This response drops off rapidly at higher frequencies [**97**, **98**, **99**].

Tuning via external-cavity length, typically controlled in the Littrow configuration with a PZT element and pivot, a typical ECDL can frequency tune the laser output by a few GHz without experiencing an external-cavity mode hop. This is due to the operating external-cavity mode tuning off of the center of the diode-chip mode sufficiently far that another external-cavity mode is preferred. If the diode-chip mode is tuned with the external-cavity length using laser diode current modulation (current feed-forward), the mode-hop-free tuning range can be increased typically by at least an order of magnitude [**178**, **179**, **180**, **164**, **165**]. For the system discussed in Ch. 3, the mode-hop-free tuning range without current feed-forward is 2.8 GHz. With current feed-forward, this value increases to >68 GHz, which is a factor of >20 larger. In this case, the system is limited by the maximum desired driving current and by the lasing threshold.

The benefits of using an AR-coated laser diode can be dramatic on increasing (1) lower-wavelength-side tuning range, (2) mode-hop-free tuning range, and (3) output power [**180**, **164**, **160**]. For (1), the threshold current of the diode chip increases significantly; this increases the density of electrons and holes in the active area of the laser diode chip, populating higher-energy and higher-momentum states in the semiconductor band structure. The laser diode with external-feedback can lase from these higher-energy electron/hole pairs. For (2), the diode-chip modes are suppressed like those shown in Fig. 2.1. With a sufficiently-low front facet reflectivity, the ECDL will lase at the peak of the grating profile, and diode-chip losses will be negligible. Because of the latter, the ECDL mode-hopping from one external-cavity mode to the next is suppressed or eliminated. For (3), the front facet is no longer an extra reflective element that partially reflects the optical power circulating in the diode chip; in the steady-state, this results in higher optical output power for a given set of laser diode operating parameters. Additionally, optimizing the feedback alignment and operating parameter space using an AR-coated laser diode is significantly more simple.

2.2. The ECDL Design

The basic design of the in-house Littrow-configuration ECDL is that of Ricci and Hänsch [**169**, **163**] modified to place the pivot point of the grating arm near the optimal point of the plane of the grating and rear facet of the laser diode for a large mode-hop-free tuning range [**181**, **182**, **183**, **184**, **164**, **185**] without using an AR-coated laser diode. In this configuration, the first-order grating diffraction is coupled back into the laser diode. So as the grating angle of incidence θ changes, the output wavelength λ will tune according

$$\lambda = 2a \sin\theta \tag{2.6}$$

where a is the grating line spacing. The resonant ECDL cavity mode is given by

$$m\frac{\lambda}{2} = L_e \tag{2.7}$$

where m is an integer and L_e is the optical path length of the external cavity, typically ∼20 mm. Therefore, if the three parameters (1) grating angle, (2) resonant ECDL cavity mode, and (3) the diode-chip mode can tune together via grating angle and diode laser drive current, mode hops between diode-chip and external-cavity modes will be minimized.

Our ECDL design is originally from Ref. [**186**, **187**], which is designed for use at 780 nm with a holographic 1800 line/mm grating for an operating grating angle of 44.6°. Additionally, the design includes a mirror, which is attached to the end of the grating arm such that it rotates with the grating and reflects the 0-order diffraction ECDL output by $\sim 90°$. Thus, the ECDL output beam pointing will be unaffected by changes in the grating angle. In other words, the output beam will displace slightly as a function of grating angle, which is preferable when long beamlines after the ECDL are involved.

The design is modified with a longer vertical grating adjustment lever arm to add clearance for the grating arm adjuster to permit a wider range of wavelengths and grating line spacings to be used. An added benefit of this longer adjustment arm is that the vertical adjustment resolution is finer. Additionally, several alignment dowel pins are

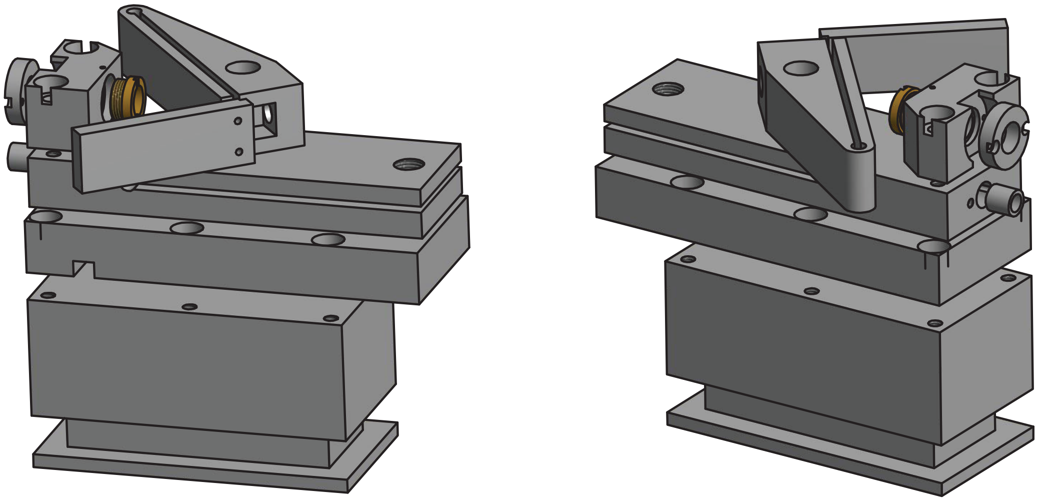


Figure 2.3. Assembly drawing of the in-house ECDL machined components. All drawings are located in App. F.1.1.

added to help constrain alignment. A view of the machined components assembly for our ECDL design is shown in Fig. 2.5. Note that the slots added to the laser base plate and grating arm are machined via electro discharge machining. All most-recent drawings are located in App. F.1.1.

2.3. Assembling an ECDL

ECDL assembly begins with the laser itself before the enclosure is built around the ECDL. Diagnostics data is then taken to characterize the performance of the individual laser diode. The external cavity is then added to the system and aligned. Further diagnostics data is collected and current feed-forward gain optimized.

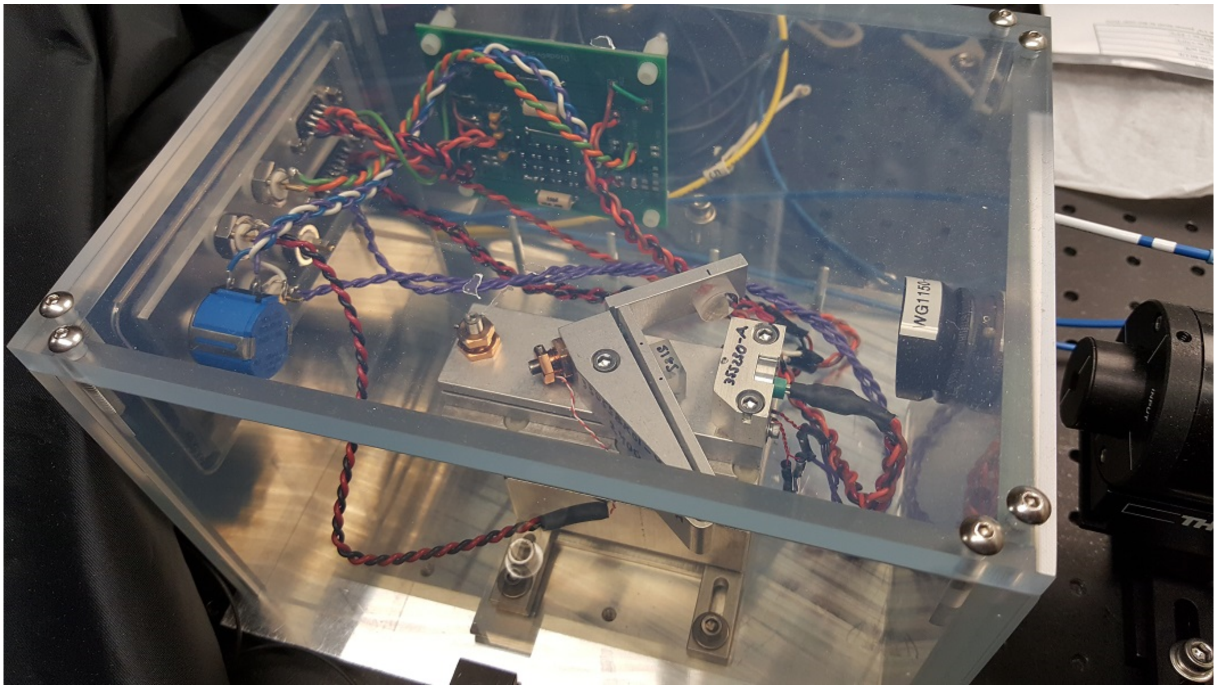


Figure 2.4. Picture of an assembled ECDL system. Note the labeling on the laser head and grating arm of the collimation lens and grating, respectively, as well as the indexing marks for the grating and corner cube mirror mount. Additionally, note the protection circuit in the upper center portion of the image.

2.3.1. The Laser

The first task to assemble a laser system is to thermal epoxy the heater into the heater plate and two thermistors into the thermistor holder. For the heater plate, wire leads are soldered to a 50 Ω resistor and leads insulated to prevent shorting to the heater plate. This is then placed inside of the pocket machined into the heater plate and area filled with thermal epoxy, which is best performed by bolting a heatsink lid to the back of the heater plate with a layer of plastic in between, filling the pocket half-way with thermal epoxy, then pressing the thermal-epoxy-dipped resistor into place. For the thermistor holder, two

10 kΩ thermistors (Amphenol RL0503-5820-97-MS) are dipped into the thermal epoxy and placed into the thermistor holder.

The second task is to compress the flexure hinges of the laser base plate and grating arm, accomplished with a large vise. This assures that the flexure adjustments have appreciable range and tension.

Once the thermal epoxy cures and flexure hinges are compressed, the rest of the system is assembled typically with the heatsink attached to the enclosure base plate (Sec. 2.3.2). A thin layer of Arctic Silver 5 thermal compound is spread on the solid heatsink where the TEC element (Laird Technologies 56460-500 CP10,127,05 with silicone sealant) is placed with cold-side facing up. The TEC element upper and lower surfaces are also layered in thermal compound. A spacer piece, made of G-10 FR-4 composite with near-identical dimensions and height at the TEC element, is placed opposite the TEC element on the solid heatsink. The heater plate with thermal compound layered where it contacts the TEC element and with heater directly above the TEC element is bolted in place using glass-filled nylon SHCSs (McMaster-Carr 91221A460) to electrically isolate the laser head from the enclosure. The laser base plate is then bolted in place with a thin layer of thermal compound between it and the heater plate and two alignment pins (McMaster-Carr 97395A397). A ThorLabs N100L5P 3/16"-100 locking bushing and F19SS075 3/16"-100 hex adjuster are then attached to the laser base plate. The thermistor holder pocket is coated with thermal compound and thermistor holder pressed into place using a SCHS and large-diameter washer.

Depending on the size of the laser diode case (either 5.6 or 9 mm), the appropriate laser head is taken and laser diode installed. Proper grounding must be used for this step or laser diode destruction will likely occur. A very small amount of thermal compound is applied to the pocket where laser diode resides, and the laser diode is installed with the diode mounting ring tightened down after the laser diode is rotated a few times to evenly distribute the thermal compound. Typically, the diode should be installed such that the output-spot long-axis is parallel to the table. This is verified in a later step.

A collimation lens (ThorLabs A110, 354330, or 355230 with appropriate AR-coating) is chosen to yield the desired output spot size of <5 mm MFD (full-width at e^{-2} of maximum) such that the output beam will not be clipped by an optical isolator aperture. The lens is then mounted in the lens casing of appropriate size and depth for the application, possibly with the use of a spacer. The lens is glued in place using DUCO cement using a cotton-tipped applicator to press the lens into the casing with a single motion; repeated motions with the same applicator can coat the lens with the cement. However, DUCO can be dissolved with methanol. Once dried, the collimation lens is screwed into the laser head, and the laser head is set aside in an electrostatically-safe location.

The next step is to enclose the system and install the laser diode protection circuit to protect the laser diode from ESD events and/or improper drive electronics connections.

If the water-cooled heatsink is to be used, the above assembly procedure is followed only with the addition of adding an o-ring to the heatsink base and a sealant around the o-ring before the TEC is placed on the heatsink lid. The nylon SHCS hold the assembly together and seal the heatsink.

2.3.2. The Enclosure

The enclosure for the ECDL system is shown in Fig. 2.5, completed ECDL system in Fig. 2.4, and all drawings are located in App. F.1.2. For reference, the heatsink will only fit in a single orientation once the enclosure walls are installed.

First, the circuit wall and feedthru wall are bolted into place on the aluminum floor. The electronics plate is then assembled using a 15-pin D-Sub receptacle for the temperature controller; a 9-pin D-Sub receptacle for the current driver; BNC connections for temperature monitoring via the second thermistor, PZT voltage, capacitive current modulation, and resistive current modulation; and a 10 kΩ potentiometer used with the protection circuit (Sec. 2.4.3), all with sufficiently-long 22 AWG wire leads of suitable color. The D-sub connectors follow the Newport 6100 controller pinout as listed in Table 2.1 with the addition of the temperature controller heater connections. The protection circuit with laser diode socket is then soldered to the electronics plate before the completed electronics plate is solvent-welded in place on the feedthru wall. Once cured, the protection circuit is bolted to the circuit wall using short standoffs.

If already installed, the laser heatsink, which is bolted to the ECDL enclosure using two heatsink brackets and four washers/SHCSs, is removed temporarily. The thermistor leads, TEC element, and heater are then soldered in before the laser heatsink is bolted back into place. The remaining walls of the enclosure, the long wall and output wall, are then installed. A ThorLabs AR-coated, 5-mm-thick window is installed into a ThorLabs SM1L03T 10° angled mount to avoid coupling back-reflections into the laser, and the mount is screwed into place with the angle facing down on the output wall inside of the

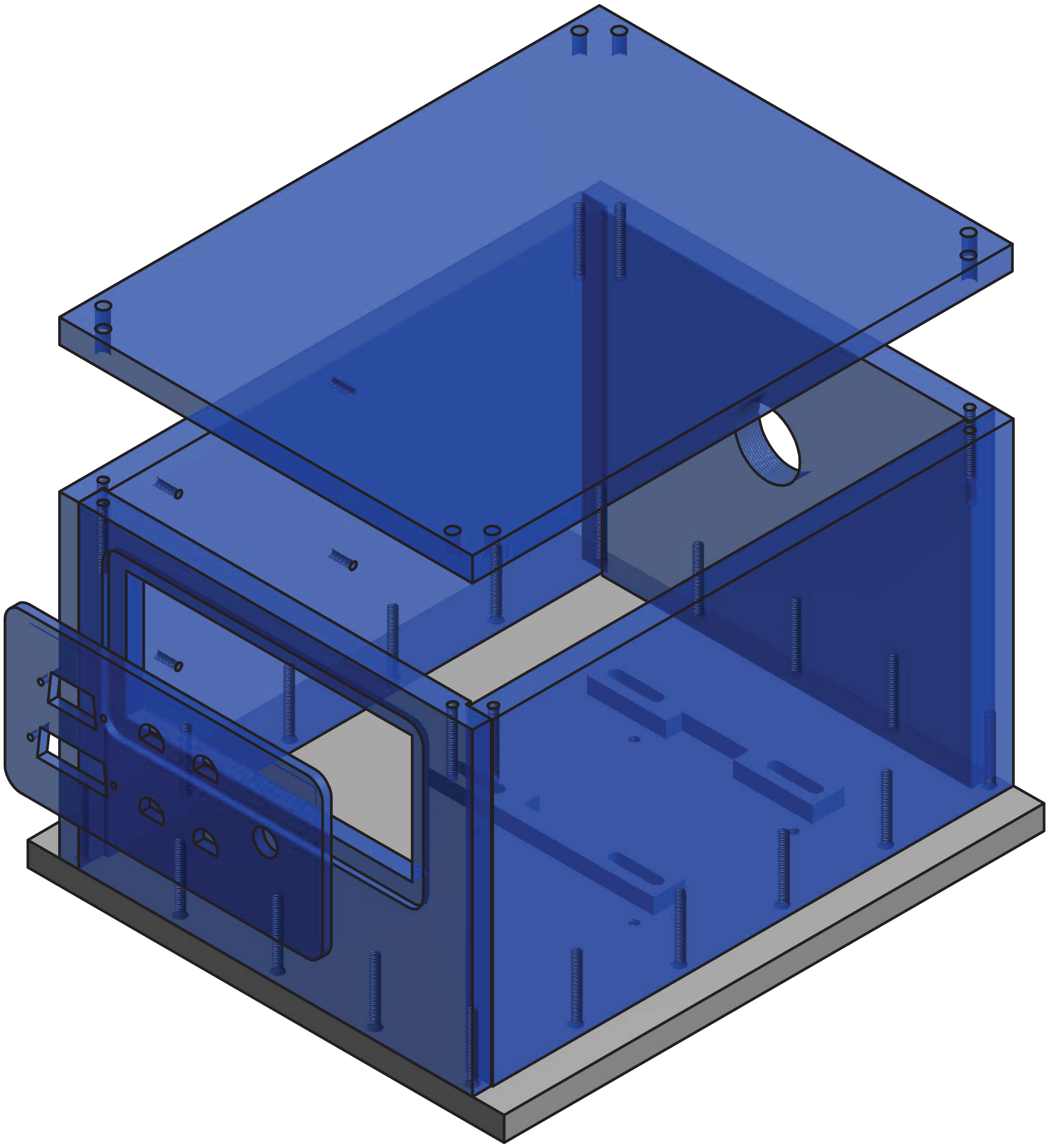


Figure 2.5. Assembly drawing of the in-house ECDL enclosure machined components. Clear blue components are static-dissipative acrylic. All drawings are located in App. F.1.2.

enclosure. A ThorLabs SM1D12SZ zero-aperture iris is then installed on the outside of the enclosure.

Table 2.1. ECDL electrical connection pinouts for D-Sub connectors for the current driver and temperature controller. Note that these are the same as the Newport 6100 controller with the addition of the heater connections. The TEC and laser diode connections each have two pins per drive polarity to decrease the current load on and voltage drop across the cabling between driver/controller and ECDL system.

D-Sub	Pin	Connection
Temperature Controller	1	TEC (+)
	2	TEC (+)
	3	TEC (-)
	4	TEC (-)
	7	Thermistor
	8	Thermistor
	9	Heater (+)
	11	Heater (-)
Current Driver	3	Ground
	4	Laser Diode Cathode
	5	Laser Diode Cathode
	6	Photodiode Cathode
	7	Photodiode Anode
	8	Laser Diode Anode
	9	Laser Diode Anode

2.3.3. Installing the Laser Head and Collimation

After the system is enclosed and the diode protection circuit (Sec. 2.4.3) is in place with potentiometer shorted as verified by a multimeter testing between the laser diode anode and cathode pins, an alignment pin is installed into the laser base plate, a thin layer of thermal compound is applied to the bottom of the laser head and top of the laser base plate, and the laser head is bolted into place. The laser diode socket is then plugged in.

In order to collimate the laser beam and collect free-running laser diode diagnostics, a grating arm and corner cube mirror mount are installed with appropriate ThorLabs BB05 dielectric mirrors glued in place with DUCO cement. These are easiest to place by turning the laser diode current on with sufficient current to see a laser spot. The mirrors can be held in place with a pair of tweezers, optimal location indexed with a pen, and glued in place sequentially. The laser spot long-axis-orientation is now verified to be parallel to the table. If it is not, the laser diode is rotated in the laser head appropriately. The heatsink position is then adjusted to place the laser output spot in the middle of the enclosure output aperture.

The collimation lens is then adjusted to place a focal spot far away from the laser system. A ThorLabs shearing interferometer is then used to collimate the long-axis of the beam to ensure optimal back-coupling of the ECDL feedback light and a narrow ECDL linewidth [**188**]. Once collimated, the collimation lens locking set screw on the laser head is tightened and collimation again verified. This is best performed at low drive currents just above threshold. If no fringes can be observed, this process can be repeated once the grating is installed.

2.3.4. Free-running Diagnostics

Three sets of diagnostics are taken to characterize the laser diode for verification of delivery specifications, setup information, and for use to verify any future laser diode damage: (1) output power, driving voltage, and photodiode current (if available) versus driving current up to the laser diode operating output power; (2) output wavelength as a function of drive current; and (3) repetition of (1) and (2) such that data is collected at 20, 25, and 30 °C. Sample data is shown in Fig. 2.6, Table. 2.2, Fig. 2.7, and Table. 2.3. Temperature of the laser system is verified using the second thermistor on the BNC feedthru. If the laser

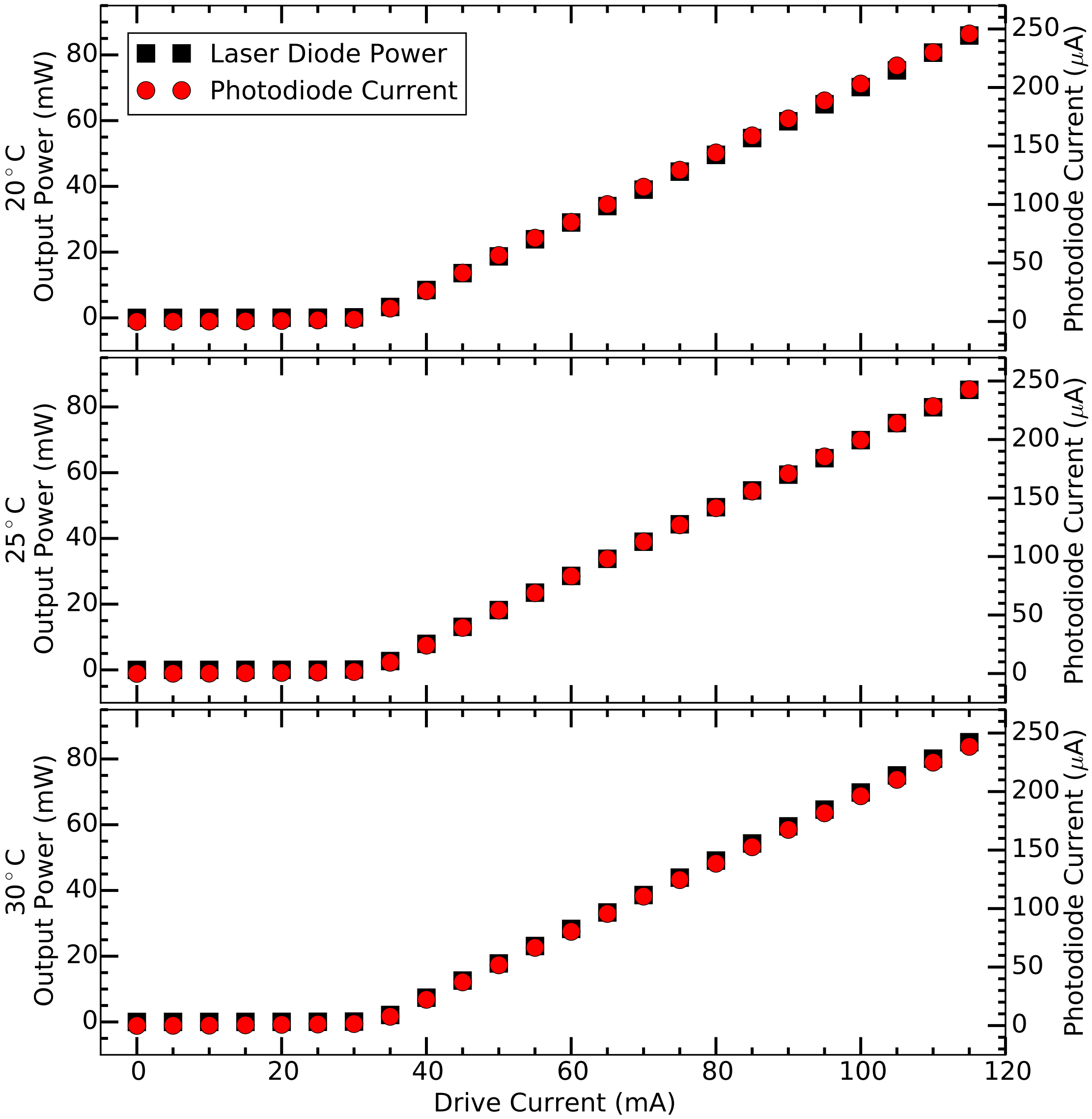


Figure 2.6. Output power and photodiode current plotted versus laser diode drive current at 20, 25, and 30 °C. This data is for the laser diode used in Ch. 3 and collected using a Newport 6100 controller.

diode is AR-coated, as much data as possible from the above sets is collected, typically only (1).

Table 2.2. Free-running wavelength and parameters at 90 mW operating power at 20, 25, and 30 °C. This data is for the laser diode used in Ch. 3 and collected using a Newport 6100 controller. From the data, this laser diode tunes 0.122 nm/°C with a 0°C wavelength of 781.23 nm as given by a linear fit to the data.

Temp. (°C)	LD Current (mA)	LD Voltage (V)	PD Current (μA)	Wavelength (nm)
20	119.34	2.069	258.1	783.6996
25	119.30	2.061	254.7	784.2108
30	119.70	2.057	251.3	784.9175

Table 2.3. Free-running wavelength versus laser diode drive current fit results at 20, 25, and 30 °C. This data is for the laser diode used in Ch. 3 and collected using a Newport 6100 controller. From the data averages, this laser diode tunes 0.0167 nm/mA and 0.177 nm/°C, which is comparable with the results in Table 2.2.

Temp. (°C)	Slope (nm/mA)	Intercept (nm)
20	0.0198	781.1748
25	0.0137	782.4239
30	0.0167	782.9452

Estimating that the laser will operate at ~30% of the free-running operating current when in the external cavity, the output wavelength when operating here should be tuned to the desired output wavelength by changing the temperature. The approximate temperature change can be inferred from the data collected via (3).

2.3.5. Grating Alignment and Wavelength Tuning

After the free-running diagnostics data are collected, the grating arm and corner cube mirror mount are installed. First, a ThorLabs N100L5P 3/16"-100 locking bushing and F19SS075 3/16"-100 hex adjuster are installed into the grating arm. A ThorLabs

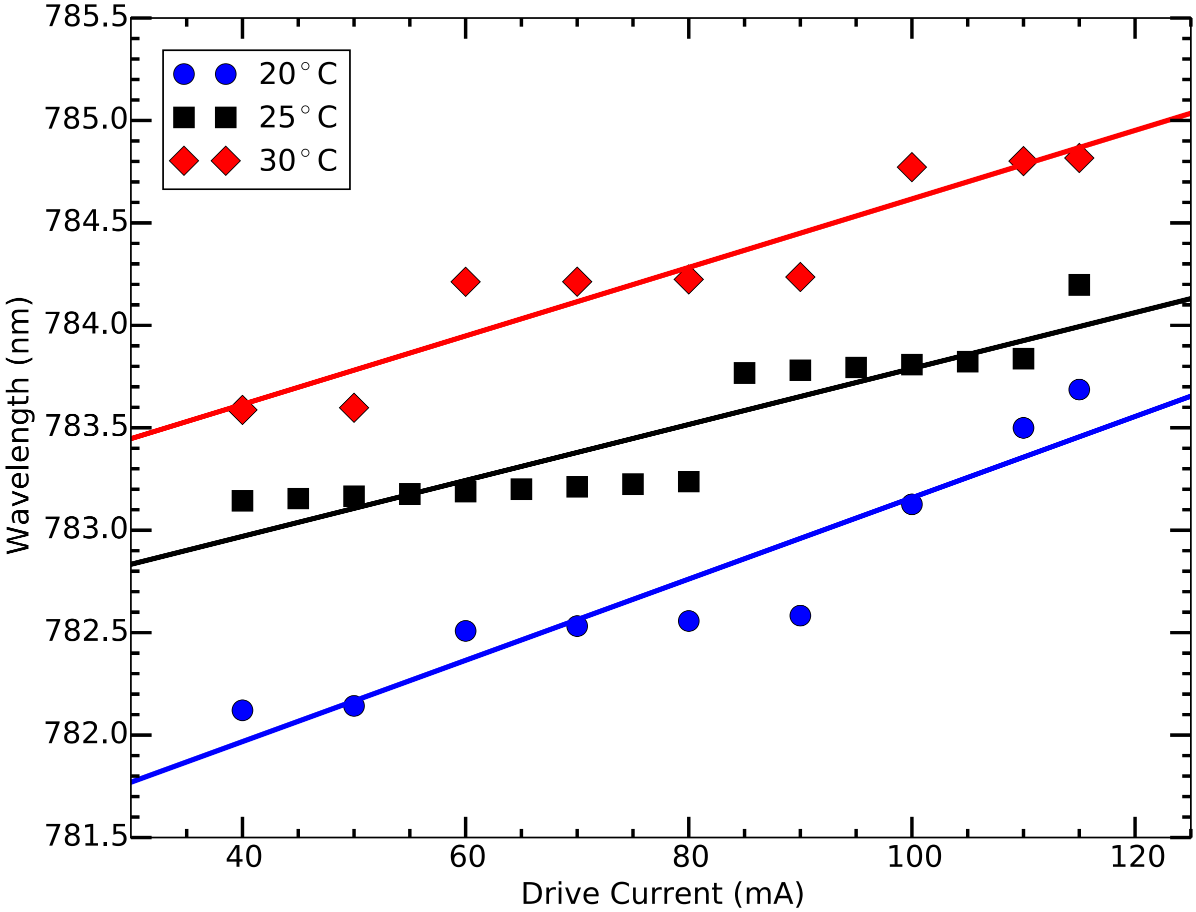


Figure 2.7. Free-running wavelength versus laser diode drive current at 20, 25, and 30 °C (data points) with linear fits to the data (solid lines). The smooth increase in wavelength is due to the gain profile of the laser diode shifting with the diode-chip cavity mode. The large jumps in wavelength correspond to a hop in the diode-chip cavity mode. This data is for the laser diode used in Ch. 3 and collected using a Newport 6100 controller.

AE0203D08F PZT actuator is then glued in place between the grating arm and hex adjuster with DUCO cement on the grating arm side. Tension on the grating arm provided by the hex adjuster conveniently holds the PZT element in place while the DUCO cement dries.

The grating is then set on the laser base plate while observing the first-order diffraction spot at $\sim$30% of the free-running operating drive current. With appropriate angle of the grating, the grating diffraction efficiency is measured, which should be $\lesssim$20%. If this value is significantly higher, it's likely that either the input polarization from the laser diode is incorrect and the laser diode should be rotated by 90°, a different grating should be used, or a HWP should be installed between laser diode and grating (Sec. 2.5.5.4).

The grating mount is then loosely bolted in place and grating set against the arm angled such that the first-order diffraction couples back to the laser diode, which is done either by visually confirming the diffraction alignment or preferably by lowering the drive current to just below threshold and adjusting the grating position until the output intensity "flashes" as the back-coupled light is sufficient to stimulate lasing of the system. The vertical adjustment hex screw will likely need to be adjusted for proper alignment. The grating arm is then indexed with a pen where the grating should be placed. The grating arm is removed and grating glued in place using DUCO cement. Once dried, the corner cube mirror mount is installed, grating arm bolted in place and roughly aligned similarly to before. A BB05 dielectric mirror is then held against the corner cube mirror mount with tweezers, ideal location indexed, grating arm again removed, and mirror glued in place.

If the laser diode contains a monitor photodiode, the photodiode current at the free-running operating output power (Sec. 2.3.4) informs the user of the maximum intra-diode-chip cavity power the laser should operate under when placed in an external cavity to assure a long laser diode lifetime. If no monitor photodiode is included, the drive current at ∼30% of the operating free-running power is used as the maximum with external-cavity feedback.

Grating alignment first proceeds by operating the laser near the expected maximum current with external-cavity feedback. When the grating alignment is close, a second spot should be visible next to the primary output spot. The goal is to overlap this spot with the main. When close, the drive current is reduced to just below threshold and vertical alignment adjusted to maximize the output power. This process is then iterated.

The heatsink is again repositioned to center the output spot on the enclosure output aperture and an optical isolator is placed immediately after the laser before powering up the ECDL to the external-cavity operating current. The output power is then controlled with a HWP and PBS before being fiber-coupled to a spectrometer or wavelength meter to finely tune the grating angle and output wavelength; the grating arm hex adjuster is tuned appropriately such that the output wavelength is within ∼10 GHz of the desired wavelength. Typically, the vertical adjustment will need to be optimized frequently during this process. Once close to the desired wavelength, the vertical adjustment is finely optimized such that mode-hops only occur in a monotonically-increasing manner as discussed in Ref. [166]. Final wavelength adjustment is performed using small changes in temperature, drive current, and PZT voltage.

Once the wavelength is set, the optical isolator is optimized. A beam profile is then taken, typically with a ThorLabs DCC1545M CMOS camera, and a 2-dimensional Gaussian fit performed to extract the output beam size and ellipticity, which is used to determine the cylindrical and, if needed, spherical lens pair required to correct the beam ellipticity and size for optimal fiber-coupling efficiency.

2.3.6. Setting the Current Feed-forward

As noted in Refs. [178, 179, 180, 164, 165], modulating the current along with the external-cavity PZT element provides a significantly larger mode-hop-free tuning range. This is implemented on the current driver modulation section and is adjusted by the resistor R18 and MOD_{GAIN} trimmer potentiometer. Two pieces of data are required to properly set the current feed-forward, which will be characterized in units of mA/$\mathrm{V_{PZT}}$. The first is the wavelength tuning with driving current measured in Sec. 2.3.4. The second is the wavelength tuning with PZT voltage, which can be measured with a wavelength meter. The two sensitivities are combined to yield the desired mA/$\mathrm{V_{PZT}}$ value.

Preferably, this value is empirically determined. In this method, the PZT driver is set to a low voltage and drive current adjusted to center an external-cavity mode near the maximum external-cavity-configuration current. The PZT voltage and drive current values are recorded. The PZT voltage is then increased until the mode hops or becomes unstable as seen on the wavelength meter. The current is then decreased to recover and center the mode. The mode is then tracked iteratively until it can no longer be sustained. The last known PZT and drive current values where the mode is centered can be combined with the first-collected data point to extract the necessary mA/$\mathrm{V_{PZT}}$ modulation.

2.3.7. Alignment of Injection-locked System

Injection-locking allows for power amplification of a master laser, often an ECDL, either directly or after a frequency modulator [189, 97, 190, 191]. If an in-house ECDL system is to be used in an injection-locking application, the grating is replaced with a mirror, PZT element is omitted, and grating arm hex adjuster is replaced with a longer ThorLabs

F19SS125 3/16”-100 hex adjuster, which is screwed in sufficiently to apply tension on the grating arm. The vertical adjuster is adjusted to place the output beam parallel to the optical table.

To injection-lock a laser diode, laser power from the master laser must be injected into the slave laser diode. This can best be performed using an optical isolator with reject ports such that light can be externally coupled in the reverse direction (i.e. ThorLabs -HP option optical isolators). In this case, a pair of steering mirrors, a PBS, and a HWP are placed at the isolator reject port. The HWP is rotated to optimize the back-coupled power, and the steering mirrors are used to couple into the laser diode, which can be observed using a spectrometer, wavelength meter, or scanning Fabry-Pérot cavity. Note that the two beam shapes and sizes must be comparable for optimal coupling efficiency.

2.4. The Electronics

Drive electronics for ECDL systems are commercially available, but both temperature controllers and current drivers can be built in-house including enclosure as shown in Fig. 2.8 with drawings in App. F.1.4. Commercial drivers are used for demanding applications (i.e. the 1228 nm laser system where the laser diode drive current is $\gtrsim$600 mA), but for most in-house ECDL applications in this work, in-house temperature controllers and current drivers are used. The most recent versions of the circuits are discussed below, but the discussion does not include the improvements described in Ch. 4. Additionally, L-Com Global Connectivity CPMS9MM-5 and CPMS15MM-5 or -10 heavy duty, doubly-shielded cables are used between driver and ECDL to shield from environmental electrical noise.

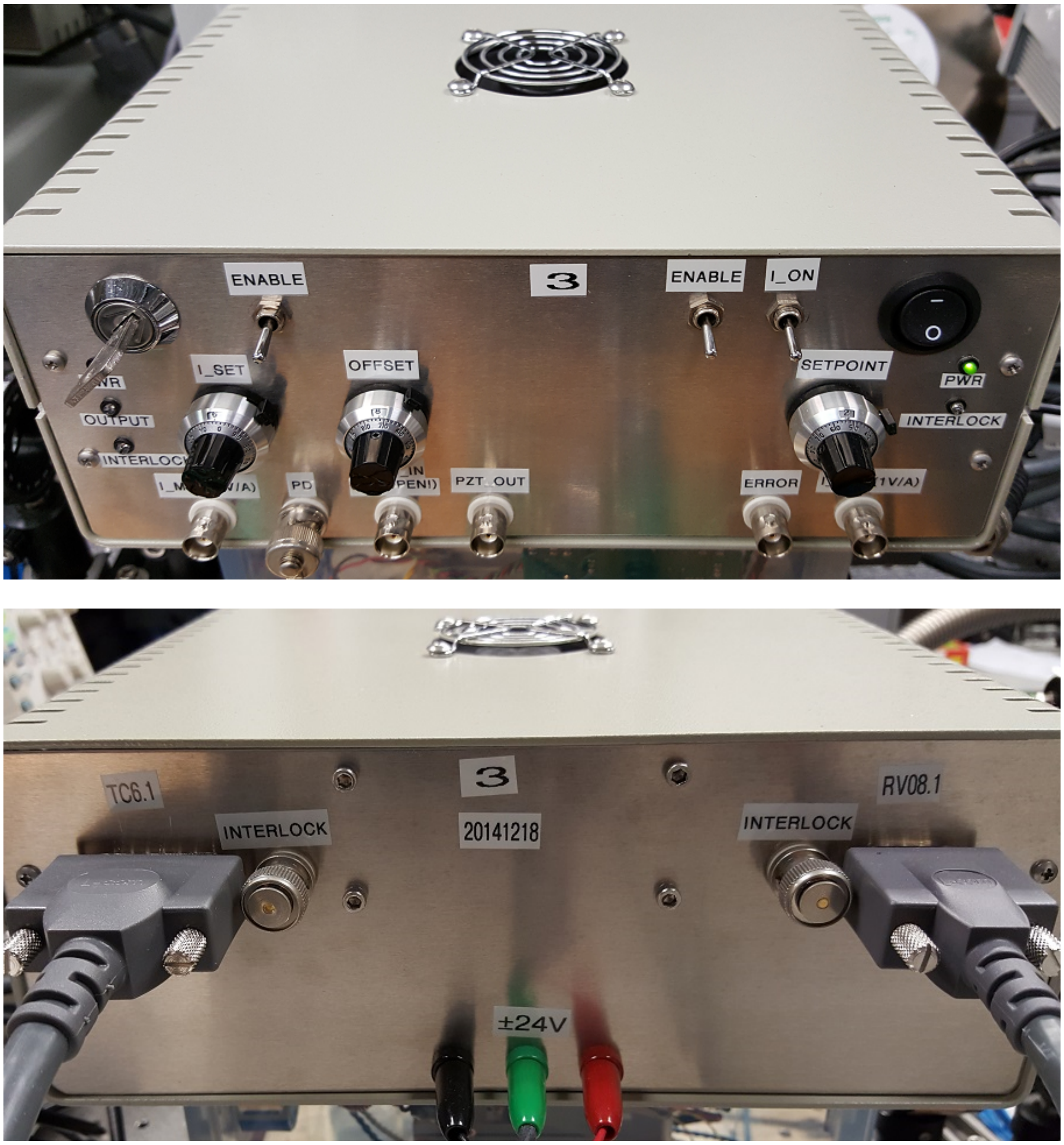


Figure 2.8. Picture of assembled ECDL electronics system. The enclosure base is a Hammond Manufacturing 1402KV. The fan on the top of the case, powered by the -24 V supply, circulates sufficient air to keep the system cool to ensure longevity of internal components. Note that the power switch to the laser driver is keyed and a functional interlock is included on the rear to conform to laser safety regulations.

2.4.1. The Temperature Controller

The temperature controller circuit is shown in Fig. 2.9 with schematic and board located in App. A.1.5. The circuit, powered by ±24 V, calculates an error signal referencing the thermistor sensor voltage to a set point provided by a potentiometer R_{box}. The voltage reference is a buffered Zener diode. The response of the circuit set point to R_{box} value is not trivial due to the way the circuit is designed. The response of the thermistor is nonlinear over a wide range of temperatures, and the nonlinear R_{box} dependence ensures that the sensitivity of the set point and sensitivity of the thermistor are comparable. At low temperatures, a small change in temperature results in a large change in resistance and vice-versa at high temperatures. A reference table of simulated (via PSpice) and measured R_{box} resistances to thermistor resistance are shown in Table 2.4 with data plotted in Fig. 2.10. Due to this, the R_{box} potentiometer is typically a 10-turn 100 kΩ potentiometer in series with a 100 kΩ resistor. Further data relating R_{box}, thermistor resistance, and thermistor voltage are in Sec. 2.5.5.3.

After error signal generation, which has a gain set by the ER_{GAIN} potentiometer (25.04 kΩ typical value), the circuit performs a PI servo loop (typical P_{GAIN} value is 65.87 kΩ and typical I_{GAIN} value is 18.81 kΩ) with resulting signal fed to a current drive loop consisting of a differential amplifier with output connected to a push-pull Darlington-pair transistor set. An interlock subcircuit prevents the circuit from driving current if the interlock pins are not shorted, which could be used as secondary protection for sensitive systems. The voltages that supply the transistors are independently regulated using LM337 and LM338 regulators to enable maximum drive voltages to the TEC element. Note that setting a TEC element maximum current is significantly more complicated due to the voltage drop

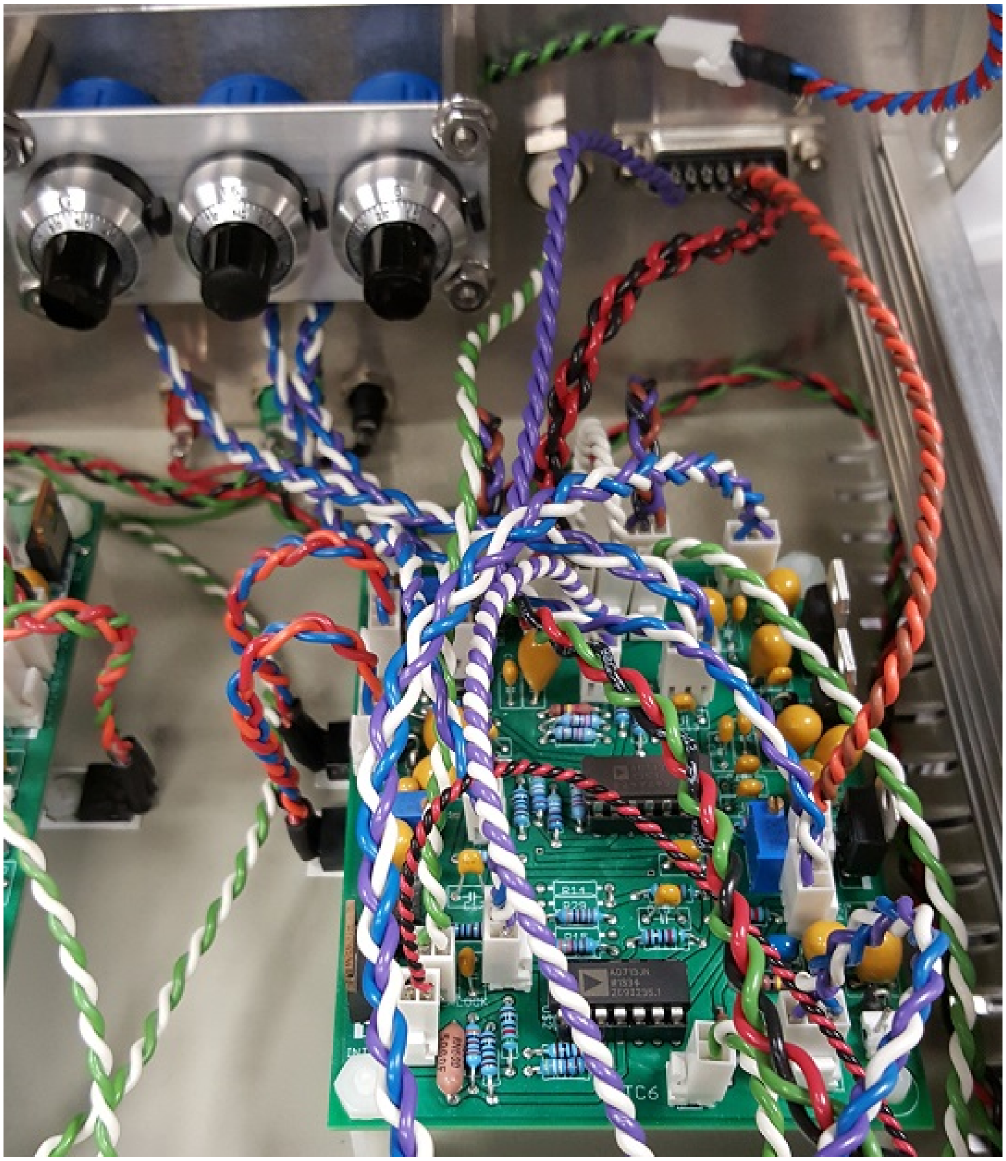


Figure 2.9. Picture of an assembled temperature controller in an ECDL electronics system. All wiring is 22 AWG. ER_{GAIN}, P_{GAIN}, and I_{GAIN} potentiometers are mounted inside of the case in the upper left of the image.

across the device depending on the temperature difference across the TEC element, so the voltage-limit control is chosen. Additionally note that this circuit can be easily modified to perform temperature control using a heater element. In this case, $U\$24$, $Q3$, and $IC6$

Table 2.4. Temperature controller R_{box} resistance to thermistor resistance and measured temperature. Temperature is measured using the second thermistor in an ECDL driven by the temperature controller.

R_{box} (kΩ)	Thermistor (kΩ)	Temperature (°C)
Calculated		
370	4	47
295	5	42
190	8	30
142	11	23
114	14	17.5
96.4	17	13
84	20	10
75	23	7
Experimental		
120	13.4	18
130	12.3	20
140	11.3	22
150	10.4	24
160	9.7	26

are omitted. In the ECDL controller, $Q2$, $Q3$, $IC3$, and $IC6$ are mounted to the case for heatsinking.

2.4.2. The Current Driver

The Libbrecht-Hall circuit, originally published in 1993, has been one of the primary circuits used for low-noise current drivers for laser diodes [**192, 193, 194**]. A detailed description of the DC current drive of the circuit is in Ch. 4. For higher-current or high laser diode operating voltage applications, R_{SENSE} can be replaced with a lower value resistor, typically either 33 or 40 Ω. Additionally, other Libbrecht-Hall implementations exist for high-current and high-compliance-voltage applications [**195**]. Our implementation of the circuit (Fig. 2.11 with schematic and board in App. A.1.3), powered by

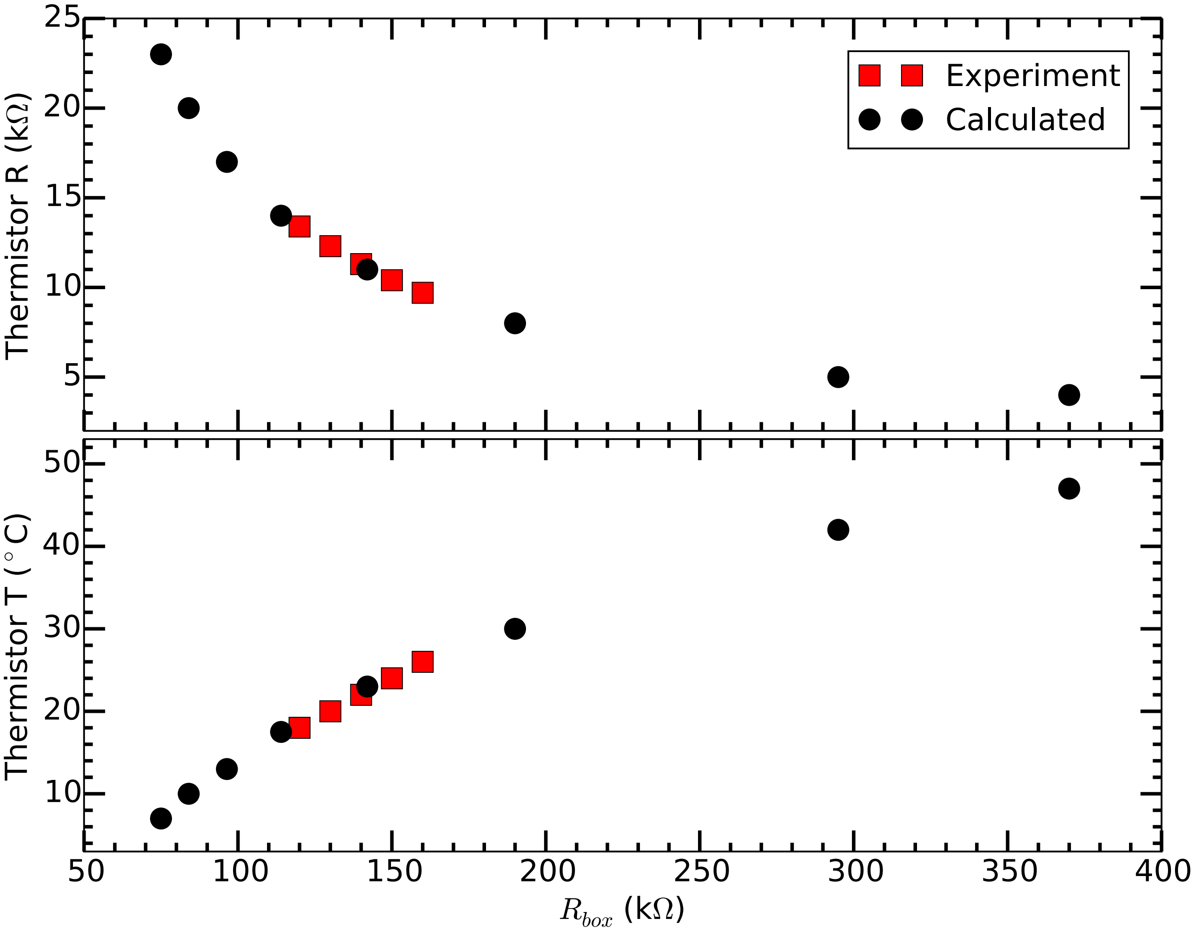


Figure 2.10. Temperature controller R_{box} resistance to thermistor resistance and measured temperature. Temperature is measured using the second thermistor in an ECDL driven by the temperature controller.

±24 V, also contains additional line regulation via a capacitor and inductor π-filter, a current-modulation section with current feed-forward, an external modulation input for laser frequency locking, and a laser output safety interlock.

The circuit uses an IRF510 MOSFET connected to ground to control the laser drive current state (on or off), which is clever since the current drive electronics stay at a steady operating current and therefore temperature regardless of if current is flowing through the

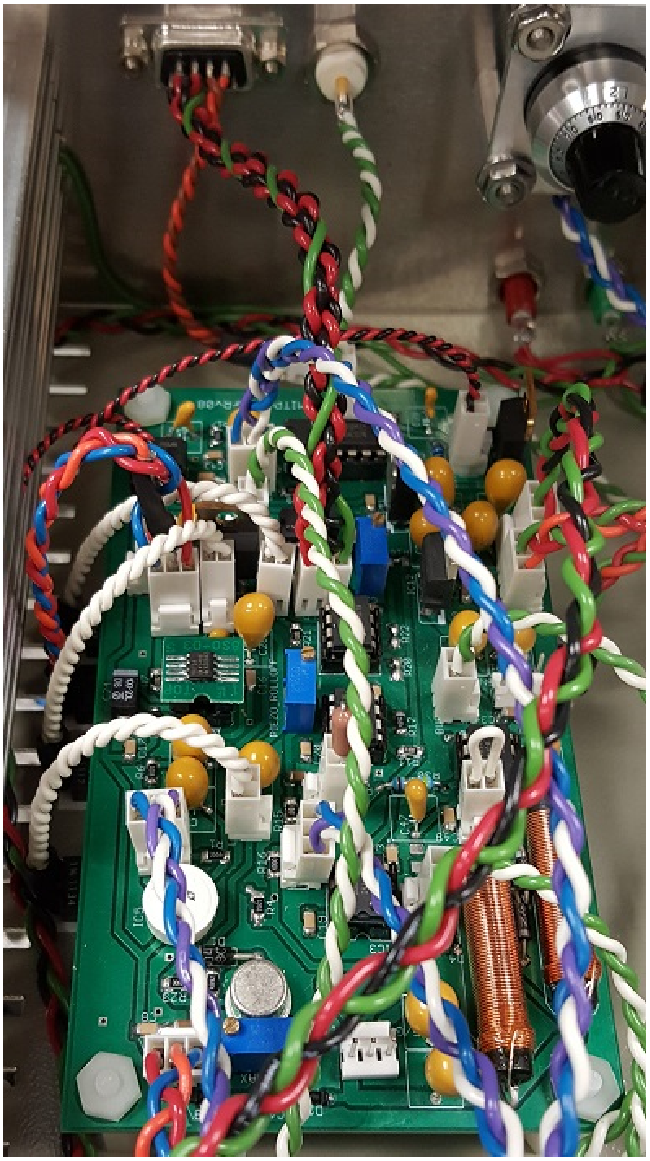


Figure 2.11. Picture of an assembled current driver in an ECDL electronics system. All wiring is 22 AWG.

laser diode. When this MOSFET is in the off-state (high-impedance from source to drain), the output voltage of the current driver will drive current through the laser diode. When this MOSFET is in the on-state, it shorts the output of the current driver circuit to ground resulting in only a small leakage current flowing through the laser diode. This switching

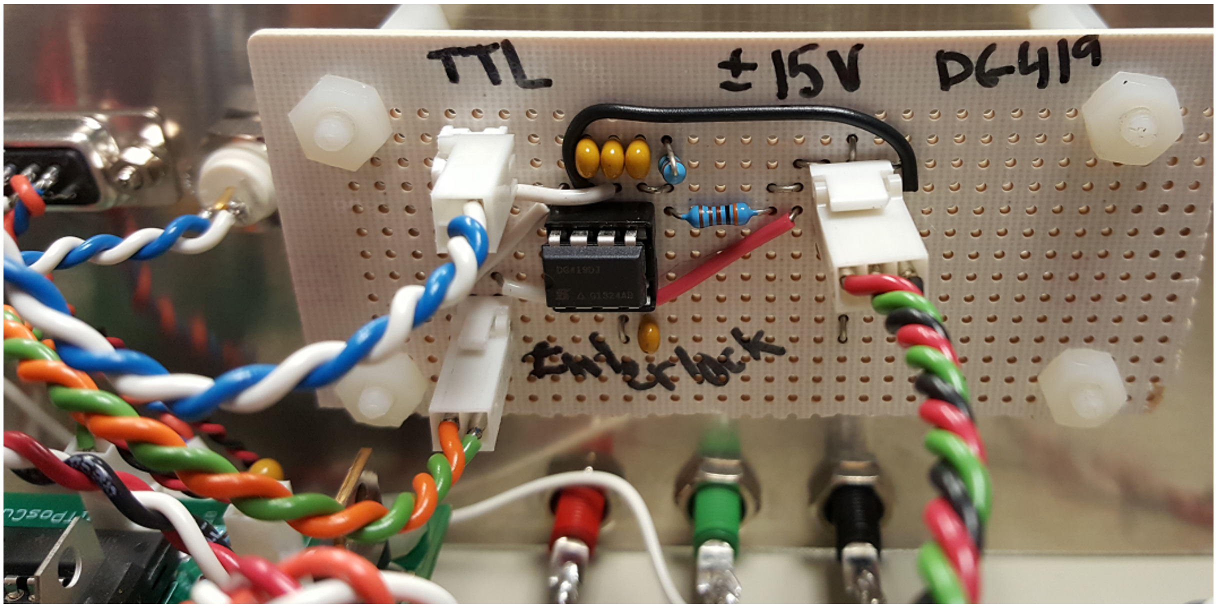


Figure 2.12. Picture of the circuit that enables current driver output switching using an input TTL signal. This specific example is integrated into the UV LED driver (Sec. 13.1.2). All wiring is 22 AWG, and the circuit diagram is located in App. A.1.10.

is controlled via a DG419 solid state switch and a low-pass filter to the MOSFET gate for a slow on/off response. The switch is controlled by the output of a NAND gate with one channel being an enable switch on the front panel and the other being the output of an interlock circuit, preventing laser output when the interlock pins are not shorted. Additionally, if the on/off state of the controller is to be externally controlled by computer or other laser when injection-locked, a secondary circuit can be used (Fig. 2.12) to turn the system on and off using an input logic signal and the interlock circuit, implemented by a DG419 chip opening or shorting the interlock connection (App. A.1.10).

The modulation section of the circuit uses an instrumentation amplifier (INA114 or INA217) with input signal to a BNC on the front panel. Because of this, the modulation input section should not be left open or the INA output will rail. A potentiometer

MOD_{OFFSET}, used for wavelength tuning by hand via the front panel, provides a maximum offset voltage of -10 V to the INA output. This signal is then split to the PZT section and current feed-forward section.

For the PZT section, the signal is inverted and low-pass-filtered with trimmer typically set to the minimum low-pass filter frequency to avoid exciting acoustic resonances in the ECDL. The resulting signal is buffered and output to the laser front panel to connect to a PZT driver (Sec. 2.4.4).

The current feed-forward section uses a voltage divider with trimmer to another INA, which uses a resistor between output pins with the reference pin shorted to the output of Q_{CTL}. Note that the current feed-forward and modulation section is designed to only *sink* current with MOD_{OFFSET}. The input modulation signal should be restricted to $\sim \pm 1$ V to prevent this section from over-driving the laser diode, which can be performed using an external BNC enclosure box with rectifier diodes.

Additionally note that the modulation sections of the circuit can be omitted when the circuit is used to drive a free-running laser diode, injection-locked laser diode, or LED. In the ECDL controller, the 7815 regulator, LM317 regulator, Q_{CTL}, R_{SENSE}, R_{MON}, and LPF_R are case mounted for heatsinking. Additionally, a 77F101J-TR-RC 100 μH inductor is soldered to the Q_{CTL} MOSFET output to prevent the possibility of an RF-modulation signal destabilizing the circuit.

2.4.3. The Protection Circuit

A laser diode protection circuit (Fig. 2.4 with schematic and board in App. A.1.4) protects the laser diode from ESD events originating from the ECDL controller. It additionally includes a 10-turn potentiometer to short the laser diode anode and cathode for added laser diode protection when cables are unplugged or new electronics are connected to the system. A 10-turn potentiometer is used because a 1-turn unit typically does not adequately short, and there are debouncing concerns if a mechanical switch is used.

To filter the electrical connection to controller, the circuit uses bypass capacitors to ground on the photodiode pins, bypass capacitor between laser diode anode and cathode, a rectifier diode from laser diode cathode to anode to protect against reverse-biasing the laser diode, multiple rectifier diodes in series to protect against excess forward voltage over the laser diode with typically one rectifier diode per 0.5 V of laser diode operating voltage, 10 μH inductors (Ferroperm 1582 2.2UH 1.8A), and a 1N5711 Schottky diode immediately before the output pins to the laser diode. Jumper pads are included for solder connections to set the photodiode anode and cathode connections appropriately; this makes the circuit modular for all laser diode pinout configurations.

A laser diode socket is used to connect to the laser diode (3M 203-6585-50-0602J, 203-6585-50-0602J, 204-6585-50-0602J, or 204-6585-50-0602J depending on laser diode case size and number of pins). Note that the socket diameter must be machined down on a lathe to fit inside of the laser diode mounting ring.

Both direct capacitive and resistive (including phase-lead components) laser diode modulation inputs are included with BNC inputs directly on the ECDL enclosure.

2.4.4. The PZT Driver and Wavelength Locking

To avoid building a low-noise, fast-response PZT driver, a ThorLabs MDT694B or MDT693B open-loop PZT controller is used. Note that some PZT channels contain an extra input low-pass filter to address acoustic resonances in the ECDL system (Sec. 3.4.1).

If a commercial current driver is used with the specific ECDL system, the PZT driver modulation port is connected directly to the wavelength lock computer (Sec. 2.6). The laser wavelength is manually tuned with PZT voltage and laser diode drive current with no current feed-forward, although this could be implemented with a secondary circuit or second DAC channel.

If a laboratory-built current driver is used, the driver's buffered PZT output connects directly to the modulation input of the PZT driver. In this case, the wavelength-locking signal connects to the current driver modulation input for the current feed-forward feature. The laser wavelength is then typically scanned using the modulation offset nob on the current driver with occasional tuning of the PZT offset voltage on the PZT driver.

2.5. Diode Laser Systems

Multiple commercial and in-house-built ECDL systems are used throughout the duration of this work. This section discusses most past, current, and near-future diode laser systems. A table of several diagnostic parameters for select in-house-built ECDL systems is in Table 2.5.

Table 2.5. Characteristics and diagnostics data for select in-house-built ECDL systems. Center wavelength (CW) is given as that operating free-running (FR) at 25°C and operating output power and mode-hop-free tuning range (MHFTR) with current feed-forward (FF). MHFTR values denoted with a ">" are due to the measured value being limited by the PZT driver voltage and/or drive current.

System (nm) Diode	CW (nm)	FR nm/°C	FR nm/mA	FF mA/V_{PZT}	MHFTR (GHz)
413 SAL-415-030	415.68	0.061	0.014		
455 NDB4216E	455.76	0.04	0.018		
493 NDS1316	493.2846	0.0067	0.030	-0.084	>58
657 HL6545MG	661.44	0.1716	0.01652	-0.35	>55
780 Reference LD785P090	784.5267	0.24	0.017	-0.46	>68
780 for JTM LD785P090	784.2108	0.12	0.016		
780 for JTM QLD-780-150S	778.0389		0.018		>7
780 for JTM LD785-SE400	785.1558		0.014	-0.55	
791 QLD-795-150S	794.0442	0.28	0.020	-0.82	>44

2.5.1. 413 nm Barium Photoionization

The 413 nm barium $^1S_0 \rightarrow {}^3D_1$ intercombination line $(1+1)$ REMPI [196, 197, 198] photoionization laser diode is a Sacher Lasertechnik SAL-415-030 AR-coated laser diode (S/N #14) with a Newport KGA110-A collimation lens and Newport 05HG3600-300-1 3600 line/mm grating. A Newport 6100 controller is used to drive the laser diode due to the high operating voltage of the InGaN semiconductor, and in-house laser control

Table 2.6. 413 nm barium $^1S_0 \rightarrow {}^3D_1$ optogalvanic spectroscopy fit results. From this data, the Doppler-broadened FWHM is 1.1 GHz.

Parameter	A	c_w	w	D
Fit Value	-2.971(16)	413.3590569(13) nm	2.72(2) × 10^{-4} nm	-0.573(13)

electronics were not yet available. The wavelength is initially set the NIST-reported value of 413.35922 nm [199] and finely-tuned using a Doppler-broadened absorption line in a Hamamatsu L2783-56NE-BA barium laser galvatron using optogalvanic spectroscopy [200, 201, 202, 203, 204, 205, 119, 206]. The wavelength is later fine-tuned using the apparatus described in Ch. 8 by maximizing the fluorescing ^{138}Ba$^+$ loading rate.

2.5.1.1. 413 nm Optogalvanic Spectroscopy. To tune to the 413 nm Ba resonance, Doppler-broadened optogalvanic spectroscopy is used (Fig. 2.13) in which the optogalvanic signal is measured using a optical chopper wheel and lock-in amplifier. A LabVIEW program (App. C.1.1.1) records the wavelength meter reading and optogalvanic signal from the lock-in amplifier using a National Instruments ADC card with laser frequency tuning performed by hand. An inverted Gaussian fit function given by

$$Ae^{\frac{-(x-c_w)^2}{2w^2}} + D \tag{2.8}$$

is used to fit the data where A is the signal amplitude, c_w is the center wavelength, w is the Gaussian half-width, and D is a background term. Fit results are given in Table. 2.6.

From this data, the barium $^1S_0 \rightarrow {}^3D_1$ transition is 413.3590569(13) nm, which is +290 MHz from the NIST-reported value of 413.35922 nm [199] likely due to the wavelength meter absolute calibration being unreliable during this laboratory epoch. Additionally, the line is substantially Doppler broadened as expected with a FWHM of 1.1 GHz.

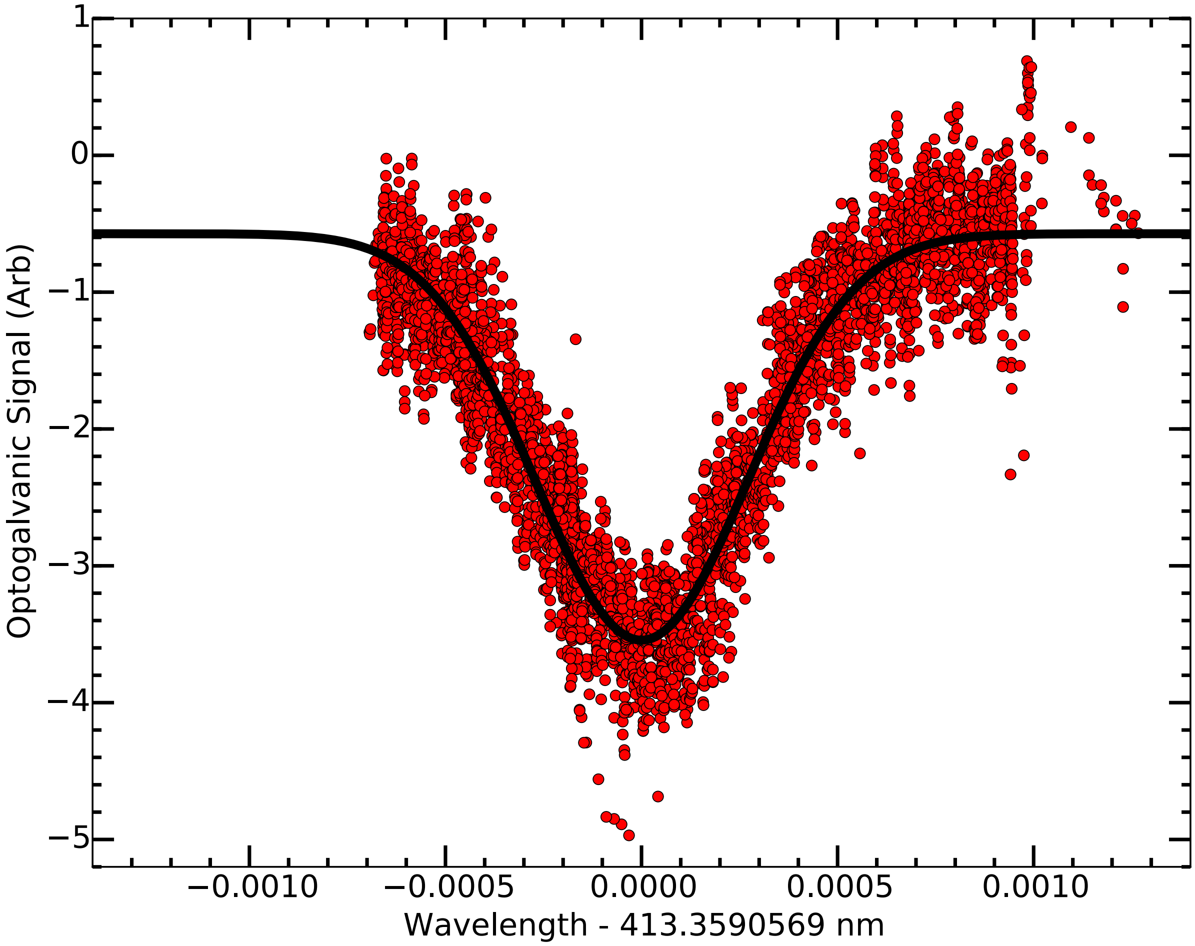


Figure 2.13. 413 nm barium $^1S_0 \rightarrow {}^3D_1$ optogalvanic spectroscopy. A fit (black line) to the data (red circles) is discussed in the text with fit results located in Table 2.6.

2.5.1.2. 413 nm Laser Diode Degradation. After a few weeks of operation, the diode underwent significant degradation with low output power and no change in drive voltage (Fig. 2.14), a dark-spot in the middle of the output beam profile (Fig. 2.15), and monotonic shifting of the output wavelength towards the blue at a given set current. With the collected data, the diode likely underwent an ESD event causing forward surge damage to the middle of the active region of the laser diode chip [**207**, **208**]. Once a defect in the semiconductor is formed, the area absorbs additional optical power, further heating the area, growing the damaged volume. This explains the slow degradation of the laser output over the course of ∼1 week. However, it is unclear when or where the initial damage occurred.

2.5.2. 455 nm Barium Shelve

The 455 nm free-running laser uses a Nichia NDB4216E laser diode (S/N 12B10AC 140) with ThorLabs 355230-A collimation lens. A grating was temporarily installed in order to narrow the laser linewidth sufficiently such that the shear plate interferometer fringes could be observed for collimation. In-house electronics drive the ECDL system, which are powered by a pair of Acopian L24-MC390-B158-C79-E23-F1-G3-K37 24 V supplies along with a UV LED (Sec. 13.1.2) and 650 nm injection-locked laser system (Sec. 2.5.6.3) when used. Since the laser diode has four pins, the socket is indexed to pin 1 and should be facing up. Because this laser system is not an ECDL, no optical isolator is used. However, back-reflections are carefully aligned to be out of the main beamline and FC/APC fibers must be used. Diagnostics information is located in Table 2.5, and the laser was tuned to the $^{138}\mathrm{Ba}^{+}$ $^{2}\mathrm{S}_{1/2} \rightarrow {}^{2}\mathrm{P}_{3/2}$ resonance using the drive current (Sec. 13.7.4). From that data, the laser linewidth FWHM is 51(2) GHz.

Additionally, if the 455 nm laser source is converted to an ECDL, the system could be locked to a Te_2 line lying ∼80 MHz from the $^{138}\mathrm{Ba}^{+}$ transition [209].

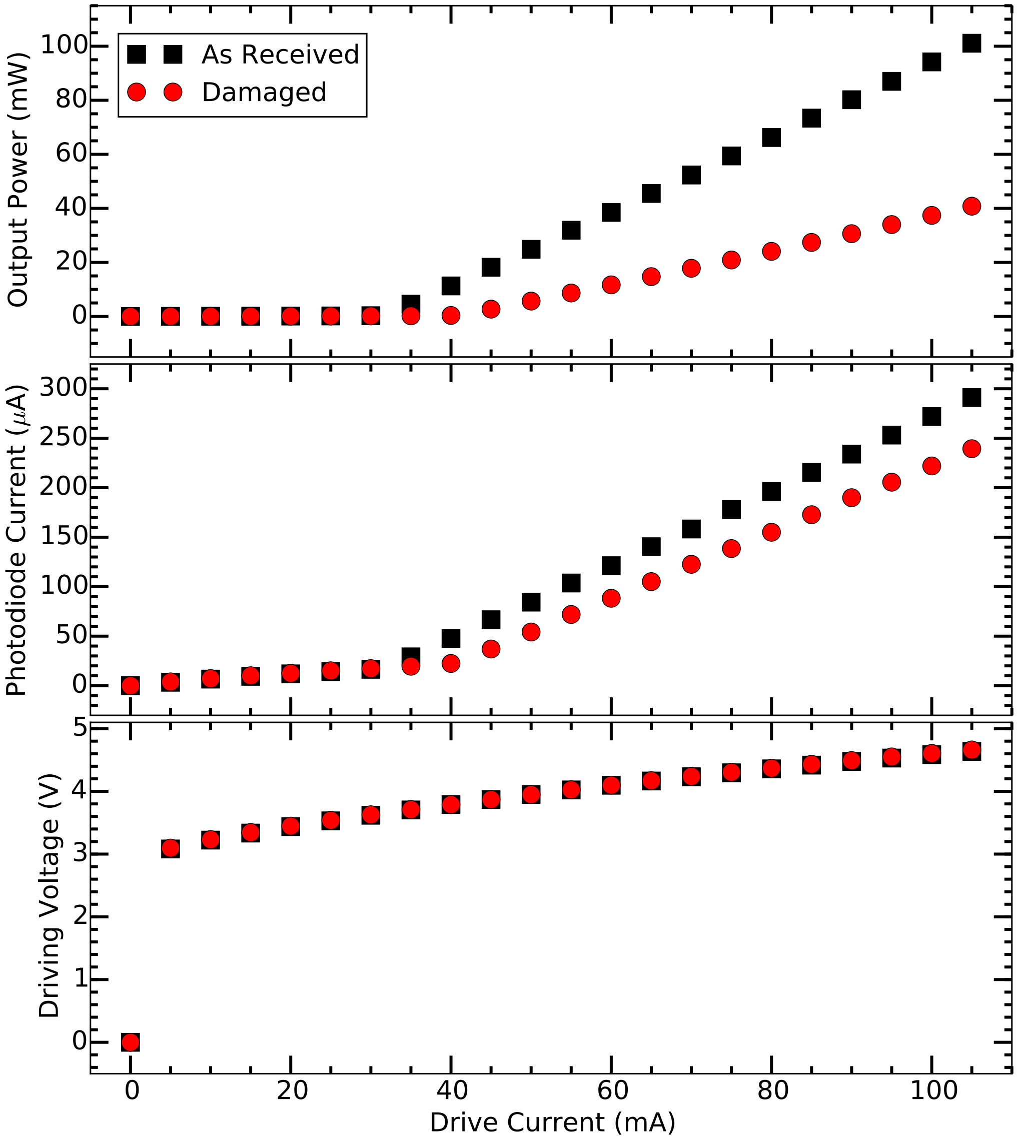


Figure 2.14. Output power, photodiode current, and driving voltage plotted versus laser diode drive current as received and after significant degradation is observed. This data is collected using a Newport 6100 controller.

2.5.3. 493 nm Toptica Barium Doppler

The 493 nm laser source for $^{138}Ba^{+}$ Doppler and near-resonant Raman sideband cooling is a Toptica 987 nm DL-Pro ECDL with BoosTA-Pro tapered amplifier system in a MOPA

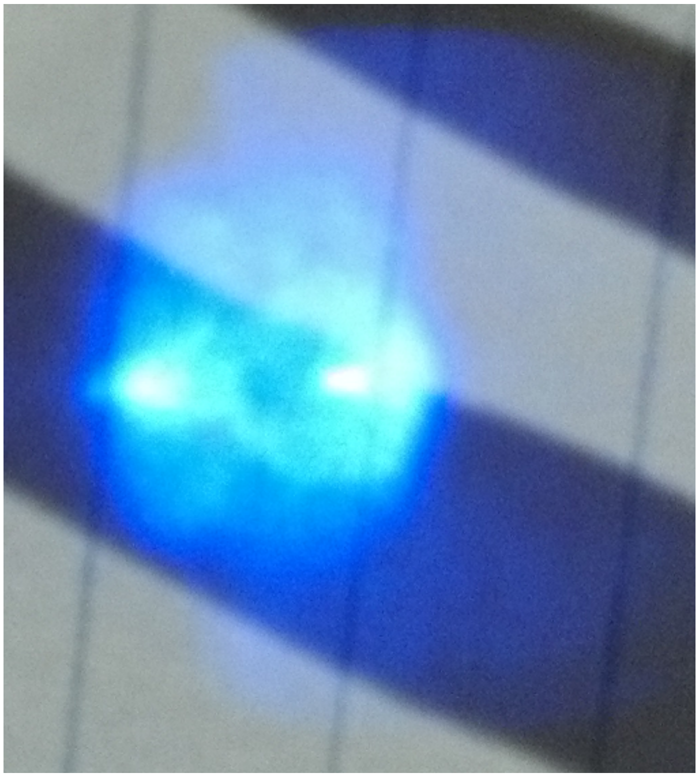


Figure 2.15. Output beam profile of the 413 nm ECDL after significant degradation. Note the large dark region in the middle of the beam profile.

arrangement with a second harmonic generation (SHG) cavity. As originally delivered, the system used a high-power 987 nm laser diode to directly pump the SHG cavity. However, as the laboratory grew, more 493 nm power was needed. Unfortunately, the original tapered amplifier housing that was designed to fit into the 987 nm ECDL head was unavailable.

Because the laser system base plate sits on vibration-isolating feet, which can shift and change the output laser position and pointing, a free-space tapered amplifier system was not considered. The decided solution was to purchase a standalone 987 nm Toptica

BoosTA-Pro system and fiber-couple the seed light from the ECDL to the tapered amplifier and from the tapered amplifier to the SHG cavity with a special fiber being used for the latter to prevent dust particles from adhering to and degrading the fiber via a laser-tweezer interaction. In total, two mirrors are removed to place the Toptica FiberDocks (recesses below the fibers in Fig. 2.16) and two holes are drilled into the external case. Note that constructing an in-house tapered amplifier system was a viable alternative [**210**].

Upgrading the system is non-trivial; the ECDL is realigned and operating parameter space again found with the maximum current lowered substantially such that the output power of the fiber to the tapered amplifier is $\lesssim$40 mW and does not exceed 45 mW. After optimization, the MHFTR is $\sim$37 GHz. The polarization configuration of the FiberDocks is fixed (as delivered, they were improperly configured), and the SHG cavity roughly aligned. The cavity alignment is optimized and locked following the system manual with photodiode gains decreased to yield similar voltage signal amplitudes as before the upgrade; however at higher pump power, the non-critical phase matching potassium niobate ($KNbO_3$) SHG crystal response is unstable, necessitating the operating temperature be decreased. This results in a lower SHG conversion efficiency (Fig. 2.17) but stable cavity operation. The loss in SHG conversion efficiency is compensated with slightly higher pump power to a point since a higher pump power results in lower crystal temperature, which results in lower SHG conversion efficiency. A solution for this would be to replace the $KNbO_3$ crystal with a different crystal that uses critical-phase matching such as beta barium borate (BBO). A cylindrical telescope would then need to be used to correct the introduced astigmatism.

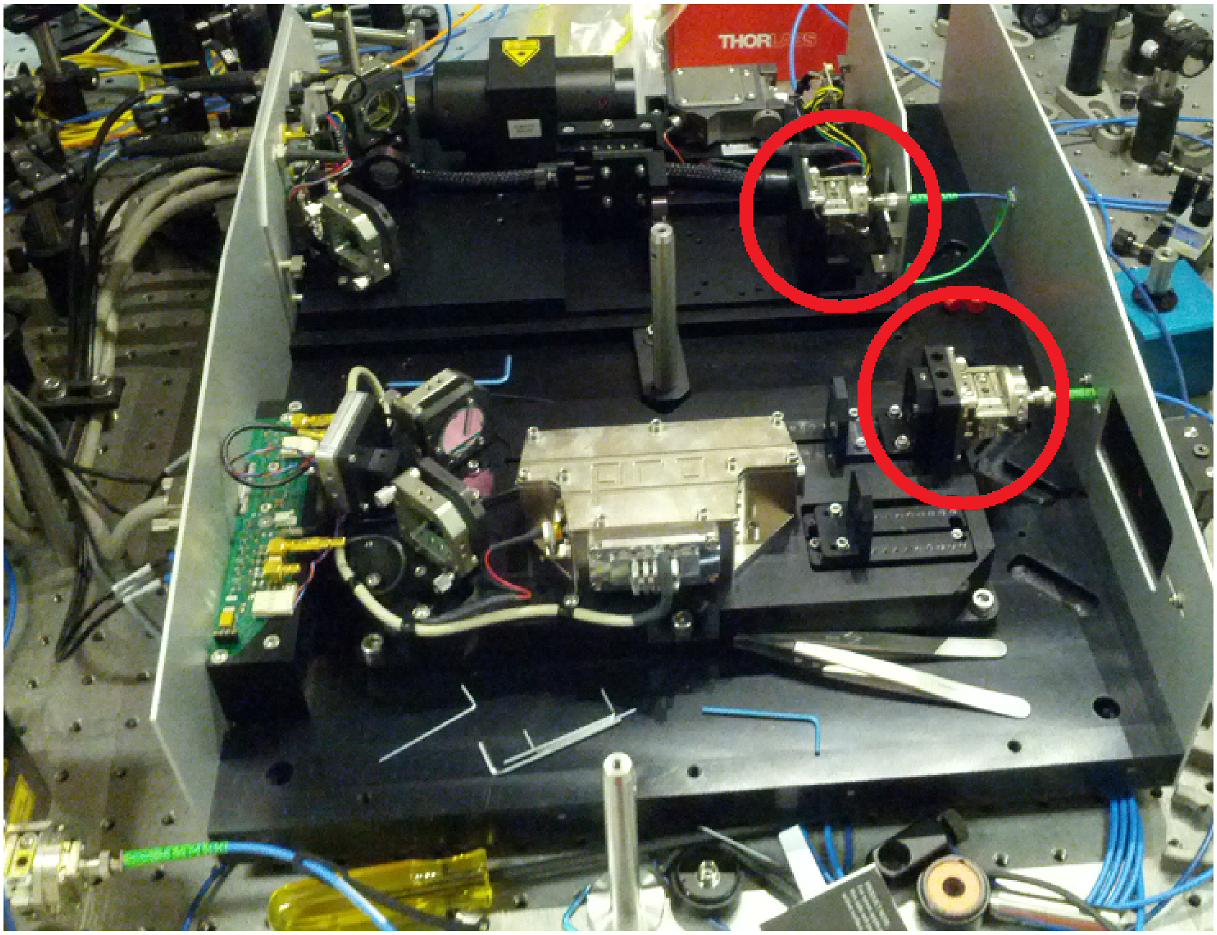


Figure 2.16. Image of the upgraded 987 nm → 493 nm laser system. Note the Toptica FiberDocks (red circles) that are installed to couple light from the ECDL to the tapered amplifier system (not shown), and couple light from the tapered amplifier system to the SHG cavity.

Note that the laser system uses a clever trick with the SHG cavity; the passive stability of the cavity at high frequencies is used to narrow the laser linewidth since the cavity finesse is sufficiently high [211]. This narrows the linewidth of the 987 nm ECDL to increase the intra-cavity power, increasing the SHG efficiency, and helps stabilize the SHG cavity lock. Additionally, the ECDL electronic interface is reconfigured such that the external frequency modulation voltage for wavelength locking is applied via the D-Sub

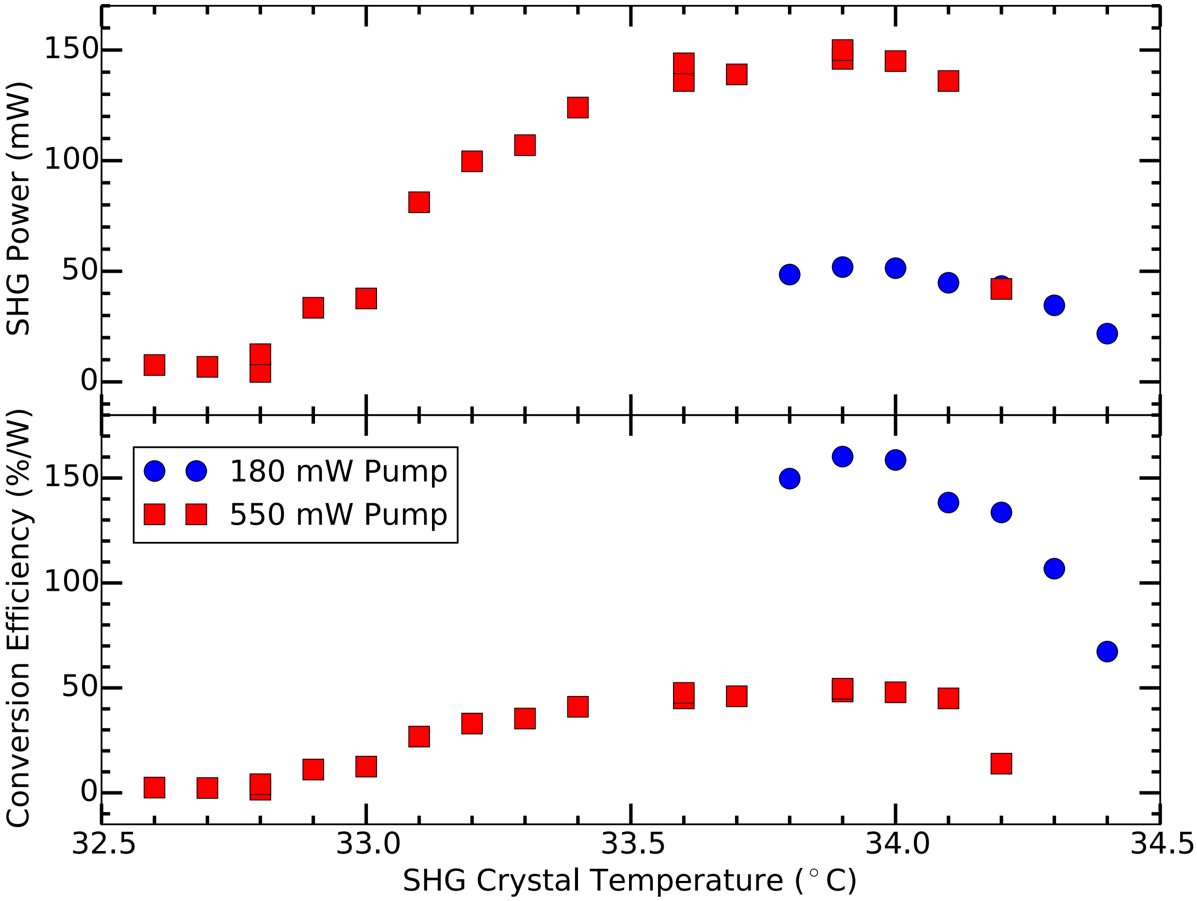


Figure 2.17. SHG output power (top) and SHG conversion efficiency (bottom) as a function of crystal temperature. Note the poor conversion efficiency at higher pump power, and that the crystal must be run at a lower temperature for stable operation.

connector while the BNC connection is wired to a system-on logic signal such that the BoosTA system powers on with the master ECDL.

Additionally, the 493 nm output beam collimation is optimized, and a $f = 100$ mm (Thorlabs LA1509-A) and $f = -75$mm (ThorLabs LC1582-A) shrinks the beam size to be optimal for fiber-coupling with a $f = 7.5$ mm collimation lens with PM460-HP fiber.

2.5.4. 493 nm Barium Far-off-resonant ECDL

The 493 nm far-off-resonant Raman laser source is an ECDL using a Nichia NDS1316 laser diode (S/N CE0000013 61). The ECDL, assembled by MGK, uses a Thorlabs 355230-A collimation lens with Richardson Gratings 33055BK01-430H 2400 line/mm grating. In-house electronics drive the ECDL system, which are powered by a PowerOne ±24 V supply. For future wavelength-locking considerations, $C6$ on the protection circuit is installed for capacitive modulation, and the four-pin laser diode socket is indexed with the laser head and should face towards the corner cube mirror mount. A ThorLabs IO-5-488-LP optical isolator is placed at the ECDL output and a cylindrical telescope, comprised of a $f = 300$ (ThorLabs LJ1558RM-A) and a $f = -100$ mm (ThorLabs LK1743RM-A) lens, partially correct the ellipticity of the output spot. Further beam shaping is not performed due to the poor output spot profile before the light is fiber-coupled using ThorLabs C230TMD-A collimation lens with PM460-HP fiber. Diagnostics information is located in Table 2.5.

At high intra-cavity powers, GaN diode lasers have a propensity to not lase on a Gaussian spatial mode. In order to operate near the desired wavelength with this low-power laser diode, it is operated at high current and thus high intra-cavity power, and the small islands of single-longitudinal mode operation are small and distantly spaced. Additionally, the laser diode chip length times refractive index is $nL = 0.861$ mm for a 174 GHz diode-chip FSR, which significantly limits the tunability of the ECDL. Because of this, the ECDL operates -59 GHz from the $^2S_{1/2} \rightarrow {}^2P_{1/2}$ transition in $^{138}Ba^+$ and the fiber-coupling efficiency is $\sim$15%, both of which are non-ideal for the experiment.

To remedy both the small tunability of the ECDL and the poor output beam shape, a replacement laser diode was purchased. The Toptica LD-0488-060-1 (S/N 77192) replacement is a Nichia NDS4116 that is hand-selected for a center wavelength near our operating wavelength (492 nm) and that has a higher operating power (60 mW free-running) for an improved output spatial mode at the same ECDL output power. Additionally, the replacement laser diode has the same pinout, simplifying the replacement.

2.5.5. 614 nm Barium Deshelve

The 614 nm deshelving laser source is the output of an AdvR Inc. SHG module pumped by an ECDL using an Innolume laser diode gain chip. Due to the low 614 nm power requirements for the experiments discussed in Part. 3, we decided that the module is a faster solution than constructing an SHG optical cavity.

2.5.5.1. 1228 nm ECDL. The in-house 1228 nm ECDL uses an Innolume GC-1220-110-TO-200-B curved stripe laser diode gain chip (S/N DO3490e-q4-BO1-A3) with ThorLabs 355230-C collimation lens and Richardson Gratings 330556K01-548H 1150 line/mm grating. Due to the demanding current requirements of the gain chip, ThorLabs commercial drivers are used (TED 200 C and LDC 210 C) with no current feed-forward. For future wavelength-locking considerations, $C6$ on the protection circuit is installed for capacitive modulation. Due to the high reflectivity rear facet and low reflectivity front facet of the chip, the output wavelength is insensitive to the driving current, and no mode-hops are observed while scanning the PZT voltage over >120 GHz. However, mode-hops may not be observable due to the longitudinal multimoding of the ECDL (Sec. 2.5.5.4). A ThorLabs IOT-5-1225-VLP tandem optical isolator is placed at the ECDL output and a cylindrical telescope, comprised of a $f = 250$ (ThorLabs LJ1267RM-C) and a $f = -50$ mm (ThorLabs LK1336RM-C) lens, corrects the ellipticity of the output spot. No further beam shaping is necessary to couple into the PM1300 fiber with a ThorLabs C230TMD-C collimation lens. Additionally, beam size measurements are performed by recording the transmitted power past a razor blade on a translation stage. An error function fit is then performed to the resulting power versus razor blade position data. Furthermore, due to the wide operating output wavelength range of the laser diode gain chip, the wavelength is initially set using an Agilent 86140B optical spectrum analyzer.

2.5.5.2. SHG Module. SHG is performed with an AdvR Inc. WSH-P13L45AL0-14101701 waveguide potassium titanyl phosphate (KTP) SHG module. The 1228 nm pump light is coupled into the waveguide via PM1300 fiber (9.5 μm MFD at 1300 nm) with an FC/APC connector. The unit itself is mounted on a modified ECDL heatsink base, which is mounted in a custom enclosure (App. F.1.5) with electrical connections for temperature control (Fig. 2.18). The input C230TMD-C collimation lens is mounted in a S1TM09 adapter in front of SM1FCA fiber adapter plate in the side of the enclosure. The resulting 614 nm output is collimated, and both 614 nm and 1228 nm beams exit the unit with the latter used to couple the 1228 nm pump light into the input fiber. A ThorLabs DMLP90 dichroic separates the harmonics, and each harmonic exits the enclosure through a ThorLabs AR-coated window mounted in a SM1L03T 10° angled mount and is shuttered by a SM1D12SZ zero-aperture iris.

According to the manufacturer, if the pump source is single-longitudinal-mode, the SHG efficiency of the SHG module decreases by roughly a factor of two (from 270 %/W to 135%/W). Furthermore, the measured 1228 nm pump output power is roughly a factor

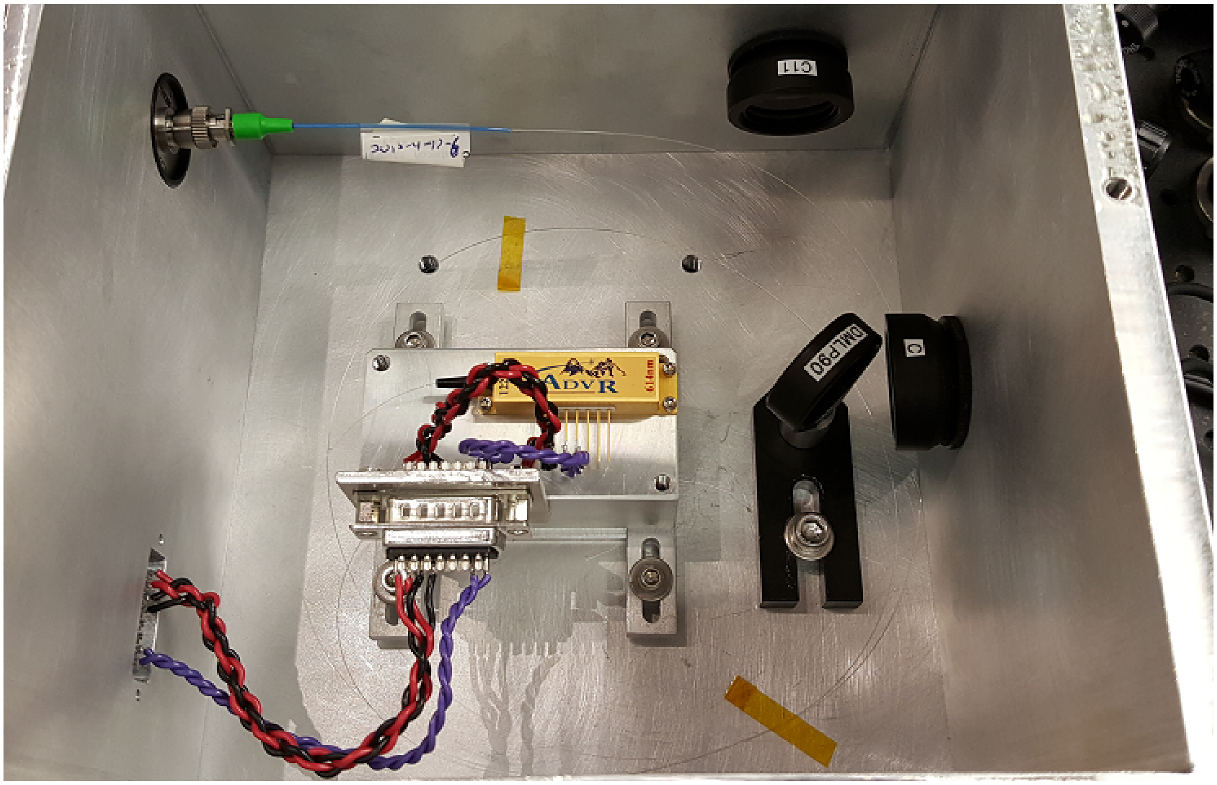


Figure 2.18. 614 nm AdvR SHG unit mounted in a custom enclosure on a modified ECDL heatsink. The fiber attaches to an adapter plate mounted on the enclosure wall. The DMLP90 dichroic separates the harmonics, transmitting the 1228 nm pump and reflecting the 614 nm second-harmonic. Enclosure drawings are in App. F.1.5.

of 2.6 lower than what is coupled to the device due to the output collimation lens not being AR-coated for the pump wavelength and additional internal coupling losses.

Despite the poor output beam profile of the SHG unit (Fig. 2.19), which is typical according to the manufacturer, the 614 nm output power out of the SHG enclosure is split to the wavelength meter using a PM460-HP fiber patch cable and to the apparatus discussed in Part. 3 using a SM600 fiber. Fiber-coupling is performed with ThorLabs CFC-5X-A collimators.

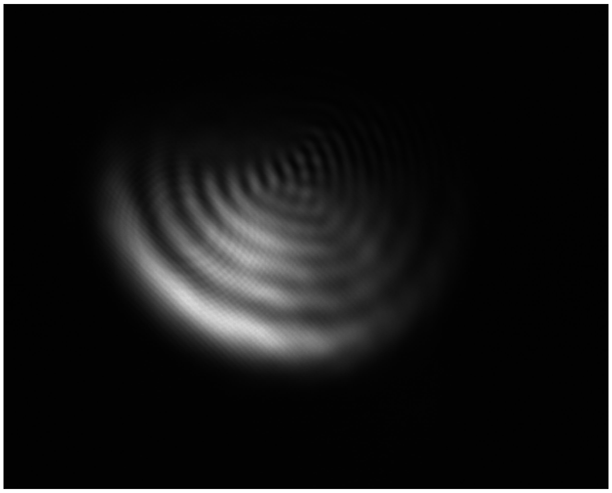


Figure 2.19. 614 nm AdvR SHG module output beam profile taken using a ThorLabs DCC1545M CMOS camera. The non-Gaussian profile is possibly caused by the nature of KTP waveguides, the angled output faced, and/or the 1228 nm laser multimoding problem (Sec. 2.5.5.4).

2.5.5.3. SHG Module Temperature Control. The SHG waveguide is temperature controlled using an internal Redfish Sensors RNBR-103F-3435 10 kΩ thermistor and Marlow Industries SP5446 TEC element. A previous-generation ECDL temperature controller circuit, called TC4SHG and mounted inside an ECDL controller enclosure, is used to employ a LT1185CT voltage regulator, which has a user-defined maximum current output limit, which is set to $\lesssim$800 mA. The circuit diagram is located in App. A.1.9 and a picture of the circuit is in Fig. 2.20. In this application, the R_{box} potentiometer is a 10-turn, 500 kΩ unit in series with a 100 kΩ resistor, the LT1185CT current and voltage limit resistor values are changed, and bypass capacitor values are increased. The potentiometer values are $ER_{GAIN} = 25.02$ kΩ, $P_{GAIN} = 14.79$ kΩ, $I_{GAIN} = 19.77$ kΩ, and $R_{box} = 483.5$ kΩ.

Unlike the ECDL systems, the SHG module does not have a second thermistor for temperature monitoring. In order to measure the temperature of the unit, the voltage across the thermistor is used. Calibration data, collected using both a potentiometer

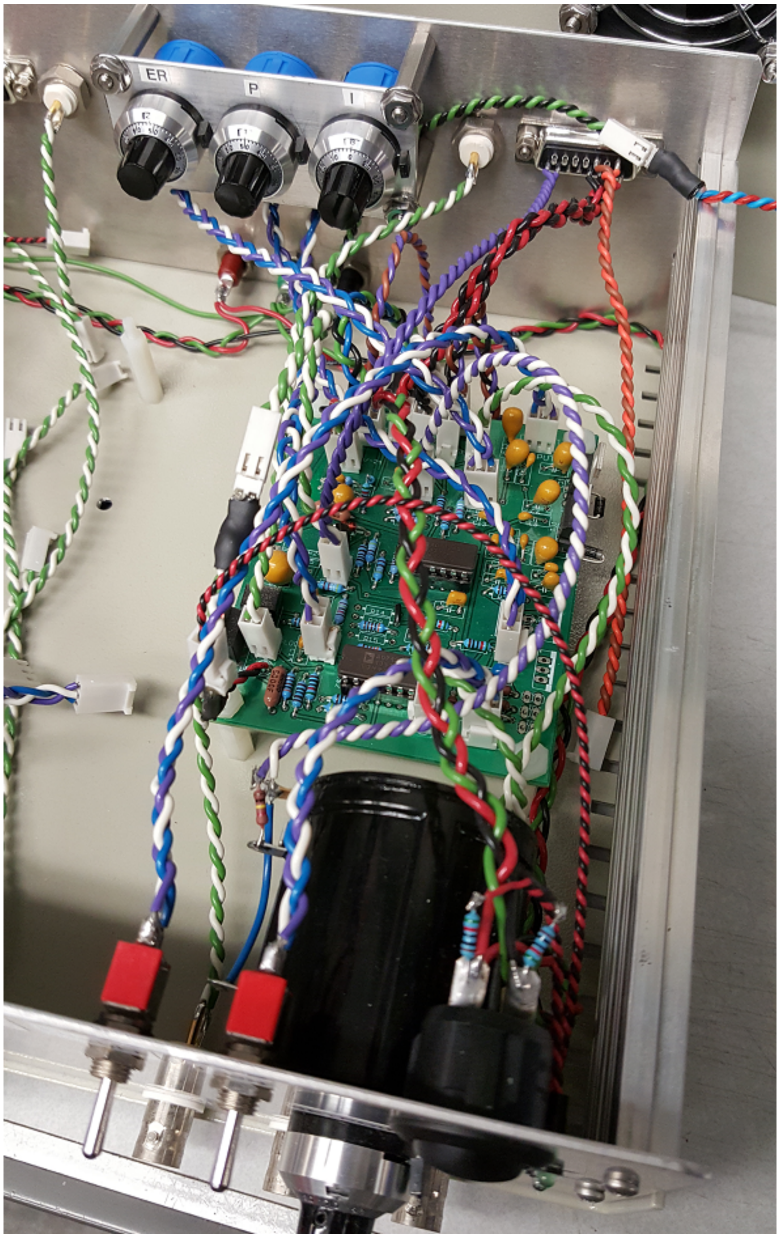


Figure 2.20. 614 nm AdvR SHG module temperature controller. The circuit is mounted in an ECDL controller enclosure. The LT1185CT regulator and Q3 are case mounted for heatsinking. All wiring is 22 AWG.

dummy load and an ECDL system, along with displayed polynomial fits to the data are shown in Fig. 2.22.

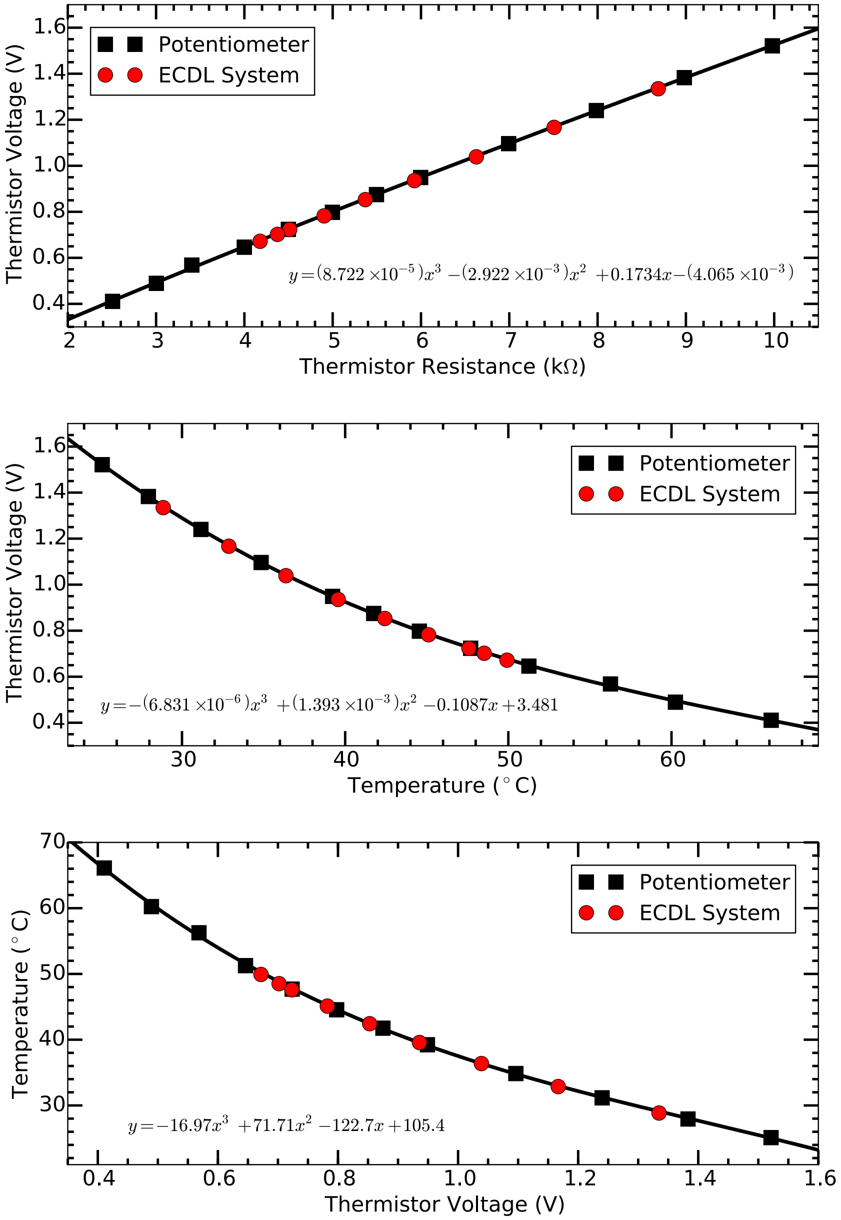


Figure 2.21. 614 nm AdvR SHG module temperature measurements relating thermistor resistance, thermistor voltage, and inferred temperature. Points are data taken either with a potentiometer dummy load or an ECDL system, and the black line is a third-order polynomial fit to the potentiometer data with results located in each figure.

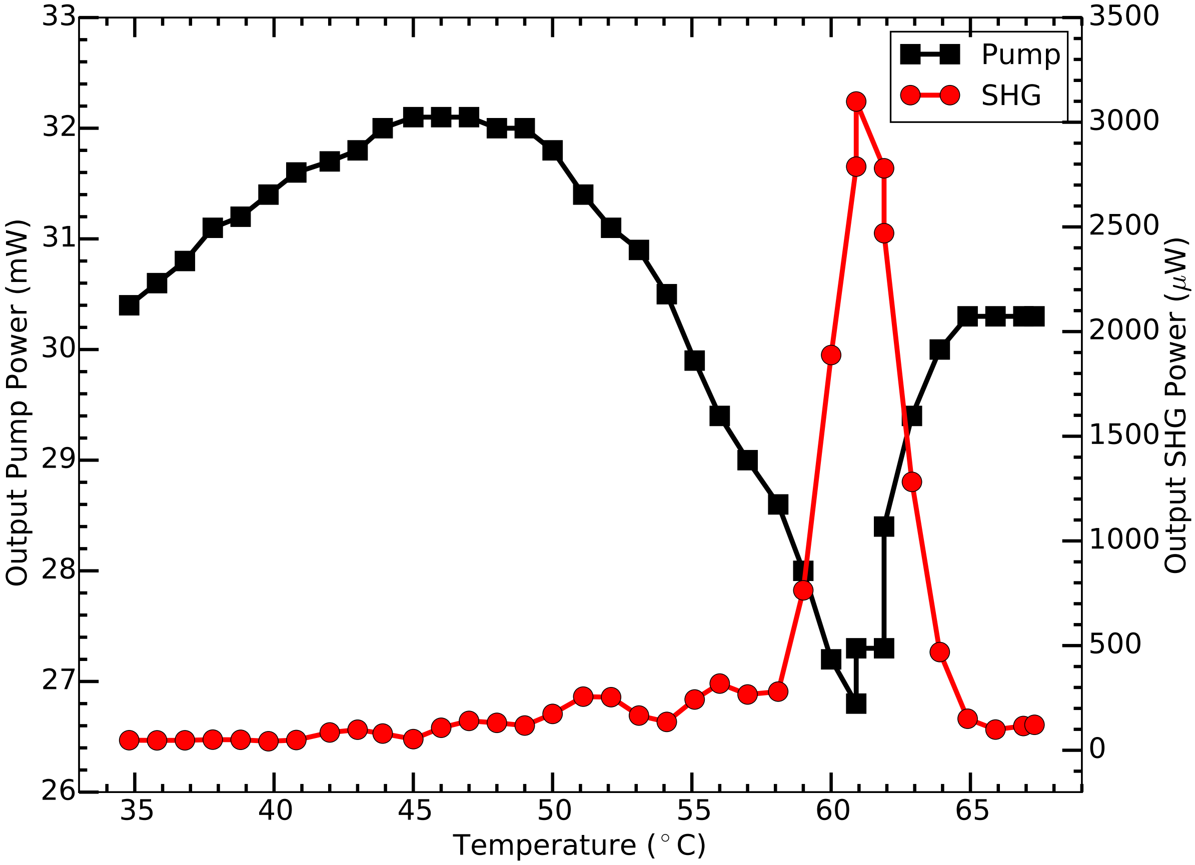


Figure 2.22. 614 nm AdvR SHG module output pump and SHG powers versus waveguide temperature. Solid lines connect data points and are meant to guide the eye. From the data, the optimal temperature is ~60.9°C ($V_{\text{therm}} \sim 0.4831$ V).

The SHG module output power as a function of temperature is shown in Fig. 2.22. From the data, the highest SHG conversion efficiency occurs at ~60.9°C at $R_{box} = 483.5$ kΩ and $V_{\text{therm}} = 0.4877$ V. Note that the waveguide maximum temperature is 75°C = 1925 kΩ = 2.2823 V_{therm}.

2.5.5.4. Longitudinal Multimoding Problem. Unfortunately, the output of the 1228 nm ECDL contains more than one longitudinal mode as can be seen on the Toptica scanning Fabry-Pérot cavity (Fig. 2.23). Since the scanning Fabry-Pérot cavity has a

1 GHz FSR, the observed modes are likely neighboring external cavity modes, which should be spaced by ∼7 GHz based on a ∼2 cm external cavity length. This multimode behavior is likely due to (1) the rear facet of the 1228 nm laser diode gain chip having a high-reflection coating and the front facet having both a high-quality AR-coating and curved-stripe geometry such that the residual reflectivity is $\sim 10^{-5}$, and (2) the grating having a low line number such that multiple longitudinal modes can couple back into the laser diode gain chip, which is assisted by the curved-stripe geometry of the device. Thus, there is insufficient loss at neighboring external cavity modes to suppress of lasing at these frequencies.

This longitudinal multimode problem can be possibly resolved in three ways: (1) the grating line number can be increased [**212**, **213**, **214**], (2) the geometry of the system can be changed, and/or (3) the gain chip can be replaced with a similar device without curved-stripe geometry [**212**]. For (1), multimoding behavior is not observed in a similar system also using an Innolume laser diode gain chip with the same ThorLabs collimation lens but operating at 1156 nm with a 1200 line/mm grating (Edmund Optics W48-464) [**212**]. Replacement gratings with 1312 line/mm spacing (Richardson Gratings 33055BK01-143H) are in-stock to implement this potential solution. For (2), the spatial extent of the output light that couples back into the laser diode gain chip can be suppressed by rotating the ECDL geometry [**215**, **213**, **214**]. By rotating the laser diode by 90° and placing a HWP (Tower Optical Z-12.7-A-500-N-1228) between the collimation lens and the grating in an ECDL waveplate mount (App. F.1.1) to rotate the polarization back by 90°, the undesired external mode coupling will be suppressed. However, this reduces the spatial resolution of the grating, and therefore would be a secondary solution to (1). For (3),

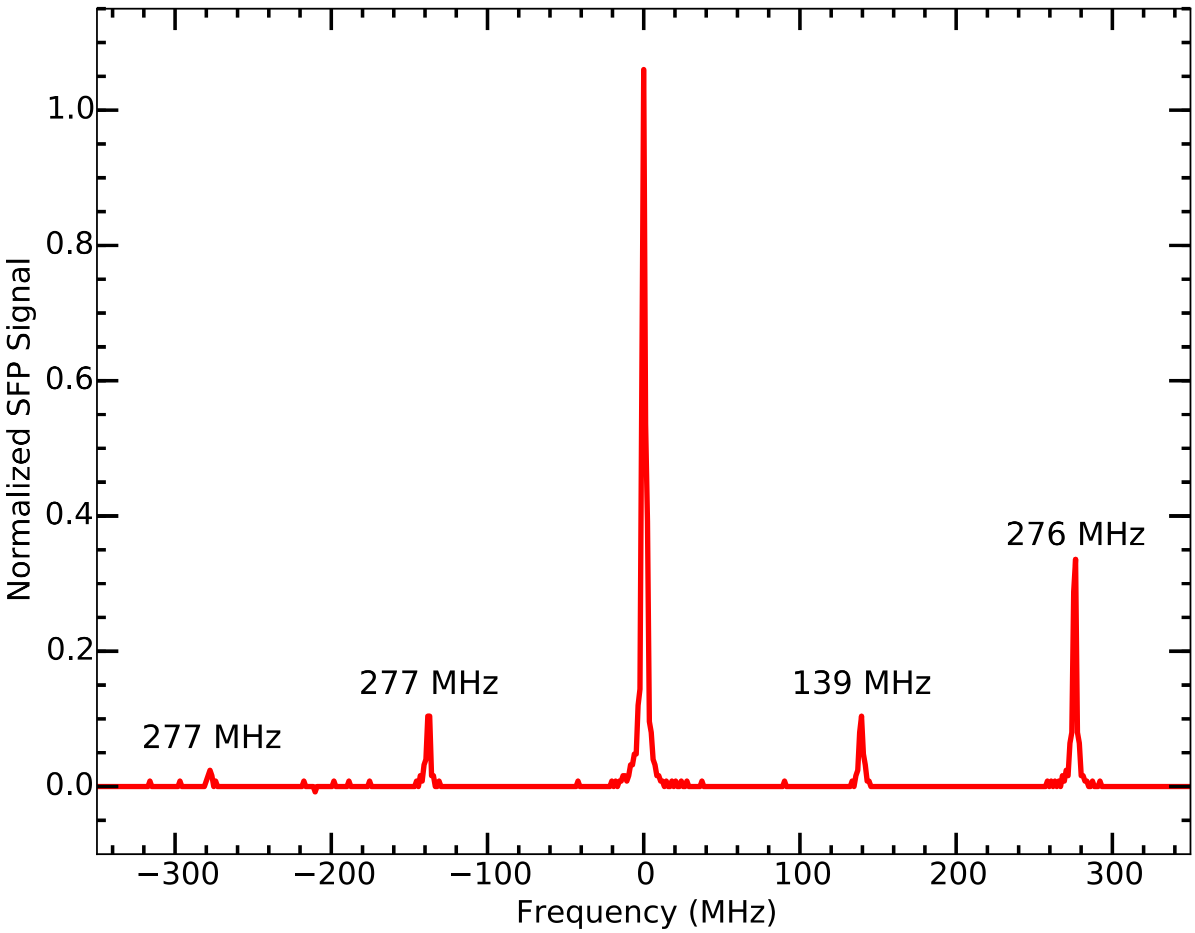


Figure 2.23. 1228 nm ECDL output spectrum as observed with the Toptica scanning Fabry-Pérot cavity. Note that while the additional lasing modes observed here are only spaced by <600 MHz, the FSR of the cavity is 1 GHz. Therefore, the modes are likely space by the observed spacing plus a multiple of 1 GHz.

using a non-curved-stripe geometry would increase the reflectivity of the front facet of the laser diode gain chip, suppressing lasing on neighboring external cavity modes.

2.5.6. 650 nm Barium Doppler Repump

The primary Ba$^+$ 650 nm $^2\mathrm{D}_{3/2} \rightarrow {}^2\mathrm{P}_{1/2}$ Doppler repump laser is a Toptica DL100 ECDL, and various additional injection-locking or ECDL options have been explored. At this wavelength, few non-AR-coated laser diode options are commercially available with $\gtrsim$10 mW of power. However, several AR-coated laser diodes operate at this wavelength.

2.5.6.1. Toptica DL100 Master. The 650 nm Doppler repump laser is a Toptica DL100 ECDL, which uses a mechanical design similar to our in-house ECDL design only with a significantly-shorter grating arm. The corner cube mirror mount and mirror are significantly lighter than in our design, which suppresses acoustic coupling to the environment and imparted jitter in the frequency spectrum. Additionally, the laser diode is AR-coated with a measured laser output beam size of 637 by 4284 μm MFD. A pair of cylindrical lenses rounds the output spot and a second optical isolator (ThorLabs IO-5-650-LP) prevents back-reflections from the FC/PC fibers from complicating the output spectrum of the laser. A fraction of the laser power is sent to the wavelength meter lock using a window and fiber-coupling after the isolator before the remaining power is distributed to various experiments via fiber.

2.5.6.2. Alternative Laser Diode Options. Several alternative high-power single-mode laser diodes are available at the nearby wavelengths of 642 and 658 nm at 25°C. With the AlGaInP semiconductor temperature tuning of ~0.2 nm/°C, these options would need to be operated above ~65°C or below ~0°C. The former temperature is typically greater than the absolute maximum rating of the device, and the latter, while non-damaging to the laser diode, is a significant technical challenge. Both options were explored to some extent without substantial success. In particular, an Opnext HL6545MG laser diode was cooled to -25°C; while the output spectrum was not suitable for injection-locking at 650 nm, the device would likely operate at 650 nm in an ECDL configuration. However, condensation on the exterior of the enclosure and output window was problematic (the interior was purged with CO_2). Because of these technical challenges, AR-coated laser diodes are employed.

2.5.6.3. Sacher Injection Lock. For a time, the apparatus in Ch. 13 used an injection-locked Sacher Lasertechnik SAL-650-10 AR-coated laser diode (S/N 4147-06) on the same optical table. The system is driven by in-house electronics systems for injection locking powered by a pair of Acopian L24-MC390-B158-C79-E23-F1-G3-K37 24 V supplies along with a UV LED (Sec. 13.1.2) and 455 nm laser system (Sec. 2.5.2).

In this arrangement, a cylindrical telescope immediately follows the laser system to decrease the ellipticity of the beam and a HWP rotates the output polarization to that needed for the OFR IO-5-650-LP optical isolator. The output polarizer of the isolator is replaced with a PBS on a kinematic platform. Because the output polarizer housing is magnetic and contributes to the total magnetic field of the isolator, the Faraday rotation is no longer ideal and the optical isolation is decreased. Light from the Toptica DL100 master laser (∼10 μW) enters the apparatus area by Thorlabs PM630-HP fiber, is collimated, passes through a HWP, and is injected into the slave laser diode using a steering mirror and the PBS behind the isolator. The ∼2 mW output of the laser is verified to be injection-locked by observing interference fringes from a Mach-Zehnder interferometer [**154**] next to the laser system. Due to significant ASE in the output due to low injection-locking power and broadband self-lasing of the AR-coated diode, this system was abandoned.

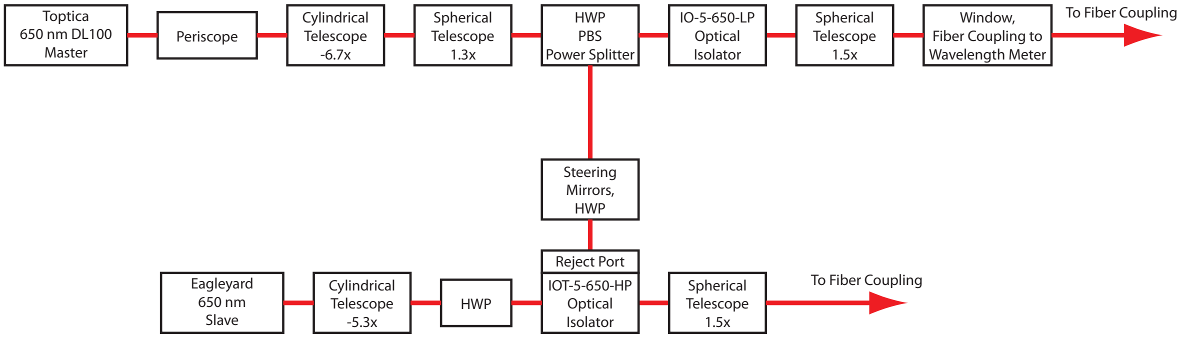


Figure 2.24. 650 nm beamline optical block diagram for both adding the Eagleyard slave laser and optimizing the output beam spot size for fiber coupling. All telescope lenses are listed in Table 2.7.

2.5.6.4. Eagleyard Photonics Injection Lock. Because non-AR-coated laser diode options at 650 nm are low-power, an Eagleyard Photonics injection-locked system is planned, which will at least double the available 650 nm power available in the laboratory. The AR-coated EYP-RWE-0650-00502-2000-SOT02-0000 laser diode gain chip (S/N AG-04364) is mounted in the in-house ECDL system optomechanics with in-house electronics, and the output is collimated with a ThorLabs 355230-B aspheric lens. Because the laser diode will not lase without external feedback, a grating is temporarily used such that shear plate interferometer fringes are observed for collimation. Additionally, the beam size taken with the grating in place measures 825 by 4330 μm MFD. For optimal fiber coupling, enlarging the beam size by 1.5x after a cylindrical telescope would be optimal for the ThorLabs $f = 6.24$ mm A110TM-A lens. The beamline is shown in Fig. 2.24 with telescope lenses listed in Table 2.7.

To injection-lock the gain chip to the Toptica ECDL master, the injection beam must be mode-matched to the gain chip beam profile. Additionally, the Toptica DL100 master beamline requires optimization for fiber-coupling with the output beam currently both

Table 2.7. Telescope details for those used in Fig. 2.24. All telescope lens distances were optimized using a shear plate interferometer. All cylindrcal lenses are mounted in ThorLabs CYLCP mounts in 60 mm cage assemblies, and all spherical lenses are mounted in SM1 lens tubes.

Telescope	-6.7x Cyl	-5.3x Cyl	1.3x	1.5 x
Input Lens	LJ1653L1-A	LJ1567L1-A	LC1582-A	LC1120-A
Input Lens f (mm)	+200	+100	-75	-100
Output Lens P/N	LK1982L1-A	LK1037L1-A	LA1509-A	LA1433-A
Output Lens f (mm)	-30	-19	+100	+150
Magnification	6.7	5.3	1.3	1.5

having a poor shape and an improper size for efficient fiber-coupling. To address the above concerns, the correct beam shaping setup for both optimal mode-matching and fiber-coupling is shown in Fig. 2.24 with telescope lenses listed in Table 2.7. Additionally, a DG419 switch circuit (App. A.1.10), switching on a powered-on logic signal from the Toptica ECDL electronics, can be installed into the in-house ECDL current driver for the gain chip such that the slave laser powers on with the master. The Toptica 987 nm ECDL does this with the BoosTA-Pro system (Sec. 2.5.3).

To verify that the gain chip will injection-lock, the gain chip is configured as a ring-laser (Fig. 2.25). In this setup, a fraction of the laser power after the beam shaping optics and the optical isolator is sampled with a HWP and PBS and is injected back into the gain chip using the optical isolator rejection port. After successful alignment, the system injection-locks with the HWP aligned for minimal feedback power (∼12%) and 30 mW of output power at the operating current of 161.1 mA as expected by the gain chip specifications. Thus, the gain chip should be readily injection-lockable to the Toptica DL100 ECDL master and double the available 650 nm laser power in the laboratory.

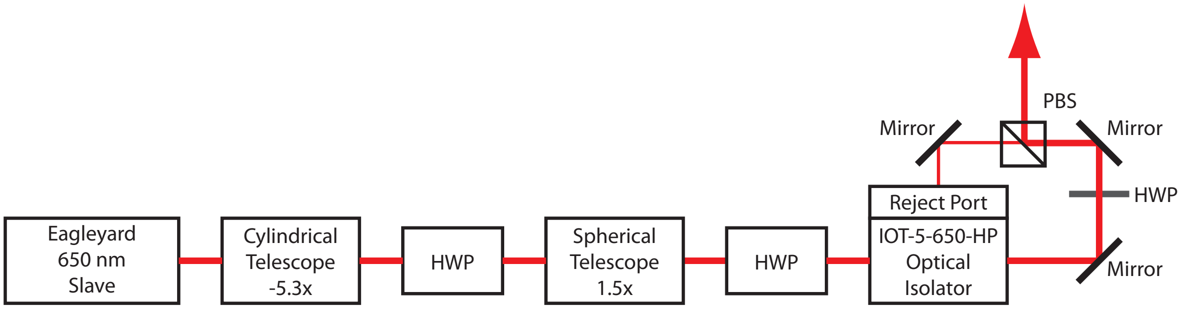


Figure 2.25. 650 nm Eagleyard ring laser demonstration optical block diagram. All telescope lenses are listed in Table. 2.7.

2.5.7. 657 nm Cavity Locking Test ECDL

This ECDL system provides a laser system for the work performed by PRS in Ref. [**216**] and discussed briefly in Sec. 16.2.3; it serves as a test system to develop the expertise and electronics necessary to lock an ECDL to an optical cavity (App. F.4.4) without using a critical laser system necessary for other experiments in the laboratory. The ∼657 nm ECDL uses an Opnext HL6545MG laser diode with ThorLabs A110-B collimation lens and 2400 line/mm grating. The system is driven by in-house electronics, which are powered by an Agilent E3620 supply at ±24 V. Both the resistive ($R1 = 2.5$ kΩ, $R2 = 100$ Ω, and $C5 = 47$ pF) and capacitive modulation ($C6 = 33$ pF) sections of the protection circuit are installed with the capacitive modulation working well by observing driven sidebands on the Toptica scanning Fabry-Pérot cavity. An OFR IO-5-650-LP optical isolator is placed at the ECDL output followed by HWP and PBS for power control. Part of this beam is directed to the Toptica scanning Fabry-Pérot cavity while the remainder is fiber-coupled into a PM650-HP fiber patch cable with output to the optical cavity Pound-Drever-Hall

Table 2.8. Iodine lines near 655 nm that could be used to quantify the temperature-controlled cavity wavelength lock with the ∼657 ECDL or the Barium 650 nm Doppler repump ECDL. The Atlas values are reported in Ref. [217], and the measured values are from Ref. [218].

Atlas	Measured	Measured
Wavenumber (cm^{-1})	Wavenumber (cm^{-1})	Wavelength (nm)
15233.3653	15233.3674	656.453674
15278.5914	15278.5743	654.511331
15297.7275	15297.7125	653.692505
15341.8700	15341.8530	651.811747
15379.2371	15379.2216	650.227967
15406.9515	15406.9375	649.058257

(PDH) and cavity mode-matching setup. Diagnostics information is located in Table 2.5 with free-running data from a different laser diode of the same model.

This wavelength region is chosen because it is sufficiently close to the 650 nm Ba$^+$ Doppler repump transition that the same optical cavity mirror coatings can be used, the laser diode is inexpensive and readily available, and several molecular iodine lines exist nearby (Table 2.8) [217, 218, 219]. The proposed scheme is to lock the ECDL to the optical cavity. Once the lock electronics perform well, the ECDL will be tuned to near an iodine line, locked to the cavity, and be used to monitor an iodine line in a ThorLabs CQ19100-I reference cell using Doppler-free spectroscopy to characterize the stability of the cavity lock.

2.5.8. 780 nm Rubidium EDCLs

Multiple 780 nm systems are constructed for Rb, one to serve as a reference laser and three to be used in experiments by JTM in another laboratory, and are briefly described in this section.

2.5.8.1. Rubidium Reference. The 780 nm Rb reference laser used for the work discussed in Ch. 3 is described in detail in Sec. 3.2.

2.5.8.2. ECDLs for JTM. Three ECDL systems are constructed for JTM: one low-, one medium-, and one high-power. All systems use a Newport KGA110-B collimation lens, Newport 05HG1800-500-1 1800 line/mm grating, and ThorLabs IOT-5-780-VLP tandem optical isolator. Additionally, all use a previous iteration of the ECDL temperature controller and current driver circuit. The PZT output of the current driver connects directly to the PZT, and the output range is increased to 12.6 V. All diagnostics data is located in Table. 2.5.

The low (medium) power ECLD uses a ThorLabs L785P090 (Q-Photonics QLD-780-150S) laser diode.

The high power ECDL uses a ThorLabs LD785-SE400 400 mW laser diode, and is tuned to ~50 GHz blue from the ^{85}Rb, $F = 2$ transition [220]. Due to the high current demands of the ECDL system, the current driver is modified: R_{sense} is a 25 and a 33 Ω resistor in parallel for an effective value of 14.2 Ω, $R_{mon} = 0.1$ Ω with $R3 = 7.4$ kΩ to compensate for the resistance decrease, $LPF_R = 1$ Ω, and the inductor on the MOSFET output is replaced with a higher-current device. The observed linewidth of this system after current driver value changes is ~10 MHz. The spot size at the laser output at 148 mW of output power is 3160 by 1440 μm. A cylindrical telescope, comprised of a $f = 150$ mm (ThorLabs LJ1629RM-B) and a $f = -75$ (ThorLabs LK1431RM-B) lens, focuses the long axis of the beam to correct the ellipticity, and a spherical telescope, comprised of a $f = 150$ mm (ThorLabs LA1433-B) and a $f = -75$ (ThorLabs LC1582-B) lens, shrinks the beam. The resulting output beam, 804 by 683 μm, is then fiber-coupled.

Table 2.9. 791 nm barium ^{1}S$_0$ → ^{3}P$_1$ optogalvanic spectroscopy fit results. From this data, the Doppler-broadened FWHM is 640 MHz.

Parameter	A	c_w	w	D
Fit Value	-0.613(2)	791.3507582(12) nm	$5.69(3) \times 10^{-4}$ nm	-0.496(3)

2.5.9. 791 nm Barium Photoionization

The 791 nm ^{1}S$_0$ → ^{3}P$_1$ intercombination line $(1 + 1')$ REMPI [**221**] photoionization laser source is either a Toptica DL-pro before an in-house ECDL replaced the repurposed Toptica laser. Early versions of this isotope-selective photoionization technique use a Stanford Research Systems NL100 nitrogen laser for the second photon. Later iterations replace the nitrogen laser with a UV LED [**222**].

2.5.9.1. Toptica DL-Pro. The wavelength is initially set to the NIST-reported value of 791.3505 nm [**199**] and finely-tuned using a Doppler-broadened absorption line in a Hamamatsu L2783-56NE-BA barium laser galvatron using optogalvanic spectroscopy [**200**, **201**, **202**, **203**, **204**, **205**, **119**, **206**]. The wavelength is later fine-tuned using the apparatus described in Ch. 8 and the apparatus described in Ch. 13 by maximizing the fluorescing ^{138}Ba$^+$ loading rate.

2.5.9.2. 791 nm Optogalvanic Spectroscopy. To initially find the 791 nm Ba resonance, Doppler-broadened optogalvanic spectroscopy is used similarly to the 413 nm ECDL (Sec. 2.5.1.1). The optogalvanic signal is shown in Fig. 2.26 and fit results are given in Table. 2.9.

From this data, the barium ^{1}S$_0$ → ^{3}P$_1$ transition is 791.3507582(12) nm, which is -120 MHz from the NIST-reported value of 791.3505 nm [**199**] likely due to the wavelength

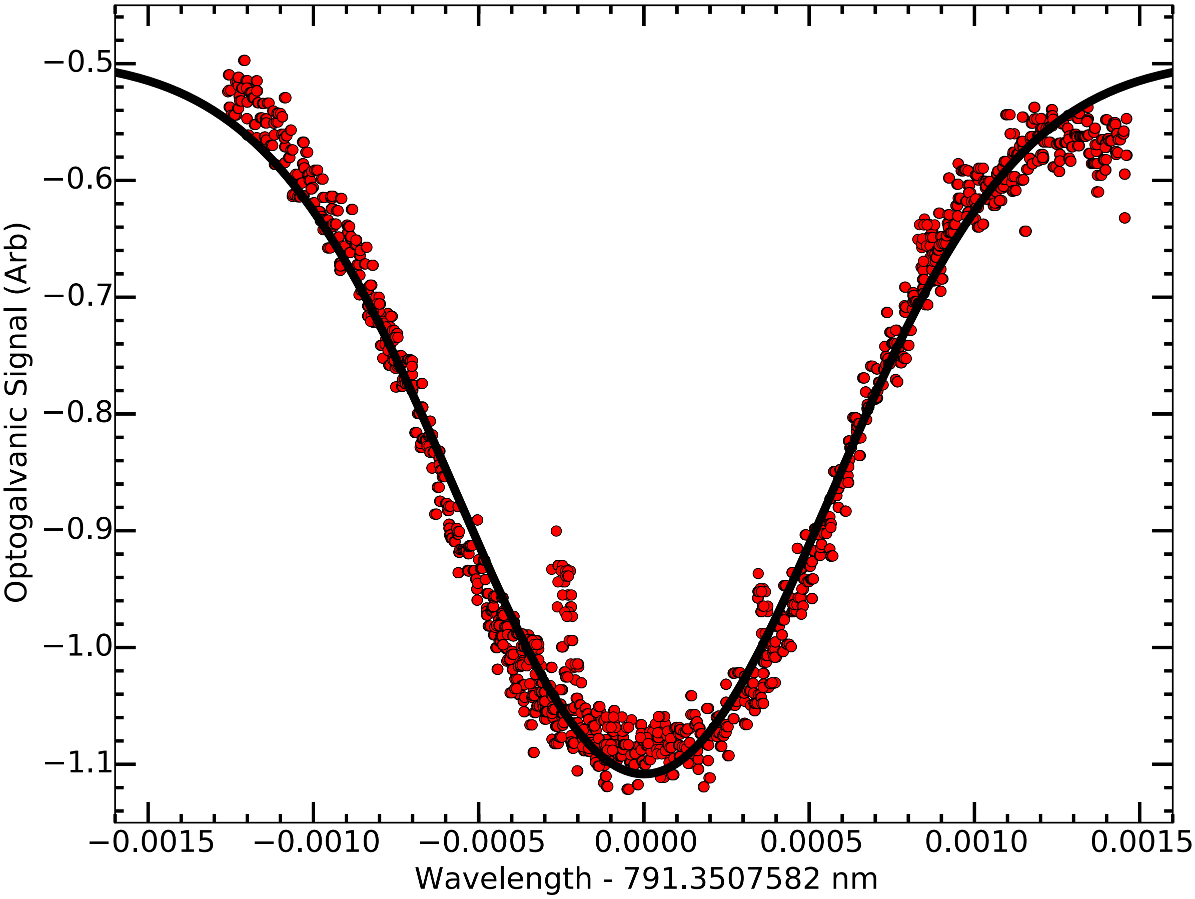


Figure 2.26. 791 nm barium $^1S_0 \rightarrow {}^3P_1$ optogalvanic spectroscopy. A fit (black line) to the data (red circles) is discussed in the text with fit results located in Table 2.9.

meter absolute calibration being unreliable during this laboratory epoch. Additionally, the line is substantially Doppler broadened as expected with a FWHM of 640 MHz.

2.5.9.3. 791 nm ECDL. The in-house 791 nm ECDL uses a Q-Photonics QLD-795-150S laser diode (S/N AX1407LD-103) with ThorLabs A110-B collimation lens and 1800 line/mm grating. In-house electronics drive the ECDL system, which are powered by a PowerOne ±24 V supply. A ThorLabs IOT-5-780-VLP tandem optical isolator is placed at the ECDL output and a cylindrical telescope, comprised of a $f = 250$ (ThorLabs

LJ1267RM-B) and a $f = -75$ mm (ThorLabs LK1431RM-B) lens pair, correct the ellipticity of the output spot. Further beam shaping from the near-circular 1.5 mm MFD beam size is unnecessary for fiber coupling using ThorLabs CFC-8X-B collimators with 780HP fiber. Diagnostics information is located in Table 2.5.

2.6. Wavelength Meter Lock Setup

Wavelength locking of all ECDLs is performed by a LabVIEW program (App. C.1.1.2) that steps through the laser colors sequentially using AOM switching (all IntraAction ATM-801A1 or A2 AOMs with DE-801T drivers) via a NI PCI-6722 static and waveform analog output card on a dedicated computer. Polarization optics and dichroics combine all beams onto a single ThorLabs 460HP single-mode fiber patch cable to the High Finesse WS-U 10 wavelength meter calibrated to a Research Electro-Optics R-32734 frequency stabilized HeNe laser. The LabVIEW program then generates an error signal, performs a PI servo loop, and uses the same PCI-6722 card to output an analog signal to each laser system's external modulation input. The speed of this wavelength locking system is limited by LabVIEW; typically, the lock voltage of each laser color is updated every ∼1 second.

2.7. SHG Crystal Length Modeling

A critical component when designing a SHG optical cavity is the SHG crystal length; if the crystal is too short or too long, the SHG efficiency will be low. Therefore, modeling the SHG efficiency as a function of crystal length, cavity parameters, and input pump power is an essential component of SHG optical cavity design. Discussion on resonant optical cavities and SHG optical cavity design is relatively common [**153**, **155**, **223**, **224**, 225, 212], so only the crystal length modeling will be discussed. Additionally note that nonlinear processes other than SHG are also widely used [226, 227, 228, 229].

2.7.1. Boyd-Kleinmann Theory

Boyd-Kleinmann theory is used to model the SHG doubling efficiency [230, 225]. The SHG harmonic power, $P_{2\omega}$, is given by

$$P_{2\omega} = \left\{ \frac{16\pi^2 d_{eff}^2}{\lambda^3 \epsilon_0 n^2 c} \right\} \cdot L \cdot e^{-(\alpha_1 + 0.5\alpha_2)L} \cdot h\left(\sigma, \beta, \kappa, \xi, \mu\right) \cdot P_\omega^2 \tag{2.9}$$

where d_{eff} is the effective nonlinear coefficient of the crystal, λ is the wavelength of the fundamental beam, ϵ_0 is the permittivity of free space, n is the refractive index of the crystal at the phase matching condition, c is the speed of light, L is the length of the crystal, α_1 (α_2) is the absorption coefficient of the fundamental (second harmonic), P_ω is the power of the fundamental beam, and h is the function of optimizable parameters, which is given by

$$h\left(\sigma, \beta, \kappa, \xi, \mu\right) = \frac{e^{\mu\alpha L}}{4\xi} \iint_{\xi(1-\mu)}^{\xi(1+\mu)} \frac{e^{-\kappa(\tau+\tau') + i\sigma(\tau-\tau') - \beta^2(\tau-\tau')^2}}{\left(1+i\tau\right)\left(1-i\tau\right)} \mathrm{d}\tau \mathrm{d}\tau'. \tag{2.10}$$

The input parameters in $h\left(\sigma, \beta, \kappa, \xi, \mu\right)$ are:

- μ, a measure of the displacement of the fundamental laser beam focus from the center of the crystal, which is quantified by f, the relative position of the focus from the entrance of the crystal. μ is given by

$$\mu = \frac{L - 2f}{L}. \tag{2.11}$$

- α, the relative absorption factor, which is given by

$$\alpha = \alpha_1 - \frac{1}{2}\alpha_2. \tag{2.12}$$

- ξ, the optimum focusing parameter that gives the optimum confocal parameter of the laser beam. For non-critical phase matching, the walk-off angle of the laser beam inside the crystal ρ is zero, and this value is $\xi = 2.84$. For critical phase matching, ξ is a function of L and the confocal parameter of the laser beam b, is optimized at $\xi = 1.39$, and is given by

$$\xi = \frac{L}{b}. \tag{2.13}$$

- κ, an absorption factor depending on b and α, which is given by

$$\kappa = \frac{b\alpha}{2}. \tag{2.14}$$

- σ, the phase mismatch factor that arises from b and the fluctuations in the wavenumber Δk of the fundamental beam in the crystal due to refractive index fluctuations Δn, which is given by

$$\sigma = \frac{1}{2}b\Delta k. \tag{2.15}$$

- β, the walk-off factor depending on ξ, L, the wavenumber k of the fundamental beam in the crystal, and ρ, which is given by

$$\beta = \frac{1}{2}\rho\sqrt{\frac{Lk}{\xi}}. \tag{2.16}$$

Eq. 2.9 can be written as

$$P_{2\omega} = \eta P_{\omega}^{2} \tag{2.17}$$

where

$$\eta = \left\{ \frac{16\pi^2 d_{eff}^2}{\lambda^3 \epsilon_0 n^2 c} \right\} \cdot L \cdot e^{-(\alpha_1 + 0.5\alpha_2)L} \cdot h\left(\sigma, \beta, \kappa, \xi, \mu\right). \tag{2.18}$$

2.7.2. Optical Cavity Fundamental Power Enhancement

As seen from Eq. 2.9 and Eq. 2.17, the second harmonic power $P_{2\omega}$ increases as the *square* of the fundamental power P_{ω}, i.e. an increase of the fundamental power by a factor of ∼50 will result in an increase in the second-harmonic power by ∼2500. This fundamental power increase factor is typically achieved using a ring or bow-tie optical cavity (Fig. 2.27) [**225**, **212**].

The fundamental power in the cavity $P_{\omega}^{\mathrm{cav}}$ is given by

$$P_{\omega}^{\mathrm{cav}} = A c_{\mathrm{eff}} P_{\omega}^{\mathrm{inp}} \tag{2.19}$$

where c_{eff} is the coupling efficiency of the on-cavity-resonance incident fundamental power $P_{\omega}^{\mathrm{inp}}$ on the input coupler, with reflectivity r, and A is the enhancement factor [**231**], which is given by

$$A = \frac{1 - r}{\left(1 - \sqrt{rv}\right)^2}. \tag{2.20}$$

v is the fractional loss of the initial fundamental laser power to the laser power after one round trip inside the optical cavity, and contains all of the losses in the cavity experienced by the fundamental beam including SHG. Therefore, v is given by

$$v = \left(1 - \alpha_1 L\right)\left(1 - \eta P_{\omega}^{\mathrm{cav}}\right)\left(1 - t\right) \tag{2.21}$$

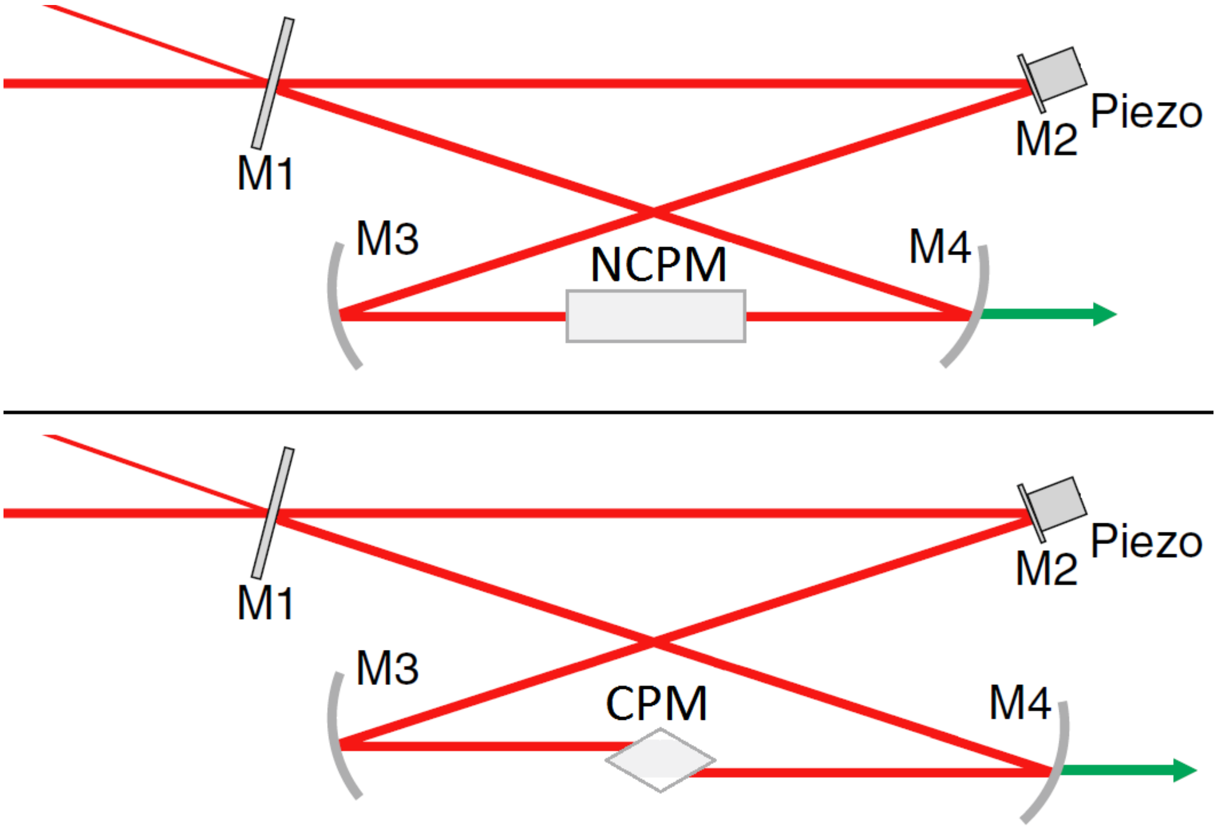


Figure 2.27. Optical ring cavities for SHG for non-critical phase matching (upper) and critical phase matching (lower). Note the additional beam walk-off that occurs in the critical phase matching case. Figure adapted from Ref. [225].

where t is an extra loss factor due to additional fundamental power losses in the cavity, i.e. the mirror reflectivities other than the input coupler and the SHG crystal input and output facet reflectivities.

Note that if the cavity is impedance-matched, i.e. the reflectivity of the input mirror r is matched with all losses in the cavity v, the value of A is maximized. While difficult to model the exact losses in the SHG cavity especially when using high-quality mirrors and AR-coatings, operating near this impedance-matched condition can increase $P_{\omega}^{\mathrm{cav}}$ by

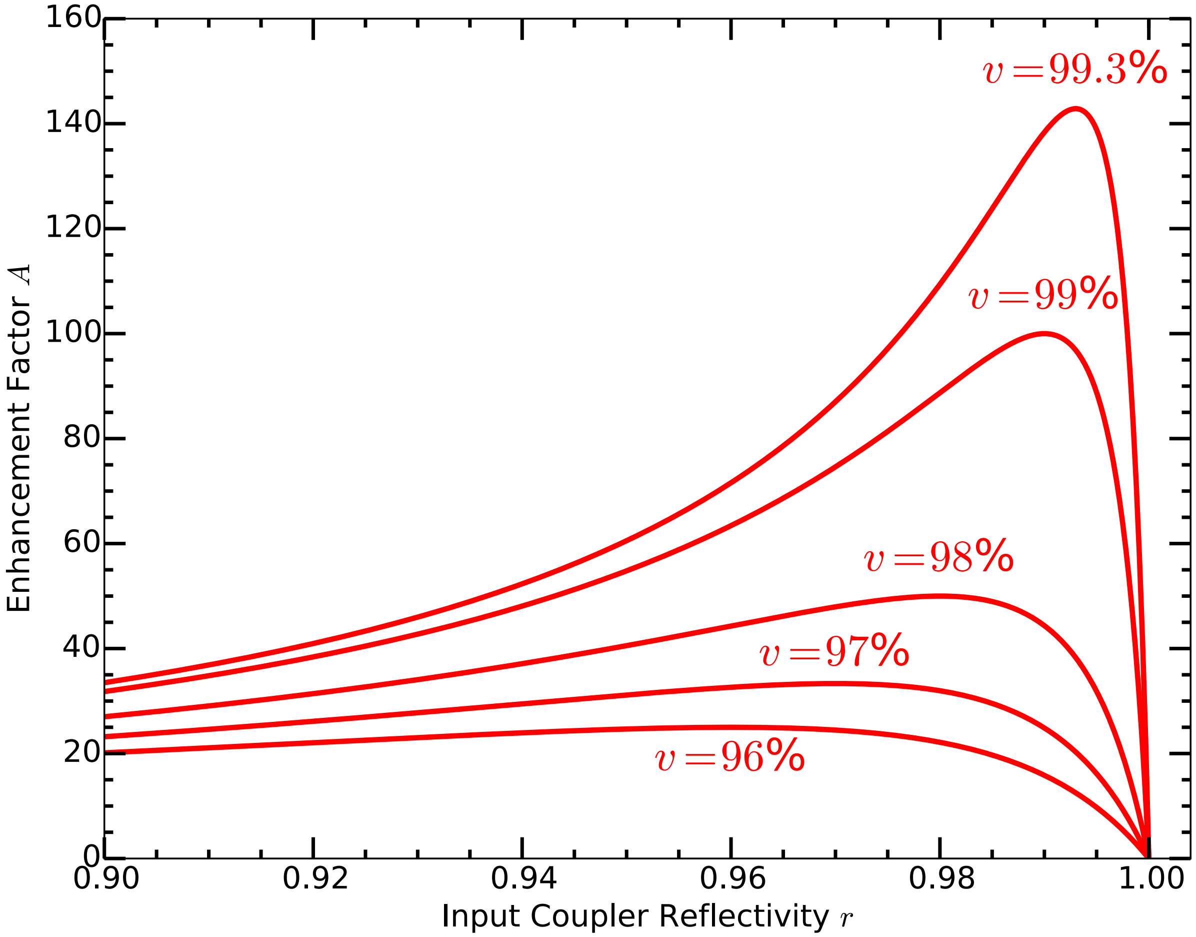


Figure 2.28. Enhancement factor A plotted as a function of input coupler reflectivity r for various internal cavity losses v. Note the optical cavity enhancement when r matches v especially at higher values of r, corresponding to the impedance-matched condition. This is why high reflectivity mirrors and excellent AR-coatings inside the cavity are crucial for efficient SHG.

a factor of $\gtrsim 2$, which would increase $P_{2\omega}$ by a factor of $\gtrsim 4$ (Fig. 2.28). Therefore, trying a variety of input coupler reflectivities is recommended.

2.7.3. Modeling Code

All modeling code runs in the Octave environment and is found in App. D.1.3. User inputs are the optical cavity parameters (r, t, c_{eff}, and P_{ω}^{inp}), maximum SHG crystal length and number of points to calculate, input beam wavelength, and crystal parameters (d_{eff}, α_1, α_2, n_1, n_2, Δn, ρ, and ξ), where n_1 (n_2) is the refractive index of the fundamental (second-harmonic) beam. Typical SHG crystal parameters are found in publication data, from crystal manufacturers, and/or from SNLO [**232**].

For non-critical phase matching, the function of optimizable parameters is separated into four one-dimensional integrals (real and imaginary parts of τ and τ'), and each is numerically integrated at every crystal length point. For critical phase matching, the function of optimizable parameters is numerically integrated in two-dimensions by brute force for each crystal length point with the summation step size varied to verify that it is sufficiently small. η is then calculated, the equilibrium fundamental power in the cavity is solved numerically, and the SHG power is calculated. Although all code in App. D.1.3 calculates the second-harmonic power output from a SHG cavity, the code is easily modified to provide single-pass second-harmonic power.

2.7.4. Non-critical Phase Matching Example

An additional 987 nm ECDL system for 493 nm SHG Ba$^+$ Doppler cooling is modeled using a potassium niobate (KNbO$_3$) SHG crystal for non-critical phase matching at ∼34°C. For the optical cavity, $r = 0.978$, $t = 0.01$, $c_{\text{eff}} = 0.80$, and $P_{\omega}^{\text{inp}} = 0.120$ W; $P_{\omega}^{\text{inp}} = 0.120$ W is a conservative estimation of the output power of an ECDL using a Sacher Lasertechnik FP-0980-0300 (S/N #17) laser diode with 300 mW free-running power. The used

Table 2.10. Non-critical phase matching modeling results using $KNbO_3$ for 987 → 493 nm SHG for Ba^+ Doppler cooling with optical cavity parameters given in the text and SHG crytal parameters given in the code (App. D.1.3.3). r value optimization is not performed.

Source	P_ω^{inp} (mW)	Shortest Optimal L (mm)	$P_{2\omega}$ (mW)
SAL-1030-60	60	36	32
FP-0980-0300	120	22	65
TA MOPA	1000	3.2	550

crystal parameters are given in Ref. [**233**] and the code (App. D.1.3.3). The modeling output is shown in Fig. 2.29. Note that r value optimization is not performed.

Additional input power options are a Sacher Lasertechnik SAL-1030-60 (S/N BA-01558) AR-coated laser diode with 60 mW of output power. This ECDL could also be used in a 1.5 W tapered amplifier MOPA system. Both of these options are also modeled with expected SHG output powers listed in Table 2.10. Note that r value optimization is not performed.

Based on the results in Table 2.10 and the expectation of using the FP-0980-0300 laser diode, a 2.3 × 3 × 20 mm $KNbO_3$ crystal was purchased from FEE (S/N BA 110646,00/39750, I137-5/1 Nr 1) with Brewster-cut facets (65.8°).

2.7.5. Critical Phase Matching Example

To generate light for SiO^+ at 385 nm [**234**] for the experiment discussed in Ref. [**235**], a 770 nm ECDL and tapered amplifier MOPA system for 385 nm SHG is modeled using a bismuth triborate (BiBO) SHG crystal for critical phase matching. The optical cavity conditions are the same as to that in Sec. 2.7.4: $r = 0.978$, $t = 0.01$, and $c_{\mathrm{eff}} = 0.80$, but $P_\omega^{\mathrm{inp}} = 1.0$ W; $P_\omega^{\mathrm{inp}} = 1.0$ W is a conservative estimation of the output power of an

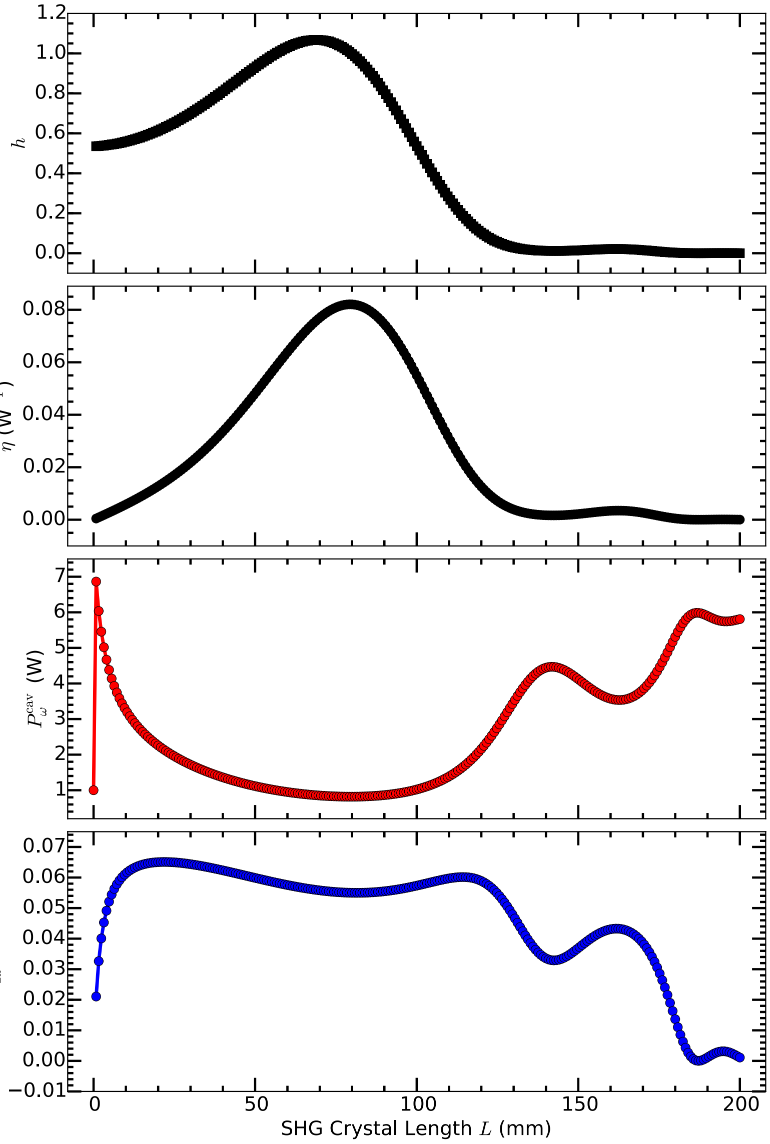


Figure 2.29. Non-critical phase matching modeling using $KNbO_3$ for 987 → 493 nm SHG for Ba^+ Doppler cooling with optical cavity parameters given in the text and SHG crystal parameters given in Ref. [**233**] and the code (App. D.1.3.3). r value optimization is not performed. The length of the crystal in this model is unrealistically long to show the behavior and maximization of various parameters at long crystal lengths.

ECDL MOPA system using an Eagleyard EYP-RWE-0780-02000-1300-SOT12-0000 (S/N AD-14005) AR-coated laser diode gain chip and an Eagleyard EYP-TPA-0765-01500-3006-CMT03-0000 (S/N EA-01641) tapered amplifier with ∼1.5 W of output power. The used crystal parameters are given in the code (App. D.1.3.2). The modeling output is shown in Fig. 2.30. Note that r value optimization is not performed.

From the modeling results, the SHG power begins to saturate at a crystal length of ∼10 mm due to the walk-off of the fundamental beam in the crystal. However, ∼300 mW of output power is expected. Additionally, the walk-off limitation can be corrected by using a longer, periodically-poled crystal.

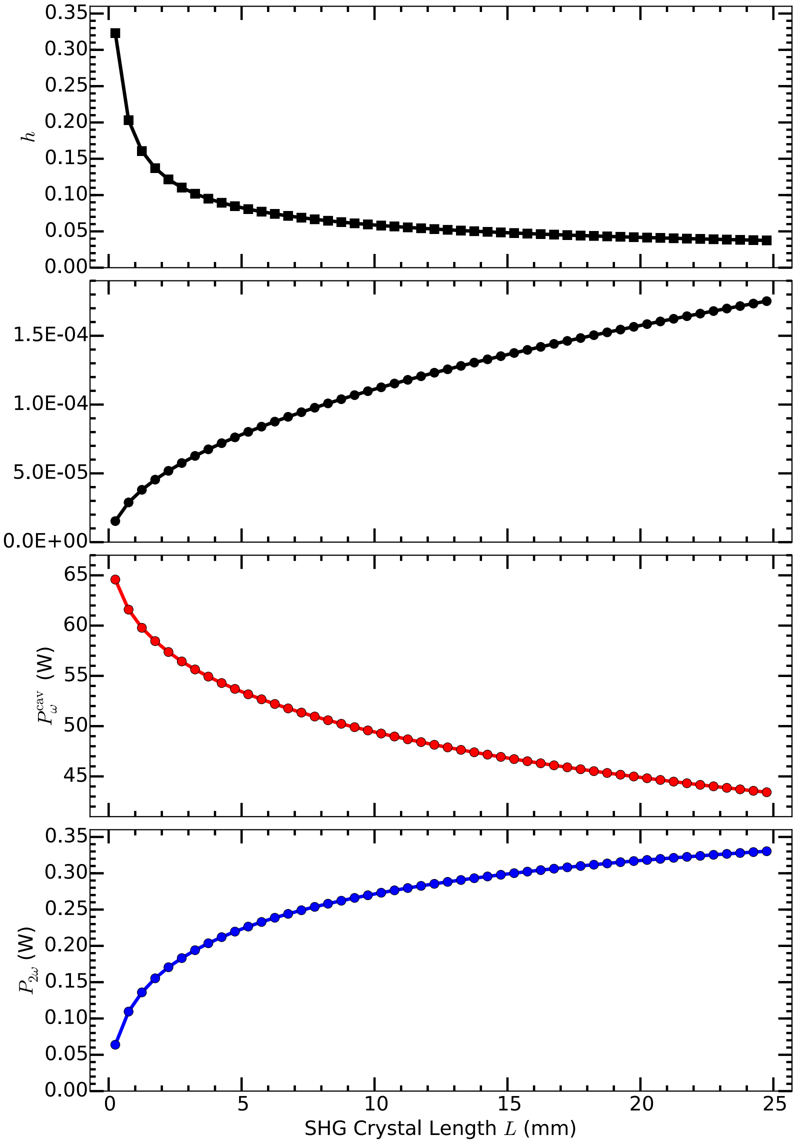


Figure 2.30. Critical phase matching modeling using BiBO for 770 → 385 nm SHG for SiO$^+$ experiments with $P_{\omega}^{\text{inp}} = 1.0$ W, optical cavity parameters given in the text, and SHG crystal parameters given in the code (App. D.1.3.2). r value optimization is not performed.

CHAPTER 3

Rb Reference Laser for Transfer Cavity Lock

This chapter summarizes the proposal of using an active transfer cavity, locked to a rubidium reference laser, to lock the frequency of other ECDLs. It proposes a cavity design and discusses the constructed reference laser including details of the Rb polarization spectroscopy technique used to generate an error signal without a lock-in amplifier or frequency dithering, the PI lock servo, and locking results. Acoustic resonances of the ECDL laser system are also quantified and discussed.

3.1. Transfer Cavity Locking Proposal

The wavelength meter lock discussed in Sec. 2.6 is used because no convenient atomic or molecular reference lines are sufficiently close to many of our required wavelengths with reference cells available only for a few Ba^+ transitions. Tellurium (Te_2) lines exist close to the Ba^+ 455 [**209**] and 493 nm [**236**] transitions, but the cell must be heated to >200 °C, which is a technical challenge. Additionally, iodine (I_2) lines exist near the Ba^+ 650 nm [**217**, **218**, **219**] transition (Table 2.8), but locking to these lines requires significant laser power to shift the laser frequency sufficiently and to lock to a line. Additionally, the lines are weak at room temperature, necessitating the cell to be heated significantly.

To lock the numerous CW laser systems required to address the many Ba^+ transitions used in this work as well as for AlH^+ fluorescence (Sec. 11.8), a transfer cavity [**237**, **238**, **239**, **191**, **240**] under vacuum is proposed (Fig. 3.1). In this application, the

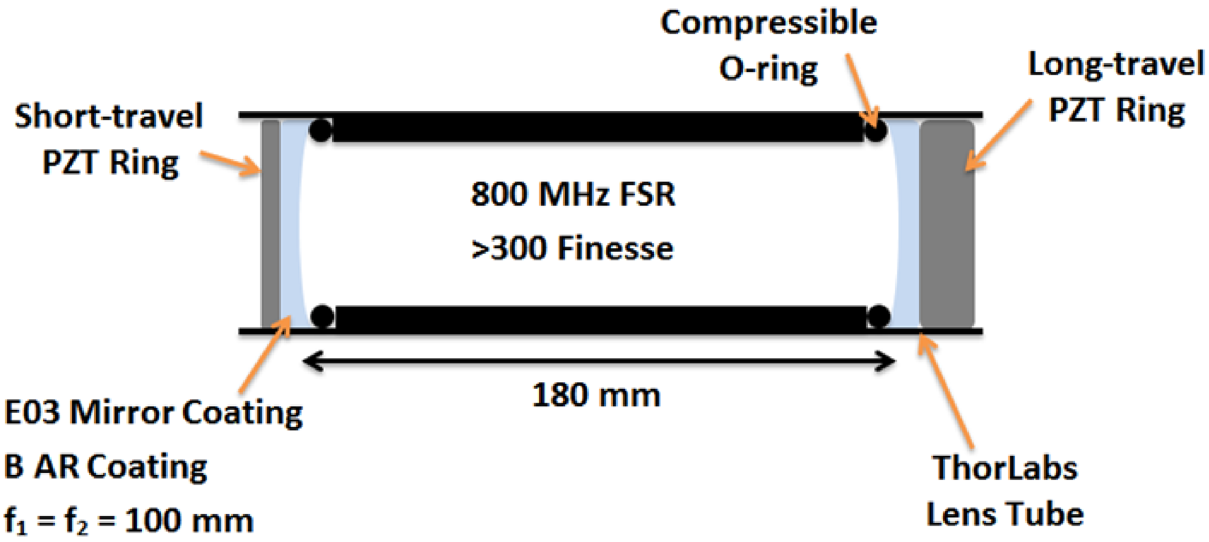


Figure 3.1. Diagram of the transfer cavity proposed to lock the Ba$^+$ and/or AlH$^+$ CW laser systems. The two PZT elements would be used to actively lock the cavity to a laser locked to a Rb reference, which would compensate for length drifts of the cavity due to temperature drifts altering the length of the lens tube housing. Note that $f_1 = f_2 = 100$ mm, and the mirrors will be separated by 180 mm ensuring that the cavity is stable.

cavity will be *actively* locked to a laser that is locked to Rb, which has been previously demonstrated [237]. Because the cavity is actively locked, the system does not need to be actively temperature controlled since any length variations in the cavity due to temperature fluctuations will be compensated using PZT elements. Additionally, the passive stability of the cavity at high frequencies can be used for laser linewidth narrowing if the cavity finesse is sufficiently high [211].

The cavity length would be actively stabilized using two PZT ring elements (Noliac NAC2125-H4mm and -H20mm) mounted in ThorLabs lens tube using custom mounts (Fig. 3.2, App. F.1.6). The long-travel PZT ring would compensate for slow-timescale cavity length drifts while the short-travel PZT would operate with a higher frequency bandwidth. The cavity length would be locked using the Pound-Drever-Hall technique

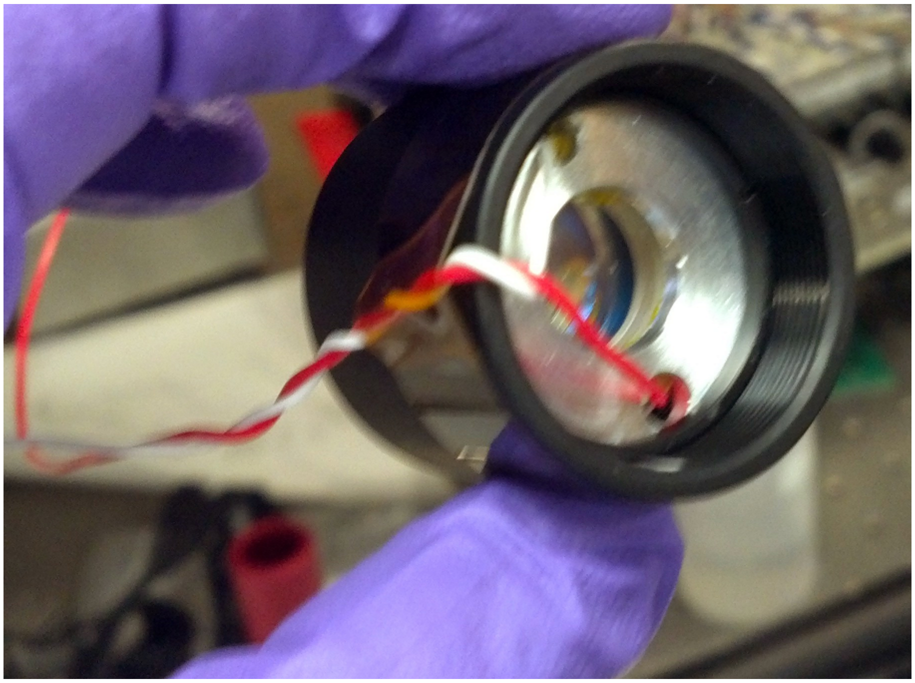


Figure 3.2. Picture of an assembled transfer cavity mirror mount. Inside of the ThorLabs SM1 lens tube are the pieces in App. F.1.6, o-ring, PZT ring, and ThorLabs 1" E03 mirror. The holes on the last piece are wire feedthrus for the PZT element.

[241, 242, 191] and PID servo loop being developed in work by PRS [216]. Note however that the Pound-Drever-Hall locking scheme requires dielectric mirrors with low optical absorption, and the broadband ThorLabs E03 mirrors proposed do not meet this criteria.

Due to a cavity resonance condition depending on the index of refraction of the filler material, and this index differentially varying with temperature, humidity, and air pressure sufficiently between various wavelengths of interest to cause drifts of ∼10 MHz over a ∼100 nm range, the cavity will be placed in vacuum. Additionally, AOMs will shift the Ba^+ and/or AlH^+ laser frequencies to a cavity resonance. Note that multiple wavelengths may be locked to the same cavity [243].

3.2. The 780 nm Reference ECDL System

The in-house-built reference EDCL system uses a Thorlabs L785P090 laser diode, Newport KGA110-B collimation lens, 1800 line/mm grating with ∼15% feedback, ThorLabs IOT-5-780-VLP tandem optical isolator, and beam-correction optics. Laser characteristics and tuning parameters are given in Ch. 2. Some of the output laser power is selected with HWP and PBS and sent to a ThorLabs P5-780PM-FC-2 fiber patch cable, which outputs to the polarization spectroscopy setup (Sec. 3.3) with a ThorLabs CFC-11X-B collimator. The remainder of the light will be used to lock transfer cavities (Sec. 3.1).

The laser electronics are an in-house temperature controller and initially a Newport current driver while the in-house current driver is developed. The initial current driver schematic and board are from the Vladan Vuletić group at MIT. We unsuccessfully redesigned the board to use non-surface-mount components; the resultant circuit oscillates dramatically, which is quantified using an Agilent N9320A spectrum analyzer without and with constructed ∼20 dB (App. A.1.1) and ∼40 dB (App. A.1.2) amplifier circuits. Later iterations use and improve upon the initial MIT board design with the amplifier circuits and spectrum analyzer measuring no current noise down to $\lesssim 5\ \mu\text{A}_{\text{RMS}}$. Polarization spectroscopy (Sec. 3.3) is then used to estimate the drive current noise by its modulation of the ECDL output frequency. The version driving the reference laser is similar to the final design located in App. A.1.3. Changes from the MIT design include the addition of ground planes, a low-pass filter on the LM317 section output, and bypass capacitors around the circuit. Additionally, the current-controlling-servo op-amp and MOSFET as well as component values and voltages used in the modulation section are changed.

All power for the electronics is isolated by a APC Smart-UPS 1000 uninterruptible power supply and in-house electronics are supplied from a PowerOne ±24 V, 2.4 A supply.

3.3. Polarization Spectroscopy

Doppler-free polarization spectroscopy [**244, 245**] is used to generate an error signal for locking the reference laser. This method avoids laser frequency modulation and lock-in detection required in frequency-modulation spectroscopy techniques. The polarization spectroscopy setup is shown in Fig. 3.3. Most of the input light (pump beam) passes through the glass slide and a QWP oriented to circularly-polarize the beam, and mirrors are used to overlap the pump beam with the probe beam in a Triad Technologies TT-RB-20X75-Q Rb vapor cell. A small fraction of the light input to the setup is picked off by the thick glass slide, forming the probe beam, and passes through an iris to block the second probe beam reflection spot and pump beam. A HWP aligns the linear polarization at 45° such that the detection PBS equally splits the probe beam power to the two ThorLabs SM05PD2A photodiodes with FL05780-10 bandpass filters and LA1540-B $f = 15$ mm lenses when the pump beam is blocked. All components unless otherwise specified are from ThorLabs.

The circularly-polarized pump beam optically pumps the Rb vapor to an extreme m_F state depending on the circular-polarization handedness, inducing a birefringence in the vapor. The counterpropagating, linearly-polarized probe beam then analyzes this induced birefringence. Because of the linearly-polarized probe beam can be expressed as a sum of left- and right-circular polarizations, and the vapor is optically pumped to an extreme m_F state, the linear polarization rotates. This is observed as a change in the intensity

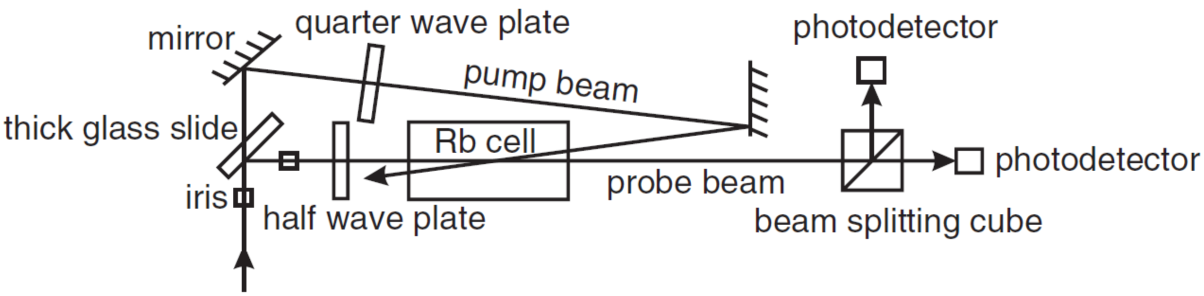


Figure 3.3. Diagram of the polarization spectroscopy setup used to lock to the ^{85}Rb, $F = 3$ line. An additional HWP and PBS (not shown) are placed before the first input iris. Figure adapted from Ref. [245].

split by the PBS to the two photodiode signals. The polarization spectroscopy signal is then the difference of the detector signals.

As noted in Ref. [245], the polarization spectroscopy signal is optimal when ambient magnetic fields are cancelled with the earth's magnetic field reducing the signal by a factor of two. At higher field strengths, the signal reduces further before distorting as the degeneracy of the Zeeman levels is lifted. Ref. [245] also notes that the polarization spectroscopy signal zero-crossing is shifted from the atomic resonance and is dependent on the pump beam power. However, as long the power input to the polarization spectroscopy setup is stable, this shift from the atomic resonance is not a concern. Additionally, this power could be servo-controlled using an AOM.

3.3.1. Polarization Spectroscopy Electronics

The circuit used for the polarization spectroscopy is located in App. A.1.6. The circuit uses transimpedance amplifiers (Sec. 3.3.1.1), optimized for this application, before adding and subtracting the two photodiode signals for transmission and polarization spectroscopy (error) signals, respectively. An additional buffered voltage reference and potentiometer

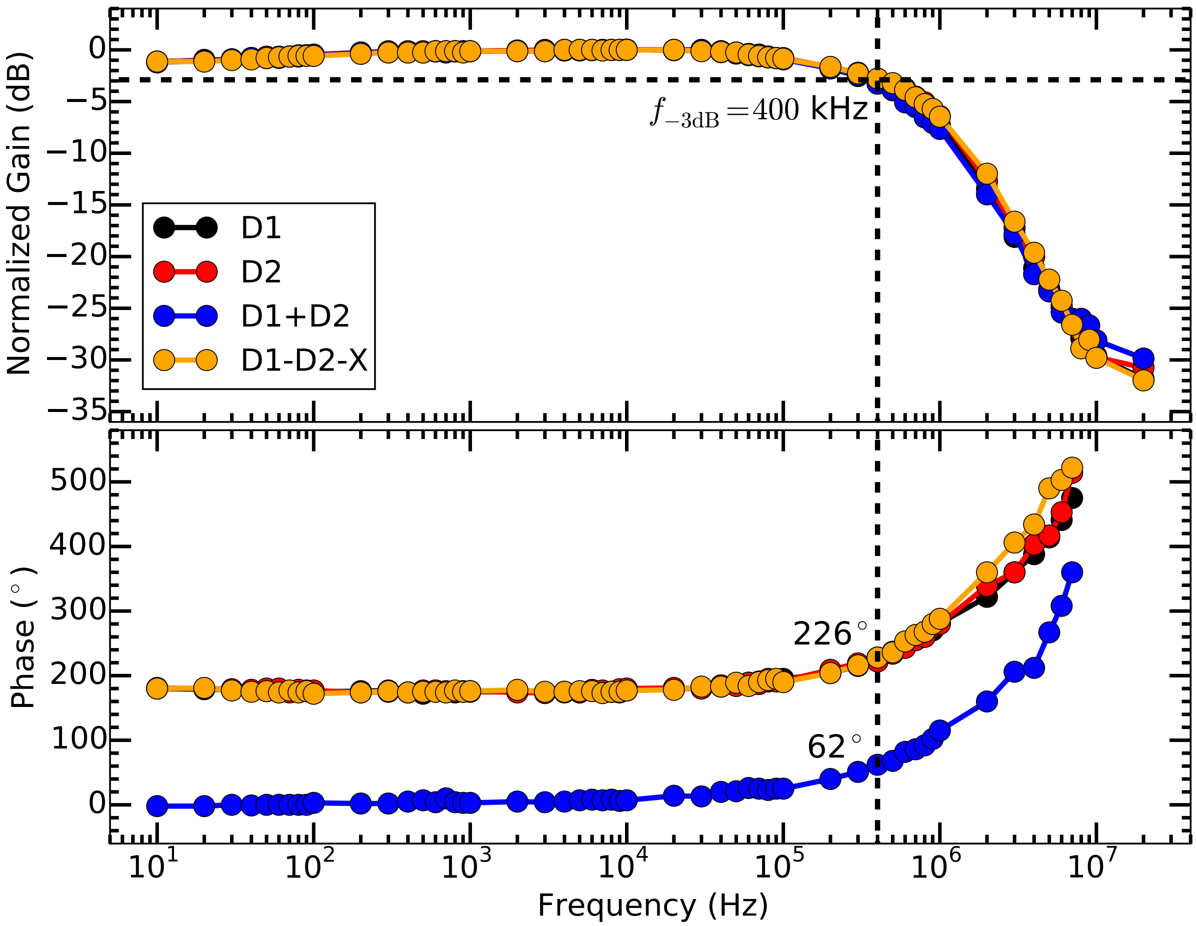


Figure 3.4. Gain and phase response of the polarization spectroscopy circuit as a function of frequency. D1 and D2 are the photodiode signals and X is the added offset voltage. The $f_{-3\mathrm{dB}}$ frequency is 400 kHz with no significant phase accumulation until >100 kHz. Without the transimpedance amplifiers, the circuit bandwidth is ∼10 MHz.

provide offset trimming of the subtracted polarization spectroscopy signal. The gain and phase response of this circuit as a function of frequency is shown in Fig. 3.4, measured using an Agilent function generator sinusoidally driving an LED placed immediately in front of a photodiode. The circuit bandwidth and phase accumulation is from the transimpedance amplifiers as the circuit with them omitted has a bandwidth of ∼10 MHz.

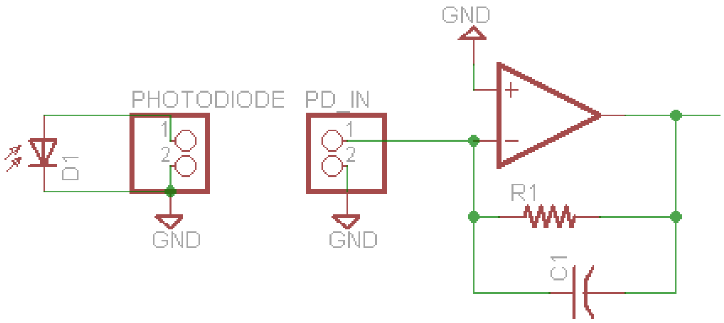


Figure 3.5. Circuit implementation of a transimpedance amplifier to convert the photodiode output current to voltage. The gain of the circuit is given by the value of R_1 while C_1 stabilizes the op-amp from the capacitive load of the photodiode.

3.3.1.1. The Transimpedance Amplifier. Transimpedance amplifiers are used to convert the photodiode output current to voltage. The basic op-amp circuit for this is shown in Fig. 3.5 [**246, 247**]. Due to the large capacitance of the photodiode and any cabling, the op-amp must be stabilized from oscillation at high frequencies with a capacitor (C_1) in parallel with the gain resistor (R_1) with theoretical value

$$C_1 = \sqrt{\frac{C_T}{2\pi R_1 \left(GBP\right)}} \tag{3.1}$$

where (GBP) is the specific op-amp gain bandwidth product and C_T is the total input capacitance of the op amp including contributions from the photodiode, connectors, cabling, and op-amp input. Because the op-amp has a finite (GBP), the -3 dB operating bandwidth of the amplifier is limited by the values of R_1 and C_T and is given by

$$f_{-3\text{dB}} = \sqrt{\frac{(GBP)}{2\pi C_T R_1}}. \tag{3.2}$$

Therefore, given a desired transimpedance amplifier gain, operating bandwidth, and photodiode, the value of C_1 and choice of op-amp can be determined. Note that (1) if the system oscillates, the value of C_1 must be increased; and (2) C_1 can be much larger than required at the expense of some operating bandwidth. Also note that electronics after the transimpedance amplifier can add additional capacitance, destabilizing the circuit; each transimpedance amplifier should either use a sufficiently large C_1 value or be immediately followed by a buffer.

3.3.2. Polarization Spectroscopy Results

The Doppler-broadened saturated-absorption transmission and polarization spectroscopy (error) signals after optimizing the polarization spectroscopy setup, electronics, and the 780 nm ECDL current driver are shown in Fig. 3.6 [**220**, **248**]. For these data, 2 mW of power was sent to the polarization spectroscopy setup with 1.60 (0.250) mW in the pump (probe) beams. The reference laser is locked to the ^{85}Rb, $F = 3$ line using the polarization spectroscopy (error) signal. From the data, the slope of the polarization spectroscopy signal is -8.3 mV/MHz and slope of the Doppler-broadened absorption edge is 4.0 mV/MHz. Both parameters are used for diagnostics and to measure gains of various parts of the system.

3.4. PID Servo Loop

Detailed discussions of electronic control systems, servo loop theory, and negative feedback loops are widely available [**249**, **250**, **251**, **252**]. The basic principle is to control a system using negative feedback. In context of electronics, this is done by measuring a

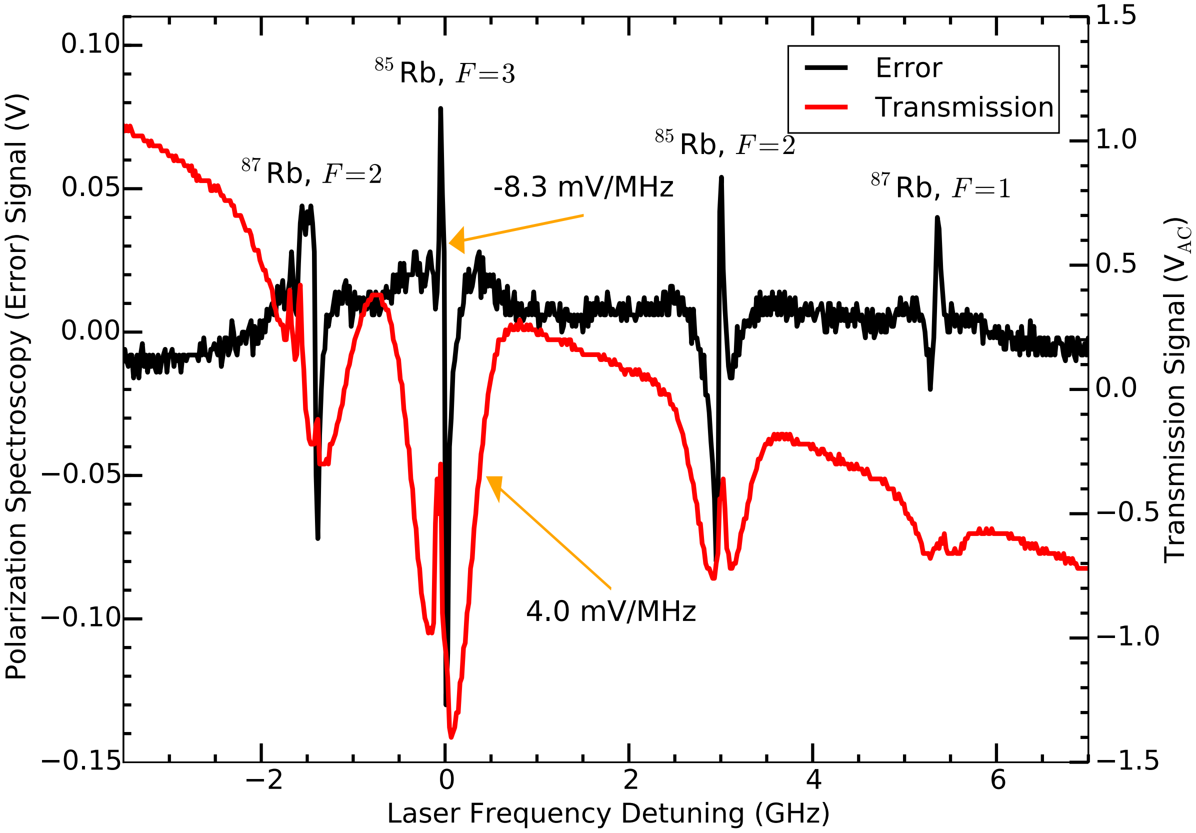


Figure 3.6. Doppler-broadened saturated-absorption transmission (red) and polarization spectroscopy (black) spectra taken with 1.750 (0.250) mW in the pump (probe) beams. Note the desired lineshape of the polarization spectroscopy signal, which is used as the error signal to PI servo and the gradual slope in the transmission signal due to the current feed-forward decreasing the laser drive current. Labeled slopes are used for diagnostics and to measure various gains of the system.

voltage related to the property of the system to be controlled, processing the input voltage and generating an error voltage, performing a operations on that error voltage, and outputting a negative-feedback correction signal. For a PID servo loop, the P-stage provides a correction signal *proportional* to the error signal; the I-stage provides a correction signal that is the *time-integral* of the error signal to suppress the error signal over time; and the

D-stage provides a correction signal that is the *time-derivative* of the error signal which is used at high frequencies for the system. The summation of these three stages form a PID servo loop whose gains must be carefully chosen to assure that the system is stable.

Summarizing from Ref. [253] and Ref. [254], which discuss the servo loop requirements for high-bandwidth locking of ECDLs to high-finesse cavities, the system response must be known to properly design a stable servo loop. Once the system response is known, the PID gains can be adjusted to shape the frequency and phase response of the servo loop in a well-defined way.

One important design consideration is sufficient phase margin, which is the phase delay of the output control signal at the closed-loop 0 dB gain point. In other words, as the phase at this point approaches 180°, the error signal begins to be *amplified*, injecting noise into the system. Insufficient phase margin (typically $\lesssim$30°) will cause the system to oscillate as the system is underdamped. A phase margin of between ~30 and ~90° is typically desired with the latter being acceptable since successive excursions of the error signal are reduced in time. This desired phase margin is provided by the response of a single-pole low-pass filter, which contributes a -20 dB/decade gain roll-off above the -3 dB frequency of the filter.

Another important design consideration is to characterize the gain, phase response, and frequencies of any resonances in the system. The servo loop must be designed such that the closed-loop gain at the resonance frequencies is <0 dB, or the servo loop will amplify the resonances, and the system will oscillate. Additionally, the phase response of these resonances is non-trivial, creating challenges for a simple implementation of a PID servo loop. One simple solution is to sufficiently low-pass-filter the servo loop such that the closed-loop gain at the first resonance frequency is <0 dB.

A common tuning method for PID servo loops is to increase the gain of each stage sequentially until system oscillations are observed. While this maximizes the gains while operating with <0 dB gain at any resonances in the system, it is often a time-consuming, guess-and-check process. Additionally, knowledge of the gain and phase response of the system can uncover problems that can be proactively addressed to improve the final servo loop performance.

Therefore, to carefully construct a servo loop for the Rb reference ECDL system, (1) the acoustic resonances are mapped and appropriate filter used to the PZT driver such that the closed-loop gain at these resonance frequencies is <0 dB (Sec. 3.4.1), (2) the combined gain and phase response of the ECDL and polarization spectroscopy circuit are measured as a function of frequency (Sec. 3.4.2), and (3) the PID servo loop gains are estimated, the circuit is constructed, and the system is locked (Sec. 3.4.3).

3.4.1. ECDL Acoustic Resonances

Observing the 780 nm ECDL output on a Toptica FPI100 scanning Fabry-Pérot interferometer with FPI-100-0750-1 mirrors, ∼20 MHz of frequency jitter is observed, coupling to acoustic noise in the room. Additionally, the spectrum analyzer notes multiple frequency resonances when the laser wavelength is tuned to the Doppler-broadened ^{85}Rb, $F = 3$ absorption edge (Fig. 3.6). In order to quantify the gain and phase response of these resonances, the grating arm PZT is used. While the ECDL wavelength is placed on the Doppler-broadened ^{85}Rb, $F = 3$ absorption edge, a function generator modulates

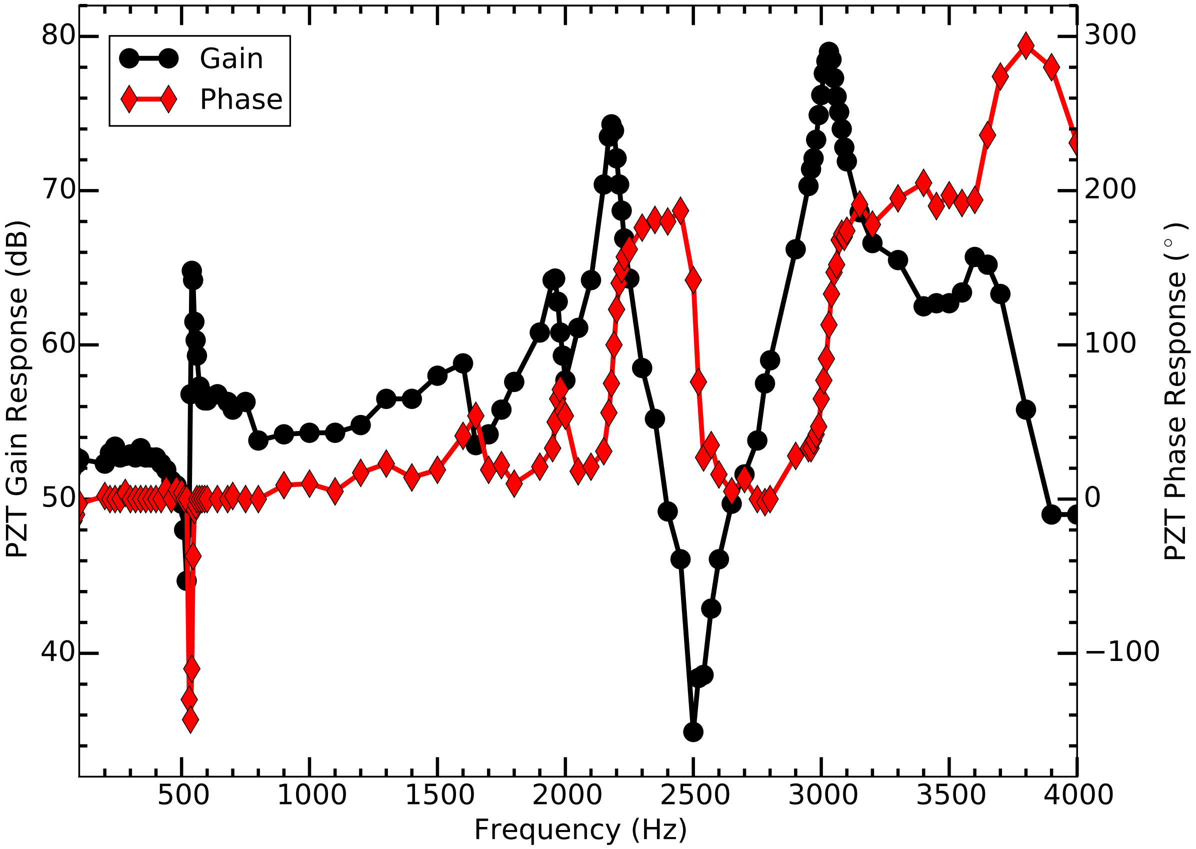


Figure 3.7. Gain (black) and phase (red) response of the ECDL while modulating the PZT. Note the first acoustic resonance is at ∼500 Hz with additional high gain resonances at ∼2.25 kHz and ∼3 kHz.

the PZT driver sinusoidally at a very-low amplitude. The resulting Doppler-broadened absorption edge voltage signal amplitude and phase response relative to the input signal is recorded to characterize these resonances. As shown in Fig. 3.7, the ECDL contains multiple acoustic resonances of the grating arm with the lowest at ∼500 Hz, which is near the frequency of human voice and ambient background noise in the laboratory space.

To prevent the servo loop from exciting these acoustic resonances, a second-order low-pass filter is added between the PZT output of the ECDL controller and modulation input of the PZT driver. Each filter uses a 500 Ω resistor and 22 μF ceramic capacitor for a 14.4 Hz roll-off frequency. Since the filter is second-order, the gain roll-off is -40 dB/decade for a servo-loop suppression of >55 dB at ∼500 Hz. To verify this will prevent excitation of the acoustic resonances, the PZT is again modulated while observing the Doppler-broadened ^{85}Rb, $F = 3$ absorption edge voltage signal (Fig. 3.8) with the function generator ramping its output frequency in time. From the data, the external modulation no longer excites the acoustic resonances, observed by a lack of a voltage response around the resonance frequencies. Therefore, the ECDL will be locked primarily by current modulation at frequencies $\gtrsim$10 Hz. Unfortunately, this filter prevents the laser from being scanned quickly over a large frequency range.

3.4.2. ECDL Modulation Response

Once the second-order low-pass filter is in place to prevent excitation of acoustic resonances, the ECDL modulation gain and phase response is investigated by connecting a function generator to the input modulation port of the ECDL current driver and sinusoidally modulating the system at a very-low amplitude while observing the response from the polarization spectroscopy Doppler-broadened ^{85}Rb, $F = 3$ absorption edge (Fig. 3.9). From this data, the modulation bandwidth of the ECDL and polarization spectroscopy circuit is ∼100 kHz before the phase delay becomes excessive. Additionally note the extra data points collected around the acoustic resonances that demonstrate insignificant gain variation and no poor phase response of the system.

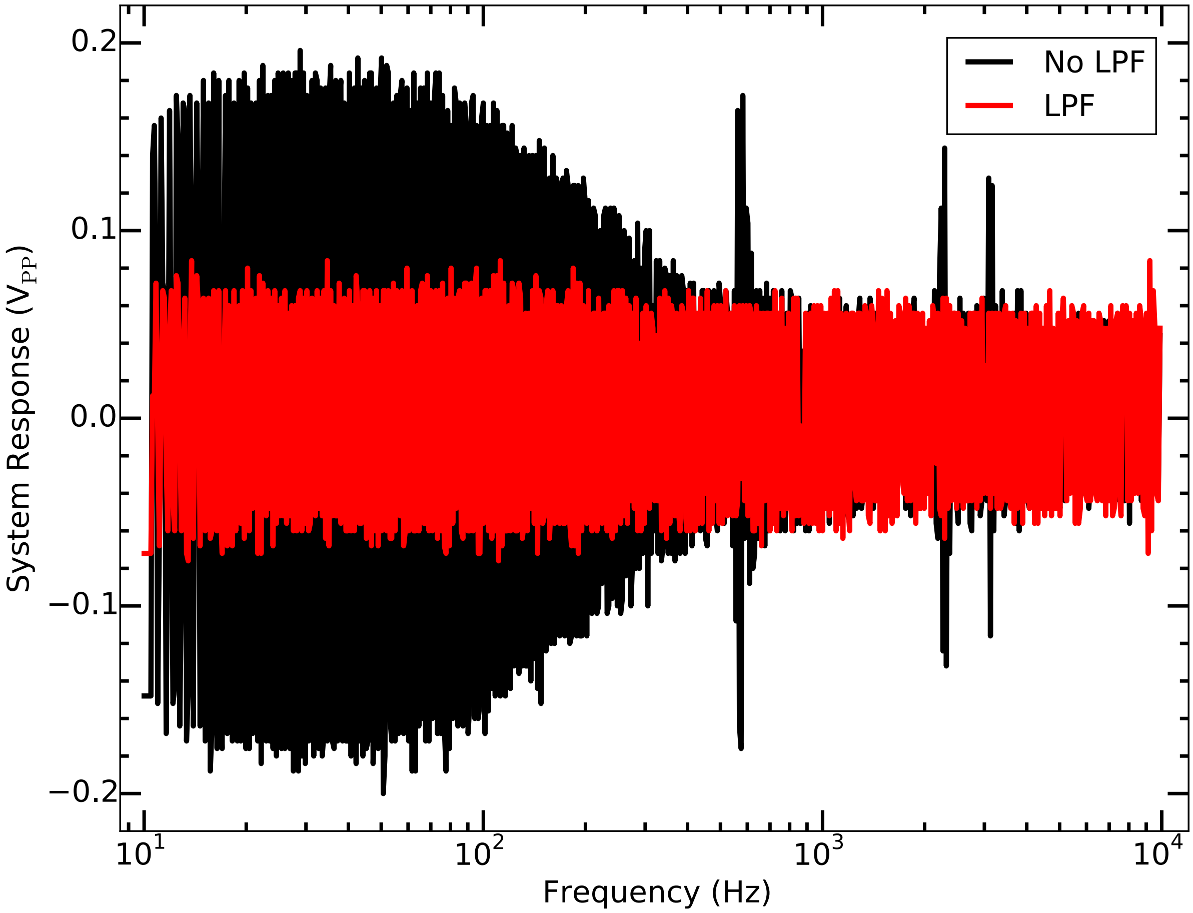


Figure 3.8. System response to a frequency sweep without (black) and with (red) the second-order low-pass filter to the PZT driver. Note that the response with the low-pass filter does not increase at the acoustic resonance frequencies unlike without the filter.

3.4.3. ECDL PI Servo Loop

Because a high-frequency response isn't required for the Rb reference laser, as linewidth narrowing is not necessary, and the technical challenge of implementing the D-stage of a servo loop, the D-stage is omitted from the circuit (App. A.1.8). The data presented

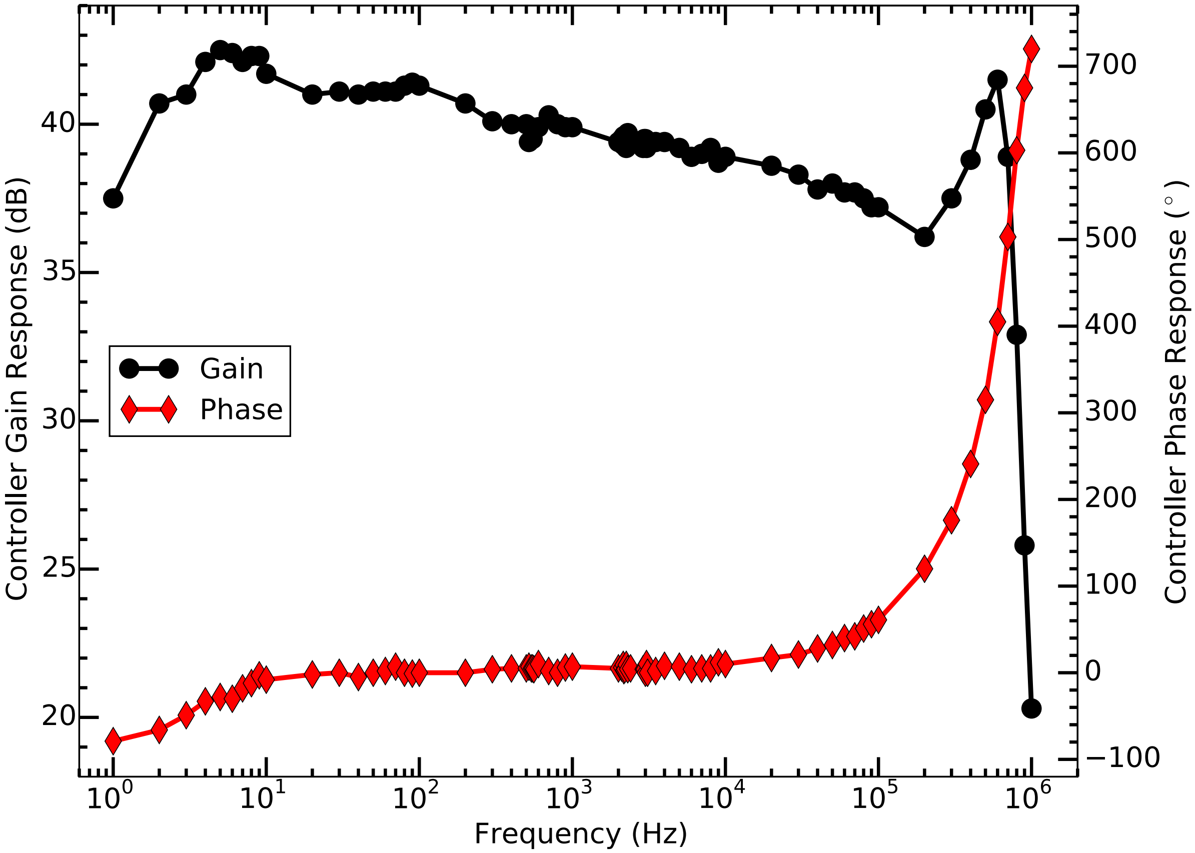


Figure 3.9. ECDL and polarization spectroscopy gain (black) and phase (red) response. Note that the modulation bandwidth is ∼100 kHz before the phase delay becomes excessive. Additionally note that no poor responses are observed around the acoustic resonances.

in Fig. 3.9 provides the data required to determine the gain of the PI-servo loop taking into account the extra factor of ∼2 gain from the increased slope of the polarization spectroscopy signal compared with the Doppler-broadened ^{85}Rb, $F = 3$ absorption edge (Fig. 3.6). Combining the measured gain and phase response of the PI servo loop (Fig. 3.10) with the data in Fig. 3.9, the closed-loop gain and phase response is estimated (Fig. 3.11). From the data, the servo loop bandwidth is ∼250 kHz with a 110° phase margin, and is barely stable. To make the system more robust, the I-stage gain is typically

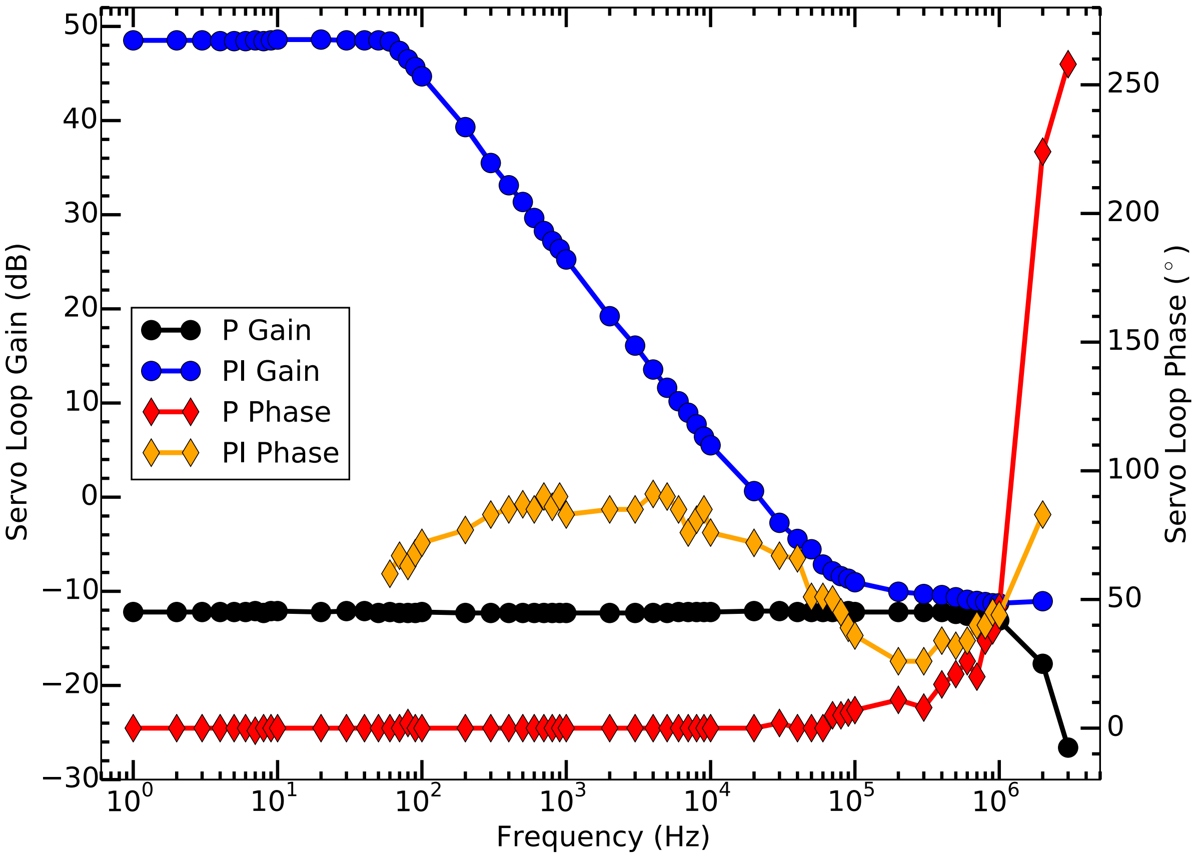


Figure 3.10. P servo loop gain (black) and phase (red) response and PI servo loop gain (blue) and phase (orange) response. Note the P gain is flat to prevent oscillation of the system at high frequencies. The added gain from the I-stage is significant and saturates at ∼10 Hz.

decreased such that the lock bandwidth is ∼100 kHz. Additionally note that there is >40 dB of negative feedback at the acoustic resonance frequencies.

3.5. Reference Laser Results

Spectrum analyzer traces of the polarization spectroscopy error signal spectral power density while unlocked around the zero-crossing and while locked are shown in Fig. 3.12. The background power spectral density trace is taken with the probe beam blocked. Using

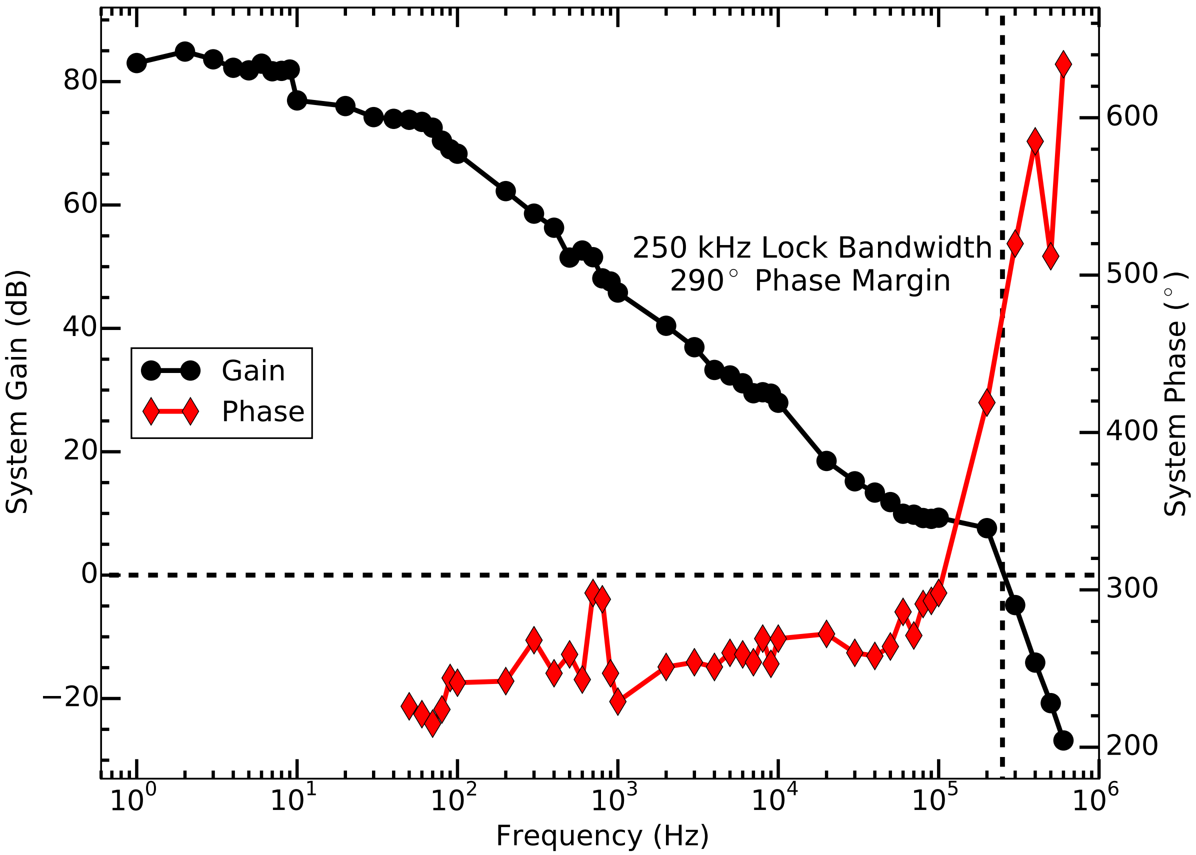


Figure 3.11. Estimated closed-loop gain (black) and phase (red) response of the ECDL, polarization spectroscopy, and PI servo loop. From the data, the lock bandwidth is ∼250 kHz with a barely-stable 110° phase margin. Typically, the I-stage gain is decreased for a ∼100 kHz lock bandwidth.

the code in App. D.1.2, the total power contained in each trace is calculated and converted to an RMS frequency variation using the error signal slope (Fig. 3.6). The residual current noise is then estimated (Table 3.1). Frequencies below 4 kHz are omitted to exclude most acoustic resonances from the estimated current noise. Additionally, the noise contributions above 400 kHz are not included because of the polarization spectroscopy response rolling off at higher frequencies.

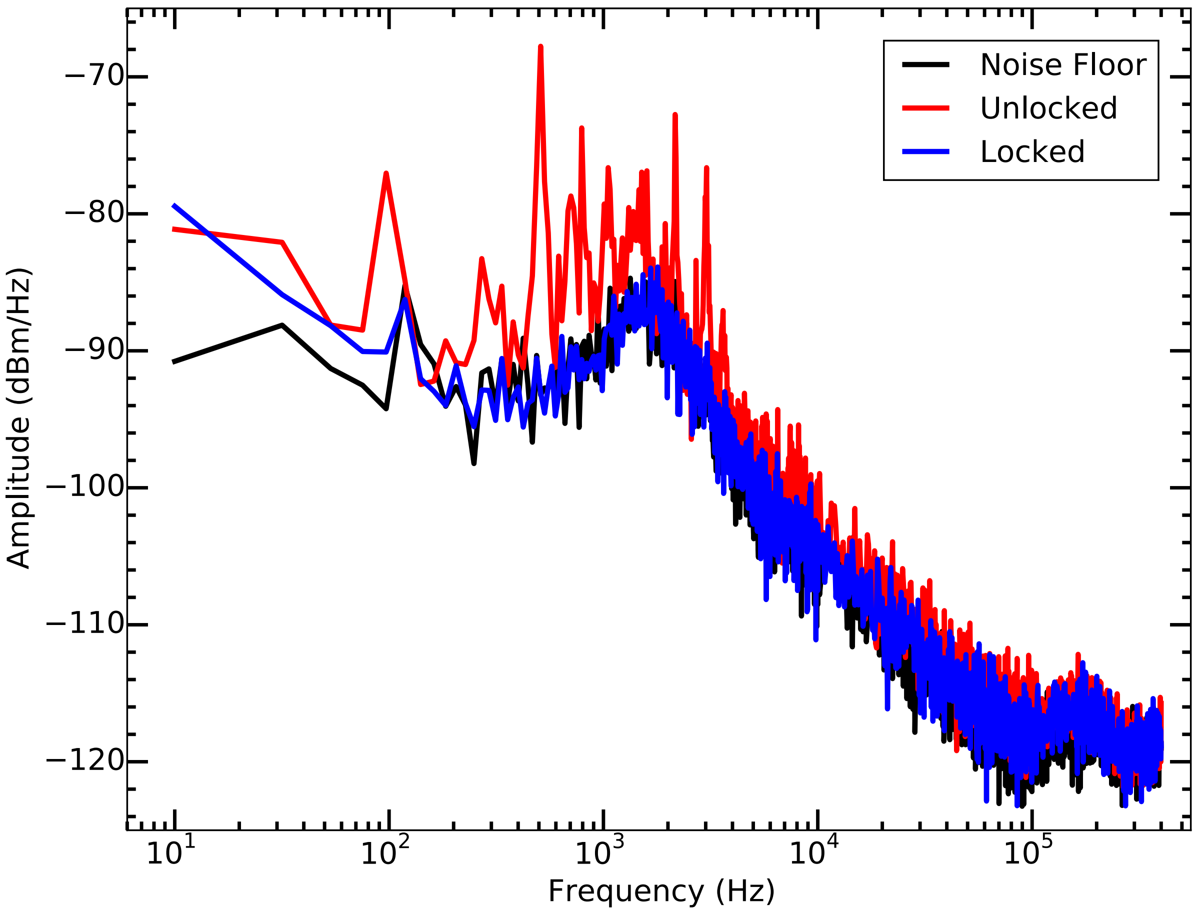


Figure 3.12. Spectral power density of the polarization spectroscopy signal while unlocked near resonance (red) and while locked (blue). The spectrum analyzer background is also shown (black).

Table 3.1. Residual RMS frequency and current noise estimated from the data in Fig. 3.12 for frequencies between 4 and 400 kHz.

	Total Frequency ($\mathrm{kHz_{RMS}}$)	Total Current ($\mathrm{nA_{RMS}}$)
Floor	61.6	117
Unlocked	117	224
Locked	56.9	108

From the data, the residual RMS frequency (current) noise when locked cannot be estimated because it is comparable to the noise floor. Therefore, this reference laser should be more than sufficient to lock the proposed transfer cavity assuming the slow-timescale-drift of the polarization spectroscopy zero-crossing is small. This can be quantified by using a second polarization spectroscopy setup in another room or by using Doppler-free frequency-modulated saturated absorption spectroscopy. Additionally, this system has been stable for $\sim$3 years with only minor optics realignment necessary to successfully lock the system.

CHAPTER 4

Further Current Driver Noise Reduction

The Libbrecht-Hall circuit is a well-known, low-noise current driver for narrow-linewidth diode lasers. An important feature of the circuit is a current limit to protect the laser diode. However, as the current approaches the maximum limit, the noise in the laser current increases dramatically. This chapter discusses this behavior and explores simple circuit modifications to alleviate this issue and is done in collaboration with the group of Dan Steck starting with an old version (MITPosCurRv03) of the circuit in App. A.1.3.

4.1. Circuit Overview

The Libbrecht-Hall circuit, originally published in 1993, has been one of the primary circuits used for low-noise current drivers for laser diodes [**192, 193, 194**]. An implementation of the circuit (Fig. 4.1) contains several distinct subsections: (1) voltage regulation, (2) active current-stabilization servo, and (3) output current monitoring. The Libbrecht-Hall circuit also contains a current-modulation section. Since this functionality is not the primary cause of the observed noise addressed in this chapter, this section has been omitted from Fig. 4.1. However, a discussion of our current-modulation section implementation, which contributes a significant amount of noise to the output current, is discussed in Sec. 5.3.2.

Section (1), the slow-turn-on, adjustable voltage source, centers on an LM317 voltage regulator that provides additional line regulation and defines a maximum current $I_{\max}$ for

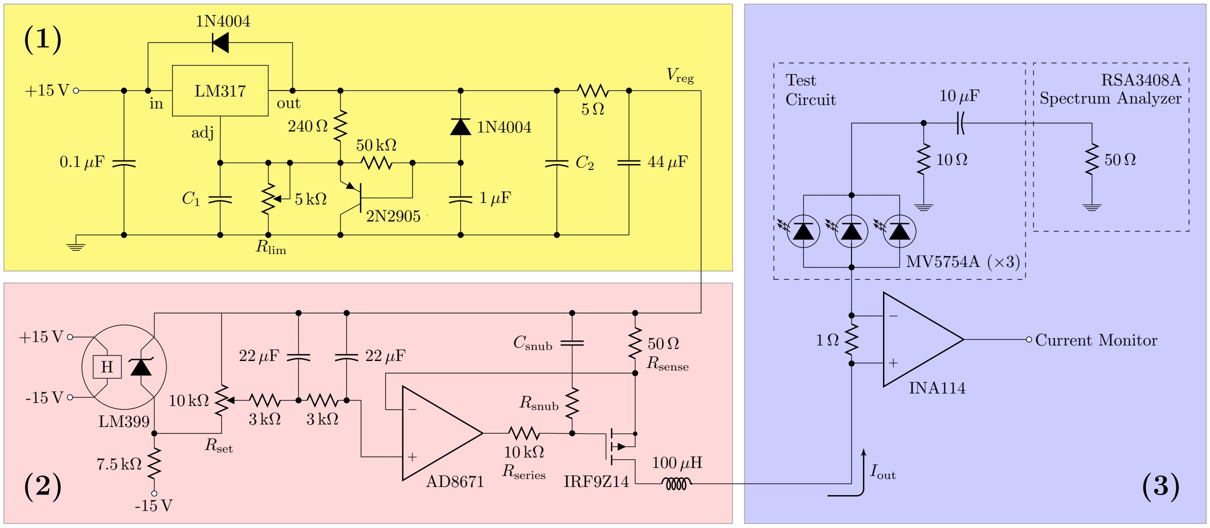


Figure 4.1. Simplified schematic of the Libbrecht-Hall-style test circuit: (1) voltage regulation, (2) active current-stabilization servo, (3) output current monitoring and noise measurement. The elements of interest in this chapter are the LM317 adjust pin bypass capacitor (C_1), the LM317 output bypass capacitor (C_2), and the snubber network (C_{snub} and R_{snub}). Not included in the schematic are filtering networks on the power lines of the AD8671, where 10 Ω series resistors with 100 μF tantalum and 0.1 μF ceramic bypass capacitors are used.

the circuit, excluding contributions from the current modulation section. This maximum current is defined by the output voltage of the regulator and the voltage drops across all components between its output and ground, mainly the sense resistor R_{sense} and the laser diode.

Section (2), the current-stabilization feedback section of the circuit, uses an AD8671 op-amp in combination with an IRF9Z14 MOSFET as a standard high-current source (see Fig. 4.12.A in Ref. [255]). A resistor R_{series} in series with the op-amp output and a snubber network, comprising capacitor C_{snub} and resistor R_{snub}, both act to stabilize the circuit at high frequencies.

In Section (3), an INA114 instrumentation amplifier measures the voltage across a 1 Ω sense resistor to monitor the output current. Ideally, this resistor is placed after the current-modulation junction to include its contribution.

The inclusion of the LM317 voltage regulator as the current limit in this circuit is clever: it effectively controls the power-supply voltage available to drive the laser, protecting it from inadvertent exposure to damagingly high currents. However, interactions between the regulator, the MOSFET, and the AD8671 op-amp turn out to be the major contributors to the current noise of the circuit as $I \to I_{\max}$. This means that the circuit cannot be used for low-noise current near $I_{\max}$, reducing the utility of the circuit, or even worse, unsuspecting users may be injecting more noise into their laser system than they realize by operating near $I_{\max}$. The purpose of this discussion is to document these noise issues and to explore how they can be mitigated via simple component changes to a greater extent than in previous work [**193, 194**]. The rest of this chapter has been organized into the following sections: (Sec. 4.2) feedback-loop noise estimate, (Sec. 4.3) voltage regulation problems; (Sec. 4.4) op-amp feedback-loop stabilization; and (Sec. 4.5) adjustments of component values and measurement results.

4.2. Feedback-loop Noise Estimate

The current noise density of the active current-stabilization servo, shown in Section (2) of Fig. 4.1, is dominated by the output noise of the op-amp and the 50 Ω sense resistor [**195, 256**]. For the discussion below, the original LT1028 op-amp will be discussed, and the expected current noise density using other low-noise op-amps is listed in Table 4.1.

The LT1028 input voltage noise density is 0.85 nV/$\sqrt{\text{Hz}}$ at 1 kHz for the better LT1028AM/AC variant. Since this is input-referenced and translated to a current noise density by the 50 Ω sense resistor, this contributes 17 pA/$\sqrt{\text{Hz}}$ of current noise density. Additionally, the input noise current density of 1 pA/$\sqrt{\text{Hz}}$ must be considered since the current flowing through the sense resistor can also flow into the op-amp inverting input, but this is a negligible additional contribution.

The Johnson-Nyquist RMS voltage noise density $V_{\text{RMS,JNn}}$ of a resistor of resistance R at a temperature $T = 25°\text{C}$ is given by

$$V_{\text{RMS,JNn}} = \sqrt{4k_{\text{B}}TR} = \left(0.128\ \frac{\text{nV}}{\sqrt{\text{Hz}}}\right)\sqrt{R}. \tag{4.1}$$

For the 50 Ω sense resistor, $V_{\text{RMS,JNn}} = 0.91$ nV/$\sqrt{\text{Hz}}$, corresponding to 18 pA/$\sqrt{\text{Hz}}$ current noise density. Adding the Johnson-Nyquist and op-amp current noise densities in quadrature yields 25 pA/$\sqrt{\text{Hz}}$, which is in agreement with Libbrecht and Hall [**192**].

Note that in this case, the amplifier and Johnson-Nyquist noises are comparable. A larger sense resistor decreases the effective amplifier noise but increases the Johnson-Nyquist noise. A smaller sense resistor increases the effective amplifier noise but decreases the Johnson-Nyquist noise. In both cases, the resulting current noise density increases. Additionally note that the op-amp noise figures at different frequencies will vary this comparable current noise density balance.

Expected current noise densities for several sense resistor values and low-noise op-amps obtained by repeating the above analysis is listed in Table 4.1, and the corresponding expected current noise from 5 to 200 kHz is listed in Table 4.2.

Table 4.1. Expected current noise densities for several low-noise op-amps using select sense resistor values using specification sheet noise figures at 1 kHz.

	Current Noise Density (pA/$\sqrt{\text{Hz}}$) using a sense resistor of				
Op Amp	50 Ω	40 Ω	33 Ω	25 Ω	14 Ω
AD797	26	30	35	44	73
AD8671	59	73	88	115	203
ADA4898-1	26	30	35	44	73
LT1028AM	25	29	34	43	70
LT1128AM	25	29	34	43	70

Table 4.2. Expected current noise using the current noise densities in Table 4.1 integrated from 5 to 200 kHz for several low-noise op-amps using select sense resistor values.

	Current Noise (pA_{RMS}) using a sense resistor of				
Op Amp	50 Ω	40 Ω	33 Ω	25 Ω	14 Ω
AD797	363	419	489	614	1020
AD8671	824	1020	1230	1610	2830
ADA4898-1	363	419	489	614	1020
LT1028AM	350	405	475	600	977
LT1128AM	350	405	475	600	977

4.3. Voltage Regulation Problem

The current-dependent inductive output impedance of the LM317 regulator is a commonly neglected issue with 3-terminal voltage regulators [**257**]. Coupled with an output capacitance, this can produce a noise peak corresponding to the LC resonance. The op-amp can only correct this to a certain extent. However, the LM317 is not the *primary* cause of the noise we observe in Fig. 4.2: as current increases, the regulator's output

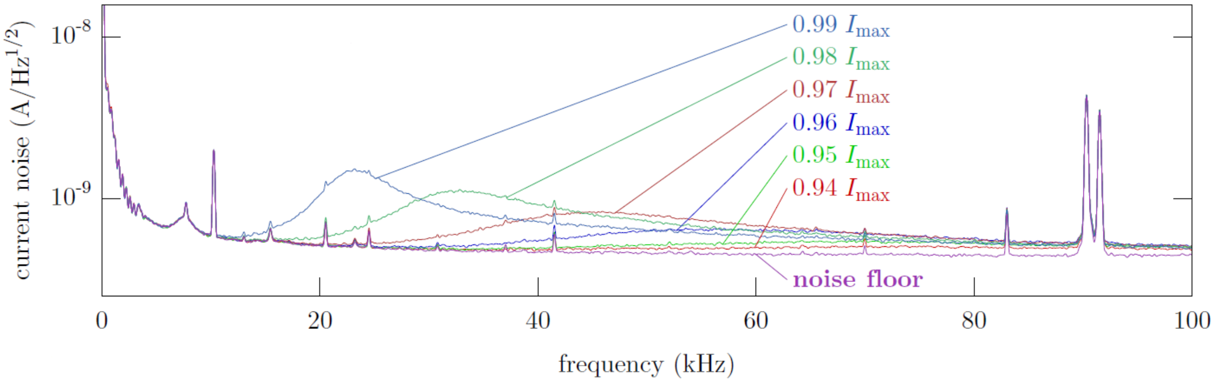


Figure 4.2. Current-noise density of the Libbrecht-Hall style circuit. $C_1 = 10\ \mu\text{F}$, $C_2 = 10\ \mu\text{F}$, and $C_{\text{snub}} = 33$ nF, and traces are plotted for several current set points (labeled as a fraction of $I_{\max}$). The noise peak grows and moves from right to left as current is increased toward $I_{\max}$.

inductance decreases [257], pushing the resonance frequency higher instead of lower as we observe.

The LM317 noise can be reduced by increasing the values of C_1 and C_2 on the adjust pin and output of the regulator, respectively [257]. In particular, as C_2 increases, the LC-resonance frequency decreases, and the op-amp can more readily prevent contamination of the output current due to the higher op-amp gain at low frequencies.

4.4. Feedback-loop Stabilization

The AD8671 stabilizes the output current of the IRF9Z14 p-channel MOSFET by controlling its gate voltage V_{G} so the source voltage V_{S} follows the set point V_{set}. Because of the finite gain and high-frequency roll-off of the op-amp, its output impedance is effectively inductive (see Sec. 4.4.2.A in Ref. [255]). This inductance, along with the gate capacitance of the MOSFET, can produce a destabilizing resonance.

External compensation components can stabilize an op-amp driving a capacitive load [192, 193, 194]. A series resistor R_{series} on the op-amp's output damps the LC resonance, and a snubber network, comprising C_{snub} and R_{snub}, reduces the bandwidth of the current source [258, 259].

This network enables stable operation of the current driver, but does not completely solve the issue. As $I \to I_{\mathrm{max}}$, the resonance behavior reemerges, although in a less dramatic fashion. An increase in the MOSFET output current I_{SD} increases the source-gate voltage V_{SG} and decreases the source-drain voltage V_{SD}. Near I_{max}, V_{SD} becomes small, and the transconductance g_{m} of the MOSFET decreases. This effectively reduces the open-loop gain and increases the op-amp's output inductance. We therefore expect the changing inductance to shift the noise *downward* in frequency. Note that a smaller V_{SD} implies a larger MOSFET gate capacitance, consistent with the decrease in peak frequency. However, this is a smaller effect. Additionally, the Q-factor of an RLC filter increases with the inductance. We observe in Fig. 4.2 a noise peak that both narrows and decreases in frequency as $I \to I_{\mathrm{max}}$. This peak is consistent with the 12-kHz bump mentioned by Libbrecht and Hall [192], which they estimate contributes 16 $\mathrm{nA}_{\mathrm{RMS}}$ of current noise. Note that the data presented in Ref. [192] was taken while operating likely at $<0.90 I_{\mathrm{max}}$ for their test circuit, and the onset of the observed noise documented in this chapter begins when the set current is $>0.90 I_{\mathrm{max}}$.

The inductor after the MOSFET also exacerbates the feedback instability. The inductor's voltage drop increases at high frequencies, further reducing the MOSFET's high-frequency transconductance. The inductor can also couple with the MOSFET's capacitance C_{SD} to create a low-impedance path for noise. The current noise can be improved by bypassing the inductor with a short or resistor [260]. However, this solution eliminates a desirable feature of the original Libbrecht-Hall design, allowing the possibility for an RF-modulation signal to destabilize the circuit.

Another simple solution to the noisy behavior near I_{max} could be to add a resistor in series with R_{set}, limiting the output current maximum to less than that provided by V_{reg}. But this is equivalent to operating away from I_{max}, and either prevents operation near the laser diode's true maximum current or undermines the protection provided by relying on V_{reg} to limit I to I_{max}.

In summary, the interaction of the op-amp and MOSFET introduces instability as $I \to I_{\mathrm{max}}$. If the destabilization overlaps in frequency with the noise of the LM317 described in Sec. 4.3, then more noise is added to the output current.

Some simple alterations to the circuit address these effects. First, we can increase C_2 to reduce the frequency of the LM317 noise peak. The op-amp has higher low-frequency gain and is thus better able to handle power supply fluctuations at low frequencies. Second, we can lower the roll-off frequency of the snubber network. This further suppresses the effect of the op-amp/MOSFET resonance at the expense of op-amp bandwidth, so the snubber frequency should not be reduced excessively. We cannot remove the inherent noisy behavior of the feedback loop because $g_{\mathrm{m}} \to 0$ as $I \to I_{\mathrm{max}}$. However, we can improve how close we can get to I_{max} before the noise appears.

4.5. Testing and Adjustments

For our op-amp and MOSFET, we have chosen components already discussed in Ref. [193]. The VP0106 FET in the original Libbrecht-Hall paper is replaced with an

IRF9Z14 MOSFET due to higher current drive capacity (higher maximum drain current and power dissipation and lower on-state resistance) while maintaining similar dynamic characteristics. The AD8671 current-feedback op-amp offers slightly lower current noise than the original design's LT1028 but higher voltage noise.

We use three LEDs (Fairchild MV5754A) connected via a two meter shielded cable as a dummy load and record current-noise data with a Tektronix RSA3408A spectrum analyzer. A capacitor blocks the DC signal and a 10 Ω resistor converts current fluctuations to voltages that are monitored by the analyzer. This method gives a higher noise floor than observed in Ref. [**194**], but enables a larger bandwidth. We average 500 traces, plot the current spectral density, and quantify the noise by integrating the signal and subtracting the integrated baseline, obtained with the testing circuit disconnected from the switched-off current driver but still attached to the spectrum analyzer. We measure the standard error of background noise to be less than 0.5 $\mathrm{nA_{RMS}}$ and assume a similar variation in the signal. Note that the spectrum analyzer LCD display contributes a large amount of noise at 40 kHz. This was ameliorated by placing the test circuit far from the spectrum analyzer.

Fig. 4.2 shows the behavior described in Sec. 4.3. As the output current approaches $I_{\max}$, the feedback loop becomes less stable. A noise peak appears around 100 kHz and shifts toward smaller frequencies. Onset of this noise begins when the set current is within $\sim$10% of $I_{\max}$. Narrow peaks in the figure are due to pickup in our measurement section, since they are present in the baseline trace as well. For these data we use capacitor and snubber values specified in Ref. [**194**] for the AD8671 and IRF9Z14 ($C_1 = 10\ \mu$F, $C_2 = 10$

μF, $C_{\text{snub}} = 33$ nF, and $R_{\text{snub}} = 100\ \Omega$). Note that the feedback circuit is not oscillating, and that the emerging noise peak is easy to overlook.

Characteristics of this noise peak are in part determined by the capacitors that help to stabilize the LM317 regulator. We found that increasing C_1 and C_2 reduced the amplitude and delayed the onset of the noise as current was increased, but capacitances larger than $C_1 = 22\ \mu$F and $C_2 = 100\ \mu$F did not further mitigate the problem. We settled on a 22 μF tantalum capacitor for C_1 and a pair of 470 μF electrolytic capacitors for C_2. Note that if little or no capacitance is used for C_1 and C_2, the circuit behaves quite poorly near I_{max}: the noise peak is broader, an order of magnitude larger in amplitude, and is easily observed in the frequency-noise spectrum of an atomic absorption signal produced with a laser driven by the circuit. We also confirmed that removing the 5 Ω, 44 μF low-pass filter at the LM317's output increases the noise, so these are important components to include.

Next we increase C_{snub}, the snubber-network capacitor, and observe that this further reduces the noise peak and delays onset of its appearance as I approaches I_{max}. To illustrate the improvements, we choose $I = 0.99 I_{\text{max}}$ and plot data in Fig. 4.3 for (i) original capacitor values from Ref. [**194**], (ii) altered LM317 bypass capacitors, and (iii) additionally altered snubber capacitor. As seen in the plot, the capacitors in (iii) have significantly alleviated the noise when compared with (i). Table 4.3 shows integrated current noise above the background for each of the three configurations at various output current levels. Note that while the larger capacitors in (ii) and (iii) certainly stabilize the circuit as I approaches I_{max}, they also reduce the integrated noise when operating far away from I_{max}. Although integrated noise values are unavailable for comparison

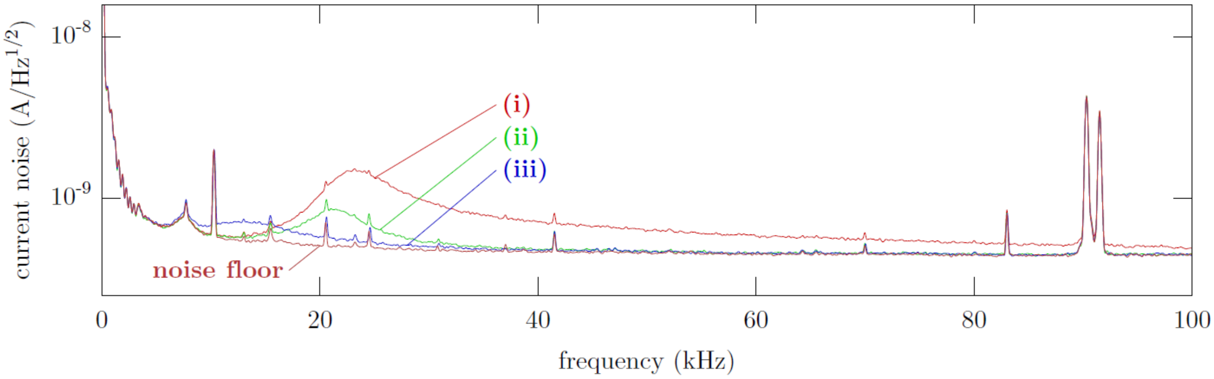


Figure 4.3. Current-noise density of the Libbrecht-Hall style circuit variations operating at 99% of the maximum current. The snubber resistor is $R_{\text{snub}} = 100\ \Omega$, and capacitor values used are (i) $C_1 = 10\ \mu\text{F}$, $C_2 = 10\ \mu\text{F}$, $C_{\text{snub}} = 33$ nF; (ii) $C_1 = 22\ \mu\text{F}$, $C_2 = 940\ \mu\text{F}$, $C_{\text{snub}} = 33$ nF; and (iii) $C_1 = 22\ \mu\text{F}$, $C_2 = 940\ \mu\text{F}$, $C_{\text{snub}} = 100$ nF.

Table 4.3. Current noise above background, integrated from 5 kHz to 200 kHz for various set points. The snubber resistor is $R_{\text{snub}} = 100\ \Omega$, and capacitor values used are (i) $C_1 = 10\ \mu\text{F}$, $C_2 = 10\ \mu\text{F}$, $C_{\text{snub}} = 33$ nF; (ii) $C_1 = 22\ \mu\text{F}$, $C_2 = 940\ \mu\text{F}$, $C_{\text{snub}} = 33$ nF; and (iii) $C_1 = 22\ \mu\text{F}$, $C_2 = 940$ μF, $C_{\text{snub}} = 100$ nF.

Current	Integrated noise (nA_{RMS})		
	(i)	(ii)	(iii)
$0.80I_{\max}$	35(6)	32(6)	28(7)
$0.94I_{\max}$	75(3)	46(4)	41(5)
$0.95I_{\max}$	90(2)	49(4)	43(5)
$0.96I_{\max}$	113(2)	45(5)	45(5)
$0.97I_{\max}$	138(1)	51(4)	49(4)
$0.98I_{\max}$	164(1)	60(3)	57(4)
$0.99I_{\max}$	175(1)	76(3)	64(3)

in Refs. [**193**, **194**], our values at $0.80I_{\max}$ are comparable with those estimated in the original Libbrecht-Hall paper [**192**].

We also thought it interesting to compare the performance of the AD8671 to some other low-noise op-amps: the original Libbrecht-Hall LT1028, the LT1128, the AD797, and

Table 4.4. Current noise above background, integrated from 5 kHz to 200 kHz for various set points. For these data we use $C_1 = 22\ \mu$F, $C_2 = 940\ \mu$F, $C_{\mathrm{snub}} = 200$ nF, and $R_{\mathrm{snub}} = 50\ \Omega$, and repeat the measurement for several low-noise op-amps.

	Integrated noise (nA$_{\mathrm{RMS}}$)				
Current	AD8671	ADA4898-1	AD797	LT1028	LT1128
$0.80I_{\max}$	18(4)	12(6)	12(6)	17(4)	15(5)
$0.94I_{\max}$	40(2)	22(3)	15(5)	33(2)	34(2)
$0.95I_{\max}$	41(2)	26(3)	28(3)	48(2)	37(2)
$0.96I_{\max}$	45(2)	39(2)	36(2)	44(2)	46(2)
$0.97I_{\max}$	53(1)	56(1)	49(1)	55(1)	48(1)
$0.98I_{\max}$	63(1)	53(1)	53(1)	54(1)	61(1)
$0.99I_{\max}$	80(1)	57(1)	51(1)	57(1)	71(1)

the ADA4898-1. When using the capacitor values from circuit (iii), however, the AD797 and LT1028 are unstable and the snubber network must be adjusted to sufficiently load the output of the op-amp [258]. We set R_{snub} to 50 Ω and change C_{snub} to 200 nF, maintaining the same RC frequency. Integrated noise values of the five op-amps are given in Table 4.4. The performance of the AD8671 with this snubber network is better at low currents than in circuit (iii), but worse at high currents. The top overall performer is the AD797, which consistently has the lowest integrated current noise. Any of the five op-amps tested give low noise far away from $I_{\max}$, and it is possible that their performance could be improved with further changes to the snubber network. However, the interaction of the capacitive load and the op-amp's output impedance create a frequency peaking unique to each op-amp/MOSFET combination, and there is no universal snubber-network solution. Additionally, the measured current noise for all op-amps is significantly higher than those estimated in Table 4.2; the circuit noise is dominated by the various inductances and capacitances in the circuit components.

4.6. Summary

Thus, more than two decades after publication, the current driver design of Libbrecht and Hall remains a dependable low-noise solution for powering laser diodes. Users, however, should be aware of the effect on performance as the circuit's output current approaches the maximum set limit. The adjustments described in this chapter are simple to implement and improve the circuit's behavior.

CHAPTER 5

Future ECDL System Development Prospects

Although the in-house ECDL system optomechanics and drive electronics are currently sufficient for most applications in the laboratory, a multitude of improvements will boost the performance and applicability of the ECDL systems. Pushing the acoustic resonances to higher frequencies will improve the passive frequency stability, reducing the frequency jitter and its coupling to the environment. Additional improvements to the optomechanics will ease collimation, while the enclosure could be redesigned to both reduce the system footprint and reduce electrical pickup noise on the laser output. Furthermore, improvements to the current driver circuit modulation section will reduce the output current noise of the driver by at least an order of magnitude.

5.1. ECDL Optomechanics and Enclosure

5.1.1. Grating Arm Acoustic Resonances

As discussed in Sec. 3.4.1, our ECDL optomechanical design contains multiple acoustic resonances, which cause frequency jitter on the laser output as the resonances couple to ambient acoustic noise. While a servo loop with ~100 kHz bandwidth provides sufficient suppression of this frequency jitter (Sec. 3.4.3), improving the passive stability of the system will reduce the demands on the servo loop as well as help prevent the servo loop from falling out of lock from loud ambient acoustic noises, i.e. a door slamming closed or setting up optics on the same optical table. Similar designs with a much-shorter grating arm length exhibit acoustic resonances at ∼2 kHz [**163, 160**]. While these systems have a slightly-higher passive stability, the non-optimized pivot position reduces the mode-hop free tuning range (Sec. 2.2).

A mechanical resonance frequency f_r of a lever arm is given by

$$f_r = \frac{1}{2\pi}\sqrt{\frac{\mu}{I}} \tag{5.1}$$

where μ is the torsion constant (stiffness of the pivot) and I is the moment of inertia of the lever arm. For the simplified case of a rod of length L and mass m pivoting about one end, the moment of inertia is

$$I_{\text{end}} = \frac{mL^2}{3}. \tag{5.2}$$

Therefore, the mechanical resonance frequency scales as $1/(\sqrt{m} \cdot L)$. Because our grating lever arm is both long and relatively heavy, the moment of inertia is large and the resulting resonance frequency is low.

To increase the resonance frequencies of the grating arm, the arm should be made (1) lighter, (2) shorter, and/or (3) stiffer. For (1), the grating and mirror substrates can be thinner and therefore lighter by at least a factor of two. For (2), the excess material after the grating can be removed, which would also help (1). However, this would eliminate the corner-cube mirror mount location, necessitating the mount be redesigned. For this, the basic idea used by Toptica in the DL100 ECDL can be adopted (Fig. 5.1); the corner cube mirror mount would be redesigned out of thin (and therefore light) sheet metal, extend from the end of the shortened grating arm at a 90° angle from the laser base, and have a bend at 45° downward to reflect the output beam in the correct direction.

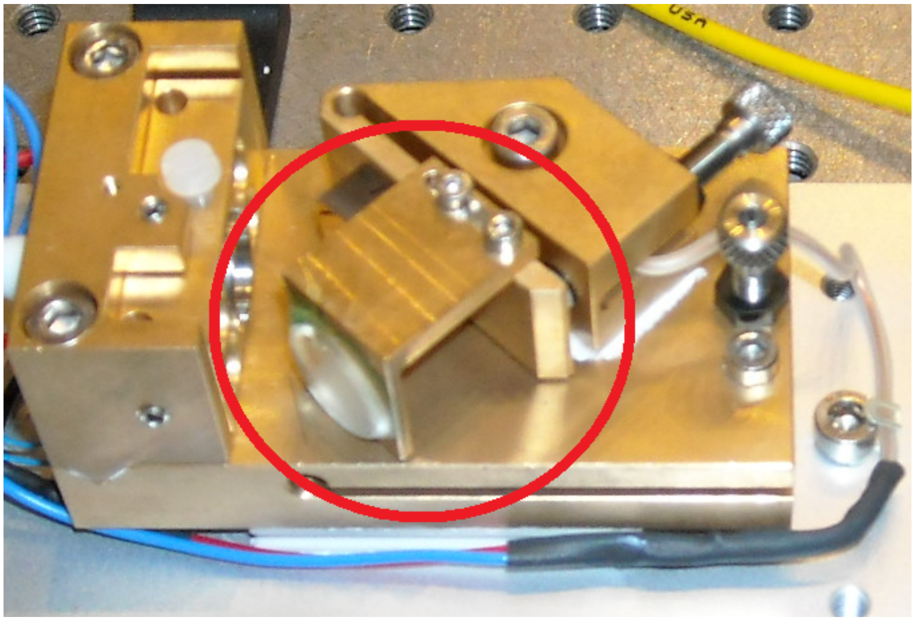


Figure 5.1. Image of the Toptica DL100 ECDL with the thin corner cube mirror mount circled in red.

Additionally, the pivot-to-grating length could be shortened by a factor of ∼2 at the expense of mode-hop free tuning range. For (3), the grating arm stiffness could be increased similar to the design in Ref. [160]. This could additionally make the arm lighter. All of the above possible solutions would greatly benefit from a finite element analysis to model the vibrational modes and optimize the resulting grating arm and corner cube mirror mount system. Furthermore, shear damping using a compressed o-ring [160] or magnetic damping of the grating arm could also be employed.

5.1.2. Grating Arm Adjusters

The implemented PZT and ultra-fine hex adjuster combination for the grating arm rough adjustment could be replaced with an ultra-fine adjuster with built-in PZT element, design optimized to constrain the PZT element such that cementing it in place is unnecessary, and/or using a low-voltage PZT stack. For the latter option, the ThorLabs PZT driver could then be omitted and PZT element driven directly by the current driver BUF634 chip up to ±15 V [160]. Additionally, the PZT element could be epoxied between two sapphire disks to help distribute the load to the grating arm [160], and the grating arm design could include a pocket for either the disk or PZT element.

5.1.3. Collimation Lens Mounting

The current collimation lens design is non-ideal with both the use of 3/8”-40 threads that are difficult to machine to a tight-fit tolerance and the requirement of cementing the lens in place. Modifying the laser head such that ThorLabs mounted aspheric lenses could be used would be a substantial improvement. Additionally, the modifications could be engineered such that the lens could mount in an external carrier (i.e. ThorLabs SM05-series lens tube adapters), which is then mounted to the laser head. This would additionally increase the adjustment range for long laser diode packages.

5.1.4. Enclosure

The ECDL enclosures are significantly larger than necessary, and the acrylic is non-conductive. Redesigning the enclosures out of metal would decrease the system footprint and provide EMI shielding, which would reduce electrical pickup frequency and amplitude noise on the laser output. A possible implementation is to use a solid base to which a front and rear panel bolts to. The remaining two sides and top can be enclosed with a single piece of material and bolted to the base, front, and/or rear panels similarly to the Toptica ECDLs.

5.2. Circuit Improvements: Libbrecht-Hall Implementation

As discussed in Ch. 4, simple component changes to our implementation of the Libbrecht-Hall circuit will decrease the primary current drive noise substantially. This includes a change in servo-loop op-amp, snubber network resistor and capacitor values, larger bypass capacitances on the LM317 voltage regulator, and additional input power filtering of the servo op-amp with 10 Ω series resistors with 100 μF tantalum and 0.1 μF ceramic bypass capacitors. Our use of diodes and large capacitors (App. A.1.3) serve to protect the op-amp inputs when the amplifier is unpowered or unusually powered, i.e. transients when turning-on or turning-off the circuit, since many ultralow-noise amplifiers use diodes to help protect the inputs rather than using current-limiting input resistors [**261**].

5.3. Circuit Improvements: Modulation Stage

As noted in Ch. 4, the modulation section is omitted by removing the instrumentation amplifier that provides the current feed-forward current (IC6) from our circuit implementation (App. A.1.3). This is because our modulation implementation using instrumentation amplifiers injects a significant amount of noise into the output of the current driver; while the current driver noise with IC6 removed is 36.2 $\mathrm{nA_{RMS}}$, the noise with IC6 in place increases to 381 (571) $\mathrm{nA_{RMS}}$ using an INA114 (INA217) instrumentation amplifier [**262**]. However, this noise can be significantly reduced with a redesign of the modulation section without eliminating the desirable features of the section.

5.3.1. Original LH Design Noise Estimate

The original current-modulation section by Libbrecht and Hall [**192**] is shown in Fig. 5.2.

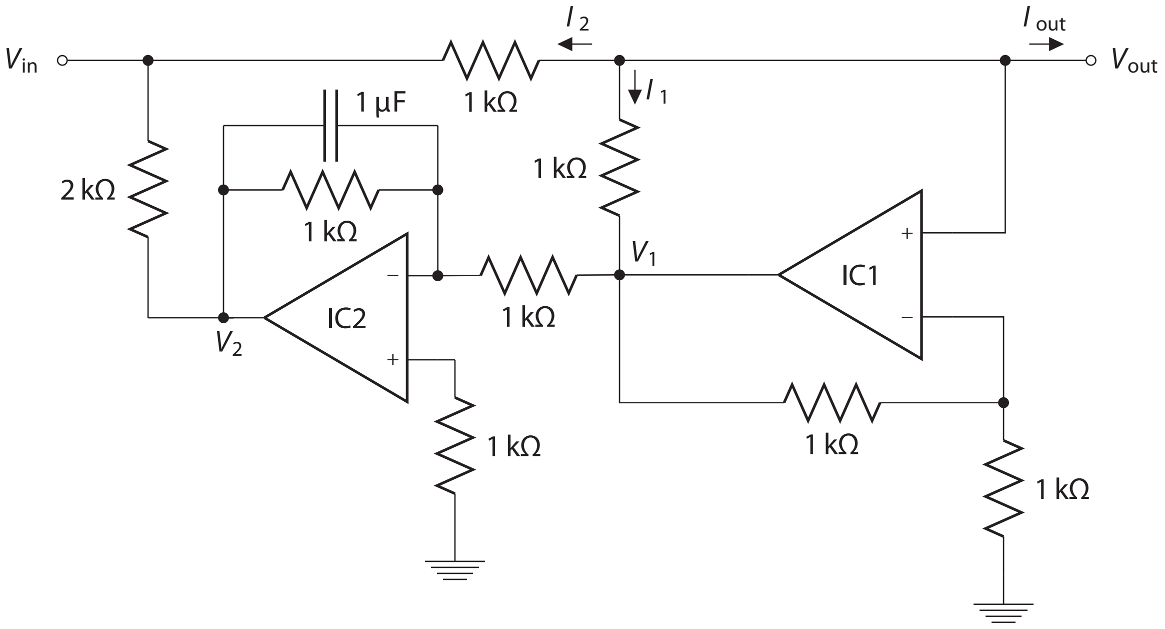


Figure 5.2. Current modulation section of the Libbrecht-Hall circuit in Ref. [192] with key currents and voltages labeled. IC1 and IC2 are OP-27 op-amps. Figure adapted from Ref. [256].

IC1, a non-inverting amplifier with a gain of 2 with non-inverting input connected to V_{out}, holds its output voltage at V_{out}, which is the operating output voltage of the constant-current drive flowing through the load of the laser diode, protection circuit, and cabling [256]. Therefore,

$$V_1 = 2V_{\text{out}} \tag{5.3}$$

and

$$I_1 = \frac{V_{\text{out}} - V_1}{1\ \text{k}\Omega} = \frac{V_{\text{out}} - 2V_{\text{out}}}{1\ \text{k}\Omega} = -\frac{V_{\text{out}}}{1\ \text{k}\Omega}. \tag{5.4}$$

The input voltage V_{in} is converted to current by the 1 kΩ resistor such that

$$I_2 = \frac{V_{\text{out}} - V_{\text{in}}}{1\ \text{k}\Omega} \tag{5.5}$$

so the total output current is

$$I_{\text{out}} = -I_1 - I_2 = \frac{V_{\text{in}}}{1\ \text{k}\Omega} \tag{5.6}$$

and is independent of the output of IC2. IC2 is an inverting amplifier with a gain of −1. Therefore with V_{in} left floating (open),

$$V_2 = -V1 = -2V_{\text{out}} \tag{5.7}$$

and

$$I_2 = \frac{V_{\text{out}} - V_2}{3\ \text{k}\Omega} = \frac{V_{\text{out}} + 2V_{\text{out}}}{3\ \text{k}\Omega} = \frac{V_{\text{out}}}{1\ \text{k}\Omega}. \tag{5.8}$$

So the total output current is

$$I_{\text{out}} = -I_1 - I_2 = -\frac{V_{\text{out}}}{1\ \text{k}\Omega} + \frac{V_{\text{out}}}{1\ \text{k}\Omega} = 0. \tag{5.9}$$

So IC2 serves to make $I_{\text{out}} = 0$ when V_{in} is disconnected [256].

Following the discussion from Sec. 4.2, the Johnson-Nyquist noise $V_{\text{RMS,JNn}}$ of the 1 kΩ sense resistor of IC1 contributes 4.1 nV/$\sqrt{\text{Hz}}$ of voltage noise density, and the two 1 kΩ resistors in parallel on the inverting input contribute 2.9 nV/$\sqrt{\text{Hz}}$. Additionally, the OP-27A/E op-amp has an input voltage noise density of 3.0 nV/$\sqrt{\text{Hz}}$ and input current noise density of 0.4 pA/$\sqrt{\text{Hz}}$. Converted to current noise by the 1 kΩ sense resistor, the total voltage noise density of 5.8 nV/$\sqrt{\text{Hz}}$ translates to 5.9 pA/$\sqrt{\text{Hz}}$ of current noise density including the op-amp input contribution. Assuming that V_{in} in a noiseless source (i.e. grounded), IC2 does not contribute to the modulation section's output current noise.

Therefore, the 25 pA/$\sqrt{\text{Hz}}$ current noise density of the main servo loop (Sec. 4.2) combines with the 5.9 pA/$\sqrt{\text{Hz}}$ from IC1 for a total current noise density of 26 pA/$\sqrt{\text{Hz}}$ [256].

If we instead assume that V_{in} is floating, IC2 contributes to the current noise density of the modulation section. The output of IC2 contributes the same total 5.9 nV/$\sqrt{\text{Hz}}$ of voltage noise density. Summing the additional Johnson-Nyquist noise of the 2 kΩ resistor (5.7 nV/$\sqrt{\text{Hz}}$) yields a total of 8.1 nV/$\sqrt{\text{Hz}}$ at V_{in}. This voltage is converted to current noise by the 1 kΩ resistor, contributing 8.1 pA/$\sqrt{\text{Hz}}$ of current noise density. Summing the contributions from IC1 and IC2 yields a modulation section output current noise density of 10.0 pA/$\sqrt{\text{Hz}}$ for a total circuit current noise density of 27 pA/$\sqrt{\text{Hz}}$ [256].

Note that lower-noise op-amps could be used in place of the OP-27s, but the overall noise reduction in the circuit is not significant compared with the main servo loop contribution of 25 pA/$\sqrt{\text{Hz}}$.

5.3.2. Instrumentation Amplifier IC Implementation Noise Estimate

The in-house current-modulation section implementation is shown in Fig. 5.3.

Both IC3 and IC6 instrumentation amplifiers operate at unity gain. The input voltage is applied to IC3 (*MOD_IN*). The *MOD_OFFSET* potentiometer provides a negative DC voltage shift to the output of IC3, adjustable between 0 and -10 V. The output of IC3 splits to an inverting unity-gain amplifier and buffer for the PZT signal while the other leg is attenuated by the *MOD_GAIN* trimmer to IC6, which converts this voltage to current using the 500 Ω resistor. This modulation stage implementation is clever since it modulates both the PZT and drive current, enabling the current feed-forward necessary for a large mode-hop free tuning range [178, 179, 180, 164, 165]. Additionally, this

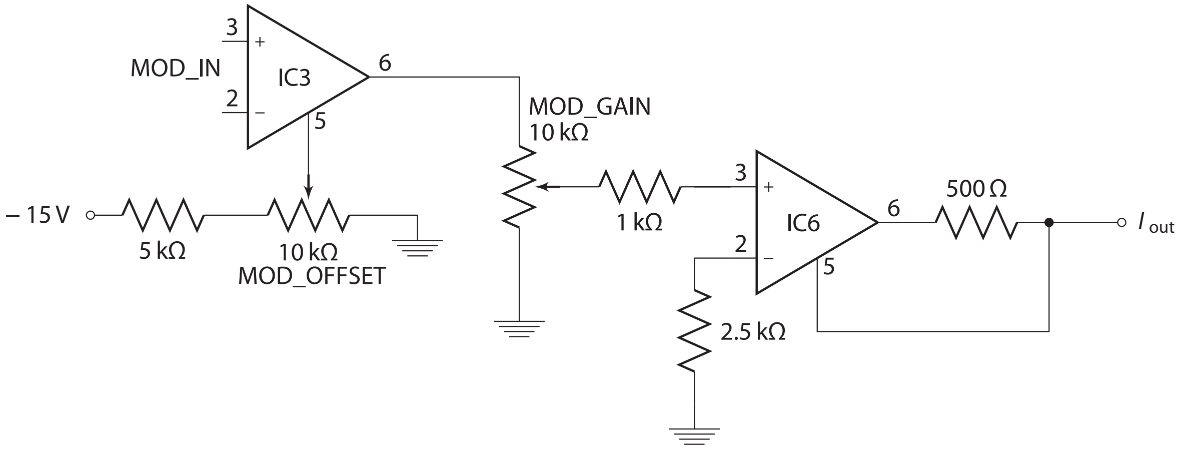


Figure 5.3. Current modulation section of the in-house Libbrecht-Hall circuit implementation in App. A.1.3. IC3 and IC6 are INA114 or INA217 instrumentation amplifiers. Figure adapted from Ref. [**256**].

feature works when adjusting the laser frequency both by hand via the MOD_OFFSET potentiometer and by an input voltage (i.e. from a servo circuit) via the MOD_IN connection.

For discussion, assume IC3 and IC6 are both INA217 instrumentation amplifiers with 1.3 nV/$\sqrt{\text{Hz}}$ of input voltage noise density, 0.8 pA/$\sqrt{\text{Hz}}$ of input current noise density, and 90 nV/$\sqrt{\text{Hz}}$ of output voltage noise density with all specified at 1 kHz. If both MOD_IN pins are shorted to ground and the -15 V supply is noiseless, the output voltage noise density of IC3 combined with the worst-case Johnson-Nyquist noise of 11 nV/$\sqrt{\text{Hz}}$ from MOD_OFFSET, 13 nV/$\sqrt{\text{Hz}}$ from MOD_GAIN, and 4.1 nV/$\sqrt{\text{Hz}}$ from the 1 kΩ input resistor totals to 92 nV/$\sqrt{\text{Hz}}$ on the non-inverting input of IC6. The 2.5 kΩ resistor on the inverting input of IC6 contributes 6.5 nV/$\sqrt{\text{Hz}}$. Summing the voltage noise densities on the non-inverting and inverting inputs with the input voltage noise density and output voltage noise density yields 130 nV/$\sqrt{\text{Hz}}$. This is converted to a current noise via the 500

Ω resistor for a total current noise density of 260 pA/$\sqrt{\text{Hz}}$ for the worst-case scenario. For the best-case scenario, both MOD_OFFSET and MOD_GAIN are shorted to ground. Therefore, the output voltage noise density of IC6 would be 90 nV/$\sqrt{\text{Hz}}$ for a total current noise density of 180 pA/$\sqrt{\text{Hz}}$.

Thus, the output current density noise from this implementation of a modulation section contributes an order of magnitude more noise to the current driver output than that from the main servo loop. The likely source of the large noise of the instrumentation amplifiers is the internal 5 or 6 kΩ (25 kΩ) resistors between internal op-amps in the INA217 (INA114) contributing to and amplifying Johnson-Nyquist noise.

5.3.3. Possible Modulation Section Solutions

One possible modulation section solution is to use the original design by Libbrecht and Hall with separate servo loops for the PZT and the laser drive current.

However, this would eliminate several desirable features of the in-house modulation section implementation: input voltage buffering, shielding the laser diode from possible ESD events, and implementation of current feed-forward. Therefore, the modulation section should be redesigned to use low-noise op-amps and low-value resistors instead of the instrumentation amplifiers.

IC3 would be replaced with a unity-gain buffer and differential amplifier to provide the DC offset voltage. MOD_OFFSET should be replaced with a 1 kΩ potentiometer referenced to an ultralow-noise voltage source, i.e. the buffered output of a LM399. MOD_GAIN should be replaced with a 1 kΩ trimmer, the 1 and 2.5 kΩ resistors removed, and IC6 replaced with three op-amps to perform the same function only with far lower noise. The value of the 500 Ω resistor should also increase, but this may require additional changes to the circuit to properly set the current feed-forward current gain. Additionally, the design could include a feature such that the current feed-forward op-amp only sinks current to prevent over-driving the laser diode. Alternatively, the system could use a current servo loop similar to the main current servo to sink current from the laser diode.

While a noise analysis of these proposed solutions are not performed since it depends on the specific circuit design, any redesign of the modulation section should include a similar noise analysis.

Part 2

Molecular Cooling

mQLS can be performed without molecular state preparation. However, this relies upon thermal redistribution to place the spectroscopy ion in the desired or required initial state to perform an experiment, and this process is random and uncontrolled. Several non-optical techniques could be employed to aid this thermal state preparation such as cryogenic buffer gas cooling or using a cryogenic ion trap apparatus. But both are technically challenging and still rely upon an uncontrollable process.

It would be far more efficient to use an optical pumping technique, using RF radiation to couple rotational levels, far-IR radiation to couple vibrational levels, or optical radiation to couple electronic levels. With a strategic choice of molecular species, radiation source, and addressed molecular states, it would be possible to achieve at least partial control of the quantum state of a molecule, in particular the rotational degree of freedom. If this could be realized in an efficient, fast manner, mQLS could be performed efficiently.

This part discusses previous molecular rotational cooling demonstrations, and extends the work performed with select hydrides using vibrational transitions to electronic. Prospects of using electronic transitions, promising in reducing state preparation timescales, will then be discussed with application to our molecule of choice, AlH^+. The experiment conducting this molecular cooling by using a spectrally-filtered pulsed laser to simultaneously drive all rotational cooling transitions will then be described. This part will then be concluded with a discussion on future experiments with AlH^+ with the goal of full molecular coherent control and non-destructive fluorescence detection.

CHAPTER 6

Molecular Cooling Prospects

6.1. Previously Demonstrated Molecular Cooling

Molecular cooling historically has been a challenge owing to (1) the lack of control of molecular vibrations and rotations and (2) the large number of thermally populated states at room temperature, each requiring a unique laser or RF frequency to be addressed. For example, AlH^+, a favorable molecule for rotational cooling, can undergo $\sim$30 electronic excitation-relaxation cycles before the molecule undergoes a vibrationally-exciting electronic decay. Additionally, because the molecule has a large vibrational energy spacing, nearly all of the population is in the ground vibrational state at room temperature. Similarly, owing to the large rotational energy spacing, roughly *only* ten rotational states are populated at room temperature [**114**, **263**]. If narrowband, CW lasers are used to address these ten thermally populated rotational levels, at least 9 laser systems would be required for rotational cooling. For diatomics with smaller reduced mass such as oxides or fluorides, the rotational energy spacing is smaller by an order of magnitude, and hence the number of thermally populated levels at room temperature increases dramatically.

Other research groups previously have demonstrated translational [**100**, **101**, **102**], vibrational [**105**, **264**, **107**, **265**], and rotational cooling [**110**, **111**, **109**, **18**, **108**] of diatomic molecules with varying degrees of success using techniques relying either on optical cycling and pumping methods, in which light sources address only transitions that result in fluorescence and/or molecular cooling, or buffer gas cooling, in which the molecules thermalize with a cryogenic buffer gas [266, 267, 268, 269, 265, 108]. Additionally, optoelectrical Sisyphus cooling has been employed to translationally cool samples of polyatomic molecules [103, 104], threshold photoionization has been used to ionize molecules into a target rovibrational state [30, 270], photoassociation of molecules from ultracold atomic gasses has been demonstrated [271, 272], and additional cooling prospects have been proposed [273].

Optical pumping for atomic species is relatively simple. In molecules, electronic transitions follow that for atomics. However, no selection rules govern spontaneous electronic decays to vibrational states, i.e. a molecule in the $v' = 0$ vibrational state in an excited electronic state has a non-zero branching ratio to all vibrational states in the ground electronic manifold. Therefore, optical pumping methods either (1) cycle photons between the ground and excited vibrational states within the same electronic manifold supplying the vibrational spacing is sufficiently large, or (2) rely on a special class of molecules that have diagonal Franck-Condon factors (FCFs). The FCF is defined as the vibrational wave function overlap between the initial and final molecular states. If diagonal, the wave function overlap between different vibrational levels and therefore transition probability is likely small. Although diagonal FCFs do not guarantee this condition due to the possibility of the electronic transition dipole moment shape and/or strength contributing significant matrix element amplitude, diagonal FCFs are generally a good indication that a molecule has favorable vibrational branching ratios upon electronic relaxation.

Additionally, buffer gas cooling is applicable for molecules that do not have suitable properties to the methods discussed above, i.e. molecules with a sizeable thermal population in vibrational states other than $v = 0$, non-diagonal FCFs, and/or inconvenient electronic, vibrational, or rotational transition wavelengths or frequencies. In this case, cryogenic buffer gas thermalizes with the molecules, reducing the number of thermally-populated states by a significant factor [266, 267, 268, 269, 265, 108].

6.1.1. Optical Translational Cooling

Translational cooling of molecules has been achieved using narrowband lasers with unwanted vibrational excitation minimized because of an unusual degree of electronic-vibrational decoupling in the selected species (diagonal FCFs) [100, 101, 102]. Optoelectrical Sisyphus cooling has also been employed to translationally cool polyatomic molecules [103, 104]. These neutral molecule experiments select molecules in a single or a few rotational-vibrational state(s) and rely on a large flux of molecules within a thermal distribution to provide the state-specific molecule numbers required.

6.1.2. Vibrational Cooling

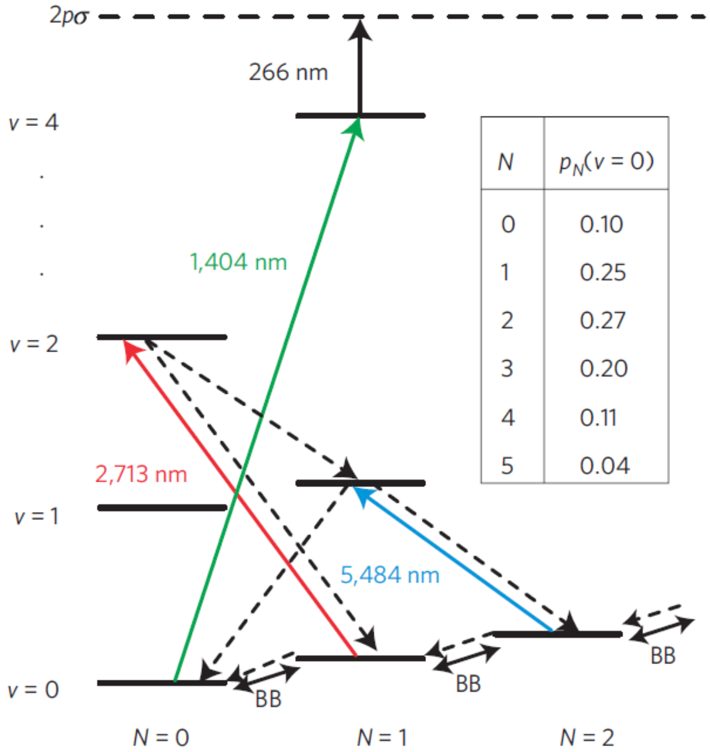
Vibrational cooling of translationally cold molecules has been performed using broadband lasers [105] and incoherent light sources [264] with spectral filtering removing light that would lead to vibrational heating. Spectrally tuned multimode diode lasers have also been used [107] as well as buffer gas cooling [265].

6.1.3. Rotational Cooling

Rotational cooling is particularly critical for many applications because of the relatively small energy of rotational excitations. Rotational cooling has previously only been accomplished using buffer gas cooling [108] and narrowband lasers optically pumping population from a small number of rotational states at one time [110, 111, 109]. A static narrowband laser was used to transfer population from two excited rotational states to the ground state in a beam experiment [18]. In an experiment on molecules formed by photoassocation from an ultracold atomic gas, cooling from the few populated rotational states to a single state was achieved by pumping from each rotational level serially using a swept narrowband laser in conjunction with a broadband laser providing vibrational cooling and repumping [109].

Rotational cooling of trapped molecular ions from room temperature to the ground state was achieved on relatively slow timescales (∼30 s) by waiting for spontaneous vibrational relaxation and blackbody-radiation-driven rotational transitions to redistribute population to levels pumped by narrowband lasers [110, 111] with the technique used in Ref. [111] shown in Fig. 6.1. In this case, applying the far-IR lasers at 2713 and 5484 nm drive rotationally-cooling vibrational transitions, pumping an initial population of 10% in the rotational ground state to 78(4)% in ∼30 s.

More recent efforts at NIST and the Georgia Institute of Technology have focused on using Raman transitions between *rotational* states with a broadband femtosecond laser for deterministic rotational state detection by using a QLS-type readout technique. Once the molecular rotational state is known, rotational cooling could then be performed [115].



N	$p_N(v = 0)$
0	0.10
1	0.25
2	0.27
3	0.20
4	0.11
5	0.04

Figure 6.1. Rotational cooling scheme used for rotational cooling of HD^+. Using the two far-IR lasers at 2713 and 5484 nm, the ground rotational state population was increased by a factor of ∼8 over the room-temperature thermal distribution in ∼30 s by optical pumping and blackbody redistribution between rotational states. The 1404 and 266 nm transitions were used for destructive state detection. Figure taken from Ref. [111].

6.2. Cooling using Electronic Transitions

The technique used in Refs. [110, 111] is applicable to many diatomic species, especially hydrides, as long as laser sources are available, the number of populated rotational states is relatively small, and the blackbody redistribution time is relatively fast. Additionally, diagonal FCFs are not required because the molecules remain in their ground electronic state. However, these vibrational transitions and blackbody redistribution ($\sim 10^1$

s) is slow compared with the timescales of electronic transitions ($\sim 10^{-6}$ s). Therefore, using electronic transitions for rotational cooling should provide orders of magnitude faster rotational cooling as long as a molecule has sufficiently diagonal FCFs to scatter enough photons to optically pump to the rotational ground state.

6.2.1. Spin-orbit-split Molecules

Molecules with a $^2\Pi$ ground state with widely-spaced spin-orbit splittings are possible candidates for rotational cooling. Due to the potential energy surfaces for the $^2\Pi_{1/2}$ and $^2\Pi_{3/2}$ states being nearly-identically shaped, the diagonal FCF condition is typically met. For example, both IF^+ and ICl^+ meet these criteria with diagonal FCFs of >0.998. Further study of these two molecular ion candidates is performed in Sec. 7.3.

6.2.2. Broadband Rotational Optical Cooling of Select Hydrides

Another set of molecules that are promising for rotational cooling is a select few hydrides such at BH^+ and AlH^+. Both are favorable species for broadband rotational optical cooling [**114**] because the $\mathrm{A}^2\Pi \leftarrow \mathrm{X}^2\Sigma^+$ electronic resonance is fairly decoupled from vibrations (diagonal FCFs), supporting around 60 (30) excitation/spontaneous emission cycles for BH^+ (AlH^+) before vibrational excitation occurs [**113**] (Fig. 6.2). It is also important to consider the complexity of the spectral filtering required to ensure that the broadband laser drives only rotational cooling transitions [**106**]. For BH^+, AlH^+, and similar species, only simple spectral filtering is needed because vibrational-electronic decoupling implies matched rotational constants in the two electronic states, and thus

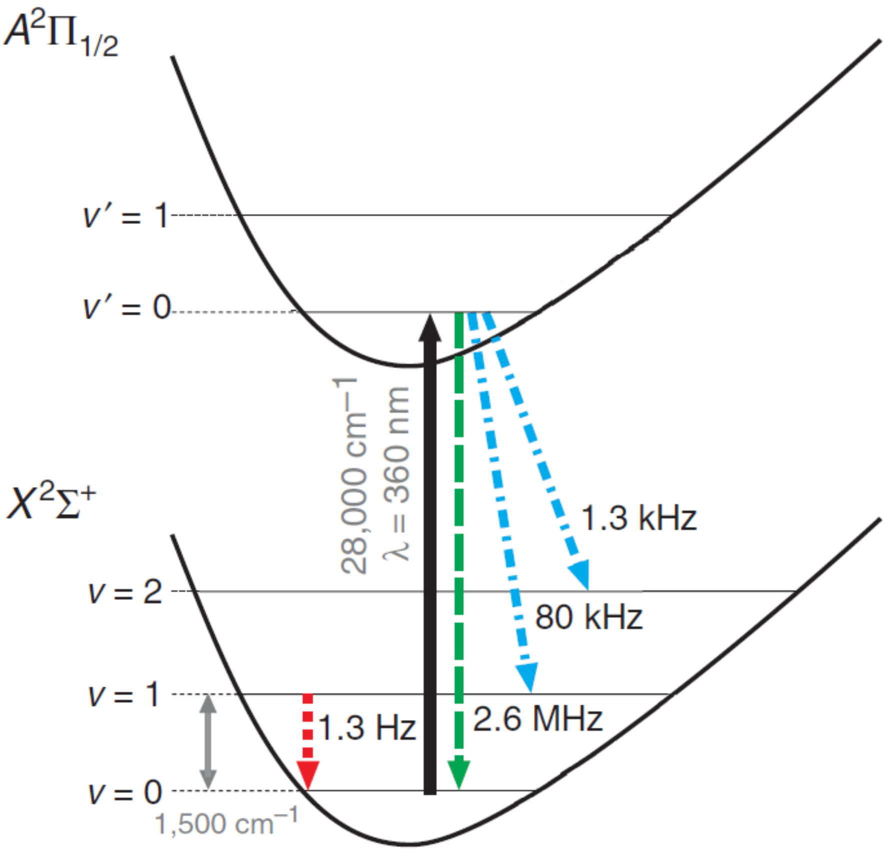


Figure 6.2. The cooling laser drives electronic transitions from $|A^2\Pi_{1/2}, v' = 0\rangle \leftarrow |X^2\Sigma^+, v = 0\rangle$ (solid black upward arrow); electronic spontaneous relaxation most often occurs without vibrational excitation (dashed downward green arrow). Dash-dotted blue downward arrows indicate slower vibrationally-exciting electronic spontaneous relaxation, and dotted red downward arrows indicate much slower vibrational spontaneous relaxation.

well-separated rotationally de-exciting (P-branch), maintaining (Q-branch), and exciting (R-branch) bands (Fig. 6.3).

Because the production of AlH [**273**] and associated REMPI has been previously demonstrated [**274**, **275**, **276**, **277**, **278**], electron-impact ionization of AlH calculated [**279**], the spectrum of AlH and AlH$^+$ is experimentally known [**116**, **117**, **118**, **280**,

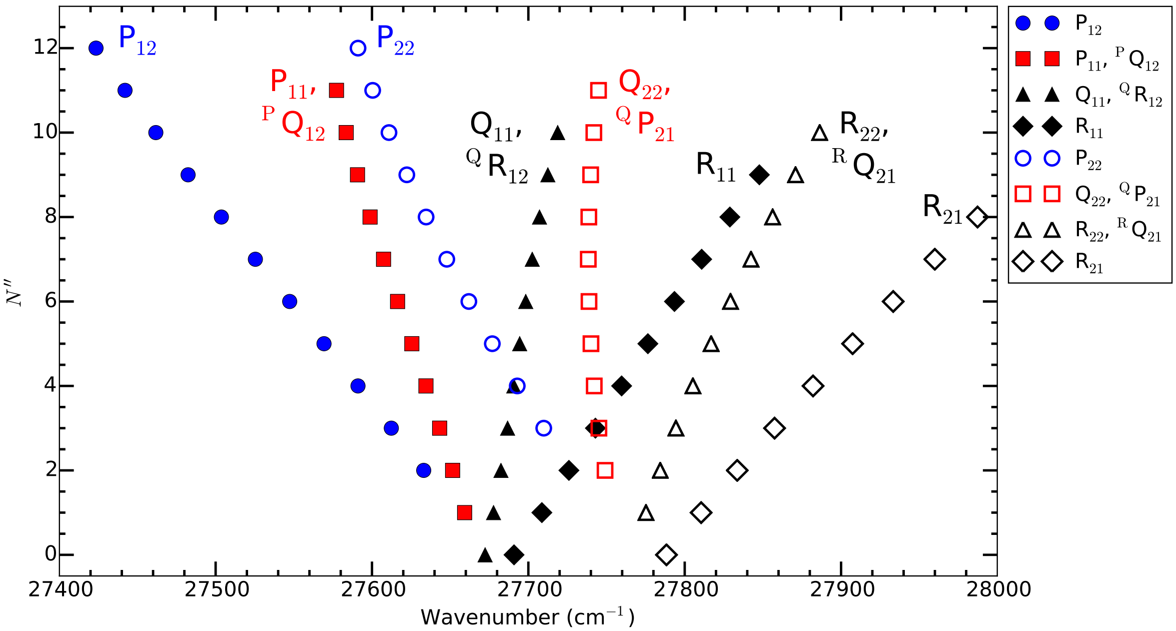


Figure 6.3. Rotational spectrum of AlH^+ for the first ~10 rotational levels. The closed (open) symbols represent the F_1 (F_2) transitions, respectively. Note the separation between the P-, Q-, and R-branches. A broadband laser source with spectral cutoff at 27655 cm^{-1} would be sufficient to drive only the rotationally de-exciting transitions, rotationally cooling the molecule.

119, **273**], theory [**281**, **282**, **120**, **113**] agrees relatively well with experimental results, and the mass of AlH^+ is closer to that of Ba^+ (a future consideration to perform mQLS), we choose this molecule to demonstrate rotational cooling using the electronic $A^2\Pi \leftarrow X^2\Sigma^+$ transition. Rate equation simulations of rotational cooling are discussed in Sec. 7.4, rotational cooling apparatus and light source in Ch. 8, rotational state readout in Ch. 9, rotational cooling results in Ch. 10, and future experiments including the possibility of fluorescence detection in Ch. 11.

CHAPTER 7

Molecular Calculations and Rotational Cooling Simulations

This chapter details calculations to determine the suitability of and rate equation simulations for some molecular candidates with the latter focused on AlH^+ rotational cooling.

7.1. Calculational Tools

To perform the calculations and simulations listed in this section, two molecule-specific programs from the University of Waterloo are used. The programs both take an input file of molecular parameters, potential energy surfaces (PESs), and a transition or permanent dipole moment (TDM or PDM) function and perform useful calculations of molecular levels and/or cross sections. Additionally, JHVN provided the rate equation simulation infrastructure described below [112, 113].

7.1.1. BCONT 2.2

BCONT, developed by Le Roy and Kraemer, calculates bound to continuum results for diatomic molecules such as photodissociation cross sections, emission coefficients, or Fermi's Golden rule predissociation rates by solving the radial Schrödinger equation for bound and quasibound levels as determined by input potential energy surfaces and a transition dipole moment function. Calculational details can be found in the BCONT manual in

Ref. [283], and its use to calculate the AlH^+ photodissociation cross section is discussed in Sec. 9.1.2.

7.1.2. LEVEL 8.0

LEVEL, developed by Le Roy, calculates the bound and quasibound states of an input molecular potential energy surface including vibrational and rotational levels, centrifugal distortion constants, FCFs, and/or matrix elements between the same (different) electronic manifold(s) using the permanent (transition) dipole moment of the state(s) involved. Calculational details can be found in the LEVEL manual in Ref. [284].

Note however that LEVEL assumes a *singlet* to *singlet* electronic transition when calculating the rotational line strength (Hönl-London) factor contribution in a given transition Einstein $\mathcal{A}$ coefficient. LEVEL uses the Hönl-London factors recommended in Ref. [285], though using the factors given in Ref. [286] would be a further improvement for multiplet transitions between Hund's cases *a* and *b*. Correcting these factors must be done by either modifying the code as described in the manual, be performed in post-processing, or simply neglected. This correction has been neglected for this work because the difference in $\mathcal{A}$ coefficient from this factor is small at low rotational quantum numbers compared with inaccuracies of the input potential energy surfaces and/or dipole moment functions.

A workhorse for the calculations below, LEVEL is used to calculate the levels, transition energies, and Einstein $\mathcal{A}$ coefficients of IF^+, ICl^+, and AlH^+. These parameters are then inputs to the rate equation simulations for ICl^+ and AlH^+.

7.1.3. Rate Equation Simulations

A rate equation approach is used to model the population dynamics of the molecular ions of interest. This model allows both investigations of rotational cooling efficiency and timescale as well as suitability for fluorescence detection and/or Doppler cooling [287, 112, 113]. The differential equation

$$\frac{d\mathbf{P}}{dt} = \mathbf{MP} \tag{7.1}$$

is numerically solved where $\mathbf{P}$ is a vector of N energy-ordered elements corresponding to different rovibrational states of the molecule, and $\mathbf{M}$ is an $N \times N$ coupling matrix where elements consist of Einstein $\mathcal{A}$ and $\mathcal{B}$ coefficients. For the calculations performed in this chapter, the rovibrational levels, Einstein $\mathcal{A}$ coefficients, and transition energies are calculated using LEVEL. From the latter two, the Einstein $\mathcal{B}$ coefficients are calculated by

$$\mathcal{B}_{ij} = \frac{\pi^2 c^3}{\hbar \omega_{ij}^3} \mathcal{A}_{ij} \tag{7.2}$$

where c is the speed of light, $\omega_{ij}/2\pi$ is the transition frequency, and ij denote the initial and final states involved. The differential equation governing the population in a specific state is then explicitly

$$\begin{aligned} \frac{dP_i}{dt} = &- \sum_{j=1}^{j=i-1} \mathcal{A}_{ij} P_i - \sum_{j=1}^{j=i-1} \mathcal{B}_{ij} \rho\left(\omega_{ij}\right) P_i - \sum_{j=i+1}^{j=N} \mathcal{B}_{ij} \rho\left(\omega_{ij}\right) P_i \\ &+ \sum_{j=i+1}^{j=N} \mathcal{A}_{ji} P_j + \sum_{j=1}^{j=i-1} \mathcal{B}_{ji} \rho\left(\omega_{ji}\right) P_j + \sum_{j=i+1}^{j=N} \mathcal{B}_{ji} \rho\left(\omega_{ji}\right) P_j \end{aligned} \tag{7.3}$$

where $\rho\left(\omega_{ji}\right)$ is the spectral energy density at a given frequency from applied laser drives and blackbody radiation at a temperature of 300 K.

The GNU Octave (MATLAB) code that performs the numerical integration is in App. D.2.2.

7.2. BaF$^+$ Ground State Preparation by Depletion

An early idea in the group to generate a collection of molecular ions in the rotational ground state, initially investigated by VNR, is by depletion/recycling [**76, 77, 78**]. In this molecular "preparation" technique, a precursor to the molecular ion species of interest is our trapped atomic ion, Ba$^+$. A high-density jet of reactive gas is then pulsed over the cloud of atomic Ba$^+$ forming the molecule of interest [**288**] with one early candidate being BaF$^+$. The molecular ion products, distributed among many thermally-populated rotational states, that are not in the desired rotational state are then selectively destroyed using REMPD. The surviving molecular ions are then in the rovibrational state of interest. After experiments, typically involving REMPD as a state readout technique, the resulting atomic ions can then be "recycled."

LEVEL is used to calculate the rovibrational levels, vibrational overtone transition energies, and Einstein $\mathcal{A}$ coefficients for the deeply-bound molecule. BCONT is then used by VNR to calculate dissociation cross sections. Note however that the molecule is so deeply bound ($\sim$52000 cm$^-$1, <200 nm) that a high vibrational level must serve as the $(1+1')$ REMPD intermediate state to assure a laser source for the $1'$ photon is available.

The LEVEL input file including ground state potential energy surface and permanent dipole moment calculated by MZI is located in App. B.1. A summary of the LEVEL and BCONT results for driving vibrational overtone transitions and subsequent photodissociation is in Table 7.1. In the table, the saturation intensity I_s for a given transition, defined as the laser intensity required for equal steady-state populations in the two coupled quantum states, is given by

$$I_s = \frac{\pi hc}{3\lambda^3}\mathcal{A} \tag{7.4}$$

where h is Planck's constant, c is the speed of light, λ is the transition wavelength, and $\mathcal{A}$ is the transition Einstein $\mathcal{A}$ coefficient. To calculate the light intensity required to drive the transition at a given rate,

$$\frac{dN_2}{dt} = N_1\mathcal{B}_{12}\rho\left(\omega_{12}\right) - N_2\mathcal{B}_{21}\rho\left(\omega_{12}\right) - N_2\mathcal{A}_{21} \tag{7.5}$$

where N_i is the population in the i^{th} level, $\mathcal{A}_{ij}$ and $\mathcal{B}_{ij}$ are the Einstein coefficients of the transition, and $\rho\left(\omega\right)$ is the energy density per unit frequency. The conversion used to light intensity is

$$\Delta\omega\rho\left(\omega\right) = \frac{I}{c} \tag{7.6}$$

where $\Delta\omega$ is the wider of the laser or transition linewidths and I is the laser intensity. If the population driven to N_2 is continuously depleted/dissociated before it can spontaneously decay or undergo stimulated emission,

$$R \equiv \frac{1}{N_1}\frac{dN_2}{dt} = \mathcal{B}_{12}\rho\left(\omega_{12}\right). \tag{7.7}$$

Assuming that the degeneracies between the upper and lower states are equal and using Eqn. 7.2,

$$R = \frac{1}{4hc}\lambda^3\mathcal{A}_{21}I\frac{1}{\Delta\omega}. \tag{7.8}$$

Table 7.1. Table of vibrational overtone transitions from $v = 0$ in BaF$^+$ to evaluate possibility to serve as the intermediate state for $(1 + 1')$ REMPD. ΔE is the vibrational overtone transition wavelength, Diss. E (width) is the peak (width) of the photodissociation cross section with maximum value of σ, τ is the estimated vibrational state lifetime, and I is the vibrational overtone transition laser intensity required to dissociate one molecule per τ. All transitions are for the first Q-branch transition. BCONT calculations are performed by VNR.

v Level	ΔE (nm)	Diss. E (nm)	Diss. Width (nm)	τ (s)	σ $\left(\mathring{A}^2\right)$)	I $\left(\frac{\text{mW}}{\text{mm}^2}\right)$
6	3276	191.2	13.6	0.523	140.7	37.0
7	2817	194.5	14.1	0.386	146.1	47.5
8	2473	198.0	14.6	0.299	149.7	58.8
9	2205	201.6	15.0	0.239	154.7	69.9
10	1991	205.3	13.0	0.197	160.3	80.3
11	1816	209.2	11.0	0.166	161.9	92.6
12	1670	211.9	11.0	0.142	168.1	103.0
13	1546	216.0	11.5	0.124	169.5	114.7
14	1441	218.8	12.0	0.109	179.0	122.0
15	1349	221.7	12.0	0.096	186.4	131.3
16	1268	224.7	12.4	0.086	176.5	152.7
17	1198	229.4	9.6	0.078	181.2	160.6
18	1135	232.6	10.0	0.071	188.4	167.4
19	1078	235.8	11.0	0.065	192.7	176.3
20	1027	239.2	7.0	0.06	193.1	187.9
21	982	242.7	7.2	0.056	187.1	204.8

From the results shown in Table 7.1, promising vibrational overtones exist. The second photon laser source would likely need to be a pulsed source for efficient SHG into the UV, requiring the vibrational overtone driving laser to be pulsed as well. With the availability of laser dyes and OPOs, light sources for both photons are readily available. One consideration however is the frequency sweep rate of the first laser in order to destroy molecules in undesired rotational levels, which is a surmountable technical challenge. Alternatively, CW or quasi-CW lasers, possibly using spectral filtering methods could be employed (Sec. 8.11).

7.3. Prospect of Spin-orbit-split Molecules

Molecules that are promising candidates for rotational cooling are those with a $^2\Pi$ ground state with a widely-spaced spin-orbit splitting. Due to the potential energy surfaces for the $^2\Pi_{1/2}$ and $^2\Pi_{3/2}$ states being nearly-identically shaped, the diagonal FCF condition is *typically* met. For example, both IF^+ and ICl^+ meet the above criteria and neutral molecule production has been demonstrated [**289**, **290**, **291**, **292**, **293**, **294**]. However, as will be shown, diagonal FCFs do not necessarily imply good branching fractions during electronic spontaneous emission due to the contribution of the transition dipole moment. Also note that transitions between the two spin-orbit-split states are only allowed due to the $^2\Pi_{1/2}$ and $^2\Pi_{3/2}$ states mixing with a higher state unequally. As such, the transition dipole moment is expected to be small and transitions between the two states are expected to have a small $\mathcal{A}$ coefficient.

Due to the limited availability of potential energy surfaces for these molecules [**295**, **296**], the preliminary potential energy surfaces and transition dipole moment calculations are provided by MZI, who generously donated some of his time to this project. Additional investigation is needed to verify the accuracy of the calculated results, but regardless of their accuracy, we gained valuable insights while exploring these species.

7.3.1. IF^+

The $X^2\Pi$ potential energy surfaces and transition dipole moment between the two spin-orbit states of IF^+ are shown in Fig. 7.1, which are used in the LEVEL input file in App. B.2. These are calculated by MZI using MOPLRO using the TZ basis set.

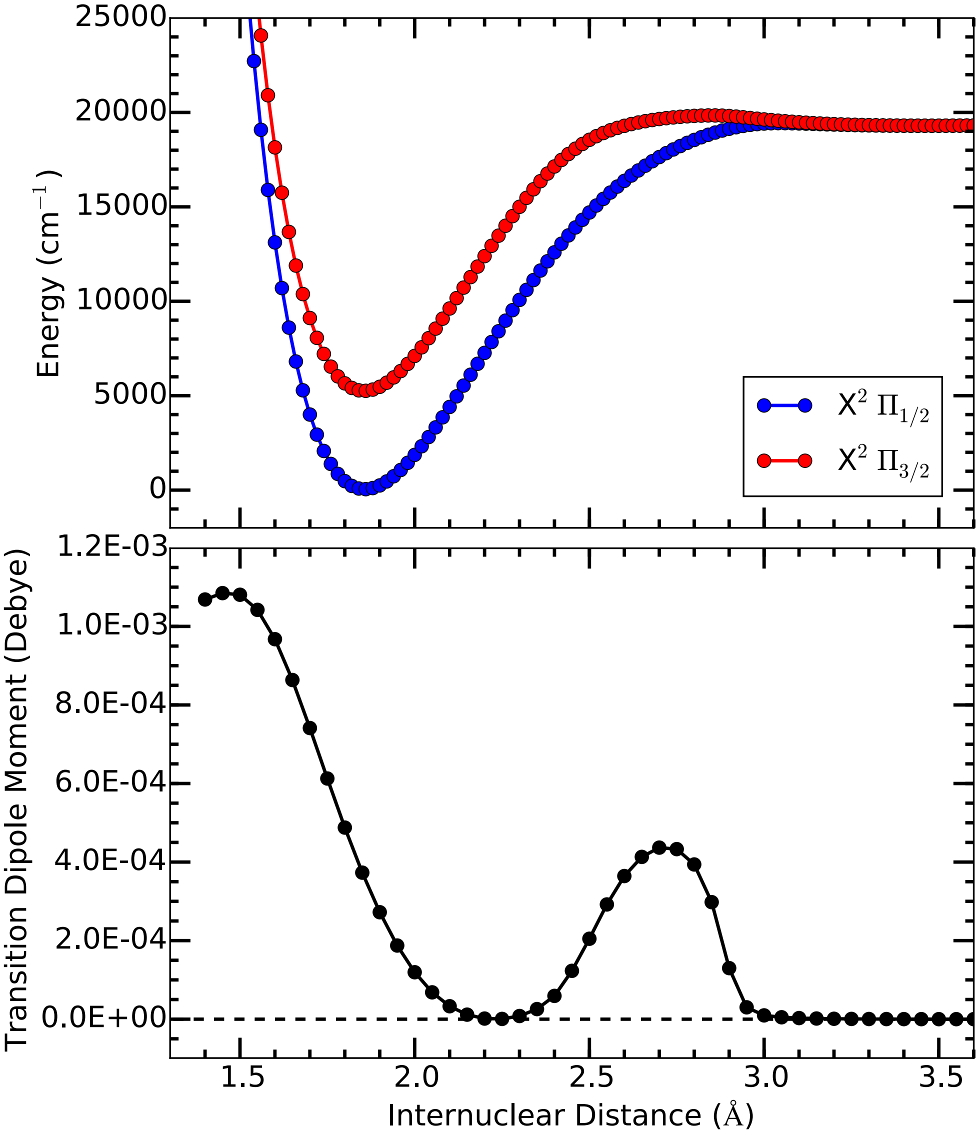


Figure 7.1. IF$^+$ potential energy surfaces of the spin-orbit-split X$^2\Pi$ state and transition dipole moment between the spin-orbit-split X$^2\Pi$ state. Note that transitions between the two states are allowed due to state mixing, hence the small transition dipole moment.

The LEVEL results for transitions between the two spin-orbit-split states is shown in Table 7.2. Note that the FCFs are quite good; ∼18000 photon scatters are expected on the $v' = 0 \leftarrow v'' = 0$ transition before a vibrationally-exciting decay occurs, which is more

Table 7.2. Table of IF$^+$ LEVEL Results using the input file in App. B.2. Note the highly favorable FCFs between the $v' = 0$ and $v'' = 1$ states, but due to the transition dipole moment function shape, the resulting $\mathcal{A}$ coefficients are only slightly favorable. All data is for the lowest Q-branch rotational transition.

$\lvert X^2\Pi_{3/2}, v'\rangle$	$\lvert X^2\Pi_{1/2}, v''\rangle$	Transition E (cm^{-1})	FCF	Matrix Element	$\mathcal{A}$ (arb)
0	0	5206	0.999	5.49×10^{-5}	426.9
0	1	4523	5.69×10^{-4}	-1.37×10^{-5}	17.25
0	2	3848	9.07×10^{-6}	2.17×10^{-6}	0.2674
1	0	5889	5.73×10^{-4}	-1.11×10^{-5}	25.02
1	1	5206	0.998	5.34×10^{-5}	402.6
1	2	4531	1.02×10^{-3}	-1.87×10^{-5}	32.64
2	0	6563	5.05×10^{-6}	1.08×10^{-6}	0.3308
2	1	5880	1.04×10^{-3}	-1.54×10^{-5}	47.95
2	2	5205	0.998	5.20×10^{-5}	380.9
3	0	7229	1.38×10^{-11}	-4.82×10^{-9}	0.0000
3	1	6546	1.69×10^{-5}	1.91×10^{-6}	1.027
3	2	5871	1.39×10^{-3}	-1.84×10^{-5}	68.80

than enough for Doppler cooling and fluorescence detection. However, when comparing the $\mathcal{A}$ coefficients between the $v' = 0 \rightarrow v'' = 0$ and $v' = 0 \rightarrow v'' = 1$ channels, ∼25 photon scattering events are expected, which may be sufficient only for rotational cooling. On closer analysis, this is because the transition dipole moment is comparably large where the $v' = 0$ and $v'' = 1$ vibrational wave functions have their most overlap, which contributes a large amplitude to the matrix element. Therefore, favorable FCFs do not *necessarily* imply favorable vibrational branching fractions from an excited electronic state.

With ∼25 scatters before undesired vibrationally exciting electronic spontaneous emission, it may be possible to perform some degree of rotational cooling of IF$^+$. The $v' = 0 \leftarrow v'' = 0$ transition wavelength is 1921 nm, which is accessible with thulium-doped fiber lasers [**297**] both operating in CW or femtosecond/broadband configurations. Although the vibrational branching fractions of IF^+ are favorable for at least some degree of rotational cooling, $\frac{1}{\mathcal{A}} \approx 4$ hours for the first Q-branch rotational transition, which is far too long to demonstrate rotational cooling within a graduate student lifetime.

7.3.2. ICl^+

The $X^2\Pi$ potential energy surfaces and transition dipole moment between the two spin-orbit states of ICl^+ are shown in Fig. 7.2, which are used in the LEVEL input file in App. B.3. These are calculated by MZI using MOPLRO. Note that using MZI's calculated PESs, LEVEL predicts a 0.5443 eV splitting between the two $X^2\Pi$ states, which differs from previous experimental results [298] by ~6%. However, MZI's PES minima/internuclear spacing seem to be much closer to the experimental results [298] than previous theory [299]. Vibrational spacings of all three results agree to within ~20%.

The LEVEL results for transitions between the two spin-orbit-split states is shown in Table 7.3. Again, the FCFs are quite good; ~850 photon scatters on the $v' = 0 \rightarrow v'' = 0$ transition are expected before a vibrationally-exciting decay occurs. However, when comparing the $\mathcal{A}$ coefficients between the $v' = 0 \rightarrow v'' = 0$ and $v'' = 1$ channels, only ~2 photon scattering events are expected! On closer analysis, this is again due to the shape of the transition dipole moment function contributing a large amplitude to the matrix element. Again, good FCFs do not *necessarily* imply favorable vibrational branching fractions from an excited electronic state.

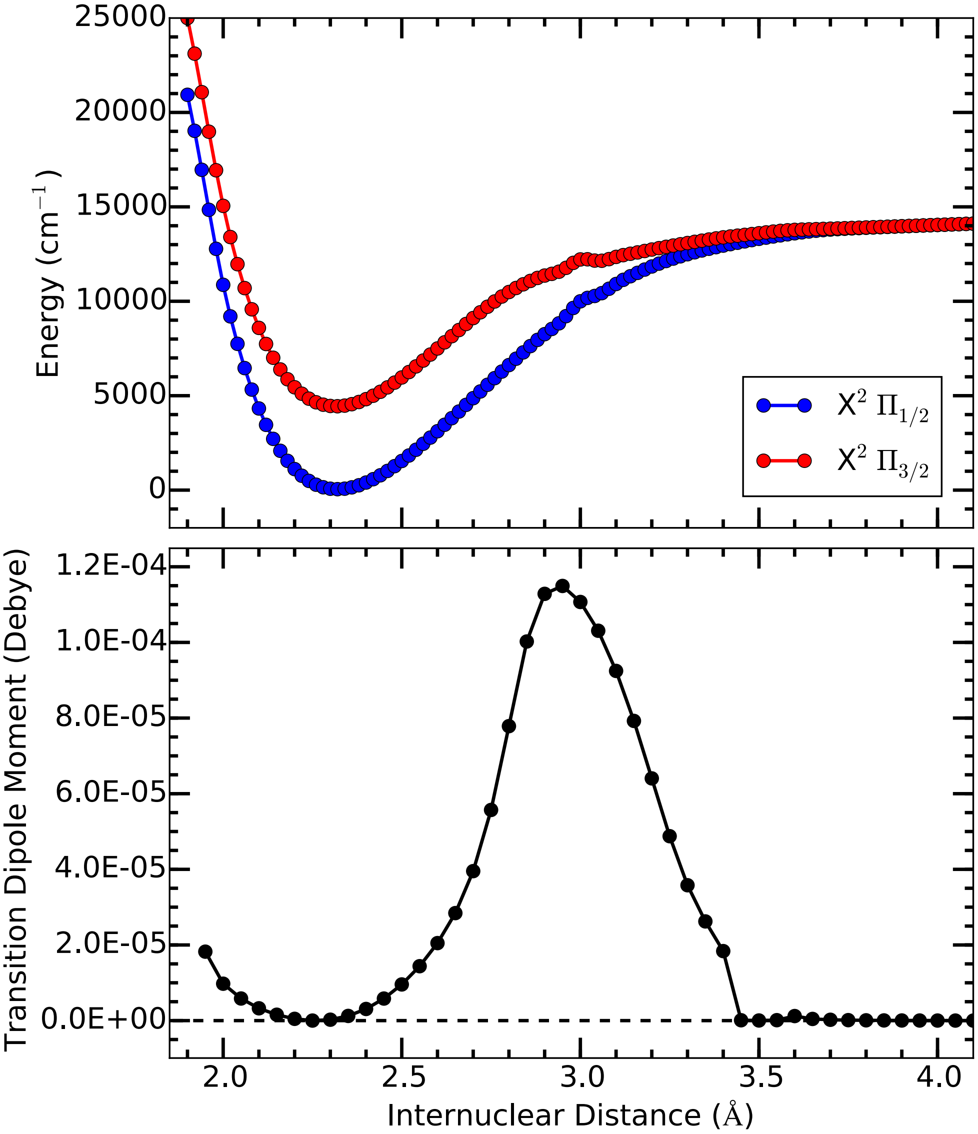


Figure 7.2. ICl$^+$ potential energy surfaces of the spin-orbit-split X$^2\Pi$ state and transition dipole moment between the spin-orbit-split X$^2\Pi$ state. Note that transitions between the two states are allowed due to state mixing, hence the small transition dipole moment.

Table 7.3. Table of ICl^+ LEVEL Results using the input file in App. B.3. Note the highly favorable FCFs between the $v' = 0$ and $v'' = 1$ states, but due to the transition dipole moment function shape, the resulting $\mathcal{A}$ coefficients are highly unfavorable. All data is for the lowest Q-branch rotational transition.

$\left\|X^2\Pi_{3/2}, v'\right\rangle$	$\left\|X^2\Pi_{1/2}, v''\right\rangle$	Transition E (cm^{-1})	FCF	Matrix Element	$\mathcal{A}$ (arb)
0	0	4390	0.999	9.25×10^{-7}	0.0723
0	1	3896	1.17×10^{-3}	9.11×10^{-7}	0.0490
0	2	3407	2.08×10^{-5}	3.76×10^{-7}	0.0056
1	0	4884	1.18×10^{-3}	9.80×10^{-7}	0.1118
1	1	4390	0.997	1.82×10^{-6}	0.2800
1	2	3901	2.08×10^{-3}	1.49×10^{-6}	0.1327
2	0	5372	8.95×10^{-6}	4.61×10^{-7}	0.0330
2	1	4877	2.13×10^{-3}	1.63×10^{-6}	0.3101
2	2	4389	0.995	2.79×10^{-6}	0.6568
3	0	5853	9.91×10^{-9}	-1.82×10^{-8}	0.0001
3	1	5358	2.23×10^{-5}	8.03×10^{-7}	0.0992
3	2	4870	2.81×10^{-3}	2.31×10^{-6}	0.6182

7.3.3. FCF versus $\mathcal{A}$-coefficient

To summarize, the FCF calculates the rovibrational wave function overlap between the initial and final states, i.e. $|\langle v', J'|v'', J''\rangle|^2$. The actual $\mathcal{A}$-coefficient matrix element is $|\langle v', J'|\, M(r)\, |v'', J''\rangle|^2$ where $M(r)$ is the transition dipole moment function. While the FCF can be >0.999 suggesting that off-diagonal vibrational branching is highly suppressed, the shape of the $M(r)$ function can degrade the actual vibrational branching by orders of magnitude. Therefore, diagonal FCFs are a good indicator that a molecule may be a good candidate for diagonal branching ratios, but a full calculation is necessary because of complications from the transition dipole moment function. Additionally, the transition dipole moment is so small that the resulting $\mathcal{A}$-coefficients in IF^+ (ICl^+) are

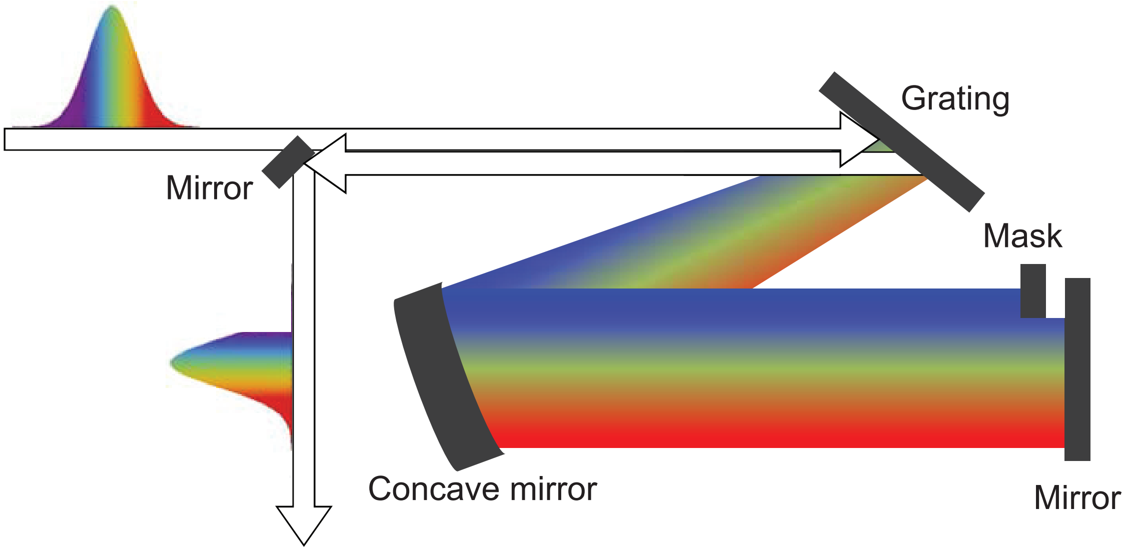


Figure 7.3. A mask at the Fourier plane of the so-called 4-f configuration commonly used in femtosecond pulse-shaping removes an undesirable portion of the broadband laser spectrum [114].

around 10^{-4} (10^{-8}) s^{-1}. However, additional calculations of $M(r)$ may improve these expected $\mathcal{A}$-coefficients.

7.4. AlH^{+}

As noted by JHVN and CYL in Refs. [113, 114], AlH^{+} is a favorable candidate for Doppler cooling and fluorescence detection experiments. Additionally, CYL in Ref. [114] performs rate equation simulations using the code discussed in Sec. 7.1.3 to show that the ion can be rotationally cooled using a spectrally-filtered broadband laser with a simplified diagram of such an apparatus in Fig. 7.3. Rate equation simulation input data generated by JHVN using the potential energy surfaces and dipole moments in Ref. [113] supplementary information is in App. D.2.2.4.

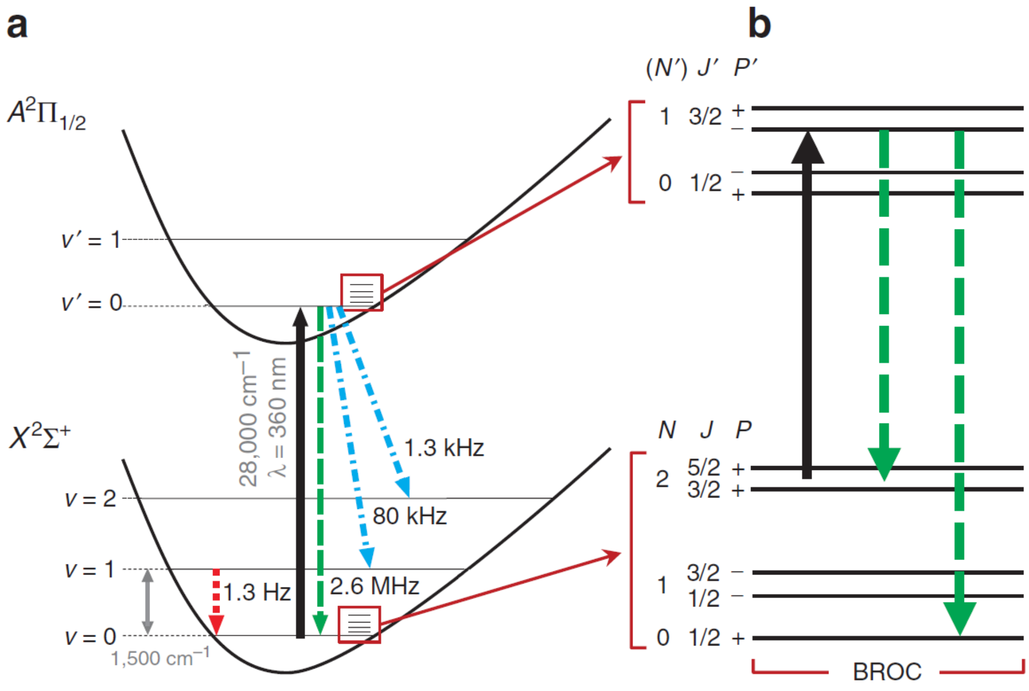


Figure 7.4. AlH$^+$ structure and broadband rotational optical cooling scheme. (**a**) The cooling laser drives electronic transitions from $\left|X^2\Sigma^+, v'' = 0\right\rangle$ to $\left|A^2\Pi_{1/2}, v' = 0\right\rangle$ (solid black upward arrow); electronic spontaneous relaxation most often occurs without vibrational excitation (dashed downward green arrow). Dash-dotted blue downward arrows indicate slower vibrationally-exciting electronic spontaneous relaxation, and dotted red downward arrows indicate much slower vibrational spontaneous relaxation. (**b**) Broadband rotational optical cooling light achieves rotational cooling by exciting from the rotational state N'' to $(N') = N'' - 1$; although rotational angular momentum is not a good quantum number in $A^2\Pi$, (N') serves as a convenient label. Parity and angular momentum selection rules allow for spontaneous relaxation only back to N'' or to $N'' - 2$. Only excitation from the lowest state cooled by broadband rotational optical cooling is drawn in the figure.

7.4.1. Broadband Rotational Optical Cooling

In Ref. [114], CYL performed rate equation simulations of AlH$^+$ rotational cooling from an initial 300 K thermal distribution by selectively driving only the $A^2\Pi_{1/2} \leftarrow X^2\Sigma^+$ P-branch transitions. The general scheme is outlined in Fig. 7.4.

The angular momentum selection rules for the $A^2\Pi_{1/2} \rightarrow X^2\Sigma^+$ transition are $\Delta J = 0$, ± 1, where $\mathbf{J} = \mathbf{N} + \mathbf{S} + \mathbf{L}$ is the total angular momentum including molecular rotation $\mathbf{N}$, electron spin $\mathbf{S}$, and electronic orbital angular momentum $\mathbf{L}$. Each electronic excitation and spontaneous emission cycle conserves molecular parity, so in the simplest broadband rotational optical cooling spectral filtering configuration, population is cooled to the lowest rotational-vibrational state of each parity, $\left|X^2\Sigma^+, v'' = 0, N'' = 0\right\rangle$ and $\left|X^2\Sigma^+, v'' = 0, N'' = 1\right\rangle$ (Fig. 7.4). Due to this parity barrier, CYL simulates driving all P-branch $A^2\Pi_{1/2} \leftarrow X^2\Sigma^+$ transitions simultaneously using a spectrally filtered broadband laser [**114**] excluding P(1) because cycling on the P(1) transition will cause undesired vibrational excitation of the odd parity states to $\left|X^2\Sigma^+, v'' = 1, N'' = 1\right\rangle$. Due to the separation of the P-, Q-, and R-branches (Fig. 6.3), configuring such a light source is relatively trivial. Simulation results are shown in Fig. 7.5.

7.4.2. Vibrationally-assisted Broadband Rotational Optical Cooling

In order to cool the entire population into the positive-parity $\left|X^2\Sigma^+, v'' = 0, N'' = 0\right\rangle$ rotational ground state, a modified vibrationally-assisted broadband rotational optical cooling scheme is used. Here, parity conservation of broadband rotational optical cooling is destroyed by forcing a vibrationally-exciting electronic decay to occur only for the negative parity by cycling on the P(1) transition that is avoided for broadband rotational optical cooling (Sec. 7.4.1) as shown in Fig. 7.6. Because of the parity barrier, the P(1) transition has no alternate channel for spontaneous relaxation within the $v'' = 0$ manifold, resulting in electronic cycling, until spontaneous electronic relaxation to $v'' = 1$ occurs. Subsequent spontaneous vibrational relaxation back to $\left|\mathrm{X}^2\Sigma^+, v'' = 0\right\rangle$ flips the parity. Population immediately afterwards is entirely in the positive-parity $\left|\mathrm{X}^2\Sigma^+, v'' = 0, N'' = 0\right\rangle$ or $\left|\mathrm{X}^2\Sigma^+, v'' = 0, N'' = 2\right\rangle$ levels, with population in the latter quickly pumped to the ground state by the broadband rotational optical cooling light. Note that only a single spectral filtering configuration is needed for vibrationally-assisted broadband rotational optical cooling from room temperature to the ground state, since the previously-discussed broadband rotational optical cooling process is included within vibrationally-assisted broadband rotational optical cooling.

Because this process relies on the spontaneous $\left|\mathrm{X}^2\Sigma^+, v' = 1, N' = 1\right\rangle \rightarrow \left|\mathrm{X}^2\Sigma^+, v'' = 0\right\rangle$ relaxation, this cooling contains two distinct timescales: (1) the broadband rotational optical cooling timescale and subsequent fast optical pumping to $\left|\mathrm{X}^2\Sigma^+, v'' = 1, N'' = 1\right\rangle$, and (2) the much slower vibrational relaxation timescale of ~1 vibrational decay time constant of 127 ms [**113**]. A summary of these two cooling timescales is shown in Fig. 7.7. The overall cooling timescale is confirmed by rate equation simulations.

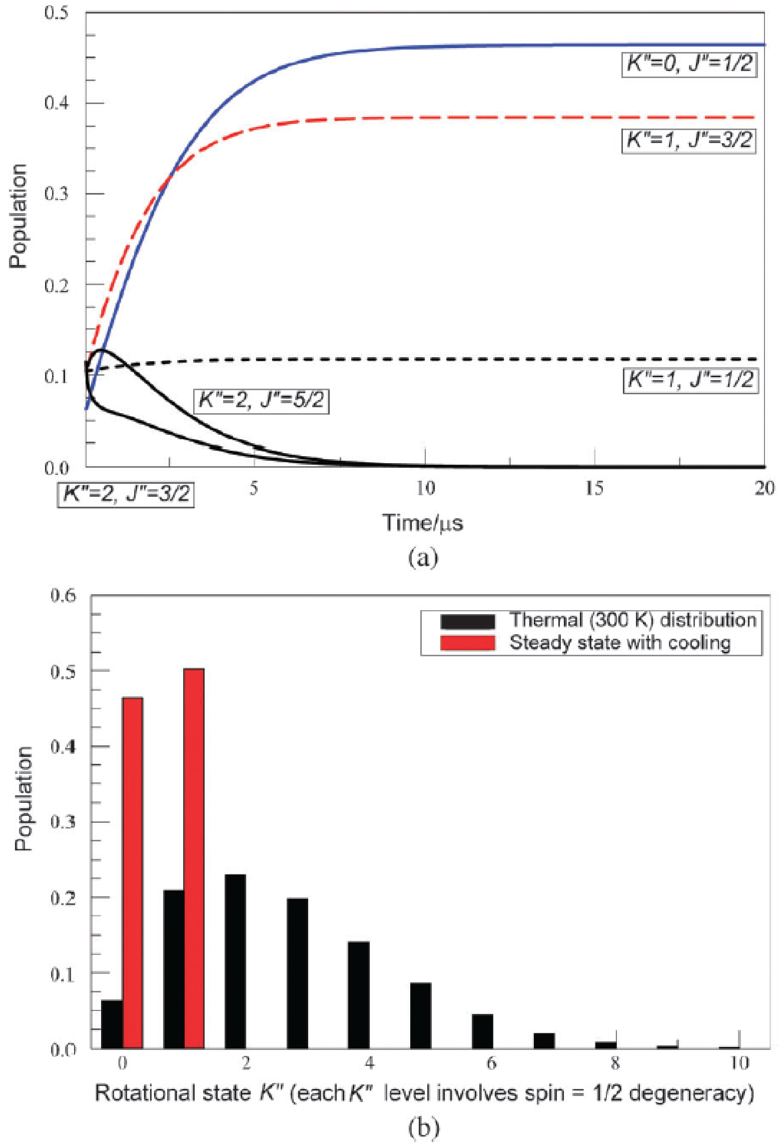
7.4.2.1. AlH$^+$ $\left|\mathbf{X}^2\Sigma^+, v' = 1, N' = 1\right\rangle$ Lifetime. From the original work by JHVN in Ref. [**113**] and CYL in Ref. [**114**], EGH calculated new potential energy surfaces and dipole moments (Fig. 7.8). Because the timescale of the vibrationally-assisted broadband rotational optical cooling is limited by the $\left|\mathrm{X}^2\Sigma^+, v' = 1, N' = 1\right\rangle \rightarrow \left|\mathrm{X}^2\Sigma^+, v = 0''\right\rangle$ relaxation, the new curves are used to update the previous result of 123 ms by Nguyen in Ref. [**113**]. The LEVEL input file using the updated data is in App. B.4.

The resulting LEVEL output in listed in Table 7.4 with the experimental result from Sec. 10.4. JHVN originally estimated a 123 ms $\left|\mathrm{X}^2\Sigma^+, v' = 1, N' = 1\right\rangle \rightarrow \left|\mathrm{X}^2\Sigma^+, v'' = 0\right\rangle$

Table 7.4. LEVEL output data used to calculate the $\left|X^2\Sigma^+, v' = 1, N' = 1\right\rangle$ $\to \left|X^2\Sigma^+, v'' = 0\right\rangle$ transition lifetime, setting the timescale for vibrationally-assisted broadband optical cooling, which is experimentally determined in Sec. 10.4.

Data Set	R(0) $\mathcal{A}$ (s^{-1})	P(2) $\mathcal{A}$(s^{-1})	$\mathcal{A}$ Sum (s^{-1})	Lifetime (ms)
JHVN [113]				123
Previous	3.06076	5.51818	8.579894	117
Updated	3.55061	6.48437	10.03498	100
Experiment				140(20)

transition lifetime τ [113]. Updated calculations both yield a slightly shorter timescale, but still ∼100 ms. The experimental $\tau = 140(20)$ ms and all calculated values are within 2σ agreement.



(a)

(b)

Figure 7.5. Simulation of AlH$^+$ rotational cooling. (**a**) Simulation results of broadband rotational optical cooling for the time evolution of the lowest five spin-rotational levels. The population distribution reaches within 99% of equilibrium in 8 μs assuming all transitions are driven at saturation. (**b**) Population distribution among rotational states before and after broadband rotational optical cooling. Figure taken from Ref. [114]. Here, the rotational quantum number is K rather than N, which is used in the rest of the text.

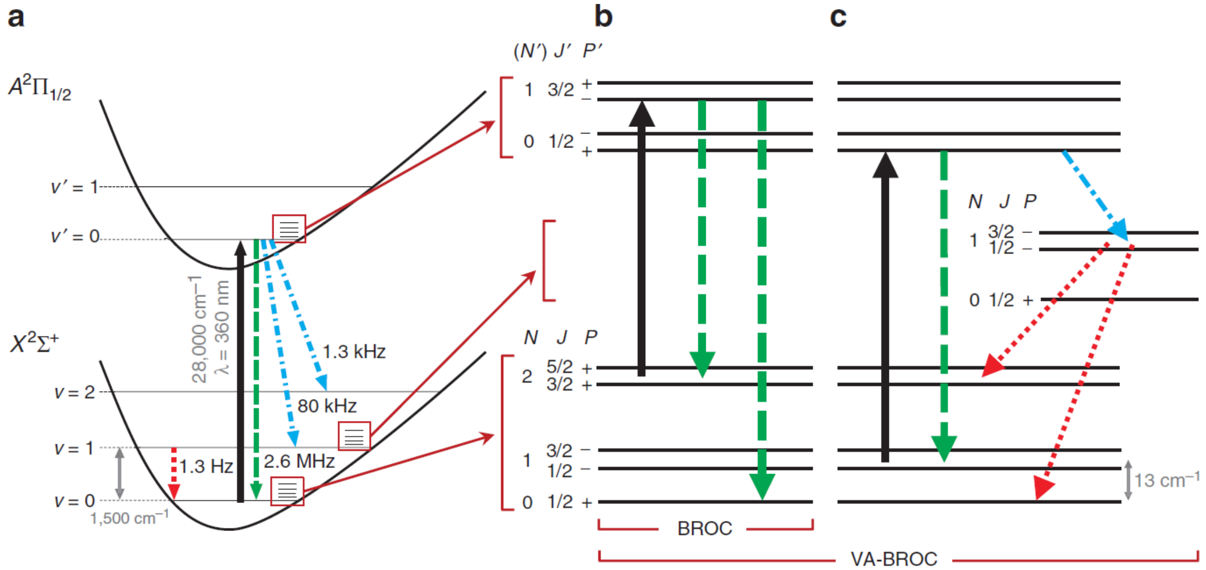


Figure 7.6. (**c**) The vibrationally-assisted broadband rotational optical cooling scheme additionally cools parity by including a spectral component to drive P(1), the first P-branch transition, driving electronic cycling until vibrational excitation and subsequent vibrational relaxation results in a parity flip.

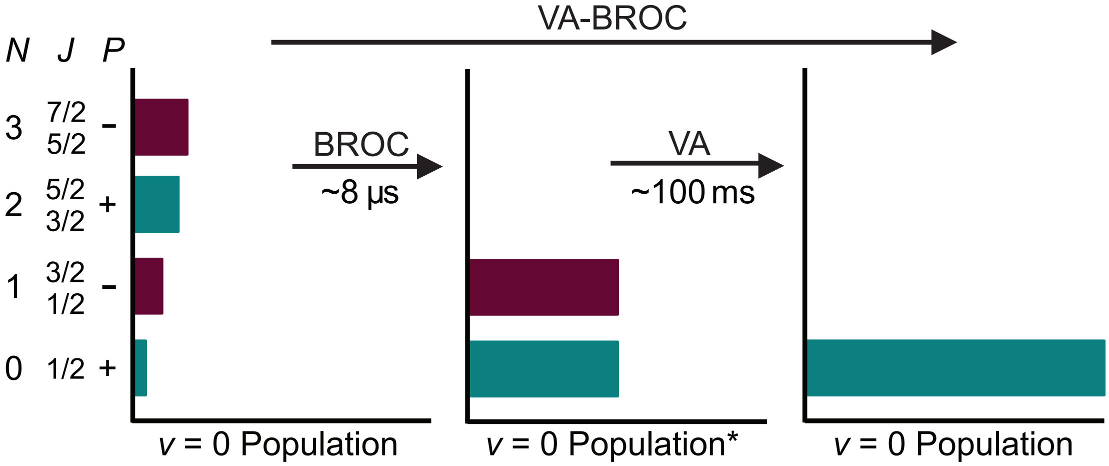


Figure 7.7. Vibrationally-assisted broadband rotational optical cooling contains two timescales: fast, parity-preserving electronic cycling and slower, vibrationally-assisted parity cooling. Broadband rotational optical cooling excites a short-timescale $v > 0$ population to an equilibrium value of 14% (denoted by the *), which then returns to ∼0% on the vibrationally-assisted broadband rotational optical cooling timescale.

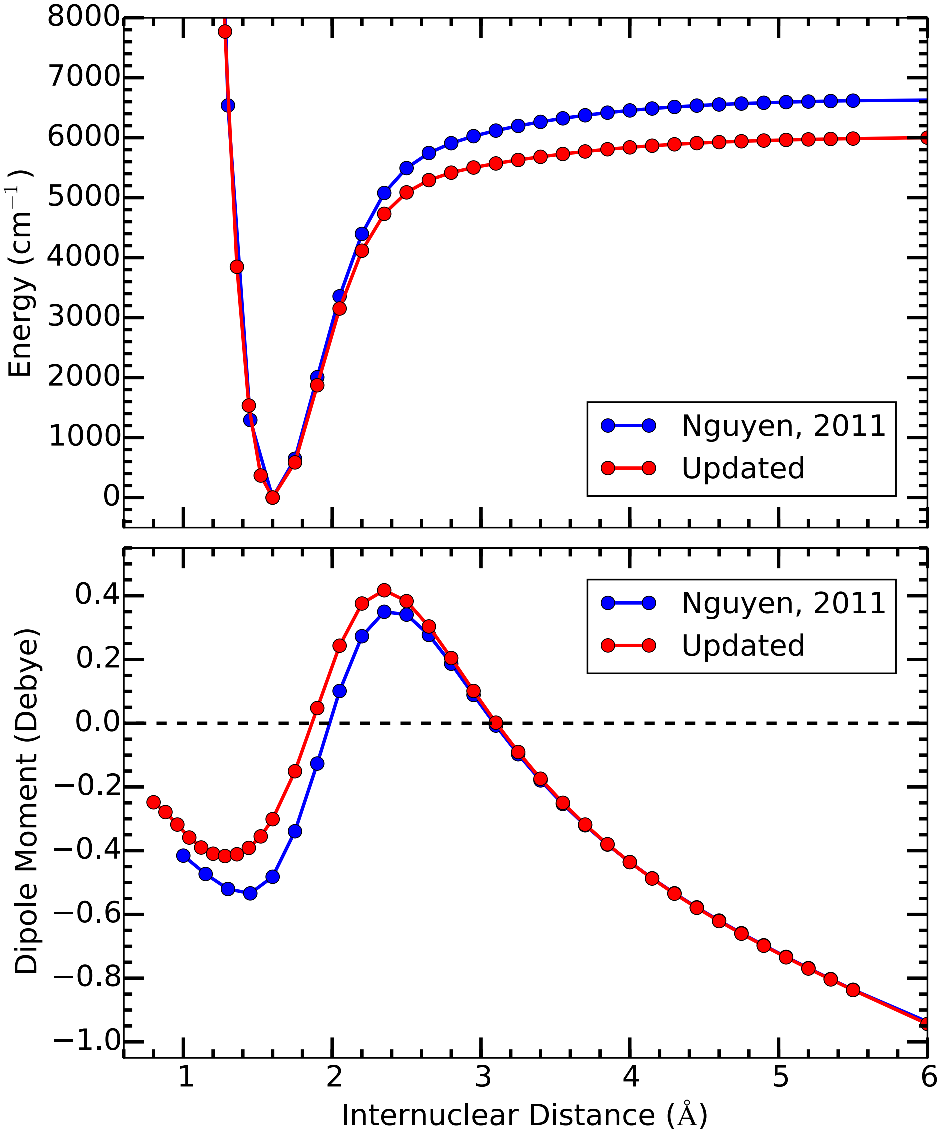


Figure 7.8. Comparison between the input data used by JHVN in Ref. [113] and updated data from EGH for the AlH$^+$ ground state potential energy surface and permanent dipole moment used to calculate the $\left|X^2\Sigma^+, v' = 1, N' = 1\right\rangle \to \left|X^2\Sigma^+, v'' = 0\right\rangle$ $\mathcal{A}$ coefficient.

CHAPTER 8

AlH^+ Rotational Cooling Apparatus, Optical Systems, and Light Source

This chapter describes the apparatus and associated details used to perform the broadband rotational optical cooling of AlH^+ to the rotational ground state. The ion trap system with integrated time-of-flight mass spectrometry apparatus, the imaging system, laser optical systems, pulsed laser systems, details of loading the ion trap, and spectrally-filtered pulsed laser optical arrangement are discussed.

8.1. The Ion Trap

The AlH^+ rotational cooling experiment is performed in the apparatus previously described in Refs. [**300**, **263**, **301**] but not in sufficient detail. The room-temperature linear RF trap, constructed by CYL, has parameters $r_0 = 3$ mm, $z_0 = 15$ mm, and $\Omega_{\mathrm{RF}} = 2\pi \times 2.35$ MHz with the RF applied to two opposing rod electrodes and the other two held at ground. A schematic of the ion trap and integrated time-of-flight mass spectrometer apparatus is shown in Fig. 8.1 with all drawings located in App. F.2.7.

A schematic of the trap RF drive system is shown in Fig. 8.2. The RF voltage applied to the trap rods is measured by mixing the directional coupler's CPL IN output with the function generator sync signal. The mixer IF port is low-pass-filtered such that the DC output voltage is related to the RF voltage applied to the trap rods. A calibration proportionality constant of -3353.68 $\mathrm{V_{PP\ RF}/V_{DC}}$, taken by measuring the applied

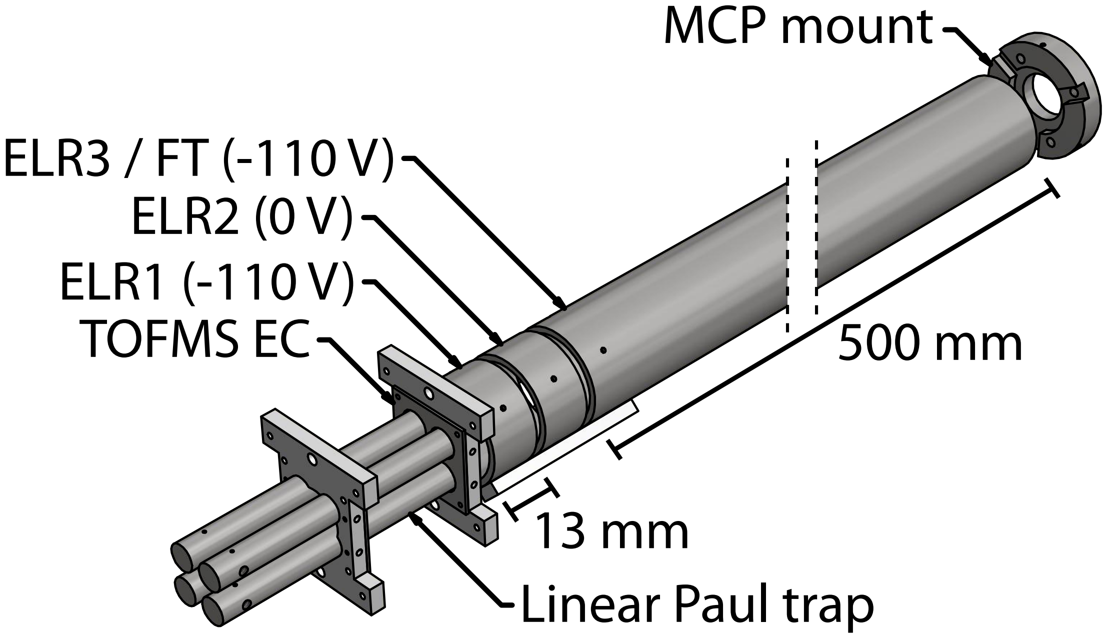


Figure 8.1. Schematic of the ion trap with integrated time-of-flight mass spectrometer apparatus. TOFMS EC labels the trap endcap that is switched at the start of time-of-flight mass spectrometry. Adjacent to TOFMS EC is the einzel lens, consisting of two rings (ELR1 and ELR2) with the third doubling as the flight tube (ELR3 / FT).

RF voltage to the vacuum electrical feedthru using high-impedance oscilloscope probes, provides the applied RF voltage. The RF frequency is tuned to the toroidal resonator resonance by minimizing the CPL OUT RF voltage. This frequency did not require regular optimization because of the low Q-factor of the toroidal resonator system providing

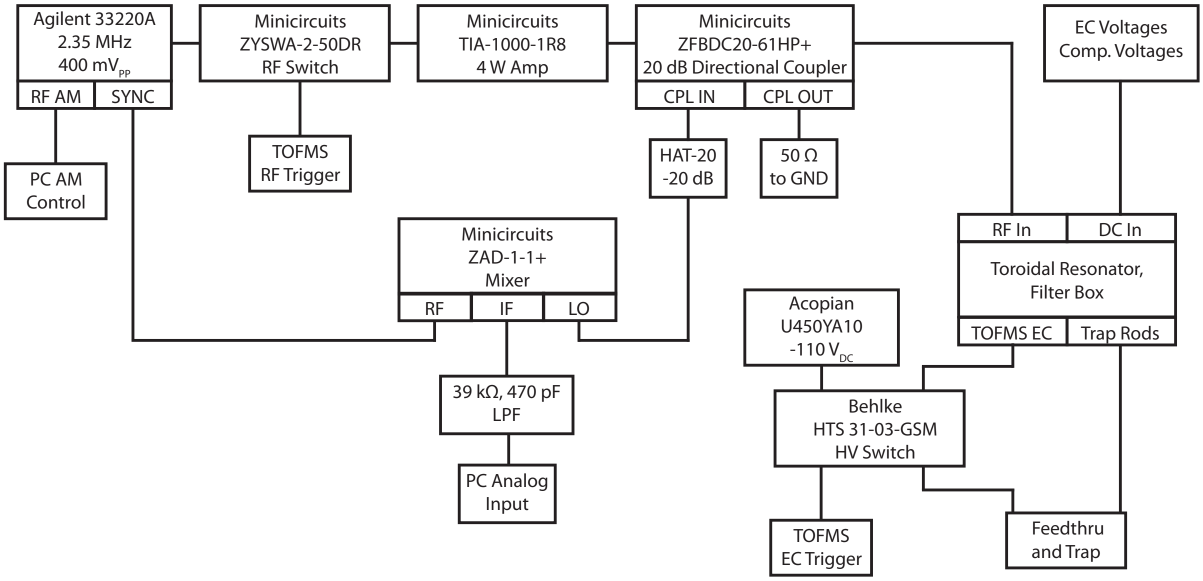


Figure 8.2. AlH$^+$ ion trap RF drive and time-of-flight switching block diagram. For active RF voltage control, the PC AM control connects to a PI servo loop (not shown) and supplies the servo loop set point. The low-pass-filtered DC voltage from the mixer is then the servo loop input signal. The RF switch allows switching off the RF drive during the time-of-flight sequence. The Behlke HV switch switches the time-of-flight endcap from 850 to -110 V in ∼10 ns and is placed after the filter box to avoid undesired capacitive loads, which would slow the switching.

a large bandwidth. The RF voltage is controlled using the RF AM input on the function generator with the generator's maximum power set to 400 mV$_{\mathrm{PP}}$.

During radial fluorescence-shaking-measurements [74] attempted and described in Ref. [301], the applied RF voltage is actively stabilized with a PI servo circuit. Additionally during these measurements, the toroid would heat significantly, reducing the resonator's Q-value and reducing the maximum RF applied to the trap rods. To reduce this heating effect, the toroid is submerged in white mineral oil (CAS 8042-47-5) using a Teflon holder in a separate enclosure (Fig. 8.3) to improve the passive cooling of the

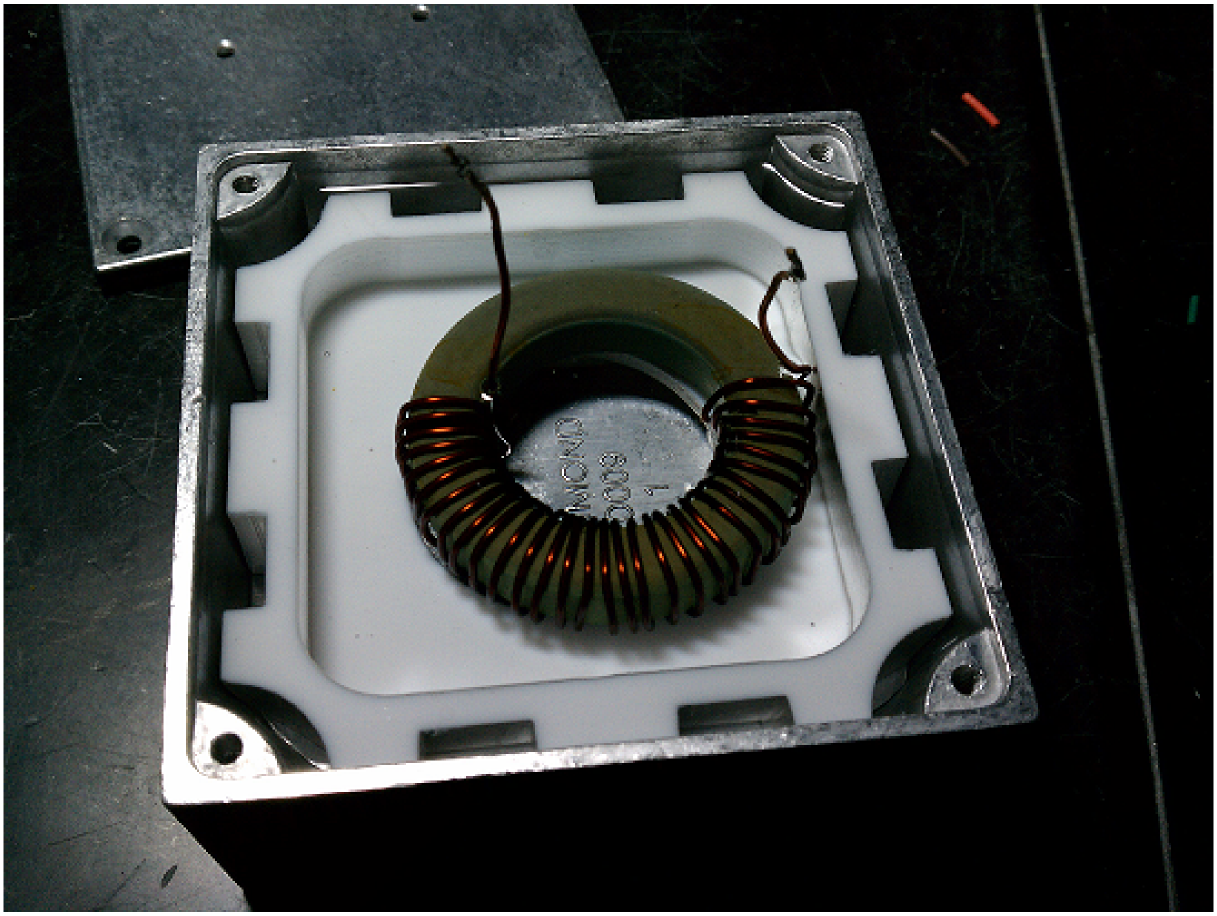


Figure 8.3. Picture of the AlH$^+$ toroid enclosure. A Teflon holder suspends the toroid in the middle of the box, and the box is filled with mineral oil to improve the thermal contact between the toroid and metal enclosure. This helps keep the toroid cool and the resonator Q-factor both high and relatively stable.

toroid. For reference, other experiments use forced-air cooling with a fan placed over the toroid.

The trap endcaps are plates with holes in the center, similar to a trap design shown in Refs. [**302, 87**], typically held at 850 V. In the early days of the apparatus, voltage was supplied using an analog amplifier circuit for each endcap (App. A.3.3) with a factor of 1000 gain using a computer-supplied control voltage. A Kepco BHK 2000-10MG power supply provides the HV for each circuit. Later apparatus iterations use a pair of the Kepco HV supplies, one for each endcap, under computer control via GPIB, to allow for the individual control of each endcap voltage for the work discussed in Sec. 8.10.1.

Compensation voltages and endcap DC voltages pass through a filter circuit (App. A.2.1) in order to filter RF and time-of-flight mass spectrometer switching voltage pickup in addition to combining the DC compensation voltages with the RF voltage before entering the vacuum chamber via a single electrical vacuum feedthru with BNC pigtails. The circuit is housed in an all-metal box that also contains the toroidal resonator enclosure with direct solder connections, and is firmly bolted to the optical table for a good ground connection. In the early days of the apparatus, a circuit supplies the compensation voltages (App. A.2.2), requiring either manual voltage control or two analog inputs for horizontal and vertical adjustment (one output channel adds the two voltages applied to opposing rods for the third trap rod). Due to technical challenges in the design and implementation, the circuit is replaced with analog output channels controlled by LabVIEW where the program calculates the combined voltage.

8.1.1. Micromotion Compensation

Any stray DC electric field due to ion trap imperfections and patch potentials on the rod surfaces displace the ion off the RF null. The RF electric field at this new position then causes the ion to oscillate at the driving RF frequency of the ion trap. Because this excess micromotion, distinguished from the secular frequency motion, is driven by the applied RF field, it cannot be significantly reduced using cooling methods [**303**]. Excess micromotion can significantly alter the atomic transition excitation spectrum (first-order

Doppler shift) and contribute significant second-order Doppler and AC Stark shifts (relevant for quadrupole transitions). Additionally, excess micromotion is a significant heating mechanism for multi-ion crystals through parametric coupling of the excess micromotion and the secular motion. In extreme cases without Doppler cooling, this will result in ions lost from the trap [303]. Multiple methods have been developed to quantify the amount of excess micromotion [303, 304, 305]. By minimizing measured micromotion as a function of DC compensation voltages, the micromotion amplitude can be minimized and ion placed at the RF null.

8.1.1.1. Ion Position. As the trap RF voltage is raised and lowered, the pseudopotential depth scales linearly with the peak-to-peak drive voltage V_{PP}. At high V_{PP}, the ion is tightly confined by the radial pseudopotential. As V_{PP} is lowered, any patch potentials will change the position of the ion, which can then be corrected with the applied DC compensation voltages. Iterating this process minimizes the effects of micromotion. While this technique is easy to implement and is generally sufficient for large-scale ion traps, sensitivity in the direction of the camera-viewing axis is limited. Therefore, more sensitive measures are often required for singly-trapped ions in tight traps [303]. Micromotion compensation is performed using this technique using the RF AM and compensation voltages control on the apparatus LabVIEW interface. Typical compensation voltages are $\lesssim |\pm 0.3|$ V.

8.2. The Time-of-flight Mass Spectrometer

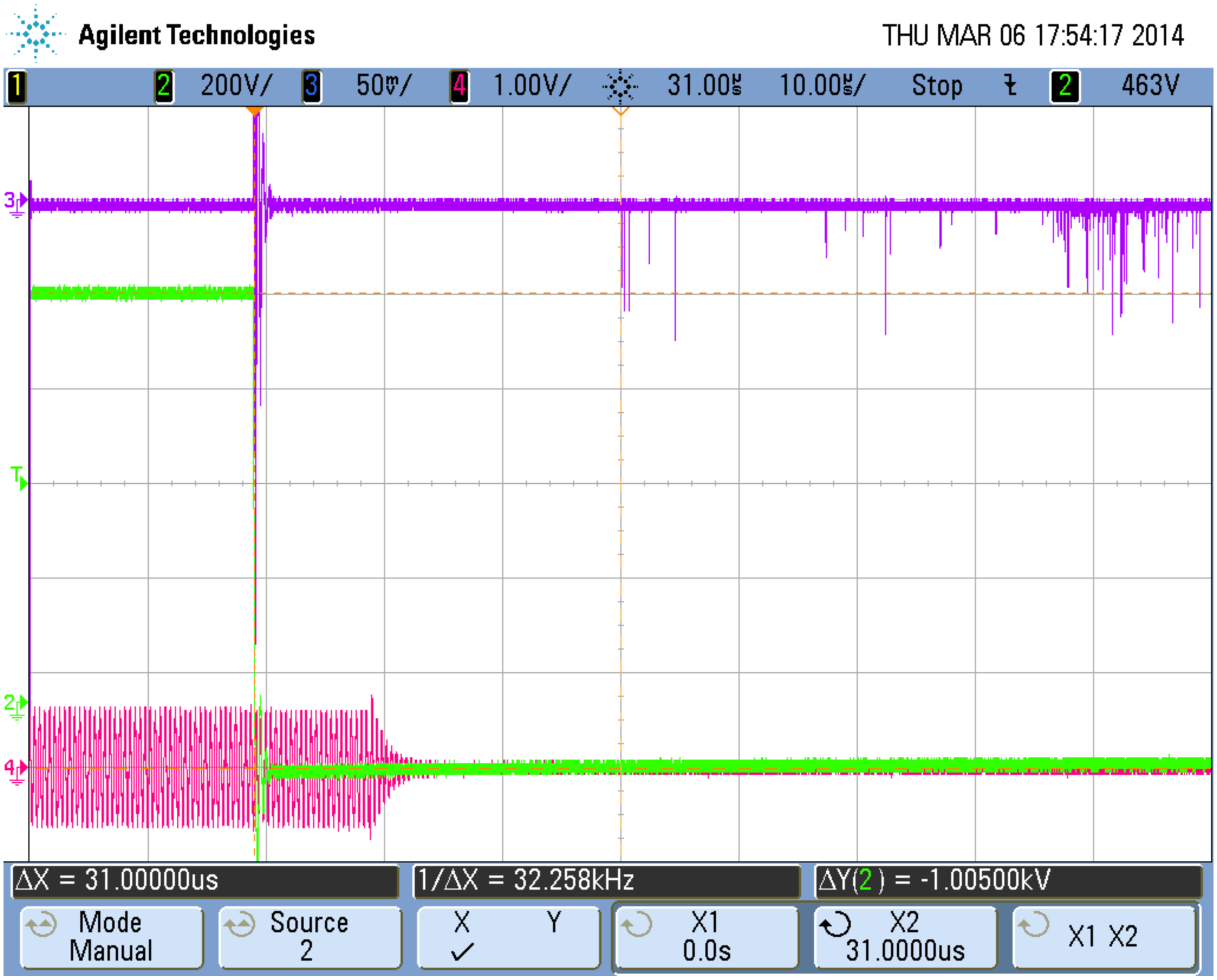
A time-of-flight mass spectrometer is integrated to one side of the ion trap with the other pointing to an electron-impact ionizer (Sec. 8.10.1). A schematic of the ion trap and time-of-flight mass spectrometer apparatus is shown in Fig. 8.1. The system is designed by CYL with details of the apparatus in Ref. [301] including design, construction, and simulations. A 3-element einzel lens utilizing a 500 mm flight tube as the last electrode spatially and temporally focuses ejected trapped ions to a MCP mounted at the end of the flight tube.

A block diagram of the electronics used to initiate a time-of-flight run is shown in Fig. 8.2. When the time-of-flight logic control switches high, the endcap neighboring the integrated time-of-flight mass spectrometer is switched from 850 V to the flight tube voltage (-110 V) using a Behlke HTS 31-03-GSM push-pull high-speed, HV switch, which is wired in between the filter circuit and vacuum system feedthru so that the HV power supply remains filtered and the capacitive load being switched is minimized. The first ion lens ring and flight tube (third ion lens element) are held at -110 V by an Acopian U450YA10 supply. The central ion lens ring is typically held at ∼0 V, but an offset voltage can be applied to assist in temporally focusing of the ions around the mass region of interest (27-28 u for Al^+ and AlH^+).

At the end of the flight tube is a Hamamatsu F4655 MCP. To apply a -0.5 kV anode bias voltage on the MCP while holding potential difference across the device at -2.5 kV, a HV voltage divider circuit is used (App. A.2.4) and mounted inside of an all-metal enclosure using MHV bulkheads. An Acopian NO3.5HA8.5T supply provides the HV with RG-8/U cable used for the -3 and -2.5 kV. This bias provides a potential difference between the MCP rear surface and the anode, which helps collect electrons on the anode, increasing total charge per ion count, and decreasing the single-ion charge distribution width (Sec. 10.2). The resistor leads are soldered such that the divider is suspended inside of the box. To further insulate the system from arcing and to help dissipate heat (1 mA current draw over 3 kV = 3 W), the voltage divider resistors and leads are covered in and enclosure is half filled with thermal potting epoxy (Resin Technology Group KONA 870 FTLV-DP, 1.44 W/Km thermal conductivity, 20.5 kV/mm dielectric strength). The analog-mode MCP signal is terminated with 50 Ω at an Agilent DSO7014A digital oscilloscope with a sample rate of 2 GSa/s, which is triggered with the time-of-flight endcap voltage switching from 850 to -110 V.

The flight tube and MCP power supplies are independently power-switched via solid-state relays (Crydom EZ240D18). A circuit with logic enable, override, and remote TTL input features controls each. This is to permit computer control of the MCP supply in particular; during REMPD of AlH^+ (Ch. 9), the 266 and 360 nm photons are sufficiently energetic to initiate an electron avalanche on the MCP. Because of the high laser intensity, the MCP supply voltage is switched off during these pulses to prevent saturating and/or damaging the device.

To extract the ions from the trap, the time-of-flight-mass-spectrometer-side trap endcap is switched from 850 V to -110 V in 10 ns (Fig. 8.4) using the HV switch. In order to guide ion extraction, the RF is left on for the beginning of the time-of-flight sequence. The timing resolution tends to be better if the RF is shut off early in the time-of-flight sequence, possibly because of deleterious effects of fringe fields as the ions approach the end of the ion trap rods. A 10 μs delay is used before the RF is switched off using a Mini-circuits ZYSWA-2-50DR switch between the function generator and amplifier with an RF ring-down time of 2 μs (Fig. 8.4). Additionally, the 493 nm Ba^+ Doppler cooling laser



ΔX = 31.00000us

1/ΔX = 32.258kHz

ΔY(2) = -1.00500kV

Figure 8.4. Oscilloscope trace of a time-of-flight sequence. Green is the time-of-flight-mass-spectrometer-side trap endcap voltage, which the scope is triggered on. Magenta is the trap RF pickup on an ion trap electrode, which is turned off 10 μs after the endcap voltage switches with a 2 μs ring-down time. Purple is the MCP signal, terminated at 50 Ω to ground with observed signal peaks corresponding to individual ion counts.

is switched off to prevent Doppler cooling from negatively influencing the time-of-flight mass spectrometry resolution.

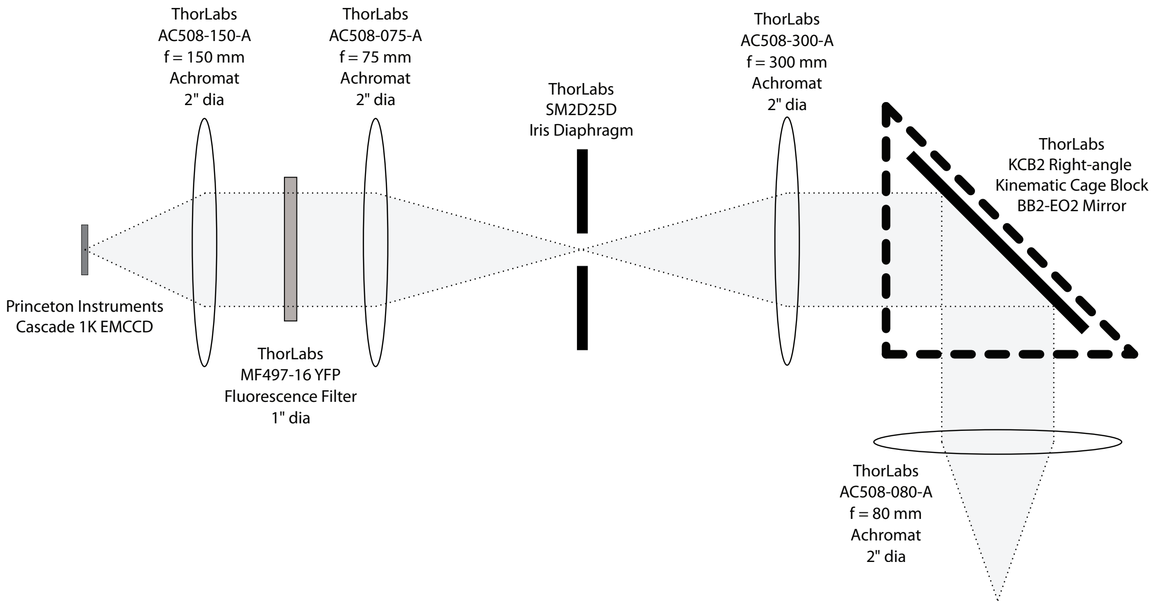


Figure 8.5. AlH$^+$ experiment optical imaging system. The system is mounted horizontally below an aluminum plate, which is attached to a ThorLabs PT3 XYZ translation stage.

8.3. Imaging System

The optical system for the experiment is shown in Fig. 8.5. The objective, a 2” diameter +80 mm achromatic doublet, sits ∼1 cm from the vacuum system viewport. An adjustable, ring-actuated iris diaphragm after the first lens pair reduces stray background light that reaches the Princeton Instruments Cascade 1k EMCCD camera. A ThorLabs MF497-16 YFP fluorescence filter sits behind the iris in order to only transmit the 493 nm Ba$^+$ fluorescence. After this pinhole is another lens pair. The right-angle allows the long-axis of the imaging system to be parallel to the flight tube to maximize usable optical table space.

The optical system is enclosed in ThorLabs 2” lens tube to block stray light and is mounted below an aluminum support plate (App. F.2.3.1) that also holds the camera.

This plate is then suspended from an angle bracket (App. F.2.3.2), which is attached to a ThorLabs PT3 XYZ translation stage that is mounted on an aluminum platform (App. F.2.3.3). The entire optical system and stage sits ∼14 inches above the optical table via internally-threaded aluminum posts.

8.4. Laser Frequency and Intensity Control via AOM

AOMs are used to shift the frequency of a laser beam and control the beam power by modulating the driving frequency and RF power, respectively [306, 307]. Since the output diffraction angle of the desired beam is dependent upon the AOM drive frequency, AOMs are commonly double-passed in a cat's eye configuration [308, 307]. In this setup, a lens of focal length f is placed a distance f from the AOM, collimating the output diverging diffraction orders. Additionally, this focuses each diffraction order, minimizing the spot size of each. The desired diffraction order is then selected and others blocked, the beam is retroreflected back into the AOM, and the desired output order again selected and others blocked. In this configuration, the lens compensates for the varying diffraction angle such that the output beam position and pointing vary negligibly as the AOM drive frequency is changed.

Two common implementations of the double-pass AOM setup use either beam walk or polarization to separate the input and output beams. Typically, our laboratory uses IntraAction ATM-series AOMs mounted to ThorLabs KM100PM mounts using an adapter piece located in App. F.4.1.

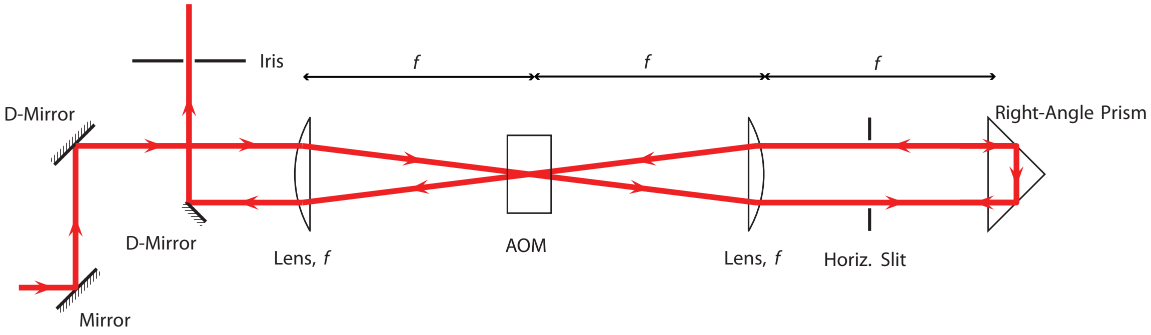


Figure 8.6. Double-pass AOM setup using beam walk to separate the input and output beams. Note that the separation between the input lens, AOM, collimating lens, and the right-angle prism is each f to be in the cat's-eye configuration. This method does not use polarization optics, but requires a horizontal slit instead of a pinhole to block the undesired AOM diffraction orders.

8.4.1. Double-pass AOM Setup with Beam Walk

The beam walk double-pass AOM setup is shown in Fig. 8.6. This implementation, while not requiring polarization optics or a linearly-polarized input beam, is significantly more difficult to align and typically has a lower efficiency than using the polarization arrangement (Sec. 8.4.2). Additionally, because a right-angle prism is used, a horizontal slit must be used such that the same diffraction order is retroreflected back into the AOM; in other words, a simple pinhole or iris cannot be used in this arrangement.

To align the setup, components are arranged with approximate locations depending on focal lengths of the lenses with a typical lens focal length of $f = 100$ mm. Without the AOM or right-angle prism in place, the distance between the two lenses is adjusted such that the output spot is collimated. The right-angle prism is placed roughly a distance f from the second lens such that the beam hits the center of the prism with optimized tip/tilt such that the beam is retroreflected. The beam is then walked using the input mirrors such that the output beam can be separated from the input with the D-mirror. The AOM is placed where the two beams overlap and optimized for the correct diffraction order. The correct output diffraction order, which does not move in position significantly as the AOM drive frequency is tuned, is then selected. The distance between the lens and right-angle prism, right-angle prism tip/tilt, and AOM angle are then optimized to maximize the output power of the setup.

8.4.2. Double-pass AOM Setup with Polarization

The polarization double-pass AOM setup is shown in Fig. 8.7. This implementation, while requiring polarization optics and a linearly-polarized input beam, is significantly easier to align and typically has a higher efficiency than using the beam walk arrangement (Sec. 8.4.1).

To align the setup, components are arranged with approximate locations depending on the focal length of the lens with a typical lens focal length of f = 100 mm. Note that the input light can either be transmitted or reflected by the PBS. Omitting the mirror, the single-pass efficiency is first optimized. The mirror is then placed, QWP angle optimized to maximize the output power from the PBS, and the second iris is placed. The correct output diffraction order does not move in position significantly as the AOM drive frequency is swept. The AOM angle, AOM drive power, and mirror tip/tilt are then optimized to maximize the output power from the setup.

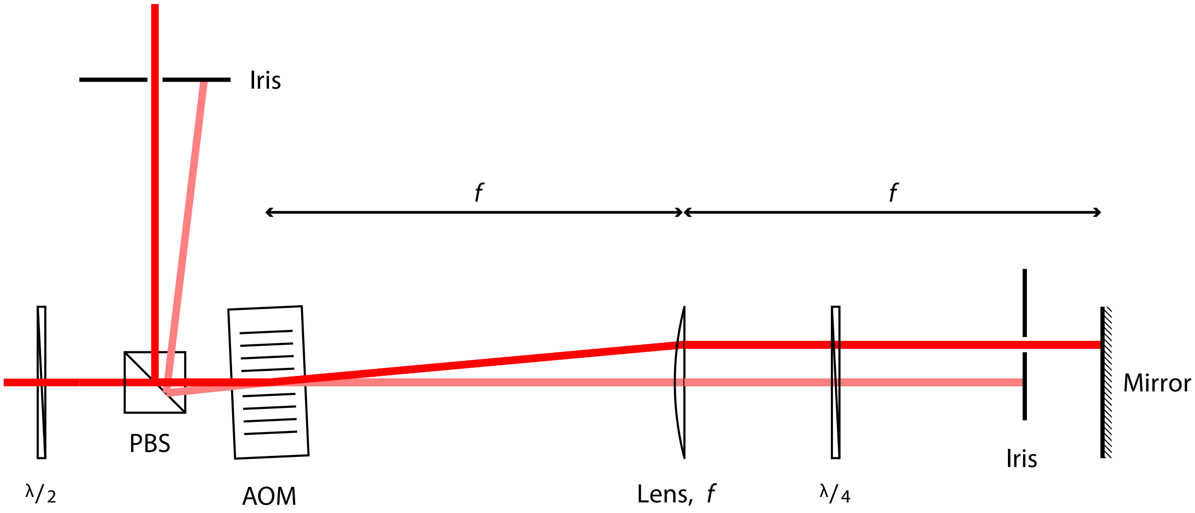


Figure 8.7. Double-pass AOM setup using polarization to separate the input and output beams. Note that the separation between the AOM and plane mirror is $2f$ to be in the cat's-eye configuration.

8.5. Laser Optical Systems

The 493 nm $^2S_{1/2} \rightarrow {}^2P_{1/2}$ Doppler cooling source enters the apparatus area via ThorLabs PM460-HP polarization-maintaining single-mode fiber. After collimation, the polarization is cleaned with a HWP and PBS. A double-pass AOM setup in the beam-walk configuration (Fig. 8.6) immediately follows using an IntraAction ATM-801A1 AOM and DE-801M driver. The 493 nm AOM setup is configured for AM and FM via PC control. Because the driver AM and FM input impedances are 50 Ω, a pair of OP07 op amp buffer circuits are used to increase the output current of the National Instruments DAC.

The 650 nm $^2D_{3/2} \rightarrow {}^2P_{1/2}$ Doppler cooling repump source enters the apparatus area via a ThorLabs PM630-HP polarization-maintaining single-mode fiber. After collimation, the polarization is cleaned with a HWP and PBS. A double-pass AOM setup in the beam-walk configuration (Fig. 8.6) immediately follows using an IntraAction ATM-801A1 AOM and DE-801M driver. This is later replaced with a single-pass setup due to 650 nm laser output power limitations. The 650 nm beamline is operated at a fixed AOM driver frequency and drive power for maximum 650 nm laser power.

Both Doppler cooling beams are overlapped using individual beam steering mirrors with the last 650 nm mirror a dichroic. A periscope raises both beams to ∼16 inches above the optical table, and is immediately followed with a 493 nm HWP and QWP (Cassix, low-order). A 650 nm HWP and QWP (Cassix, low-order) is placed before the 650 nm steering mirrors to correct the 650 nm polarization after the 493 nm waveplates to prevent pumping Ba^+ to a dark state.

Initially, the beams enter the ion trap axially using a ∼1 mm spot size. After installation of the time-of-flight mass spectrometer, the beams enter the ion trap ∼45° from the ion trap z-axis. Because of the aspect ratio of the Coulomb crystal [27, 28, 29, 30, 31], the beams are enlarged with cylindrical lenses before focusing into the vacuum system. Because this beam enlargement reduces the laser intensity, the beam is retroreflected with a mirror on the opposite side of the vacuum system.

When used, the 413 nm ${}^1S_0 \to {}^3D_1$ intercombination line, isotope-selective photoionization laser source enters the apparatus area via a ThorLabs S405-HP single-mode fiber and is shuttered with a zero-aperture iris. The beam is roughly aligned to trap center by eye and did not need further optimization.

When used, the 791 nm ${}^1S_0 \to {}^3P_1$ intercombination line, isotope-selective photoionization laser source enters the apparatus area via a ThorLabs 780HP SM fiber and is shuttered with a zero-aperture iris. This is roughly aligned to trap center by eye and did not need further optimization. The output of a SET Inc. UVTOP-310TO39HS 310 nm UV LED, mounted in a ThorLabs S1LEDM mount and driven by a laboratory-built diode laser current driver with ±15 V regulators, current modulation, and PZT sections removed, is aligned counter-propagating to the 791 nm output collimator to ensure overlap of the two at trap center.

8.6. High-power Pulsed Laser Systems

Several additional pulsed laser systems are used in the rotational cooling experiment. A Spectra Physics Mai Tai HP femtosecond laser system is used as the spectrally-filtered pulsed laser, and is discussed in Sec. 8.11.

For the $(1+1')$ REMPD state detection technique used to probe the rotational distribution of AlH^+, the $1'$-photon light source is the 266 nm fourth harmonic of a Coninuum Minilite Nd:YAG laser. This is mounted on an optical breadboard at ∼15 inches above the optical table, and the 532 nm second-harmonic beamlines is ∼1 inch above the top viewport of the vacuum system octagon.

As received, this laser is configured for 532 nm output only. A FHG crystal and mount (Continuum 640-1110) are installed with Continuum-supplied dichroics separating the 532 nm from the 266 nm output. The 266 nm outputs at the front of the laser system with ∼4 mJ per pulse. A mirror mount (App. F.2.4) adapter is mounted inside of the laser head with a 532 nm harmonic separator that transmits the remaining 1064 nm power. The side of the laser cover is drilled where the 532 nm beam exits; this hole is aligned by ablating the inner surface of the cover after internal upgrades. A picture of

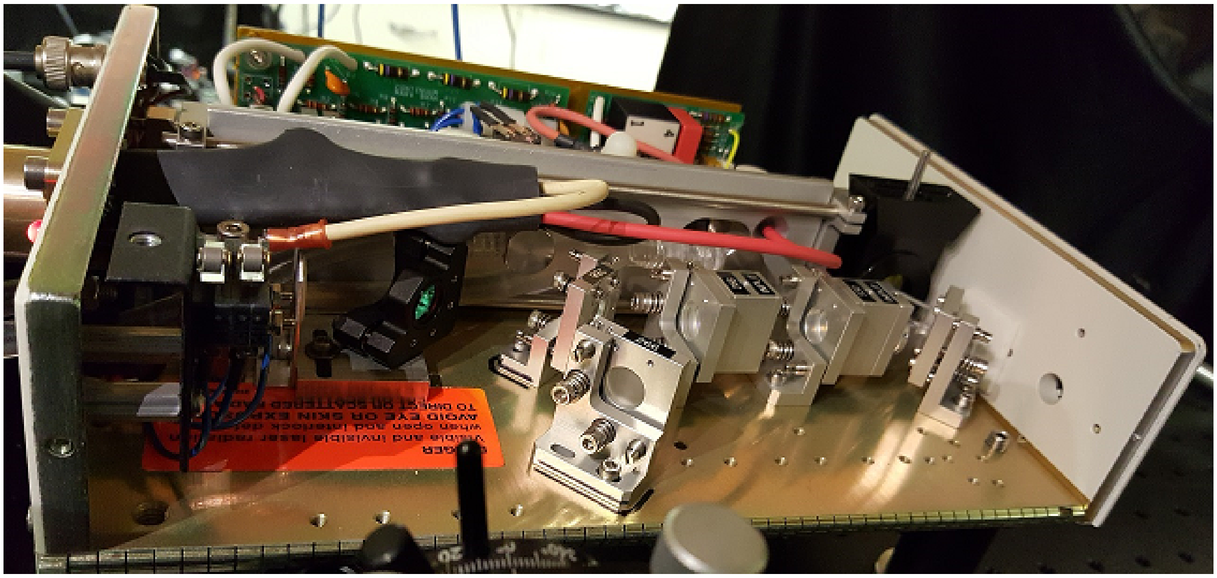


Figure 8.8. Picture of the Continuum Minilite laser head after the 266 nm FHG upgrade and 532 nm harmonic separator installation. The ThorLabs HBSY052 harmonic separator is mounted in a KM05 kinematic mirror mount, which is bolted to the laser head with an adapter (App. F.2.4). This allows both 532 nm and 266 nm harmonics to be used from the Minilite system.

the modifications is shown in Fig. 8.8. A periscope then directs the 266 nm beam to the proper height below the optical breadboard, is recollimated with a long-focal-length lens, passes through a ThorLabs 266 nm HWP and ThorLabs GLB10 α-BBO polarizer for a REMPD cross section measurement (Sec. 11.2), and is focused into the chamber using a $f = 1000$ mm spherical lens far from the vacuum system and a $f \sim 250$mm cylindrical lens nearly touching the vacuum chamber viewport.

The 266 nm beam is aligned with trap center with the aid of a ThorLabs DCC1545M CMOS camera and Ba$^+$ ions. The camera is aligned with the Doppler beam reflection spots off the 6 inch viewport. The 266 nm beam is aligned in the vertical direction to match this reflection spot height. The horizontal is then adjusted while monitoring the time required for the trapped Ba^+ to heat sufficiently to melt the Coulomb crystal (typically ∼30 s). When this time is minimized, the 266 nm beam is aligned with the ions. Once the 266 nm exposure is stopped, the ions return and again crystallize in ∼30 s. The likely mechanism is the high-intensity 266 nm light off-resonantly exciting the 265 nm Ba^+ $6s^2P_{1/2} \rightarrow 8s^2S_{1/2}$ transition. The Ba^+ ions then spontaneously decay to the low-lying metastable $6s^2D_{5/2}$ state, which has a ∼35 s lifetime [**134**].

The Continuum Minilite laser also provides the 532 nm ablation light source. A HWP and PBS provide power control. A second HWP and PBS split the remaining 532 nm power into a weak and a powerful beam. These beams pass through a pair of laboratory-built solenoid shutters to control which if either beam is transmitted. A large mirror then reflects these beams to a reflective ND filter wheel, spherical lens ($f = 150$ mm), and an additional mirror to direct the beam vertically downward into the vacuum system through the top viewport to hit ablation targets below the ion trap. Note that during alignment of the ablation beam, all ion trap and time-of-flight mass spectrometer power supplies and amplifiers must be powered off; if the ablation laser hits an ion trap electrode, it will destroy the RF amplifier and can damage other components.

For the $(1 + 1')$ REMPD state detection technique used to probe the rotational distribution of AlH^+, the 1-photon light source is the second harmonic of a Sirah Precision-Scan LG-18-US dye laser pumped at 532 nm by a Spectra Physics Quanta-Ray Pro-270 Nd:YAG laser with all components for operating at the 355 nm third-harmonic installed in both lasers. The dye laser contains an amplifier stage for high-power operation, frequency conversion unit, and Pellin-Broca prism harmonic separation stage. These lasers sit on a separate table from the rest of the apparatus. The output of the dye laser is recollimated to the apparatus optical table with irises blocking undesirable portions of the laser beam profile. Power control is performed using a 355 nm Nd:YAG zero-order HWP and ThorLabs GLB10 α-BBO polarizer to prevent power broadening from dissociating molecules in states neighboring the target state. A cylindrical lens expands the beam horizontally before being reflected at ∼45° into the vacuum system. Because the two tables shift appreciably, the dye laser beam is frequently realigned with the 266 nm beamline. Additionally, the Nd:YAG pump laser is initially used as the 266 nm photon source using a BBO crystal in a custom mount (App. F.2.6).

8.6.1. Fixing Laser Frequencies

For this experiment, all CW laser frequencies are actively locked including the 493 and 650 nm Doppler lasers and the 791 and/or 413 nm photoionization lasers when used. Wavelength stabilization of all ECDLs is performed as described in Sec. 2.6.

From other experiments in the laboratory, the 493 nm $^2\mathrm{S}_{1/2} \rightarrow {}^2\mathrm{P}_{1/2}$ and 650 nm $^2\mathrm{D}_{3/2} \rightarrow {}^2\mathrm{P}_{1/2}$ Doppler cooling wavelengths are known to within ∼10 MHz by $^{138}\mathrm{Ba}^+$ fluorescence.

The 413 nm $^1\mathrm{S}_0 \rightarrow {}^3\mathrm{P}_1$ intercombination line $(1+1)$ photoionization wavelength was previously measured using a Doppler-broadened absorption line in a Hamamatsu L2783-56NE-BA barium laser galvatron using optogalvanic spectroscopy (Sec. 2.5.1.1). The wavelength is fine-tuned using the apparatus described here by maximizing the fluorescing $^{138}\mathrm{Ba}^+$ loading rate while minimizing the number of dark ions loaded into the trap.

Table 8.1. Dye laser calibration to known AlH$^+$ A$^2\Pi_{1/2} \leftarrow$ X$^2\Sigma^+$ Q-branch transitions from Ref. [119]. Uncertainty in the wavelength meter reading is given as the standard deviation of the 600-shot-each trials. Data for the Q(2) measurement is not adequately recorded.

Transition	(Ref. [119])/2 (cm^{-1})	Set (cm^{-1})	Wavelength Meter (cm^{-1})	Total Shots
Q(0)	13836.230	13852.4	13836.27(3)	3000
Q(1)	13838.865	13854.9	13838.84(4)	5400
Q(2)	13841.240	13857.5		
Q(3)	13843.365	13859.1	13843.1(2)	13800
Q(4)	13846.290	13862.3	13846.324(10)	6600
Q(5)	13848.610	13864.7	13848.63(2)	4800
Q(6)	13851.050	13867.3	13851.045(3)	1800
Q(7)	13853.715	13869.9	13853.668(7)	4800

Similarly, the 791 nm ^{1}S$_0 \rightarrow {}^3$P$_1$ intercombination line $(1+1')$ photoionization wavelength was previously measured using a Doppler-broadened absorption line in a Hamamatsu L2783-56NE-BA barium laser galvatron using optogalvanic spectroscopy (Sec. 2.5.9.2). The wavelength is fine-tuned using the apparatus described here by maximizing the fluorescing ^{138}Ba$^+$ loading rate while minimizing the number of dark ions loaded into the trap.

To assure that the dye laser supplying the 360 nm 1-photon for $(1+1')$ REMPD of AlH$^+$ is on resonance with the desired rotational line, the 720 nm fundamental output of the dye laser is coupled into a ThorLabs HPSC25 multimode fiber to the wavelength meter with sufficient averaging and tuned to the lines reported in Ref. [119]. Table 8.1 contains the desired AlH$^+$ A$^2\Pi_{1/2} \leftarrow$ X$^2\Sigma^+$ Q-branch transitions and set dye laser wavelength.

8.7. PC Control and Timing

All PC control is performed using the NI-DAQmx hardware platform with a PXI-1033 chassis and interface cards listed in Table 8.2 within the LabVIEW environment. This

Table 8.2. National Instruments PXI I/O cards used for apparatus control

PXI Slot	Card	Function
2	PXI-6251	Analog input, 16-bit, 1 MS/s (multichannel), 16 channels
3	PXI-6602	8-channel counter/timer with digital I/O
4	PXI-GPIB	High-Performance GPIB Controller
5	PXI-6722	Static and waveform analog output, 13-bit, 8 channels
6	PXI-6602	8-channel counter/timer with digital I/O

includes EMCCD control and image quantification, setting the ion trap RF and DC voltages, running the pulse valve (Sec. 8.10.1), triggering time-of-flight mass spectrometry, and controlling the 493 nm AOM settings (App. C.2.2); shaking measurements; programming the single-shot pulse sequence for time-of-flight mass spectrometry; synchronizing the Continuum and Spectra Physics Nd:YAG pulses (App. C.2.3); rotational cooling and REMPD light shutter control; and oscilloscope data collection automation and control.

8.8. Magnetic Field

Since the AlH$^+$ A$^2\Pi$ state upper state degeneracy is smaller than that of the lower X$^2\Sigma^+$ state when driving P-branch transitions, dark states must be considered. The unpaired electron spin in the X$^2\Sigma^+$ state allows modest magnetic fields to cause sufficient Larmor precession rates and prevent a slowdown of rotational cooling. In the experiment, a $\sim$ 2 Gauss magnetic field is used perpendicular to the cooling laser polarization using a single, $\sim$100-turn coil with outer diameter of $\sim$7 inches.

8.9. Loading Ba$^+$ into the Ion Trap

Three Ba$^+$ loading techniques were used with this apparatus. The first two use $(1+1)$ or $(1+1')$ isotope-selective REMPI of neutral barium emanating from an oven under trap center. The third, direct ablation loading, is used for the AlH^+ rotational cooling data collection.

8.9.1. Photoionization of Barium

Two of the ion trap Ba^+ loading methods rely on photoionization of neutral barium emanating from an oven under trap center. The oven, a Goodfellow 713-654-51 alumina tube, ∼2 cm long with one end melted and sealed with a oxy-acetylene torch, is wrapped in a tungsten filament. This filament is spot welded to MDC vacuum electrical feedthru rods, which are connected via barrel connectors to the feedthru pins. The stiffness of the vacuum feethru pins and tungsten filament suspends the oven (Fig. 8.9) below a skimmer (App. F.2.2) [**309**, **310**], which includes two pockets to hold ablation targets.

The oven is stuffed with small strips of metallic barium (ESPI Metals) cut from 3-6 mm pieces under dry nitrogen in a glove box. The oven is the last component installed into the vacuum system before pumping down. The oven is quickly carried from the glove box, installed in the vacuum system, and the 6 inch viewport is quickly sealed with a few bolts before turning on the roughing vacuum pump. While pumping down, the viewport is further tightened. If the system is vented, dry nitrogen is used to back fill the vacuum system, and once opened, a steady flow is used to prevent barium oxidation. During the final stages of vacuum system baking, the oven is turned on and allowed to outgas for several hours at the expected operating current, typically ∼1.30 A.

The first photoionization technique is a $(1+1)$ isotope-selective REMPI process using the 413 nm $^1S_0 \rightarrow {}^3P_1$ intercombination line [**196**, **197**, **198**]. This isotope-selective photoionization technique *requires only a single light source*. Unfortunately, the 413 nm

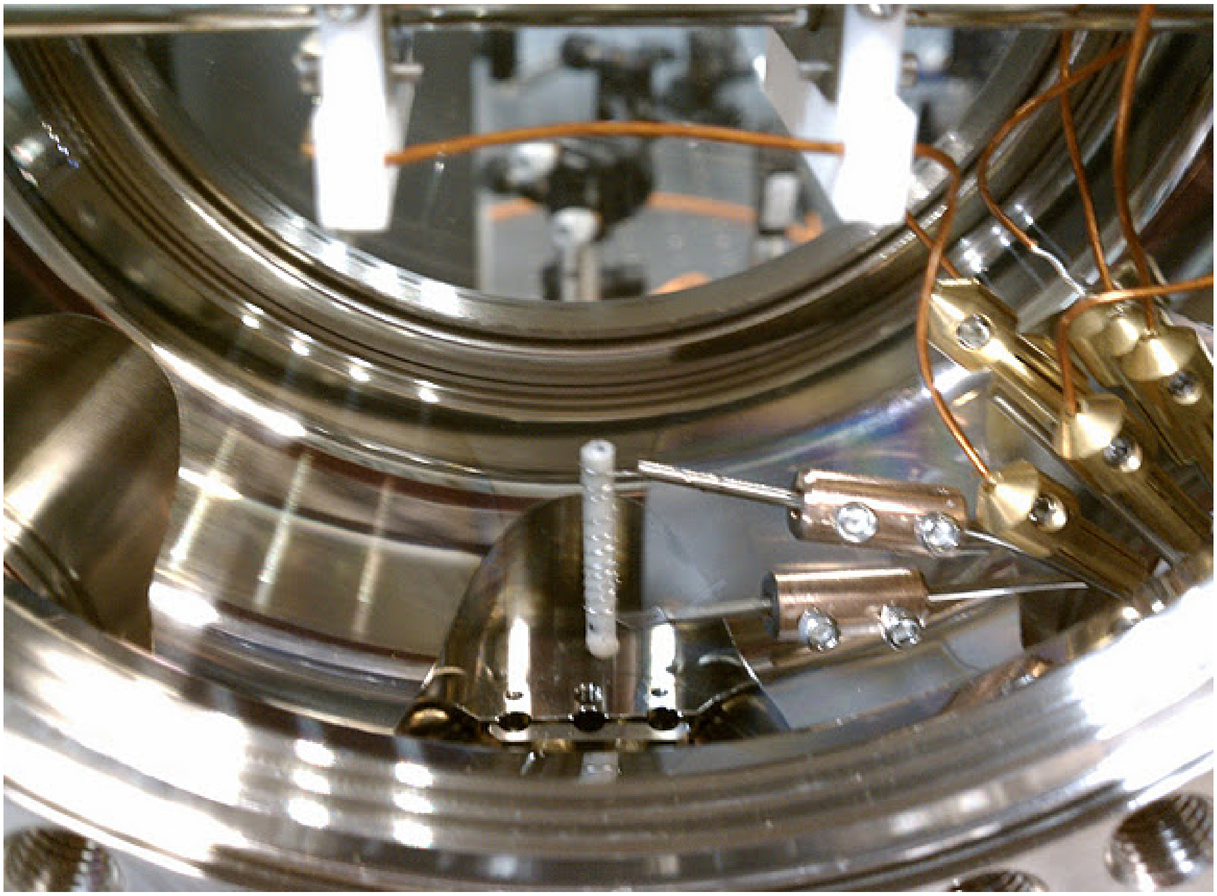


Figure 8.9. Installed barium oven with skimmer removed. One of the groove-grabbers that mounts the skimmer is visible on the very bottom center of the image.

laser diode underwent rapid degradation before the isotope-selectivity could be quantified using the ion trap; however, note that the method is qualitatively significantly faster and more isotope-selective than the 791 nm and UV LED $(1 + 1')$ technique.

Quantifying this photoionization scheme would proceed by loading for a specific amount of time and counting fluorescing $^{138}\mathrm{Ba}^+$ ions. Then the procedure would be repeated for $^{136}\mathrm{Ba}^+$ and $^{134}\mathrm{Ba}^+$ using the known isotope shifts for the Doppler cooling beams. Dark ions loaded that are not accounted for by the even-isotope investigations above would be the odd-isotope ^{135}Ba$^+$ and ^{137}Ba$^+$. The power-dependence of the loading rate could then be used to determine the effective photoionization cross section.

The second photoionization technique is a $(1 + 1')$ REMPI process using the 791 nm ^{1}S$_0 \rightarrow {}^3$P$_1$ intercombination line [221] and the 310 nm output of an SET Inc. UVTOP-310TO39HS UV LED [222] to avoid using a pulsed nitrogen laser. As observed with this apparatus with the same barium oven, this photoionization technique is qualitatively significantly slower than the 413 nm photoionization technique and less isotopically-selective.

8.9.2. Laser Ablation of Metallic Barium Target

Due to the long warm-up time of the barium oven (∼1 minute), concerning when collecting experimental statistics, concern of the neutral barium flux causing rotationally-changing collisions in the AlH$^+$ sample, and loss of the 791 nm Toptica DL-Pro laser source to another experiment in the laboratory with laser operating at 770 nm, the barium oven and associated photoionization techniques are replaced with direct laser ablation. In this scheme, a 532 nm laser pulse from a Continuum Minilite is focused directly onto a piece of metallic barium (ESPI Metals) sitting directly in one of the skimmer pockets under trap center.

The resulting ablation plume, containing electrons and a mess of atomic and clusters of charged and neutral barium, expands from the ablation spot. Any initially-produced ions that pass through the trap *would not be* trapped if the trap potential is unaffected from the ablation plume. However, two processes assist with ion trap loading: (1) the ablation plume partially shorts the trap RF voltage, which traps ions within the trapping volume as the RF voltage recovers; and (2) a fraction of the neutral atomic flux will be in long-lived, high-lying electronic states (Rydberg states), which are then field-ionized by the trap voltages. Direct ablation loading permits fast loading of the trap, with typical experiments loading a sufficient number of $^{138}\mathrm{Ba}^+$ ions cooled into a Coulomb crystal in $\sim$15 seconds.

8.10. Loading AlH^+ into the Ion Trap

Loading AlH^+ into the ion trap is an experimental challenge. The first technique attempts to produce a beam of neutral AlH, which is then ionized by electron-impact ionization. The second technique ablates a solid aluminum target followed by reaction with background gas.

8.10.1. Neutral AlH Production and Electron Impact Ionization

The first technique for producing AlH^+ uses a source similar in design to that in Ref. [276]. A tungsten filament wrapped with aluminum wire is placed inside an alumina tube (App. F.2.1) in front of a Parker Hannifin Corporation 099-0402-900 pulse valve backed by $\sim$2 ATM of H_2. The pulse valve is mounted on a 4 to 1.75 inch flat reducer flange with a Swagelok SS-4-UT-A-8 vacuum fitting welded internally similar to what is described in Ref. [311]. The alumina tube is then fitted in this fitting with a UHV-compatible o-ring. When sufficiently heated, the aluminum wets the tungsten filament, and the pulse valve introduces a high-density of H_2 gas. The resulting AlH, carried in the H_2 carrier gas, passes through a conical skimmer (App. F.2.9) [309, 310, 312, 313] into the main science vacuum chamber. Because the filament source is fragile and requires frequent replacement, an inline valve seals off the production side of the chamber. Using a Stanford

Research Systems RGA300 residual gas analyzer on the opposite side the science chamber octagon, signal attributed to AlH is observed as the signal varied with filament current, is distinct from an observed Al signal, and only appears when the electron-impact ionization energy of the RGA300 was set above the ∼8 eV expected ionization energy for AlH [279].

To ionize the neutral AlH via electron-impact ionization [279], an electron-impact ionizer (Fig. 8.10, App. F.2.5) similar in design to that of the RGA300 is designed, built, and installed between the source chamber and science chamber octagon. Due to the concentric design with electron-producing thoriated tungsten filament [314] held between mesh grids, each electrically-isolated, the maximum electron energy is well-controlled; an electron can only gain a maximum energy equal to the potential difference between filament and inner grid, which is necessary to avoid ionizing species other than AlH such as equal-mass CO. Additionally, the outer grid can be biased to prevent electrons from escaping the ionizer, and an ion lens element is added at the exit of the ionizer to assist in guiding ions into the ion trap if necessary.

A Stanford Research Systems RGA300 residual gas analyzer with the RGA ionizer removed and ion trap RF rods operating to help guide ions from the electron-impact ionizer to the RGA300 is used to verify AlH^+ production. The signal at 28 u shows a threshold behavior at ∼8 eV, which is the expected ionization energy for AlH [279]. Later attempts to successfully trap AlH^+, done by lowering the voltage on the endcap closest to the ionizer to just slightly higher than the expected ion kinetic energy while relying on trapped, laser-cooled Ba^+ to remove sufficient energy to keep produced AlH^+ trapped, is a significant challenge despite previous work of trapping externally-produced ions by previous groups [315, 316, 317, 318, 319]. This work uses a secular frequency excitation

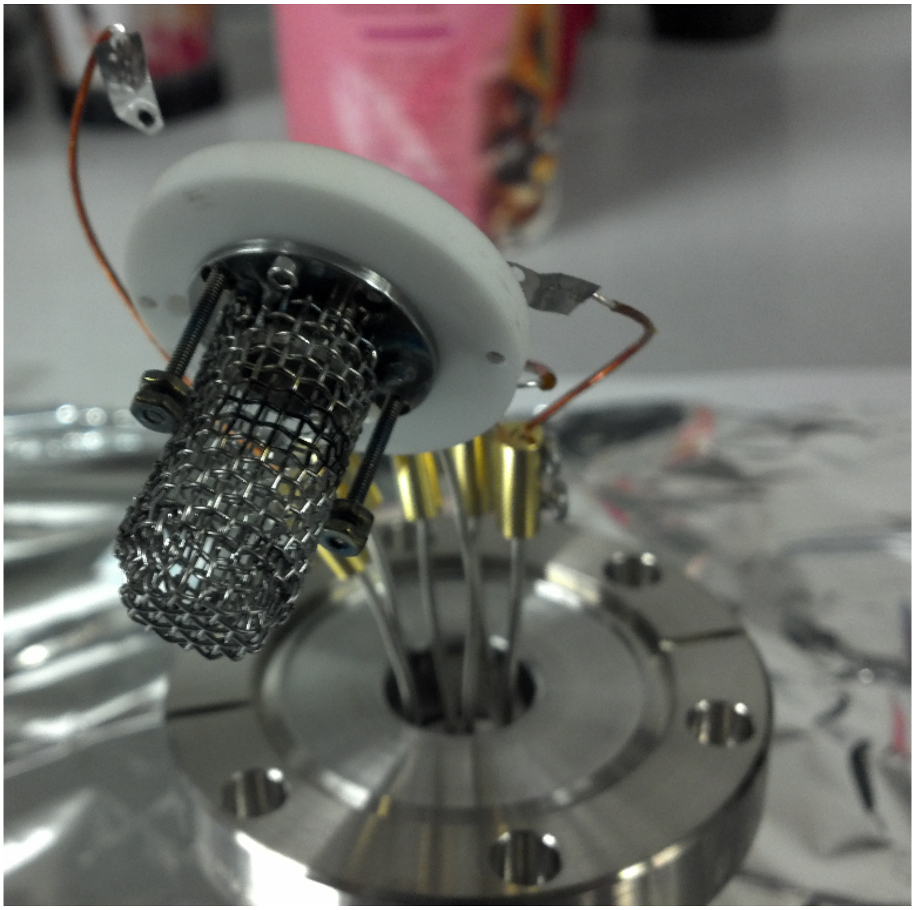


Figure 8.10. Picture of the electron impact ionizer with outer mesh grid removed. All metal components are mounted to a Macor base and are electrically isolated. A thoriated tungsten filament, mounted on the long screws between two nuts mid-way up the inner mesh grid, supplies electrons with a maximum energy set by the potential difference between filament and inner mesh grid.

fluorescence detection technique [74] discussed in Ref. [301]. Furthermore, this challenge may have been due to operating the ion trap too close to the stability limit for Al^+.

The fragile filament proved to be an additional challenge. This is replaced by a Ted Pella alumina-coated basket oven just below the line-of-sight of the pulse valve and filled with pieces of aluminum cut from a 99.999% pure aluminum evaporation pellet (Ted Pella

29-13). The alumina tube is also removed. The oven is run, producing a flux of neutral Al. The hope is that hot, neutral Al will react with the H_2 carrier gas despite being thermodynamically unfavorable [320]. Despite the oven being significantly more robust, AlH production was not observed.

8.10.2. Laser Ablation of Metallic Aluminum Target

To remedy both of the significant challenges mentioned above, and with the transition to loading Ba^+ by direct ablation, the apparatus is reconfigured for direct ablation loading of Al^+ from a 99.999% pure aluminum evaporation pellet (Ted Pella 29-13) sitting in one of the skimmer ablation target pockets under trap center. Ablation of Al requires significantly more power than that required for loading Ba^+ such that successfully loading Al^+ would frequently eject all ions from the trap. However, a small island of operation permitted co-trapping Ba^+ and Al^+ as shown in Fig. 8.11. It would be later determined that this is from operating the trap too close to the stability limit for Al^+, and increasing the trap drive frequency remedy the problem.

Using time-of-flight mass spectrometry, Al^+ loading is confirmed with typically a small AlH^+ population as well. This small AlH^+ population is attributed to the ablation process producing some Al^+ ions in a highly-excited electronic state with energy sufficient to break the hydrogen-hydrogen bond in H_2 [320]. A following collision with background H_2 gas in the chamber produces the small AlH^+ population, which is routinely used to calibrate the time-of-flight discriminator time between Al^+ and AlH^+.

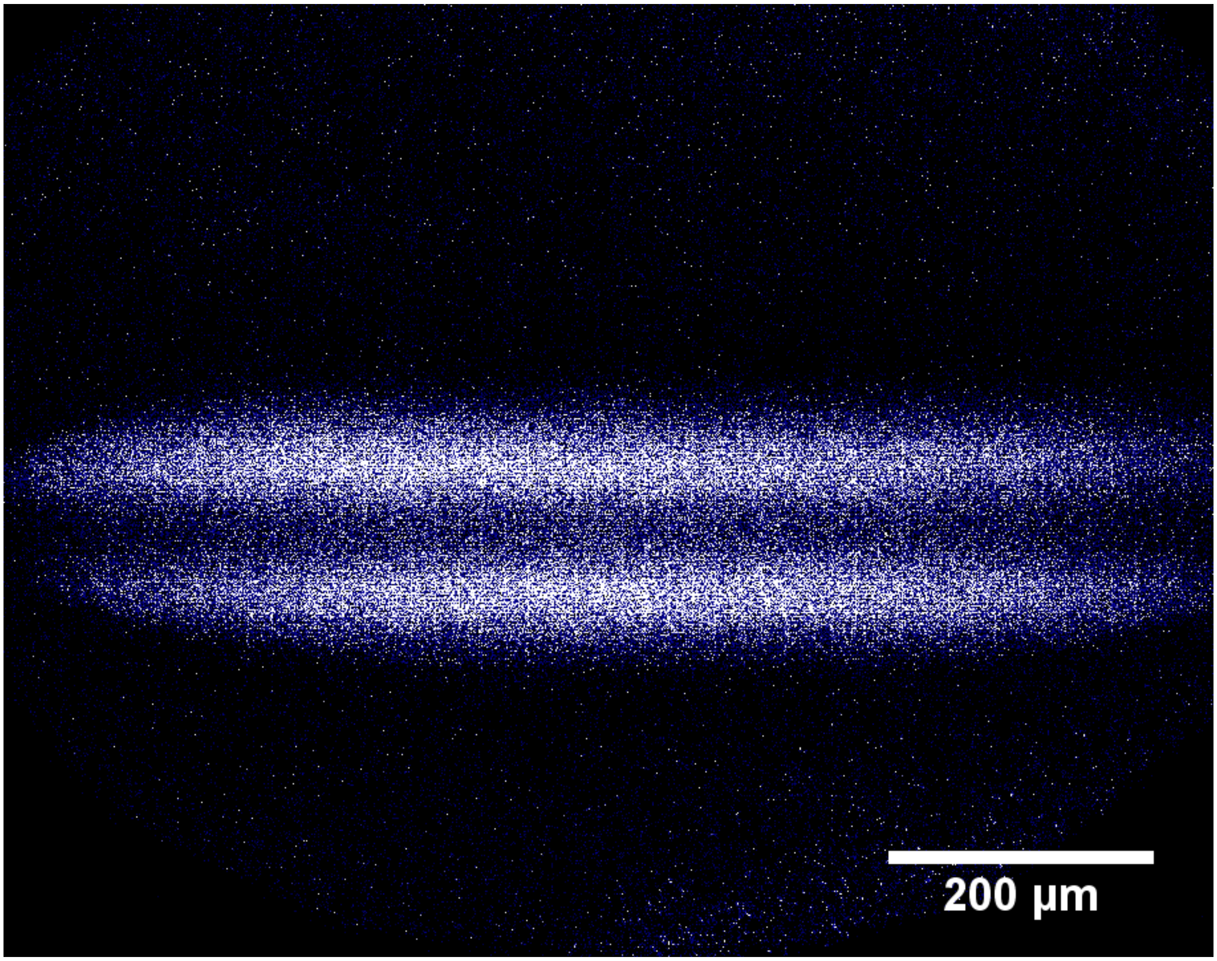


Figure 8.11. EMCCD bicrystal image. The fluorescing ions are $^{138}\mathrm{Ba}^+$. The dark core consists of other Ba^+ isotopes and AlH^+.

8.10.3. Background Gas Reaction to AlH^+

While calibrating and optimizing the time-of-flight-mass spectrometry apparatus, the Al^+ would slowly react to form AlH^+ with a reaction rate dependent on how open an angle valve is to a non-evaporable getter pump. Noting this, the experiment is operated with an intentionally degraded vacuum (3×10^{-10} Torr) so that AlH^+ forms on a one-minute timescale by reaction of ablation-loaded Al^+ with background gas. Note that no common residual gas in UHV systems can account for this reaction due to the energetics involved [320]. However, radicals can be produced by the non-evaporable getter pump, which could account for the AlH^+ production.

8.10.4. Background Gas Reaction Repeatability

After writing Ref. [263], the background reaction ceased, attributed to the non-evaporable getter and ion pump depleting the background gas providing the favorable reaction. The vacuum system is vented to atmosphere, the degraded MCP is replaced, an RGA300 is added to identify the reactant gas if the reaction returns, and the UHV system is baked. Unfortunately, this did not replenish the favorable $\mathrm{Al}^+ \rightarrow \mathrm{AlH}^+$ background gas reactant. Multiple common gases present in UHV systems, including hydrogen, methanol, ethanol, acetone, water, and laboratory air, introduced via the pulse valve, do not recover the reaction. Additionally, the electron-impact ionizer is used with the pulse valve backed by hydrogen gas in an attempt to generate atomic hydrogen [321, 322, 323], which will readily react with Al^+ to form AlH^+ [320]; however, this does not recover AlH^+ production.

Additionally, no laser sources in the laboratory are capable of sufficiently driving the $^1\mathrm{S}_0 \rightarrow {}^3\mathrm{P}$ transitions in Al^+; these states have sufficient energy to react with molecular hydrogen gas to form AlH^+. Flooding the vacuum system with H_2 and exposing the Al^+ the 2 mJ, 266 nm output of the Continuum Minilite for times up to ~60 seconds do not result in any significant AlH^+ production.

8.11. The Spectrally-filtered Pulsed Laser

The light source for broadband rotational optical cooling of AlH^+ is the second-harmonic of a SpectraPhysics Mai-Tai HP femtosecond laser yielding 900 mW at the 360 nm $A^2\Pi_{1/2} \leftarrow X^2\Sigma^+$ transition. The laser is spectrally filtered after the SHG stage; if the spectral filtering is performed before SHG, the different frequency components would mix, convoluting the resulting output spectrum and destroying the spectral filtering.

Spectral filtering is accomplished using a home-built device, consisting of a diffraction grating, mirror, and cylindrical lenses in the 4-f Fourier-transform optical layout [**324**] and a razor blade mask to block the high-frequency part of the spectrum. The broadband spectral-filtering apparatus and its application to rotational cooling is described in Ref. [**114**], while the details of the experimental apparatus are discussed below.

The experimental spectral-filtering apparatus is shown in Fig. 8.12 with a list of components in Table. 8.3. L_1 tightly focuses the fundamental-wavelength beam into the SHG crystal for efficient SHG. After L_2, the beam is ∼2.5 cm in diameter (MFD) with a slightly elliptical profile rotated slightly from the horizontal. The pair of 355 nm Nd:YAG line mirrors serves as a beam dump for the remaining fundamental power, reflecting only the 360 nm light. The 355 nm zero-order HWP rotates the polarization of the beam to be S-plane to the holographic grating for maximum efficiency. L_3, aligned to focus along the horizontal plane and sitting roughly one focal length from the grating, collimates the horizontal divergence of the beam after the grating caused by the broadband spectrum. The mask (a razor blade) and M_3 are positioned roughly one focal length from L_3 and are mounted on a translation stage to finely-adjust the mask position/spectral cutoff location. Additionally, M_3 reflects the beam slightly downward such that the output beam

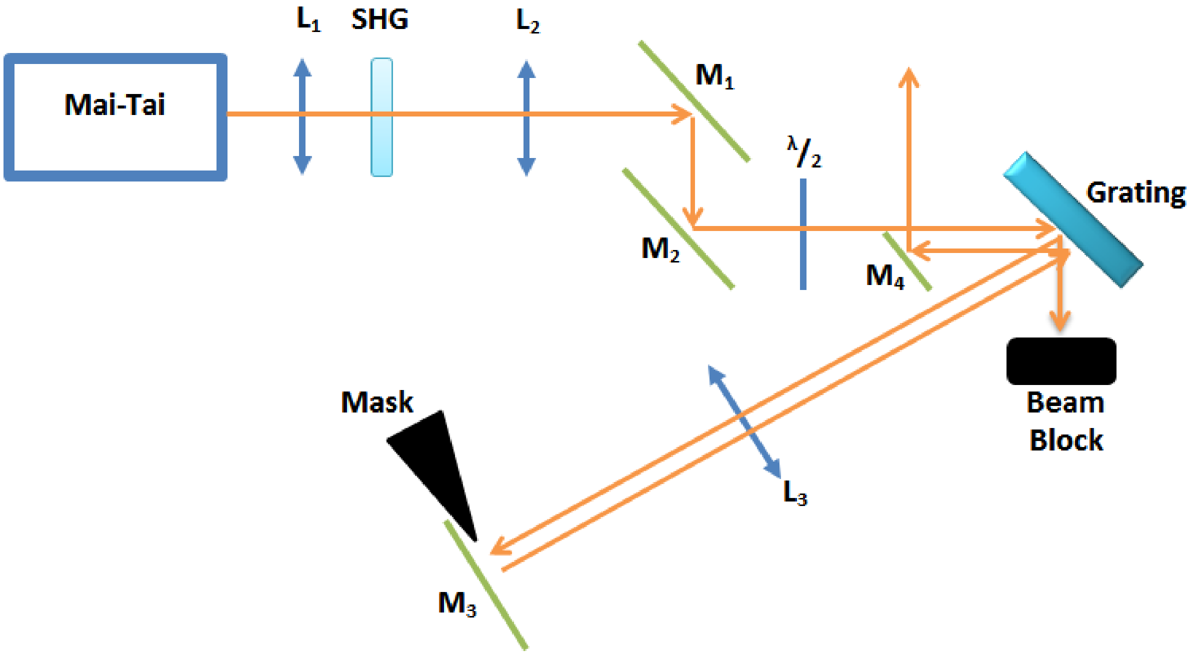


Figure 8.12. Diagram of the spectrally-filtered pulsed laser apparatus. The list of components is in Table 8.3. M_3 slightly displaces the return beam along the vertical, and M_4 picks off the slightly-displaced beam from below the input beam.

is spatially separated from the input beam for M_4, directly below the input beam, to pick off.

The setup is aligned using an Ocean Optics HR4000 high resolution spectrometer with 380 nm center wavelength grating installed with 5 μm slit width. Light is collected with a FOIS-1 integrating sphere and fibered to the spectrometer with a QP400-2-UV-VIS two meter fiber. Using a 405 nm laser pointer with expected linewidth of $\lesssim$10 GHz = $\lesssim$0.33 cm^{-1}, the spectrometer measures a feature 8 pixels wide (Fig. 8.13), which is more than sufficient to observe a spectral cutoff sharp enough for broadband rotational optical cooling of AlH$^+$.

Table 8.3. Components used in the spectrally-filtered pulsed laser apparatus. All mounts are from ThorLabs.

Comp.	Supplier	P/N	Mount	Description
L_1	ThorLabs	LA1805-A	LMR1	$f = 30$ mm
SHG	Gooch&Housego		KS1RS	BBO, 0.5 mm thick, 32° cut
L_2	ThorLabs	LA1256-A	LMR2	$f = 300$ mm
M_1	ThorLabs	NB1-K08	KMS, MH25T	355 nm YAG mirror
M_2	ThorLabs	NB1-K08	KMS, MH25T	355 nm YAG mirror
$\frac{\lambda}{2}$	ThorLabs	WPH10M-355	RSP1	355 nm zero-order HWP
Grating	Newport	53-*-530H	KM100S	3600 lines/mm Grating
L_3	ThorLabs	LJ1144L2-A	KM100C	$f = 500$ mm cylindrical
M_3	CVI	Y3-2037-0	KM200	355 nm YAG mirror, 0°
M_4	CVI	MPQ-245-390 -SQM-2506M	KM100S	Dielectric mirror

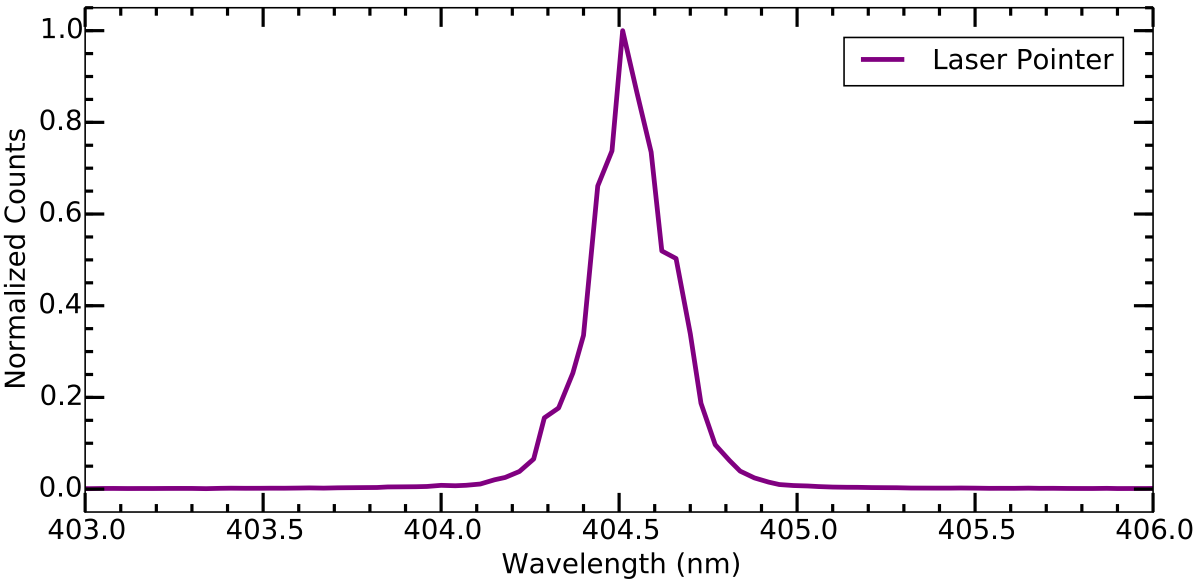


Figure 8.13. Measured output spectrum of a 405 nm laser diode in a $\lesssim$5 mW laser pointer. The expected laser linewidth is $\lesssim$10 GHz = $\lesssim$0.33 cm^{-1}, which corresponds to 8 pixels on the spectrometer. Thus, the spectrometer should have sufficient resolution to observe a spectral cutoff sharp enough for broadband rotational optical cooling of AlH$^+$.

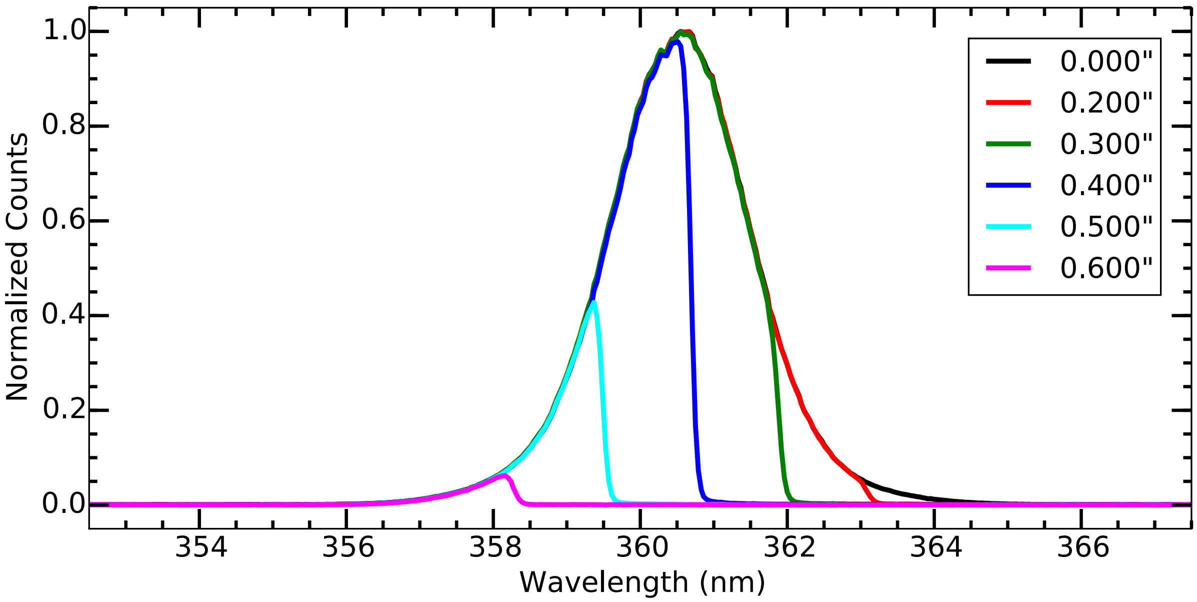


Figure 8.14. Measured spectra from behind the razor blade mask as a function of translation stage micrometer position. Note the sharp cutoff observed here is sufficient for AlH^+ rotational cooling and may be sharper than the spectrometer resolution.

The spectral-filtering apparatus is aligned component-by-component with the critical components being L_2, L_3, and the mask/M_3. L_2 is placed to provide a collimated output beam by eye over several meters. L_3 is placed using the spectrometer with the mask being a single 30 AWG wire. The wire position and L_3 are shifted slightly to maximize the contrast of the dip in the broadband spectrum as observed with the spectrometer. Once this contrast is maximized, L_3 is the correct distance from the grating and the wire is the correct distance from L_3. The wire is then replaced with the razor blade mask. The transmitted spectrum as a function of razor blade translation stage micrometer position in shown in Fig. 8.14 and recombined output spectrum after M_4 in Fig. 8.15.

To confirm that the output spot does not have different spectral components across the output beam profile, which would be problematic for rotational cooling since AlH^+

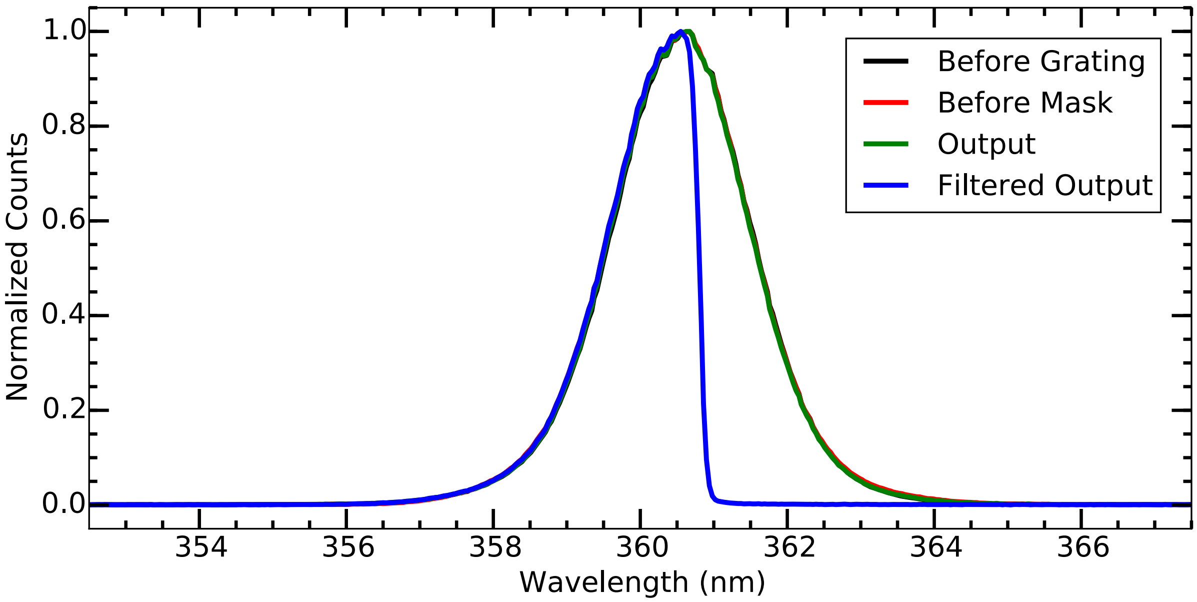


Figure 8.15. Measured output spectra before the grating, before the mask, and after M_4. From these spectra, the spectral-filtering apparatus with no mask does not significantly change the spectrum. When the mask is in place, only the spectral components not blocked by the mask are output. Note the sharp cutoff observed here is sufficient for AlH^+ rotational cooling and may be sharper than the spectrometer resolution.

molecules on different sides of the Coulomb crystal would be exposed to different rotational cooling light spectra, the output spot is clipped from left to right and right to left while recording spectra with the spectrometer (Fig. 8.17). From these spectra there is some position dependence on the spectra; however, the spectral cutoff is not affected and the spectrum does not narrow sufficiently to be problematic.

An additional concern is that the spectrally-filtered spectrum will change with distance to the ion trap. This is a concern because the beam additionally passes through a 3-element telescope and two lenses before the vacuum chamber. The former telescope is immediately after M_4, and is comprised of a $f = +200$ mm spherical lens (ThorLabs

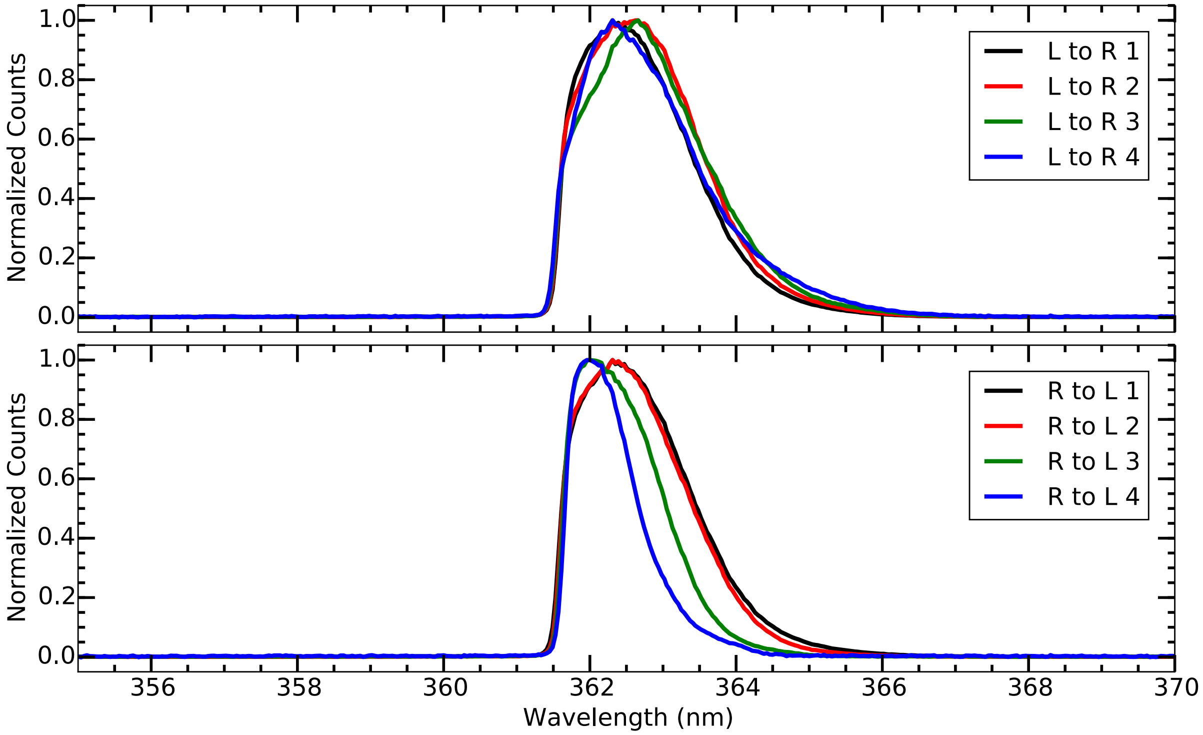


Figure 8.16. Measured output spectra after M_4 while clipping the spot from the left to the right (L to R, top) and from right to left (R to L, bottom) as a function of amount of clipping. From this data, there is some position dependence on the spectra, particularly on the left-side. However, the spectral cutoff is not affected and the spectrum does not narrow by a sufficient amount to be problematic. Note the sharp cutoff observed here is sufficient for AlH^+ rotational cooling and may be sharper than the spectrometer resolution.

LA4102-UV), a $f = 100$ cylindrical lens (ThorLabs LJ4395-UV) to compensate the astigmatism present in the beam that is placed very close to the second spherical lens, and a $f = -30$ mm spherical lens (ThorLabs LC4252-UV). The resulting beam is well collimated over >1.5 meters. Additionally, a $f \sim 250$mm UV fused silica cylindrical lens placed nearly-touching the vacuum chamber viewport focuses both the spectrally-filtered beam and 266 nm beam into the vacuum system. Spectral deviations of the spectrally-filtered beam are investigated using the spectrometer before and after the ion trap vacuum

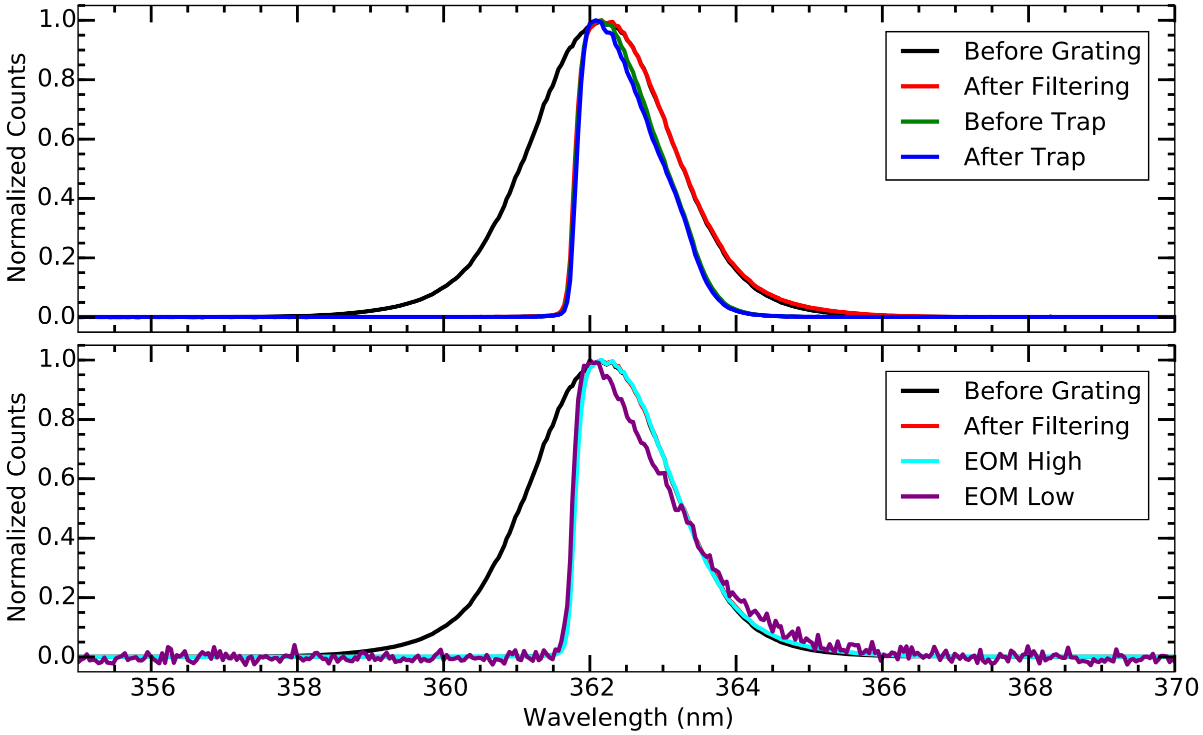


Figure 8.17. Measured output spectra before and after the ion trap vacuum system (top) and before and after the EOM in the on- and off-state (bottom). From these spectra, there is some spectral clipping at the long-wavelength side before and after the ion trap vacuum system, likely due to the reflectivity of the 355 nm Nd:YAG line mirrors that form a periscope and direct the light into the vacuum system. The EOM does not alter the spectrum, and the leakage light through the device closely matches the input light. Note the sharp cutoff observed here is sufficient for AlH$^+$ rotational cooling and may be sharper than the spectrometer resolution.

system (Fig. 8.17). A minor difference is observed as the longer-wavelength edge of the spectrum is attenuated, likely due to the wavelength dependence of the three 355 nm Nd:YAG laser line mirrors used to direct the spectrally-filtered light to the ion trap.

Additionally, the EOM (Sec. 8.12) introduces dispersion into the the femtosecond beam (Fig. 8.17). Observing the output spectra of the spectral-filtering apparatus, no detrimental alterations to the output spectrum are observed when the EOM output is high, and the leakage light through the EOM when not enabled similarly shows no detrimental alterations.

The spectral filtering necessary for broadband rotational optical cooling (Sec. 7.4.1) and vibrationally-assisted broadband rotational optical cooling (Sec. 7.4.2) of AlH^+ is shown in Fig. 8.18. The spectral-filtering cutoff is sharper than observable in the commercial spectrometer data and is further investigated and quantified by YWL.

Using a similar femtosecond laser and spectral-filtering apparatus operating at a slightly different wavelength (385 nm), YWL uses two different analysis techniques to more precisely characterize the spectral-filtering resolution. First, a home-built spectrometer is created by inserting a line-CCD camera at the optimal mask position (Fourier plane). A CW laser, the SHG output of an acoustically-quiet Toptica DL-Pro and BoosTA MOPA system, calibrates the frequency reading and resolution of the spectrometer. Second, the narrowband laser is beat against the filtered broadband light on a photodetector. The lowest-frequency beat note occurs between the narrowband laser and the nearest comb tooth from the femtosecond laser. From this beat note intensity, the spectral density of the filtered broadband light at the narrowband reference laser's frequency is determined. The filtering behavior is then mapped as the cutoff is scanned across the reference. Both techniques yield a consistent result of 2 cm^{-1} filtering cutoff sharpness at 10 dB extinction.

The ultimately achievable spectral resolution is set by the diffraction limit; for the grating (3600 lines/mm) and beam waist (16 mm), this limit is 0.1 cm^{-1} at FWHM [**114**]. The effects preventing diffraction-limited spectral-cutoff resolution will be discussed in the dissertation by YWL [**235**], and are mostly due to optics aberrations in the spherical singlet lenses used. The spectral separation between rotationally heating

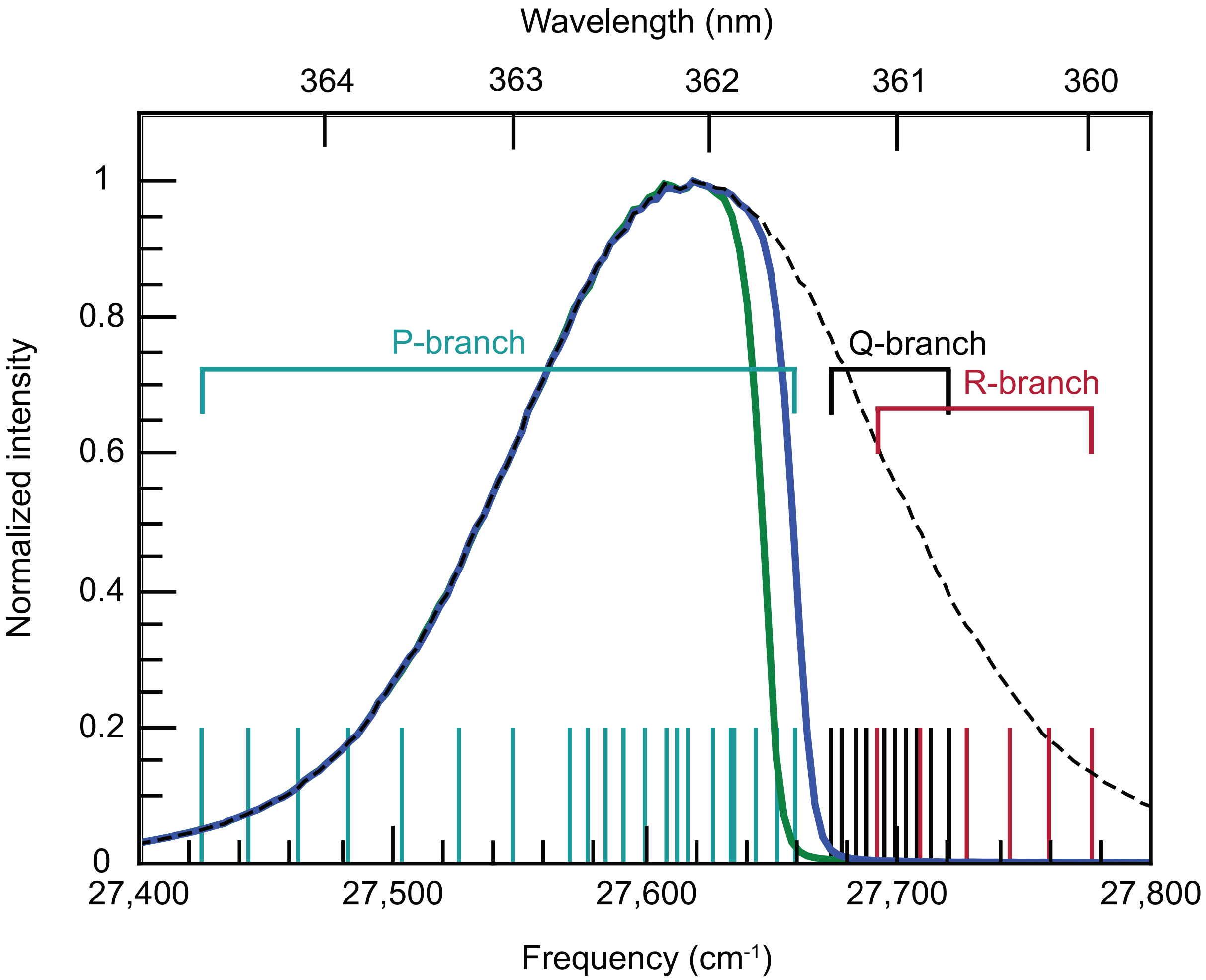


Figure 8.18. Stick spectrum of the AlH$^+$ $A^2\Pi_{1/2} \leftarrow X^2\Sigma^+$ transition, with the unfiltered femtosecond laser (dashed black), broadband rotational optical cooling (solid green), and vibrationally-assisted broadband rotational optical cooling (solid blue) spectra measured by the spectrometer. The actual spectral cutoff resolution is sharper than the commercial spectrometer data shown in the figure. Since the spectrometer absolute calibration is not reliable to the required level, both a mercury calibration lamp and rotational cooling performance is used to fine-tune the mask position.

and cooling transitions is approximately $2B_e$ ($4B_e$) for species with (without) allowed Q-branch transitions, where B_e is the rotational constant for the molecule. For AlH$^+$,

$2B_\mathrm{e} = 13\ \mathrm{cm}^{-1}$, so even the currently-achieved spectral-filtering resolution is sufficient for efficient cooling.

The spectrum of the femtosecond laser is actually a series of "comb teeth" spaced by the laser repetition rate of 80 MHz, so a mechanism is needed to guarantee that cooling transitions do not accidentally fall between the teeth. In our case, ion translational heating occurs for the radially-extended Coulomb crystal due to coupling to the RF trapping voltage [**93**], yielding sufficient Doppler broadening to ensure spectral overlap [**114**]. Alternately, frequency modulation of the cooling laser could be used to modify its tooth structure [**325**], eliminating the need for Doppler broadening.

8.12. Timing with EOM and shutters

All pulsed lasers used in the apparatus are shuttered. The 360 nm REMPD 1-photon dye laser, 266 nm REMPD 1′-photon laser, and spectral-filtering apparatus output are shuttered using ThorLabs SH1 or SH05 shutters with SC10 drivers controlled externally with TTL signals from LabVIEW. The timing delay for the spectral-filtering apparatus output light using a SH1 shutter is shown in Fig. 8.19, and the minimum reliable exposure time is ∼100 ms. This later is replaced with a SH05 shutter with a minimum reliable exposure time of ∼50 ms. The 532 nm ablation pulses for Ba^+ and Al^+ loading are shuttered using Magnet-Shultz of America 6683/1242 solenoids with sheet metal beam blocks bolted to the end. These are driven with 15 V from a TTL-enabled solid-state relay system controlled by LabVIEW.

Additionally, an EOM is placed after the output of the spectral-filtering apparatus to avoid possible complications of the dispersion introduced by the device on the SHG stage.

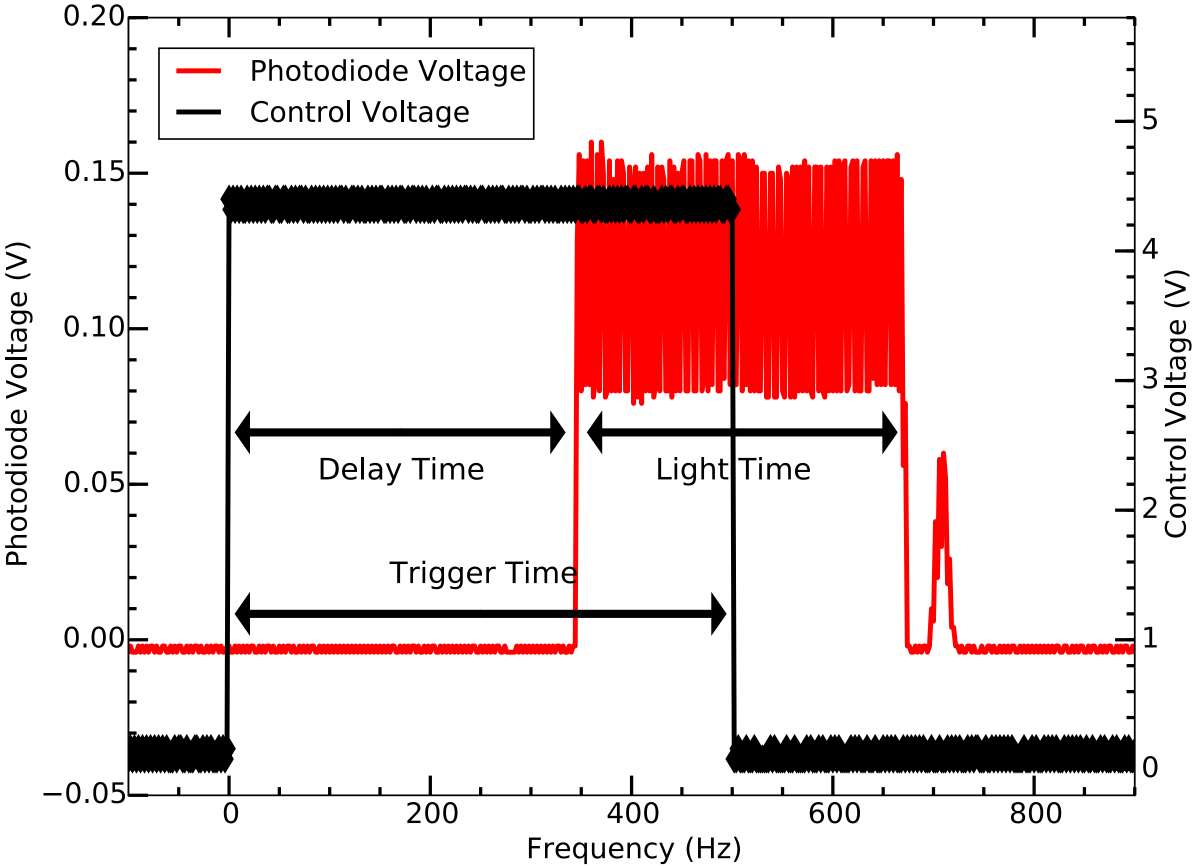


Figure 8.19. ThorLabs SH1 shutter timing on the spectrally-filtered pulsed laser output after EOM using a ThorLabs PDA1A amplified photodiode with 50 Ω termination. The TTL-to-light delay time is consistently 345 ms, and the most reliable shortest exposure time that the mechanical shutter can provide is 100 ms. The second small observed light exposure is due to a second, short TTL pulse from improper software programming.

The ConOptics 350-50-UV EOM connected to a 302A DC-to-1 MHz, 350 V amplifier receives an analog 0-2 V input signal. To convert the control-TTL logic to a variable, square-pulse analog signal, a circuit is used (App. A.2.3) that includes an enable and manual override switch on the front panel for user convenience. Note that the input and output loads on the BUF634 chip are carefully chosen to minimize the rise time and ringing in the circuit when switching.

Unfortunately, the extinction of the EOM is insufficient to prevent significant rotational cooling of AlH^+; ~500 μW of leakage light passes through the EOM as measured using the photocurrent of a ThorLabs SM05PD1A photodiode as the power is too low for the energy meter typically used to measure the Mai Tai output. Placing a ThorLabs GLB10 α-BBO polarizer before the EOM and irising all but the central part of the input beam provides only a modest improvement.

This leakage light is $\sim 10^{-5} I_s$ at the location of the ions, which is sufficient to drive broadband rotational optical cooling on the ~1 ms timescale. Since this timescale is significantly shorter than the mechanical shutter minimum open times, the EOM cannot be used to investigate the broadband rotational optical cooling time dynamics. Regardless, the EOM is kept in place and is turned on (off) a few ms before (after) the shutter opens (closes), acting as a variable laser power attenuator to prevent damaging the mechanical shutter.

All timing is performed with LabVIEW with the spectral-filtering apparatus output light shutter and EOM on one control and both REMPD shutters on another, each exposure triggered manually. Each program opens the shutters for a set time with the REMPD laser exposure sufficiently long to expose the ions to a few laser pulses to ensure complete dissociation of target-state molecules.

8.13. Experimental Procedure

In each experimental run, laser ablation is used to load laser-cooled Ba^+ ions along with typically 50 Al^+ ions; the Al^+ numbers are kept small to maintain sufficient time-of-flight mass spectrometry resolution due to the axial time-of-flight ejection [326]. AlH^+

forms on a one-minute timescale by reaction of Al^+ with background gas as the apparatus is operated with an intentionally degraded vacuum (3×10^{-10} Torr). Coulomb repulsion keeps ions several microns apart, preventing ion-ion chemical reactions and allowing the laser-cooled Ba^+ ions to sympathetically cool AlH^+ ions into a Coulomb crystal [27, 28, 29, 74, 30, 75, 76, 77, 31, 78, 79] with a translational temperature < 1 K. The large inter-ion separation and ultrahigh vacuum conditions result in decoupling of the AlH^+ translational, rotational, and vibrational temperatures. Optical rotational cooling could also be performed on translationally warmer samples, but crystallization is advantageous because it localizes the molecules, allowing for higher laser intensity and improving time-of-flight mass spectrometry resolution.

After loading the trap, AlH^+ rotations and vibrations equilibrate to room temperature via blackbody radiation. At room temperature, 99.9% of the AlH^+ population is in the lowest vibrational state $v = 0$, with significant population distributed among the first ten rotational levels, $N = 0 - 9$, and 4% in $N \geq 10$. The rotational population distribution is destructively probed by state-selective $(1 + 1')$ REMPD [121, 300] (Ch. 9), which converts AlH^+ only in a target rotational state into Al^+, and is analyzed by time-of-flight mass spectrometry. In each experimental iteration the REMPD probe is tuned to a single rotational level, and Al^+ and AlH^+ populations are counted.

CHAPTER 9

AlH$^+$ Rotational State Readout

The AlH$^+$ rotational population distribution is destructively probed [327, 75] by state-selective $(1 + 1')$ REMPD [121, 10, 11, 300], and is discussed in this chapter. We perform *ab initio* calculations relevant to predict the cross section of an experimentally-accessible $(1 + 1')$ REMPD pathway in AlH$^+$. Experimenting on AlH$^+$ ions held in a linear ion trap, we confirm dissociation via this channel with analysis performed using time-of-flight mass spectrometry, demonstrate the use of REMPD for rotational state analysis, and measure the rotational distribution of trapped AlH$^+$ to be consistent with the expected thermal distribution.

9.1. State Readout via REMPD

9.1.1. *ab initio* Calculational Details

Potential energy surfaces and transition dipole moments for AlH$^+$ are computed with the multireference singles and doubles configuration interaction (MRSDCI) method. All computations are performed with the MOLPRO quantum chemistry package [328]. The molecular orbitals used in the MRSDCI computations are obtained from the state-averaged complete active space self-consistent field (SA-CASSCF) method [329, 330]. The SA-CASSCF orbitals are optimized with an active space consisting of three electrons in the

$4\sigma 2\pi_x 2\pi_y 5\sigma 6\sigma$ orbitals while averaging over the lowest four electronic states. In the MRSDCI wave function, the 1s Al orbital is constrained to always be doubly occupied. The aug-cc-pCVQZ basis set is used for all computations [331, 332].

In a previous study [113], we found the results obtained from MRSDCI to be in excellent agreement with approximate coupled-cluster singles, doubles and triples (CC3) [333, 334] as well as the experimentally-determined bond length and vibrational frequency of AlH^+ [118]. In the present work, the $C^2\Sigma^+$ and $D^2\Pi$ states could not be studied with CC3, since current implementations are not fully spin-adapted (*i.e.* CC3 wave functions are not proper spin eigenfunctions). This led to spurious mixing of the two higher-lying doublet states with quartet states. For this reason, the present work relies on MRSDCI computations exclusively. A drawback of this approach is that the energy spacings at dissociation do not match experimental results. In order to fix the energy spacings, constant shifts were applied so that, at dissociation, the gap between the $X^2\Sigma^+$ and $A^2\Pi//B^2\Sigma^+$ is 266.9157 nm and $X^2\Sigma^+$ and $C^2\Sigma^+/D^2\Pi$ states is 167.0787 nm [335]. We expect that the nonparallelity errors of the MRSDCI method are small, and the application of a small constant shift is reasonable. Calculated potential energy surfaces and transition dipole moment functions are shown in Fig. 9.1 and Fig. 9.2, respectively

9.1.2. Photodissociation Cross Section Calculational Details

Photodissociation cross sections are calculated using the program BCONT 2.2 [283], using as input our calculated surfaces and transition moments. In Fig. 9.3, the experimentally relevant photodissociation cross sections from the $A^2\Pi$, $v = 0$ state are plotted as a function of photon wavelength. Experimentally, as shown in Fig. 9.1, 266 nm light is used

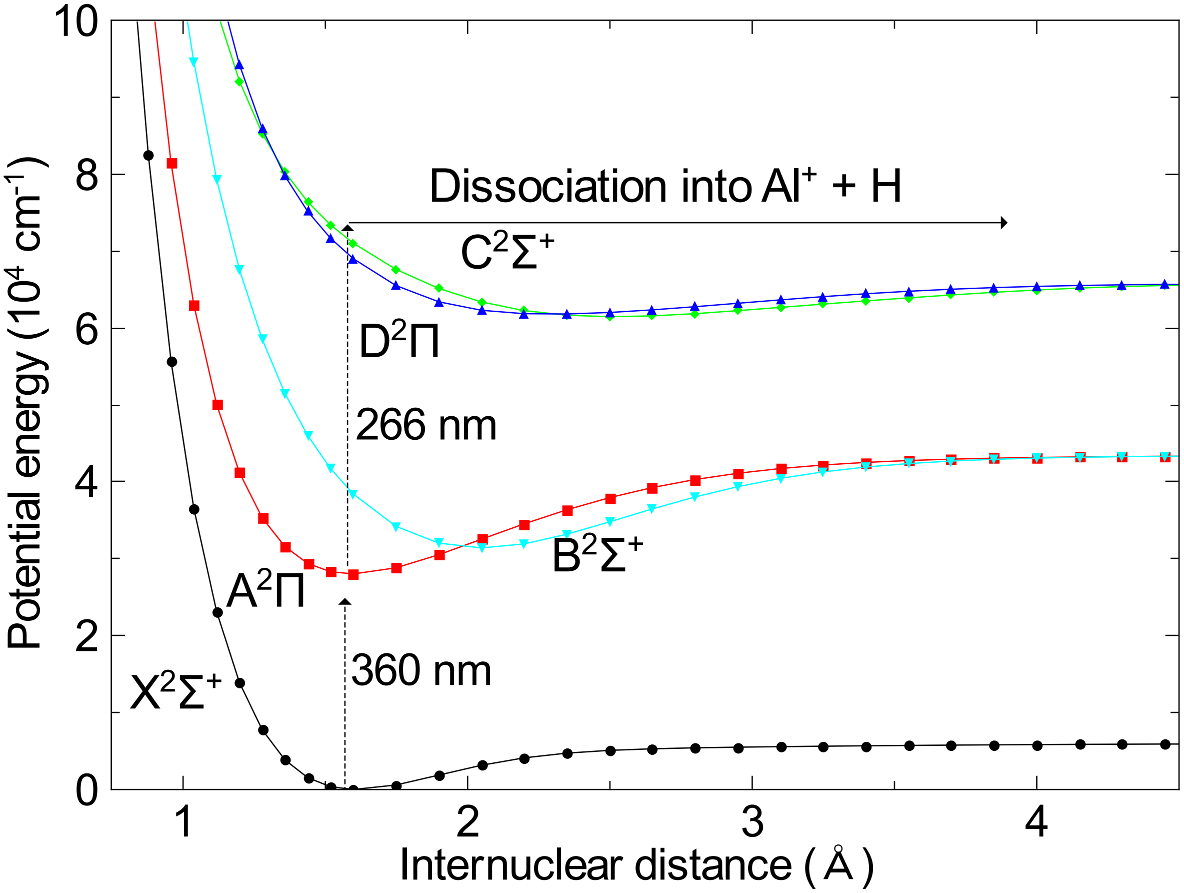


Figure 9.1. Calculated potential energy surfaces of the lowest several electronic states of AlH$^+$. The (1+1′) REMPD pathway used in the experiment is also shown.

for the dissociating photon. At this wavelength the cross section is dominated by the D$^2\Pi \leftarrow A^2\Pi$, $v = 0$ transition and is calculated to be $\sigma_0 = 6 \times 10^{-18}$ cm^2.

It could also be convenient experimentally to use 532 nm light to dissociate from the A$^2\Pi$ state to the B$^2\Sigma^+$ continuum. However, even though this cross section is near its maximum at this wavelength, the calculated cross section of 3×10^{-22} cm^2 is inconveniently

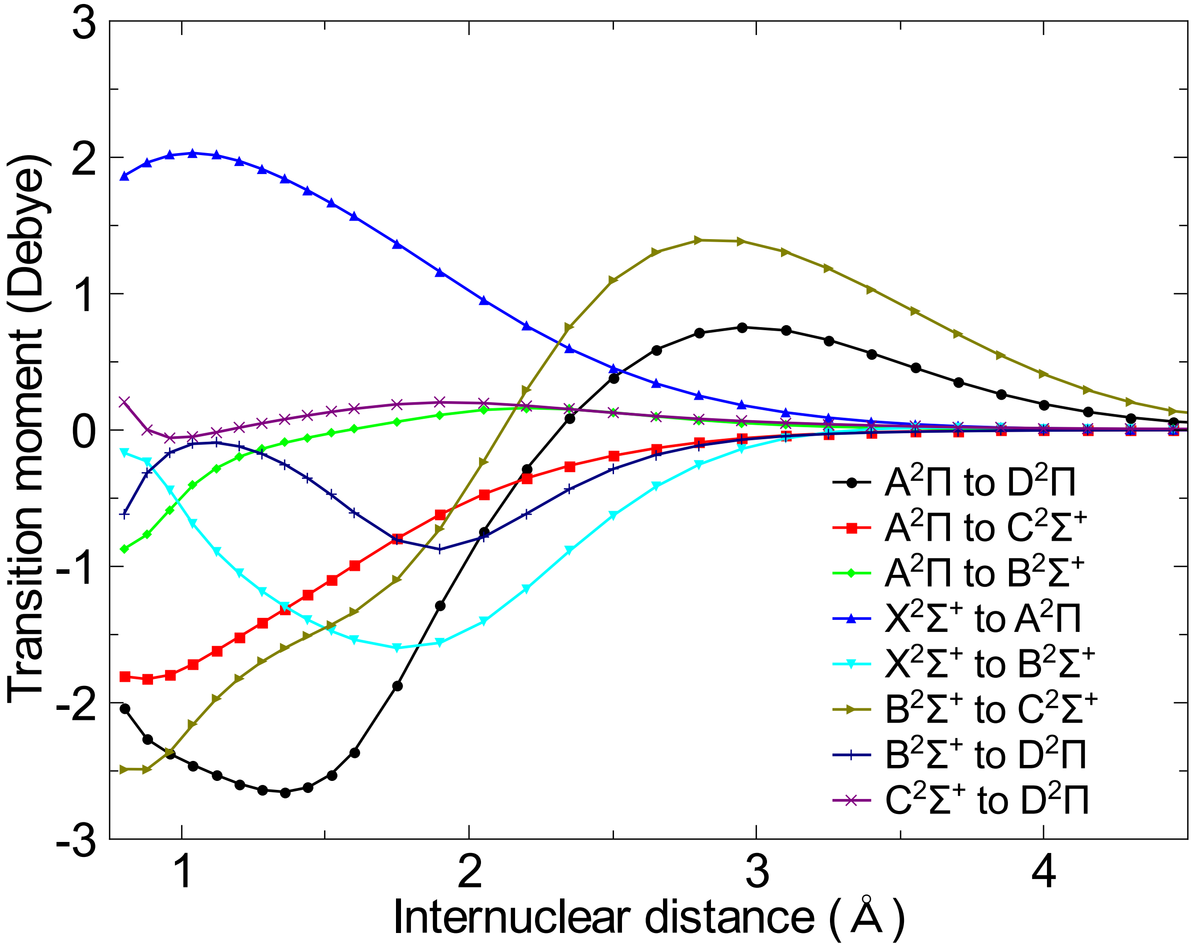


Figure 9.2. Calculated transition dipole moment functions of selected electronic transitions in AlH^+.

small. Another experimental consideration is whether the 266 nm light might drive bound-bound transitions from $X^2\Sigma^+$ to $A^2\Pi$ or $B^2\Sigma^+$. Although such processes are energetically allowed, the vibrational overlap is expected to be small, and we see no experimental evidence for such transitions.

Rotational state analysis is accomplished destructively by $(1 + 1')$ REMPD with the two photon energies shown in Fig. 9.1. In order for dissociation to proceed, the first photon must be resonant with an $A^2\Pi \leftarrow X^2\Sigma^+$ bound-bound transition, and the target

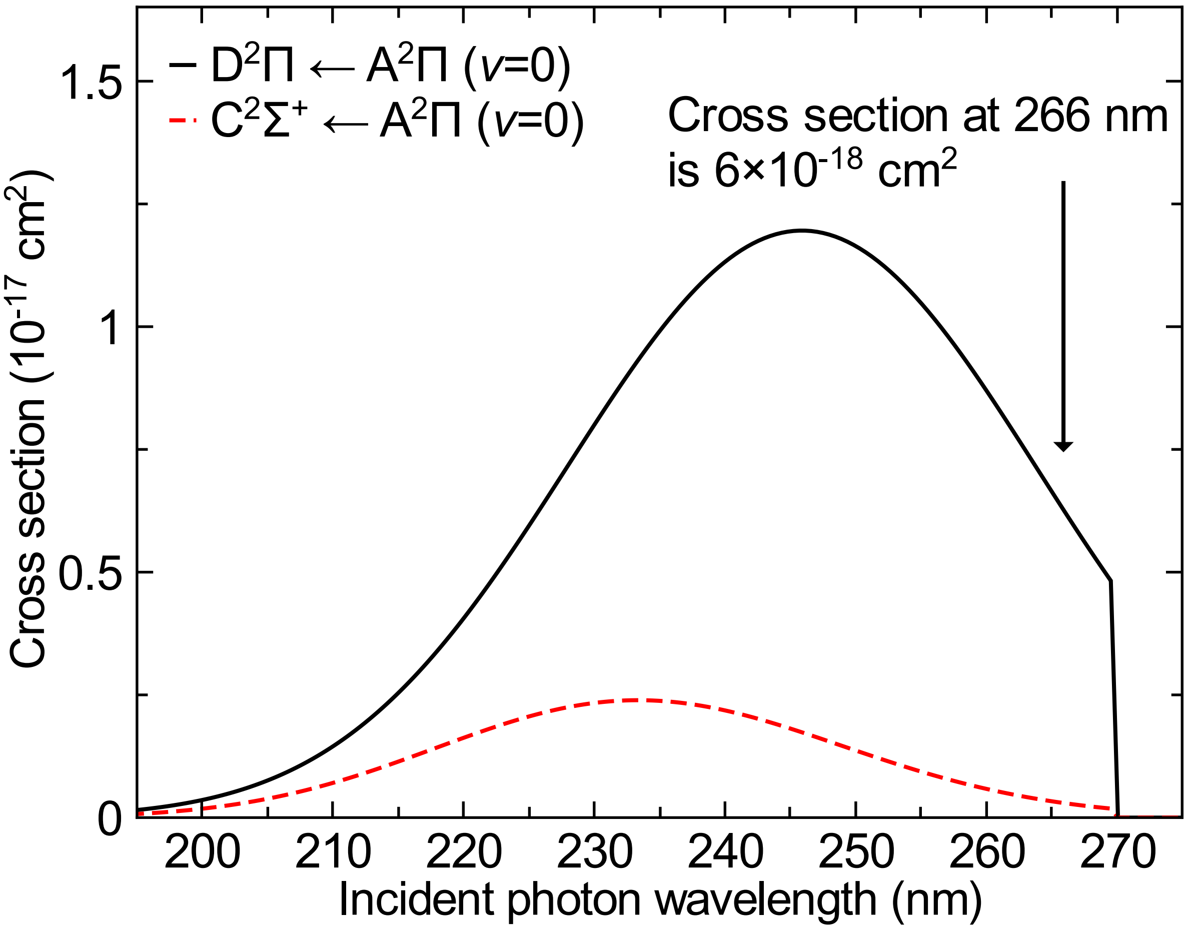


Figure 9.3. Calculated photodissociation cross sections from the AlH^+ $\mathrm{A}^2\Pi$, $v = 0$ state. The sharp cutoff near 270 nm corresponds to a photon energy insufficient to excite to the continuum.

AlH^+ ion must be in the addressed rotational state. Since the kinetic energy released in dissociation is small compared with the trap depth, successful dissociation appears as conversion of AlH^+ to trapped Al^+. Thus, for fully efficient REMPD, the resulting ratio of Al^+ to AlH^+ measures the fraction of AlH^+ initially in the probed rotational state.

In our experiment, we tune the 360 nm laser to different lines in the Q-branch of the AlH^+ $\mathrm{A}^2\Pi \leftarrow \mathrm{X}^2\Sigma^+$ transition. Transitions originating from different $\mathrm{X}^2\Sigma^+$-state rotational levels are separated from each other by at least 160 GHz, so they are well resolved above the natural linewidth (typically 1 MHz) and the linewidth of the 360 nm laser (1.8 GHz). We reduce the laser intensity sufficiently to ensure that power broadening effects do not complicate the rotational state analysis. The time-of-flight mass spectrometry sequence is triggered quickly following the REMPD pulses, much faster than the timescale of Al^+ reaction with background gas.

The REMPD 360 nm average pulse energy is 50 μJ with a spot size of 3.0 mm × 1.5 mm FWHM at trap center, providing sufficient intensity for AlH^+ ions to undergo 100 Rabi oscillations over a single pulse duration. The 266 nm photon source is synchronized and counterpropagating with the dye laser with a pulse energy of 2.2 mJ and a spot size of 1000 μm × 270 μm FWHM at trap center. The two REMPD pulses are fully overlapped spatially and temporally at trap center. Beam characterization is performed using a ThorLabs DCC1545M CMOS camera.

We use REMPD/time-of-flight mass spectrometry to map the population distribution of trapped AlH^+ in the first few rotational states (Fig. 9.4) to verify our REMPD and time-of-flight mass spectrometry analysis technique. Data analysis is described in Ch. 10. Since the AlH^+ production time is much longer than equilibration times with room temperature blackbody radiation, we expect a thermal distribution. For those molecules in the probed rotational state, the calculated cross section yields a 98% single-shot AlH^+ dissociation probability for our pulse parameters; in our current implementation we apply 10 REMPD pulses per experiment. Several experimental runs probing a given rotational state are integrated to create the figure. The agreement is good between our measured rotational

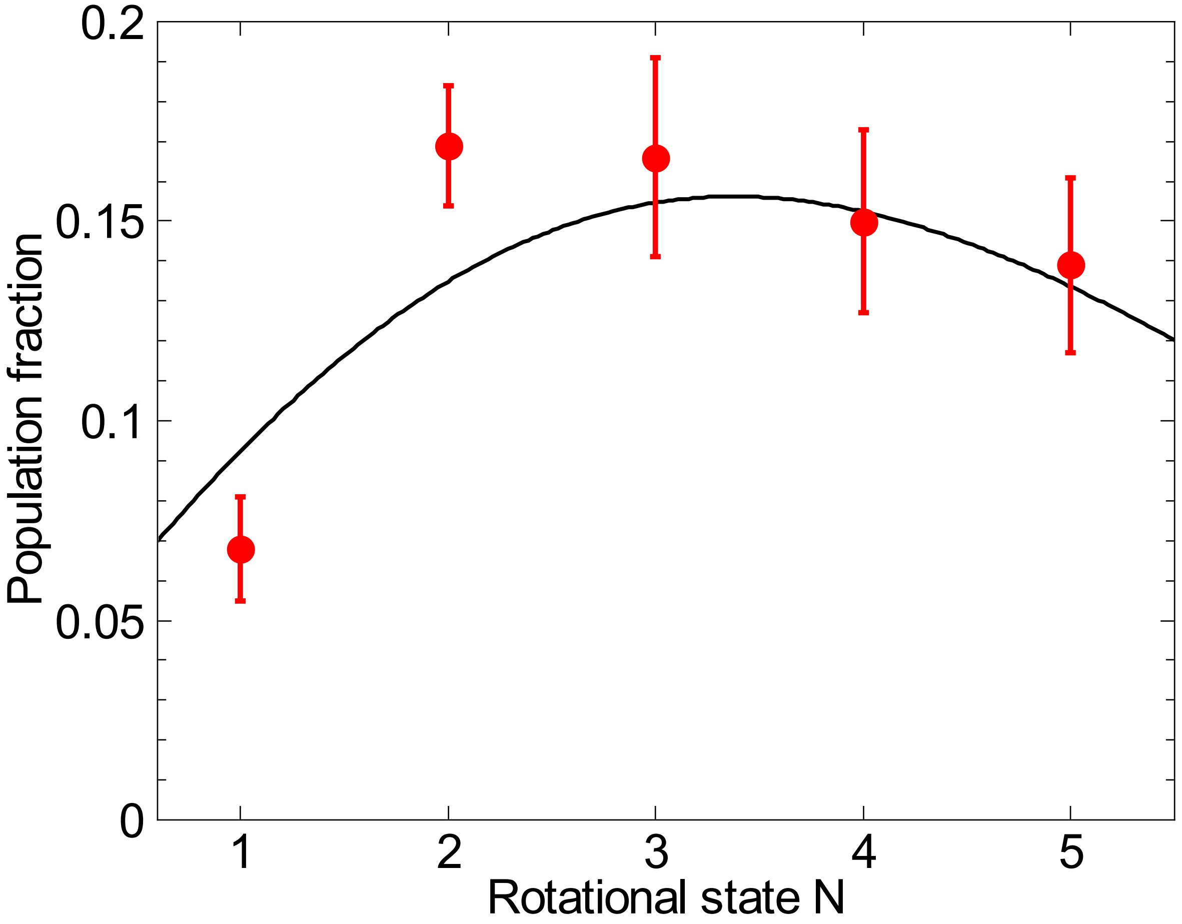


Figure 9.4. Thermal population distribution measured with REMPD and time-of-flight mass spectrometry to verify the rotational state distribution measurement technique. The black line is the expected thermal distribution at 293 K, with no free parameters. The red data points are the measured population in each rotational state with uncertainties from counting statistics. Data analysis was performed as described in Ch. 10.

distribution and the expected room temperature distribution, validating our rotational distribution measurement and data analysis techniques.

CHAPTER 10

AlH$^+$ Rotational Cooling Results and Data Analysis

After MCP signal collection on the oscilloscope, the data is processed to yield total ion numbers and the Al$^+$ population fraction. These Al$^+$ ions correspond to the number of AlH$^+$ ions that were destroyed via REMPD in the targeted rotational state. We do not attempt to independently analyze the number of AlH$^+$ loaded into the trap, but rather we determine the degree of rotational cooling by the time-of-flight mass spectrometry measurements of the Al$^+$ to AlH$^+$ ratio after REMPD. This chapter details how the data is processed, the quantification of the integrated charge per ion, and the rotational cooling results. The chapter concludes discussing the Monte Carlo model used for uncertainty analysis and determining temperature and loss limits.

10.1. Numerical MCP Charge Integration

To quantify the number of ions detected by the MCP in each experimental run, the time-of-flight mass spectrometry oscilloscope trace is analyzed by the code in App. D.2.4. The code first extracts the time and MCP data from the oscilloscope binary file using a debugged version of code supplied by Agilent (App. D.2.4.1). The code then divides the dataset into discrete subsets: (1) the first background window from 24 μs to 2 μs before the Al$^+$ widow; (2) the Al$^+$ signal window, which is 2 μs long with an end time dictated by the user-specified Al$^+$ and AlH$^+$ discriminator time t_0, typically 31 μs; (3) the AlH$^+$

signal window, which is also 2 μs long with a start time dictated by the user-specified t_0; and (4) the second background window from the end of the AlH$^+$ signal window to 38 μs.

The DC background, calculated as the mean signal from both background window subsets, does not vary over experimental repetitions significantly unless a significant number of signals are observed within one of these windows; a single or a few signal events within these background windows are averaged out and do not significantly change the calculated background level. The DC background is then subtracted from the Al$^+$ and AlH$^+$ signal windows.

A discriminator level of -10 mV is set to eliminate low-level background noise. If the MCP signal is smaller than the -10 mV discriminator, the signal value is set to zero. If the MCP signal is larger than the -10 mV discriminator, the value is passed through. Each signal window is numerically integrated to yield the total charge. See Fig. 10.1 for a typical data set before and after signal processing.

Since the time-of-flight mass spectrometry timing varies over long timescales with patch-charging of electrodes, particularly from ablating aluminum with high ablation power, every experimental run is preceded by a time-of-flight mass spectrometry run with roughly 50% of each species present in order to optimally set t_0. Data collection is typically performed by randomizing thermal, rotational cooling, and calibration runs.

10.2. Charge Analysis

To quantify the integrated charge per ion event, the individual signal peaks in the time-of-flight mass spectrometry data within the Al$^+$ and AlH$^+$ time windows are integrated from the zero value before the signal to the next zero value after the signal to include any

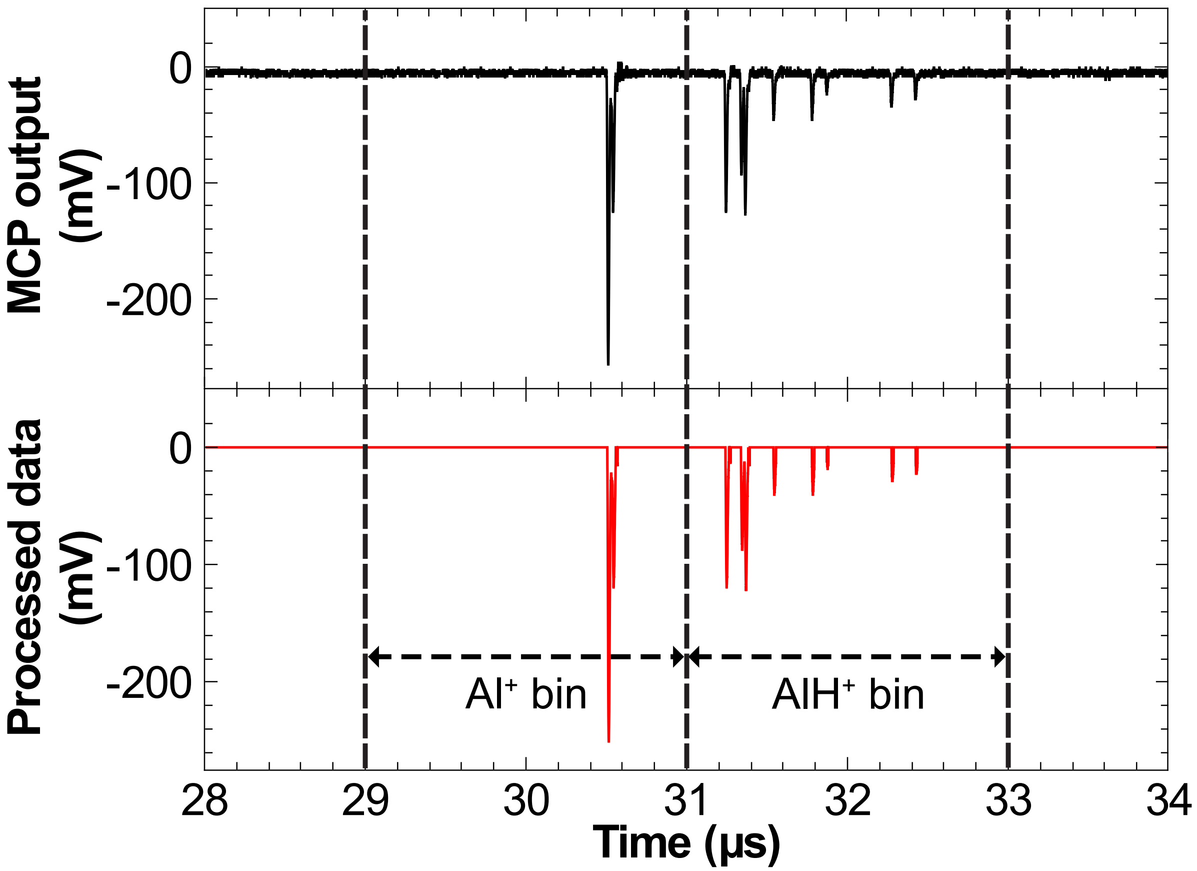


Figure 10.1. Sample time of flight mass spectrometry spectrum. The upper plot (black) is the MCP output measured by the oscilloscope. The lower plot (red) is the processed data with DC background and low-level noise removed. For this spectrum, the time of flight mass spectrometer measures 13 ± 4 Al^+ and 14 ± 4 AlH^+ ions. The single-species timing and the inter-species separation agree well with the values from a simulation. In spectra with a large Al^+ fraction, as in the figure, we find that the timing dispersion of AlH^+ is increased compared to pure Al^+ or pure AlH^+ samples, likely due to Coulomb repulsion from the faster (i.e. lighter) species adversely affecting the timing of the slower (i.e. heavier) species in a mixed sample.

events that contain more than a single ion using the code in App. D.2.4.3. Because the detector is operated in the analog-mode, multiple events can pile up as seen in Fig. 10.1, but the signal integration will account for this.

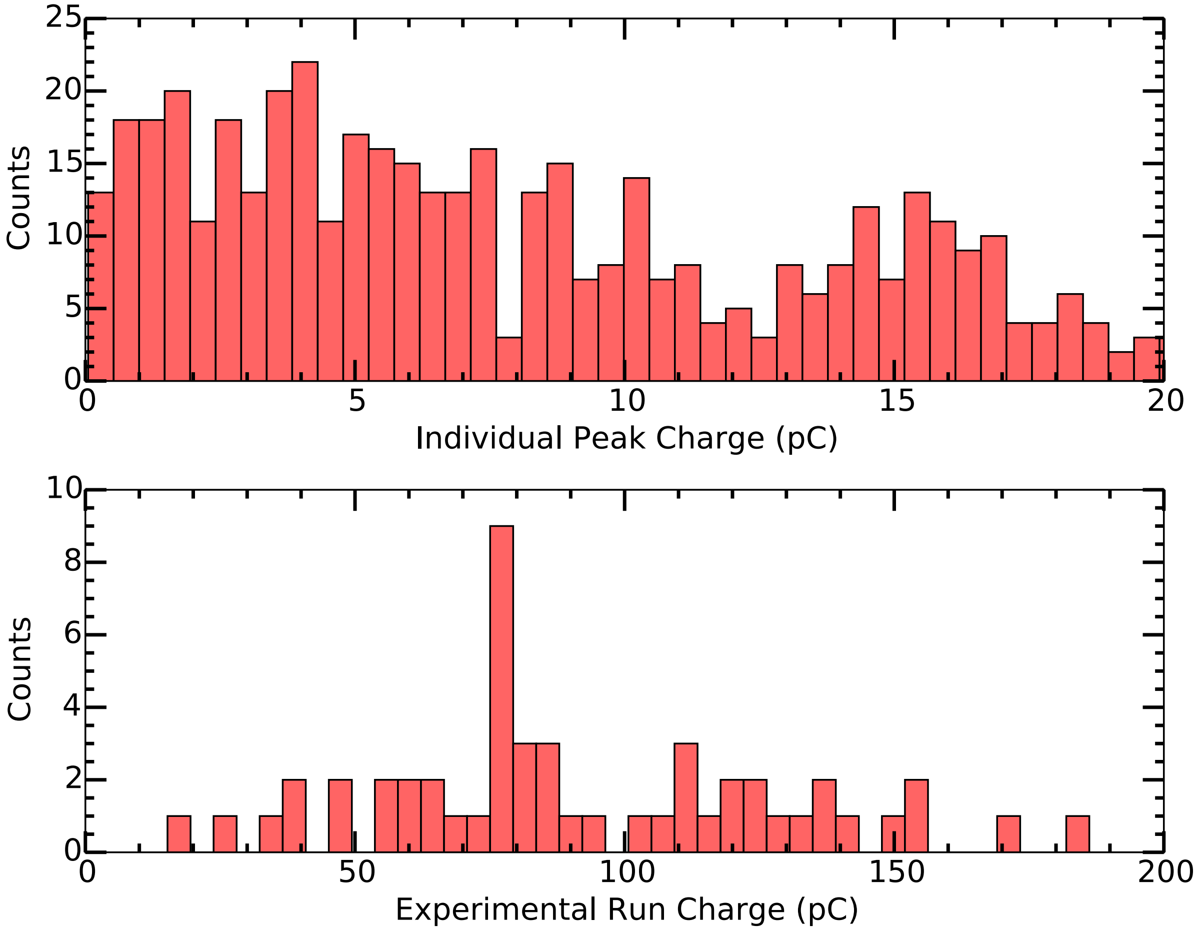


Figure 10.2. Histogram of (top) individual signal peaks before applying the -0.5 kV anode bias in the Al^+ and AlH $^+$ time windows and (bottom) the total charge per experimental run in the Al^+ and AlH $^+$ time windows. From this data, we estimate the charge per ion is ∼2.67 pC with a typical experimental run accumulating ∼80 pC of charge corresponding to ∼30 ions.

Before the MCP anode bias of -0.5 kV is applied, the single-ion charge is estimated to be 2.67 pC per ion, although the charge per ion varies significantly as seen in the histogram data in Fig. 10.2. The total charge per experimental run is typically ∼80 pC, corresponding to ∼30 ions.

After the MCP anode bias of -0.5 kV is applied, the single-ion charge is estimated to be 8 pC per ion, and the charge per ion is clearly resolvable as seen in the histogram data in Fig. 10.3. This result is consistent with the MCP gain specifications when operating with a potential difference of -2.5 kV with -0.5 kV anode bias. The total charge per experimental run is typically ∼360 pC, corresponding to ∼45 ions.

Note that when calculating the Al^+ ion fraction as the number of Al^+ ions to number of total Al^+ and AlH^+ ions, the charge per ion term drops out of the fraction, and only the integrated charge data is relevant. Therefore, the charge per ion is not critical in discussing the experimental results below.

10.3. Experimental Results

The experimental results for the measured thermal rotational distribution as well as rotational distributions after exposure to broadband rotational optical cooling and vibrationally-assisted broadband vibrational optical cooling light are discussed below. Additionally, the time dependence data of the vibrationally-assisted broadband rotational optical cooling is presented.

10.3.1. Thermal Population

The thermal population data, taken two minutes after ablation-loading of Al^+ is completed for 100% reaction to AlH^+ and to assure thermalization, is listed in Table 10.1 and plotted in Fig. 10.4. The expected populations are given by a 293 K Maxwell-Boltzmann distribution using molecular constants for AlH^+ times a degeneracy factor (App. D.2.3).

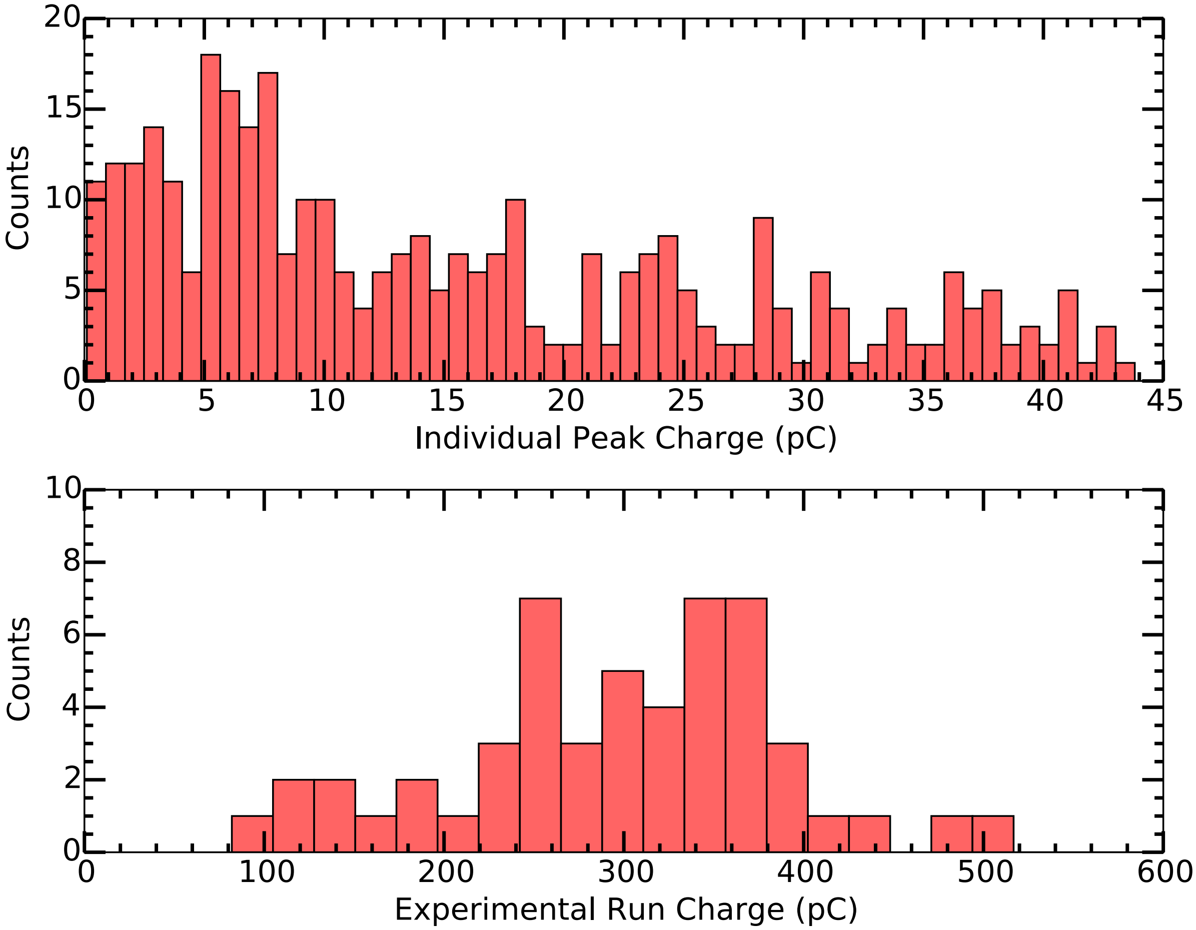


Figure 10.3. Histogram of (top) individual signal peaks after applying the -0.5 kV anode bias in the Al^+ and AlH $^+$ time windows and (bottom) the total charge per experimental run in the Al^+ and AlH $^+$ time windows. From this data, we estimate the charge per ion is ∼8 pC by the repeating histogram clusters at integer multiples of this value. A typical experimental run accumulates ∼360 pC of charge corresponding to ∼45 ions.

The experimental data is consistent with the expected thermal distribution, validating our data analysis techniques.

Table 10.1. Processed data of the thermal AlH$^+$ rotational population distribution before broadband rotational optical cooling. Expected population is given by a 293 K Maxwell-Boltzmann distribution times a degeneracy factor. The experimental data is consistent with the expected thermal distribution, validating our data analysis techniques.

N	Al$^+$	AlH$^+$	$\frac{\text{Al}^+}{\text{Al}^+ + \text{AlH}^+}$	Expected	σ Difference
0	19.6(4.5)	329.7(18.2)	0.056(13)	0.03293	1.85
1	27.8(5.4)	378.2(19.5)	0.068(13)	0.09243	1.89
2	99.3(10.0)	487.3(22.1)	0.169(16)	0.13487	2.21
3	36.6(6.1)	184.0(13.6)	0.166(25)	0.15467	0.44
4	34.8(6.0)	198.0(14.1)	0.150(24)	0.15241	0.12
5	33.1(5.8)	205.8(14.4)	0.139(23)	0.13359	0.23
6	39.0(6.3)	320.2(17.9)	0.108(17)	0.10594	0.15
7	26.9(5.3)	320.6(17.9)	0.077(15)	0.07675	0.05

10.3.2. Broadband Rotational Optical Cooling Data

The rotational state distribution data after five seconds of broadband rotational optical cooling exposure is listed in Table 10.2 and plotted in Fig. 10.4. The expected populations are given by the results of rate equation simulations (Sec. 7.4).

Broadband rotational optical cooling cools each parity independently and yields 48(4)% population in $\left|X^2\Sigma^+, v=0, N=0\right\rangle$ and 46(3)% in $\left|X^2\Sigma^+, v=0, N=1\right\rangle$. These data are statistically consistent with fully efficient broadband rotational optical cooling as 94(5)% of the population is in the lowest rotational state of each parity. Additionally, REMPD probes find no population in higher rotational states. Table 10.2 and Fig. 10.4 do include data from an anomalous experimental run yielding a non-zero $N=3$ population; however, as discussed below, this signal is most likely from detector noise.

10.3.2.1. Anomalous Data Point. In one REMPD time-of-flight mass spectrometry probe of the $N=3$ population after broadband rotational optical cooling, for which there

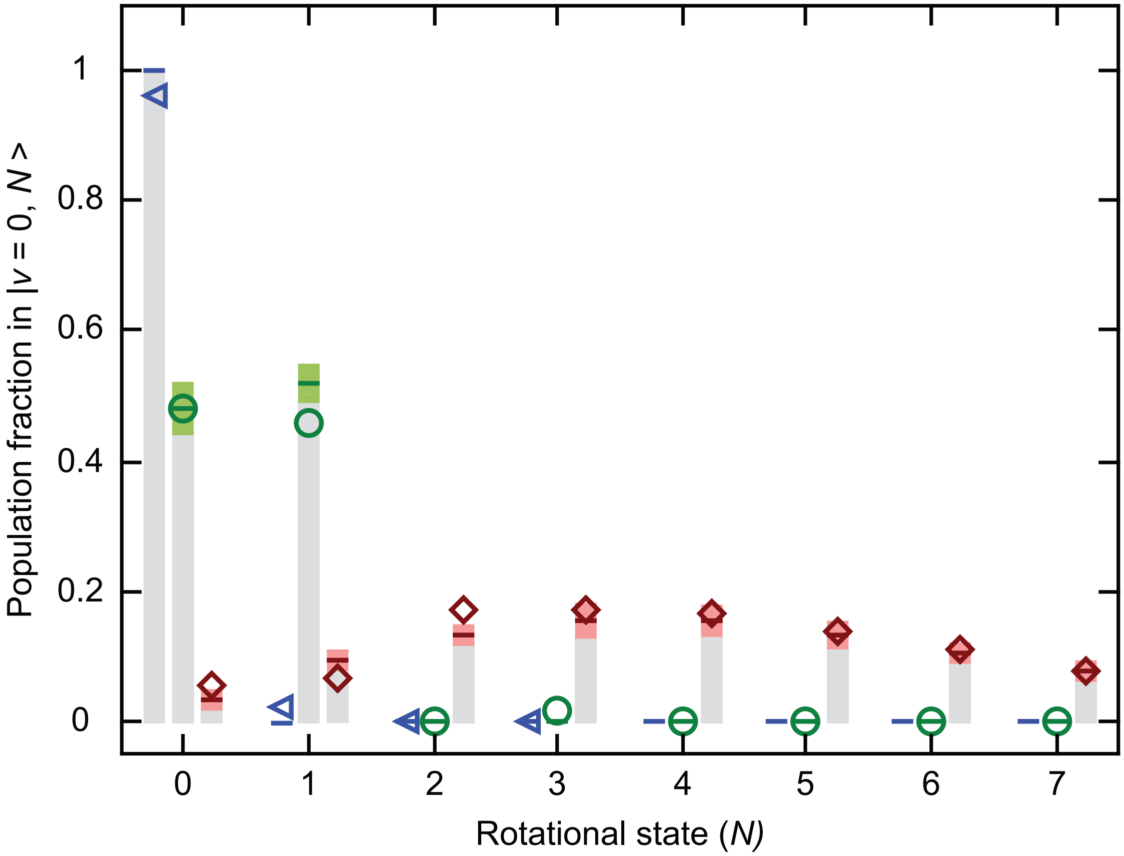


Figure 10.4. Measured rotational populations for the initial thermal distribution (red diamonds), and after 5 s of cooling with spectral filtering in the broadband rotational optical cooling (green circles) and vibrationally-assisted broadband rotational optical cooling (blue triangles) configurations. Expected thermal or simulated populations are indicated by horizontal lines with colored bands denoting the 68% confidence regions for statistical uncertainties for the experimental sample size as determined by simulation. Estimated systematic uncertainties are negligibly small in comparison. Gray vertical bars are drawn to guide the eye.

should be no Al^+ signal if broadband rotational optical cooling works perfectly, we observe a structureless peak in the Al^+ bin of the time-of-flight spectrum (Fig. 10.5). The peak has a total charge of 14.8 pC. As this data collection occurred *before* the -0.5 kV anode bias is applied, this signal is consistent with a 5.5(2.6)-ion event. If we consider the case

Table 10.2. Processed data of AlH^+ rotational population distribution after 5 seconds of broadband rotational optical cooling. Note that the uncertainties where zero ions are measured is set to 1.0. The observed signal in $N = 3$ is due to a single, structureless peak and is likely detector noise (Sec. 10.3.2.1). Expected population is given by rate equation simulations (Sec. 7.4).

N	Al^+	AlH^+	$\frac{Al^+}{Al^+ + AlH^+}$	Expected	σ Difference
0	63.8(8.0)	68.2(8.3)	0.48(4)	0.483	0.01
1	145.4(12.1)	173.5(13.2)	0.46(3)	0.517	2.18
2	0.0(1.0)	82.0(9.1)	0.00(1)	0.00	0.00
3	5.5(2.6)	290.7(17.1)	0.019(9)	0.00	2.19
4	0.0(1.0)	138.4(11.8)	0.00(1)	0.00	0.00
5	0.0(1.0)	166.0(12.9)	0.00(1)	0.00	0.00
6	0.0(1.0)	273.9(16.6)	0.00(1)	0.00	0.00
7	0.0(1.0)	100.0(10.0)	0.00(1)	0.00	0.00

in which the charge per ion is ~8 pC, this single, structureless peak is consistent with a 1.8(1.4)-ion Al^+ signal. Multiple-ion, structureless peaks occur in our time-of-flight mass spectrometry data both with and without structure; the peak is also consistent with structureless detector noise spikes observed occasionally at random times. Given our overall sample size and typical time-of-flight timing distribution, it is highly unlikely that the only observed signal from a residual population in $N \geq 2$ occurs as a single, multiple-ion event rather than as a small number of single-ion events. Nonetheless, we take the conservative approach of treating this point as $N = 3$ population resulting from imperfect broadband rotational optical cooling; as such it is included in Fig. 10.4 and in all analysis, except where otherwise noted.

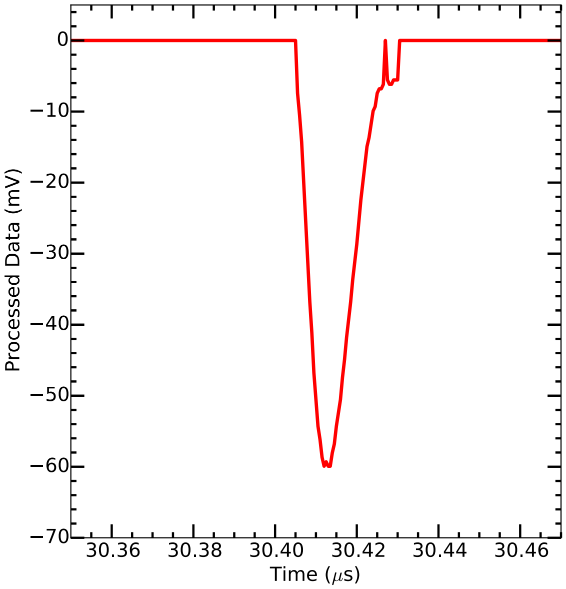


Figure 10.5. Full (right) and zoom-in (left) time-of-flight spectrum 2569.bin from 20131029 with the structureless, multiple-ion peak in the Al^+ time bin. Given our small sample size and typical timing distribution, it is highly likely that this peak is due to an MCP noise event. For this spectrum, t_0 = 30.80 μs (dashed vertical line) and the discriminator level is -5 mV.

10.3.3. Vibrationally-assisted Broadband Rotational Optical Cooling Data

The rotational state distribution data after five seconds of vibrationally-assisted broadband rotational optical cooling exposure is listed in Table 10.3 and plotted in Fig. 10.4. The expected populations are given by the results of rate equation simulations (Sec. 7.4).

After vibrationally-assisted broadband rotational optical cooling, the population measured in $\left|\mathrm{X}^2\Sigma^+, v = 0, N = 0\right\rangle$ is $95.4^{+1.3}_{-2.1}\%$; probing $\left|\mathrm{X}^2\Sigma^+, v = 0, N = 1\right\rangle$, we measure a population of $2.0^{+3.3}_{-1.4}\%$, and we observe no population in $N = 2$ or $N = 3$. These results are consistent with all population after vibrationally-assisted broadband rotational optical cooling being in the two lowest rotational states, with the population ratio corresponding

Table 10.3. Processed data of AlH$^+$ rotational population distribution after 5 seconds of vibrationally-assisted broadband rotational optical cooling. Note that the uncertainties where zero ions are measured is set to 1.0. Expected population is given by rate equation simulations (Sec. 7.4).

N	Al$^+$	AlH$^+$	$\frac{\text{Al}^+}{\text{Al}^+ + \text{AlH}^+}$	Expected	σ Difference
0	104.1(10.3)	4.7(2.4)	0.96(2)	1.00	2.02
1	1.1(1.5)	48.5(7.0)	0.02(3)	0.00	0.78
2	0.0(1.0)	10.8(3.4)	0.00(9)	0.00	0.00
3	0.0(1.0)	15.9(4.1)	0.00(6)	0.00	0.00

Table 10.4. Processed data of AlH$^+$ rotational population distribution after an exposure time of vibrationally-assisted broadband rotational optical cooling light.

Time (ms)	Al$^+$	AlH$^+$	$\frac{\text{Al}^+}{\text{Al}^+ + \text{AlH}^+}$
5000	104.1(10.3)	4.7(2.4)	0.96(2)
500	22.9(4.9)	2.1(1.8)	0.92(7)
200	41.0(6.5)	4.1(2.3)	0.91(5)
100	47.7(7.0)	14.9(4.0)	0.76(6)
50	16.6(4.2)	6.8(2.8)	0.71(10)

to a rotational temperature of $3.8^{+0.9}_{-0.3}$ K with temperature and uncertainties determined using a Monte Carlo analysis method (Sec. 10.5.4.2).

10.4. Rotational Ground State Preparation Timescale

The $N = 0$ rotational state population fraction as a function of vibrationally-assisted broadband rotational optical cooling exposure time is listed in Table 10.4 and plotted in Fig. 10.6.

Simulations (Sec. 7.4) predict that the broadband rotational optical cooling phase of cooling occurs as fast as 8 μs with our current laser intensity. However, measurements of the rotational state population dynamics are limited by the 10 ms speed of the mechanical

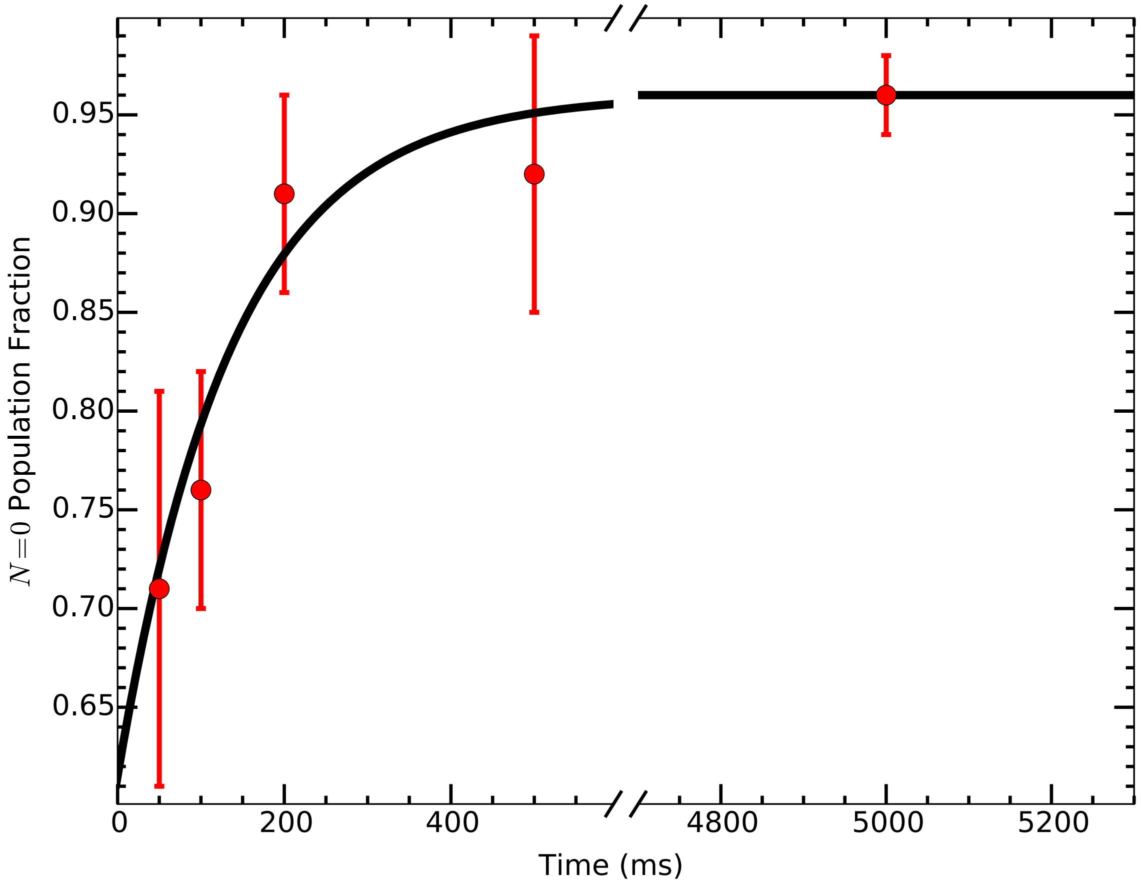


Figure 10.6. Measured population fraction in $N = 0$ versus duration of vibrationally-assisted broadband rotational optical cooling illumination. A single-parameter fit (black line) to the data (red circles) yields a vibrational spontaneous emission lifetime of $\tau = 140(20)$ ms. Vertical bars denote the $\pm 1\sigma$ statistical confidence intervals.

shutter gating the broadband rotational optical cooling light (Sec. 8.12). Nonetheless, this timing resolution allows us to analyze the slower of the two vibrationally-assisted broadband rotational optical cooling phases associated with the vibrational spontaneous relaxation event (Sec. 7.4.2). Fig. 10.6 shows data taken for various cooling times with the population measurement occurring 1 s (several vibrational-relaxation lifetimes) later.

Similarly to the approach used in Ref. [336], the vibrational lifetime τ can be extracted from the time-dependent $N = 0$ population.

The detected population P_0^d in $N = 0$ is expected to obey

$$P_0^d = P_0^f - P_1^i B_2 e^{-T/\tau} \tag{10.1}$$

where P_0^f is the measured asymptotic population in $N = 0$, P_1^i is the initial population in $N = 1$ following the faster broadband rotational optical cooling phase, taken here from simulation, $B_2 \approx 2/3$ is the calculated branching fraction for spontaneous emission from $\left|\mathrm{X}^2\Sigma^+, v = 1, N = 1\right\rangle$ to $\left|\mathrm{X}^2\Sigma^+, v = 0, N = 2\right\rangle$, $1/\tau$ is the overall vibrational spontaneous relaxation rate from $v = 1$, and T is the duration of the vibrationally-assisted broadband rotational optical cooling light exposure. Note that the $N = 0$ population at the shortest times probed is expected to be elevated from the ∼50% from broadband rotational optical cooling because vibrationally-assisted broadband rotational optical cooling quickly pumps population from $|v = 0, N = 1\rangle$ into $|v = 1, N = 0\rangle$, which then relaxes to either $|v = 0, N = 0\rangle$ or $|v = 0, N = 2\rangle$ before analysis.

The fit above yields $\tau = 140(20)$ ms, in good agreement with the theory value of 127 ms [113]. Additionally, the fit value is exactly 2σ from the updated theory value calculated in Sec. 7.4.2.1. Blackbody-radiation induced rotational and vibrational transitions from low-lying rotational states occur on ∼100 s timescale, so these processes do not significantly alter the cooling efficiency or analysis.

10.5. Monte-Carlo Analysis

A "toy" Monte Carlo technique is used to perform statistical error analysis to avoid binomial confidence interval approximations [**337**, **338**] outside of the central limit approximation. The method determines statistical properties by simulating repetition of a given experiment a large number of times, each time with the same total AlH^+ number but with statistical fluctuations using binomial random number generation creating a distribution of simulated observations. An n-tuple array using binomial-distribution random number generation is created; each row in the array represents a set of independent measurements, e.g. $N = 0$ and $N = 1$ population fractions. Using variations of this analysis, experimental consistency with our model can be verified, and we can determine statistical bounds without relying on binomial confidence intervals far from the central limit approximation. All code is located in App. D.2.1.

10.5.1. Model Consistency

Perfect broadband rotational optical cooling would result in 100% of the AlH^+ population in the $N = 0$ and $N = 1$ states. In our experiment, the combined population measured to be missing from these two states 6(5)%, consistent with the expectation of fully efficient cooling and no loss from molecular predissociation [**113**]. Apart from the suspect $N = 3$ data point discussed in Sec. 10.3.2.1, no REMPD probes after broadband rotational optical cooling detected population in $N > 1$, which is also consistent with fully efficient broadband rotational optical cooling. To verify agreement of the broadband rotational optical cooling measurements with our rate-equation simulation (Sec. 7.4.1), an n-tuple is generated that uses the experimental total ion numbers for our $N = 0$ and $N =$

1 broadband rotational optical cooling data sets, along with the broadband rotational optical cooling population fractions of $f_0 = 48.3\%$ in $N = 0$ and $f_1 = 51.7\%$ in $N = 1$ predicted from simulation. For each of 100,000 entries in the n-tuple, the population fraction ratio, defined as f_0/f_1, is calculated and histogrammed (Fig. 10.7). For this sample size, the data are well described by a Gaussian distribution; the experimentally-observed ratio is consistent with the simulation, differing by 1.3σ.

10.5.2. Vibrationally-assisted Broadband Rotational Optical Cooling Loss

To determine the limits on possible population loss to unprobed states during vibrationally-assisted broadband rotational optical cooling, we consider the statistics of the population successfully cooled to $N = 0$, and we ask whether it is consistent with the observed population remaining in $N = 1$. We define a loss fraction f_x as the population in any state other than $N = 0$ and insert this loss term into the Monte Carlo simulation. The 68% confidence level limits on f_x were found by adjusting f_x until only 16% of the simulated distribution fell on either side of the observed value of 4.6% missing from $N = 0$; a sample n-tuple distribution at one of these limits is shown in Fig 10.8. The resulting value for vibrationally-assisted broadband rotational optical cooling loss is $4.6^{+2.1}_{-1.3}\%$. A separate analysis using a similar technique for the population in $N = 1$ yields $2.0^{+3.3}_{-1.4}\%$. We conclude that our results are consistent with all population after vibrationally-assisted broadband rotational optical cooling being in either $N = 0$ or $N = 1$ with no evidence for loss to other states.

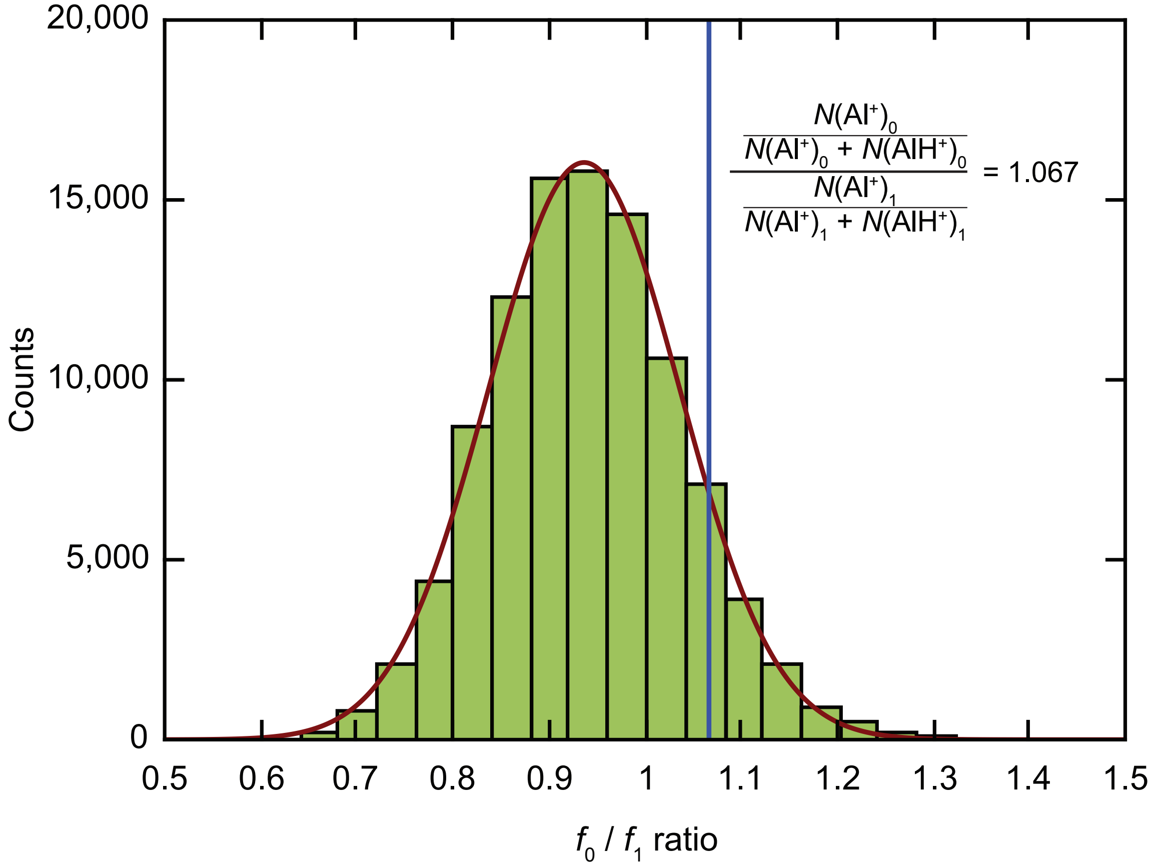


Figure 10.7. Typical distribution of the simulated ratio of the population fractions found in the two lowest states after broadband rotational optical cooling. The expression for the ratio in terms of the four observed ion counts, the observed population fraction (vertical line), and a Gaussian fit to the simulated distribution are also shown.

10.5.3. Predissociation Fraction

The energetically allowed process of predissociation, in which AlH^+ in the $\mathrm{A}^2\Pi$ state dissociates into Al^+ and H, is expected to be rare [113]. Predissociation during broadband rotational optical cooling would yield a signal of excess population in all rotational states, since it would provide an additional channel besides REMPD for dissociation of AlH^+

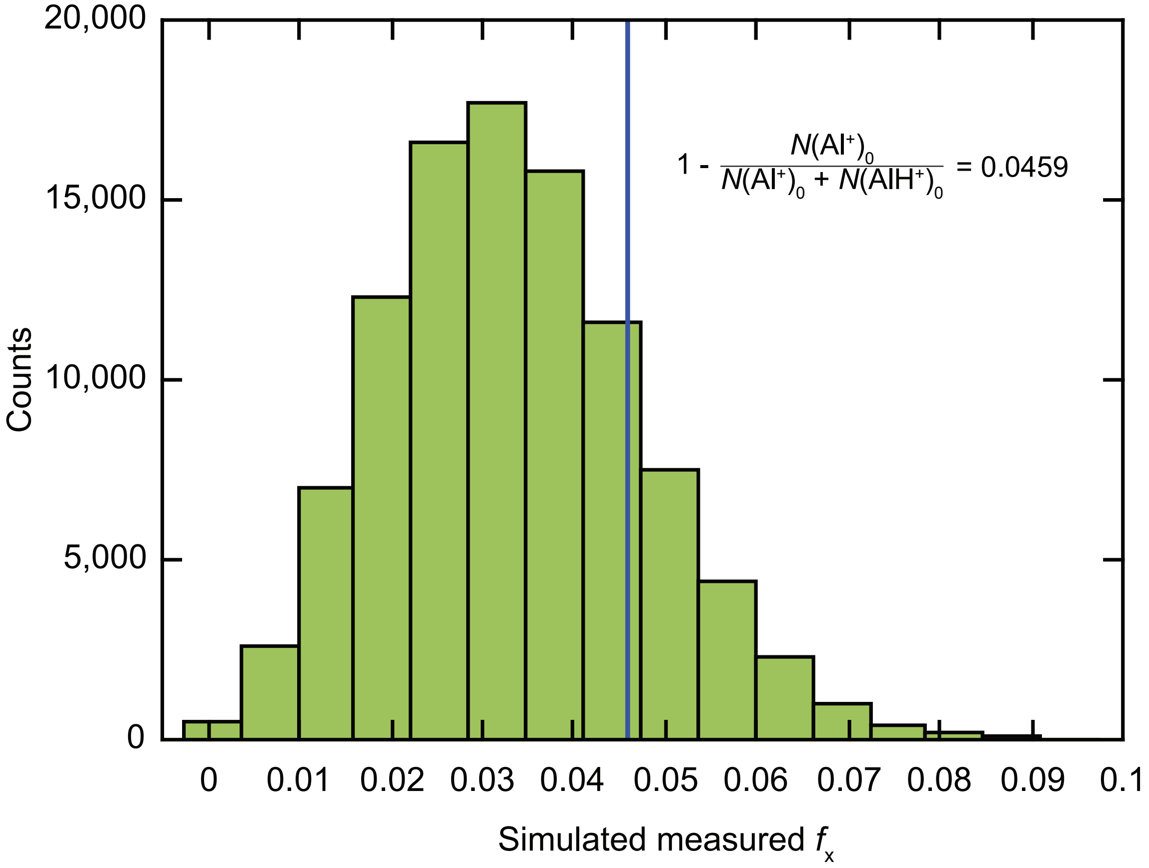


Figure 10.8. Typical histogram for determination of vibrationally-assisted broadband rotational optical cooling inefficiency f_x, defined as the fraction of population found to be missing from $N = 0$. The expression for f_x in terms of two observed ion counts and the observed value (vertical line) are also shown. This histogram shows the Monte Carlo simulation results with the inserted inefficiency parameter turned down to find the lower bound of its 68% confidence band, based on the measured f_x and histogram areas. The asymmetric distribution arises from small-number binomial statistics.

to Al^+. To set an upper limit on the predissociation fraction, the broadband rotational optical cooling total ion number detected in $N \geq 2$ is used, discarding the data set with suspect $N = 3$ signal discussed in Sec. 10.3.2.1. Using five n-tuples of 100,000 entries, an additional non-zero predissociation population is added to the Monte Carlo simulation;

this parameter is varied by hand until only 10% of n-tuple entries result in the observed value of 0 predissociated ions, from which we conclude at the 90% confidence level that less than 0.2% of the population predissociates during broadband rotational optical cooling. We estimate from the rate-equation simulation (Sec. 7.4) that there are on average five excitations to $\mathrm{A}^2\Pi$ for broadband rotational optical cooling from a room temperature distribution. Combining these results, we conclude at the 90% confidence level that the predissociation branching ratio from $\mathrm{A}^2\Pi$ relative to spontaneous emission is <0.04%, corresponding to a 90% confidence level upper limit on the predissociating decay rate of $2\pi\times$ 800 Hz.

10.5.4. Temperatures and Limits

10.5.4.1. Broadband Rotational Optical Cooling Temperature Limits. A similar method is used to set the broadband rotational optical cooling temperature upper limit for each parity state. An input temperature is used to calculate the thermal probabilities for being in the first excited state of each parity. This probability is then used to generate five 100,000-entry n-tuples for the experimental sample size. The temperature is increased until 10% of the n-tuple entries contain 1 molecule in the second-lowest rotational state of each parity.

Because the spacing between consecutive states of the same parity is large, 50 K (90 K) between the lowest two states of positive (negative) parity, only relatively weak constraints can be set on the temperature achieved by broadband rotational optical cooling. The 90% confidence level temperature upper limits are 13 K for positive parity and 19 K for negative parity. The negative-parity limit is obtained when the suspect $N = 3$

signal is neglected; if that signal is attributed to imperfect broadband rotational optical cooling, then the corresponding measured temperature for the negative parity is 19 K. The numerical correspondence to the limit value is coincidence.

10.5.4.2. Vibrationally-assisted Broadband Rotational Optical Cooling Temperature. The population ratio between $N = 1$ and $N = 0$ is used to determine the vibrationally-assisted broadband rotational optical cooling experimental temperature. The 68% confidence level limits, determined similarly to the vibrationally-assisted broadband rotational optical cooling loss and using five n-tuples of $100,000$ entries, yields a vibrationally-assisted broadband rotational optical cooling temperature of $3.8^{+0.9}_{-0.3}$ K. An alternative analysis technique would be to use the populations rather than the ratio to determine a temperature. For example, using the measured ground state population of 95% would yield a rotational temperature of 4.4 K. However, since extraction of the temperature from the population ratio makes no assumptions about detection efficiency, we consider it to be the more reliable technique.

10.6. Rotational Cooling Results Discussion

Apart from the suspect data point discussed in Sec. 10.3.2.1, there is no other evidence that imperfect broadband rotational optical cooling performance leaves population in $N \geq 2$. However, we observe a clear signal of population not fully cooled to $N = 0$ by vibrationally-assisted broadband rotational optical cooling (Fig. 10.4 and Table 10.3). This type of asymmetry between broadband rotational optical cooling and vibrationally-assisted broadband rotational optical cooling performance is expected because the spectral cutoff of the latter is necessarily closer to the Q-branch heating transition frequencies. A rate-equation simulation predicts that for operation below saturation on all transitions, if 1% of the intensity pumping P(1) leaks through the filter to pump Q(0), the result will be 6% population in $N = 1$, similar to the measured quantities discussed below.

We have not attempted to make a serious model for spectral leakage, but a simple analysis suggests plausibility for the presence of percent-level leakage intensity at Q(0). The separation between the P(1) and Q(0) transitions is 13 cm^{-1}, which is smaller than the accuracy of our absolute spectral calibration. The 2 cm^{-1} width of this cutoff is large enough compared with the P(1)-Q(0) separation that percent-level leakage at Q(0) for non-optimized mask placement would be unsurprising. Further investigation is required to confirm expectations that improved frequency tuning and cutoff resolution of the filter will lead to lower rotational temperatures after vibrationally-assisted broadband rotational optical cooling.

It is instructive to compare the performance of broadband rotational optical cooling with a few other approaches for rotational cooling of trapped molecules. The only previous laser cooling of rotations from room temperature used narrowband excitation of vibrational resonances, yielding ground state populations of 36.7% (20 K) for MgH$^+$ (Ref. [**110**]) and 78% (27 K) for HD$^+$ (Ref. [**111**]); the process-limiting timescale of several seconds is associated with rotational transitions. In comparison, vibrationally-assisted broadband rotational optical cooling of AlH$^+$ achieves a 95% ground state population and a temperature of 3.8 K with a process-limiting timescale of $\sim$ 100 ms associated with a vibrational spontaneous relaxation. Comparing generality of the approaches, the narrowband technique is readily applicable to any polar hydride but has limited utility for species with larger reduced mass (i.e. fluorides or chlorides) due to their slower vibrational and rotational transition rates [111]. Broadband rotational optical cooling is most readily applicable to specialized molecules with diagonal FCFs, but they can be heavier reduced-mass species. Finally, compared with direct cryogenic buffer gas cooling of molecular ions in a 4 K apparatus [108], we obtain in our room-temperature apparatus a temperature a few times colder for a single molecule (and many times colder for larger collections of molecules) on a similar ∼100 ms timescale.

Several other neutral and ionic candidate species for the simplest implementation of broadband rotational optical cooling have been identified [339, 112, 113]. It is interesting to consider the spectral filtering resolution required to rotationally cool diatomic species with heavier reduced mass and correspondingly smaller rotational constants and more congested spectra. Working with the same beam size but near the diffraction limit, we could expect to achieve enough resolution for efficient cooling of fluorides to their ground state and partial cooling of chlorides. Further resolution improvements could be obtained by using a finer grating and possibly of utilizing interference filters to handle transmission of the spectrum around the cutoff. Another option for rotational cooling of species with very small rotational constants would be to use a swept CW laser to cool population already near the ground state, as used in Ref. [109], but adding to their technique a broadband rotational optical cooling laser to cool higher-lying population.

CHAPTER 11

AlH^+ Future Experimental Prospects

With rotational cooling to the ground state demonstrated, one of the milestone prerequisites for implementing efficient mQLS has been achieved. The mQLS experiment can now proceed to motional ground state cooling. However, more beneficial science and important results can be performed with the apparatus once a reliable AlH^+ production method is developed (Sec. 17.4).

Due to poor EOM extinction of the rotational cooling light, the pulse-shaped femtosecond laser optical arrangement can be modified to investigate the parity-preserving, fast-timescale cooling. Faster ground state preparation could also be achieved using either a $(1+1)$ or $(1+1')$ two-photon process. Additionally, all discussion to this point has neglected the hyperfine structure of AlH^+ with the Al nuclear spin of $\frac{5}{2}$ and the H nuclear spin of $\frac{1}{2}$; however, the hyperfine structure spacing is expected to be $\lesssim$2 GHz, which can be experimentally achieved with EOM and/or AOM modulation. Regardless, optical pumping to a single hyperfine state should be achievable. Furthermore, if the ions could be prepared in the parity-trapped $N=1$ state, AlH^+ could be observed directly via fluorescence detection and possibly directly laser-cooled.

11.1. Lower Final Rotational Ground State Temperature

Due to the loss of AlH^+ production by cessation of the favorable background gas reaction (Sec. 8.10.4), the mask position may not be optimal to provide the lowest final rotational temperature. The separation between the P(1) and Q(0) transitions is 13 cm^{-1}, which is smaller than the accuracy of our absolute spectrometer calibration. Further investigation is required to confirm expectations that improved spectral filtering cutoff frequency tuning will lead to lower rotational temperatures after vibrationally-assisted broadband rotational optical cooling.

11.2. REMPD Cross Section Measurement

With a sample of AlH$^+$ prepared in the rotational ground state with a small spatial extent in the ion trap, the REMPD 266 nm 1$'$ dissociation cross section can be measured relatively easily. The 266 nm spot size, and therefore intensity at the ions, can be carefully measured by geometrically placing a detector (either a razor blade and energy meter or other beam profiler) at the location corresponding to trap center along the reflected 266 nm beam from the main viewport. The AlH$^+$ single-shot dissociation probability P_D should then obey the relation [340, 10]

$$P_D = e^{-(\text{Fluence})\times\sigma} = e^{-\frac{E\lambda}{hcA}\times\sigma} \tag{11.1}$$

where λ is the photon wavelength, σ is the photodissociation cross section of interest, h is Planck's constant, c is the speed of light, E is the 266 nm laser pulse energy, and A is a geometric factor related to area to calculate the pulse energy per unit area at the ion location.

By recording the dissociated ion fraction as a function of the 266 nm pulse energy, the photodissociation cross section can be extracted. Note that the sample would either need to be small such that the AlH^+ ions all experience the same fluence, although Gaussian beam averaging techniques could be used [10].

11.3. Broadband Rotational Optical Cooling Timescale

The imperfect spectrally-filtered pulsed laser EOM extinction (measured $\gtrsim$1000:1) and the resulting leakage light is likely sufficient to drive rotational cooling on ∼1 ms timescales, which is significantly faster than our mechanical shutter. An EOM operating at the 360 nm second harmonic of the Mai Tai is chosen to avoid the dispersion that a similar modulator would introduce at the 720 nm fundamental, which would detrimentally decrease the peak intensity of the femtosecond pulses and decrease the SHG conversion efficiency. We hoped that the resulting leakage light would not be problematic. Note that an AOM is not selected due to (1) the limited diffraction efficiency of the technology, reducing the laser power, and (2) concerns of the spatial dispersion introduced, convoluting the spectrum spatially across the output beam profile.

However, we can use the SHG nonlinearity to our advantage. By placing an EOM for use at 720 nm (ConOptics 350-105-01), we expect to *square* our extinction after the SHG stage. If we achieve the EOM specification of 500:1 extinction before SHG, this will be $(500:1)^2 = 250000:1$ after SHG, which should be more than sufficient to not drive rotational cooling on a timescale shorter than our mechanical shutters. The penalty for this increased extinction is lower SHG power for rotational cooling. Once the leakage light is not a problem, the fast, parity-preserving time dynamics of broadband rotational optical cooling can be investigated. With an implementation of a fast two-photon excitation from $\mathrm{A}^2\Pi \leftarrow \mathrm{X}^2\Sigma^+$ for the parity cooling (discussed below) necessary for rotational ground state cooling, we expect that AlH^+ can be prepared in the rotational ground state in $\ll$1 ms, which is two orders of magnitude improved over using the vibrationally-assisted technique.

11.4. Faster Ground State Preparation

Driving a single two-photon $\text{A}^2\Pi \leftarrow \text{X}^2\Sigma^+$ transition would allow parity to be cooled without needing to resort to the relatively slow vibrational spontaneous emission event used in vibrationally-assisted broadband rotational optical cooling. For instance, population pumped by broadband rotational optical cooling into $\left|\text{X}^2\Sigma^+, v = 0, N = 1\right\rangle$ could be driven by a two-photon excitation to $\text{A}^2\Pi$, with subsequent electronic spontaneous relaxation accomplishing a parity flip in less than a microsecond. Driving this transition in $\sim$1 μs could be accomplished with a single commercially available pulsed laser, using two photons of the same wavelength in a $(1 + 1)$ process with an intermediate state far from any resonance. If continuous parity cooling is desired, CW or quasi-CW lasers near-resonant with an intermediate state would be required. Using one of the first few vibrational levels of the $\text{X}^2\Sigma^+$ state as an intermediate state, commercially available lasers could drive the transition on microsecond timescales even if detuned from resonance by $\sim$100 MHz. Although the hyperfine structure of AlH^+ relevant for such narrowband excitation is not yet known, all hyperfine states should be addressable via power-broadening [341] and/or EOM/AOM modulation.

11.5. Vibrational Repumping

The problem of imperfect vibrational branching ratios upon $\text{A}^2\Pi$ spontaneous emission leads to $\sim$ 14% transient population excited to the $v > 0$ on the BROC timescale during vibrationally-assisted broadband rotational optical cooling. To achieve near-unity efficiency cooling on the broadband rotational optical cooling timescale, a second femtosecond laser can be added to drive some number of $\Delta v = -1$ vibrational cooling transitions $(\left|\mathrm{X}^2\Sigma^+, v\right\rangle$ to $\left|\mathrm{A}^2\Pi, v-1\right\rangle)$. This laser need not be spectrally-filtered for typical hydrides since vibrational heating and cooling transitions are generally separated by more than the femtosecond laser bandwidth. Although this is not the case for AlH^+, other broadband rotational optical cooling-candidate molecules such as SiO^+ [112] have structure permitting a single laser to simultaneously drive vibrational cooling transitions from several excited vibrational levels. Additionally, we note that including a vibrationally cooling femtosecond laser might also be desirable for heavier-reduced mass species in order to address their higher thermally-excited vibrational population at room temperature.

11.6. AlH^+ Hyperfine Structure Estimation

The hyperfine structure of AlH^+ is neglected up to this point since the broadband (pulsed) lasers used for rotational cooling and REMPD detection have sufficient bandwidth (power broadening) to span the hyperfine-structure-split transitions. However, future experiments such as CW fluorescence detection of AlH^+ would require at least some knowledge of the hyperfine structure as any CW laser sources would need sufficient modulation to address the necessary hyperfine levels.

Many investigations of molecular hyperfine structure have previously been discussed in theory and experiment [342, 343, 344, 345, 346, 347, 348, 349, 290, 350, 54, 351, 352, 339, 278, 353, 354, 100, 341, 355, 356, 357].

11.6.1. The HD$^+$ X$^2\Sigma^+$ State

Due to the simple, accurately-calculable structure and experimental interest by the Schiller group, the hyperfine structure of HD$^+$ is well known [344, 353, 354, 341, 355]. From experimental investigations, the strong $\left|\mathrm{X}^2\Sigma^+, v'=1, N'=1\right\rangle \leftarrow \left|\mathrm{X}^2\Sigma^+, v=0, N=0\right\rangle$ hyperfine transitions lie within 60 MHz with all transitions lying within 1 GHz [341]. From Ref. [353], the simplified effective spin Hamiltonian H_{eff} is

$$H_{\text{eff}} = b_1 \mathbf{I}_p \cdot \mathbf{s}_e + c_1 I_{pz} s_{ez} + b_2 \mathbf{I}_d \cdot \mathbf{s}_e + c_2 I_{dz} s_{ez} + \gamma \mathbf{s}_e \cdot \mathbf{L} \tag{11.2}$$

where $\mathbf{I}_p$ ($\mathbf{I}_d$) is the nuclear spin of the H (D) nucleus, I_{pz} (I_{dz}) is the nuclear spin z-axis projection of the H (D) nucleus, $\mathbf{s}_e$ is the electron spin, s_{ez} is the electron spin z-axis projection, $\mathbf{L} = \mathbf{\Omega} + \mathbf{N}$ is the total orbital angular momentum, and b_1, c_1, and γ are coupling constants whose strength determines the various splittings and dictates the optimal choice of angular momentum coupling operators:

$$\mathbf{F} = \mathbf{I}_p + \mathbf{s}_e \tag{11.3}$$

$$\mathbf{S} = \mathbf{F} + \mathbf{I}_d \tag{11.4}$$

$$\mathbf{J} = \mathbf{L} + \mathbf{S} \tag{11.5}$$

The resulting hyperfine level energy diagrams of the first two rotational levels are shown in Fig. 11.1.

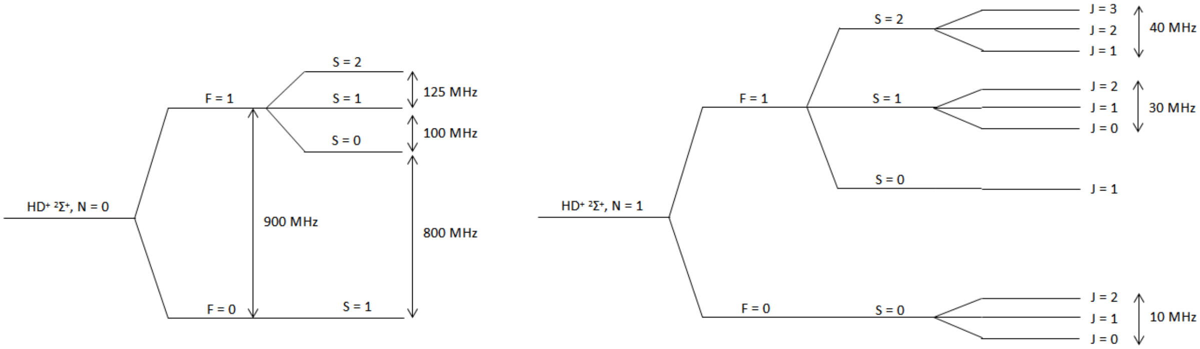


Figure 11.1. HD$^+$ hyperfine structure of the first two rotational sates of the molecule.

Table 11.1. Hyperfine constants from electron spin resonance measurements of HD$^+$ and AlH$^+$ isolated in noble gas matrices where H corresponds to the hydrogen nucleus and X corresponds to either the deuterium or aluminum nucleus. Values are from Ref. [358, 352].

Constant	HD$^+$	AlH$^+$
A_{iso}(H)	888	442
A_{dip}(H)	32.6	2
A_{iso}(X)	132	1586
A_{dip}(X)	4.3	49

The isotropic and dipolar hyperfine parameters, $A_{iso}(X)$ and $A_{dip}(X)$ measure the amount of spin-density and asymmetry of the spin-density distribution at center X, respectively [352, 355]. For HD$^+$ and AlH$^+$ isolated in noble gas matrices, these values are listed in Table 11.1.

From Ref. [353], the three dominant effective Hamiltonian terms of HD$^+$ in order of energy are $E_4 = \vec{I_p} \cdot \vec{s_e}$ (the proton-electron spin interaction), $E_5 = \vec{I_d} \cdot \vec{s_e}$ (the deuteron-electron spin interaction), and $E_1 = \vec{L} \cdot \vec{s_e}$ (the spin-rotation interaction). From Ref. [351], the two nuclear interaction terms that dominate the HD$^+$ hyperfine structure are the Fermi contact terms, which have the Hamiltonian

$$\mathcal{H}_{\text{Fermi Contact}} = \sum_{i,\alpha} \frac{g_s \mu_B g_\alpha \mu_N}{4\pi\epsilon_0 c^2} \frac{8\pi}{3} \delta^{(3)}\left(\vec{r}_{i,\alpha}\right) \vec{s_i} \cdot \vec{I_\alpha} \tag{11.6}$$

where the summation is over the electron i and the nucleus α, g_s is the electron g-factor, μ_B is Bohr magneton, g_α is the nucleon g-factor, μ_N is the nuclear megneton, ϵ_0 is the permittivity of free space, c is the speed of light, $\vec{r}_{i,\alpha}$ is the vector between the nucleon and the electron, $\vec{s_i}$ is the electron spin vector, and $\vec{I_\alpha}$ is the nucleon spin vector. Expanding for HD$^+$,

$$\begin{aligned} \mathcal{H}_{\text{Fermi Contact}} = \frac{1}{4\pi\epsilon_0 c^2} \frac{8\pi}{3} [&g_e \mu_B g_H \mu_N \delta^{(3)}\left(\vec{r}_{e,H}\right) \vec{s_e} \cdot \vec{I_H} \\ &+ g_e \mu_B g_D \mu_N \delta^{(3)}\left(\vec{r}_{e,D}\right) \vec{s_e} \cdot \vec{I_D}]. \end{aligned} \tag{11.7}$$

From Ref [353],

$$A_{iso} = \frac{8\pi}{3} g_e g_N \beta_e \beta_N \left\langle \delta\left(r\right) \right\rangle \tag{11.8}$$

where the average denotes an integral over the spin density. From Ref. [348],

$$\beta_e = \frac{e\hbar}{2m_e c} \tag{11.9}$$

$$\beta_N = \frac{e\hbar}{2m_p c} \tag{11.10}$$

are the Bohr and nuclear magneton in CGS units where e is the elementary charge, m_e (m_p) is the electron (proton) mass, and c is the speed of light. Therefore, the interaction terms, once integrated over the spin density, are

$$\mathcal{H}_{\text{Fermi Contact}} = \frac{1}{4\pi\epsilon_0 c^2} \frac{8\pi}{3} \left[A_{iso}\left(H\right) \vec{s_e} \cdot \vec{I_H} + A_{iso}\left(D\right) \vec{s_e} \cdot \vec{I_D} \right]. \tag{11.11}$$

11.6.1.1. Hyperfine Coupling Constants Relations. A_{iso} and A_{dip} are related to the Frosch-Foley coefficients [**359**, **352**, **355**] b and c by

$$A_{iso} = b + \frac{c}{3} \tag{11.12}$$

$$A_{dip} = \frac{c}{3} \tag{11.13}$$

and the Breit-Pauli Hamiltonian coefficients b_F and d_1 by

$$A_{iso} = b_F \tag{11.14}$$

$$A_{dip} = \frac{d_1}{3}. \tag{11.15}$$

Additionally, electron spin resonance measurements quantify the absolute value of the hyperfine coupling constants $A_\parallel$ and $A_\perp$, which are related to the isotropic and dipolar hyperfine parameters by

$$A_{iso} = \frac{1}{3}\left(A_\parallel + 2A_\perp\right) \tag{11.16}$$

$$A_{dip} = \frac{1}{3}\left(A_\parallel - A_\perp\right). \tag{11.17}$$

11.6.2. The AlH$^+$ X$^2\Sigma^+$ State

Despite the similar A_{iso} and A_{dip} coupling ratios between HD$^+$ and AlH$^+$, the same angular momentum coupling scheme cannot be used for AlH$^+$ due to the third term in Eq. 11.2; for HD$^+$, this term is small compared to the Fermi contact terms. However, this term dominates in AlH$^+$. Therefore, the angular momenta scheme for HD$^+$ is not applicable for AlH$^+$.

However, if this coupling scheme is used, the splitting from the A_{iso} values should increase the first splitting by 1.4x for the dominant **F** Fermi contact contribution (Al nucleus in AlH$^+$ and H nucleus in HD$^+$) and by 4x in the less-dominant **S** term. The **F** splitting in AlH$^+$ would further be increased by the significantly-larger nuclear spin of the Al nucleus compared to the H nucleus in HD$^+$ by a factor of 5x. Thus, the splittings in the AlH$^+$ X$^2\Sigma^+$ state would be $A_{iso}(Al) \sim$ 1.2 GHz and $A_{iso}(H) \sim$ 550 MHz. A full hyperfine structure calculation is performed by MGK and will be featured in future work.

11.6.3. Hyperfine Structure Spectroscopy

Although the hyperfine structure of AlH$^+$ can be roughly estimated from electron spin resonance experiments, no experimental measurements for the A$^2\Pi$ state exist. A fantastic experimental technique for measuring the hyperfine structure of this state and map the A$^2\Pi \leftarrow$ X$^2\Sigma^+$ hyperfine transition lines is Doppler-free spectroscopy of AlH$^+$ in a hollow-cathode cell [**200**, **201**, **202**, **204**, **205**, **119**, **206**] similar to the Hamamatsu L2783-56NE-BA barium laser galvatron. In this application, the hollow-cathode would be replaced with one made from aluminum, and the vacuum cell would be filled with 2.5 Torr Ne with 0.2 Torr NH$_3$, which was found to produce the strongest AlH$^+$ signal in Ref. [**119**].

A CW dye laser either directly producing 360 nm light or producing 720 nm light and using a SHG cavity would be an ideal laser source for this application due to the large tunability of the system compared with that of diode lasers. Additionally, laser diodes at 360 nm are unavailable, and few viable options are available at 720 nm with sufficient power for efficient SHG in a cavity. THG could be used with two optical cavities with a fundamental of 1080 nm, and such THG systems have been previously demonstrated [360]. However, this is a significant technical challenge.

Using a $3f$ laser frequency modulation detection method would be ideal for this application [361]. In most Doppler-free saturated-absorption experiments, the laser frequency is dithered at a frequency f, which is detected and used as the frequency reference for lock-in detection. If a parabolic power-absorption profile is assumed, the background slope error can be eliminated if the $3f$ signal is used for lock-in detection [362] at the cost of some signal-to-noise ratio. In other words, using the third harmonic of the modulation frequency for the lock-in detection removes the linear and quadratic background terms of the Doppler-broadened line profile.

11.7. AlH$^+$ Hyperfine State Pumping

11.7.1. Proof of Concept Experiment

Despite not knowing the exact hyperfine structure of AlH$^+$, it should be possible to optically pump a sample after vibrationally-assisted broadband rotational optical cooling to a single hyperfine state. This can be achieved by adding a second 360 nm dye laser beamline that is circularly-polarized with frequency tuned to drive the Q(0) $A^2\Pi \leftarrow X^2\Sigma^+$ transition. By aligning the applied magnetic field to the ions parallel to this circularly-polarized beam to make it optimally σ-polarized, optical pumping should be achievable. Conveniently, ^{138}Ba$^+$ can serve as our magnetometer for aligning the magnetic field supplying that a 493 nm Doppler cooling beam is circularly-polarized and co- or counter-propagating to the dye laser optical-pumping beamline. An example of this magnetic field alignment is in Sec. 13.8.

In this experiment, the rotational ground state cooling would proceed as before. Once the AlH$^+$ population is prepared in the single-parity rotational ground state, the ions will be exposed to the σ-polarized dye laser light. If we assume that after each pulse half the population scatters a photon, and that we need ∼5 photon scatters, this optical pumping should take ∼10 laser pulses for an exposure time of 1 second, which is significantly shorter than the blackbody rotational redistribution time.

REMPD state readout would again be employed. Using the σ-polarized dye laser output as the 1-photon source, only molecules that are not in the stretched state will be dissociated. However, the original dye laser beam (with significant π or opposite-σ polarization) will interact with all molecules in the $N = 0$ state, resulting in dissociation. Therefore, the signal of single-hyperfine-state preparation would be an asymmetry in REMPD signal between using the σ-polarized or original beamline 1-photon dye beams.

11.7.2. Fast Hyperfine State Pumping

Once hyperfine-pumping is initially demonstrated, the ConOptics UV EOM can be used as a variable power splitter to split power between the rotational cooling beamline and a second spectral filtering beamline that includes the Q(0) transition. This new, circularly-polarized beamline would then be aligned with the magnetic field in a similar fashion (σ-polarized) and serve as the hyperfine pumping laser source. We expect the combination of faster, single-parity rotational ground state cooling and this hyperfine pumping to occur on the $\lesssim$1 ms timescale since only a few $A^2\Pi \leftarrow X^2\Sigma^+$ transitions should be required.

Table 11.2. Transition wavelengths and natural linewidths of $\left|\mathrm{A}^2\Pi_{1/2}, v=0, N=0\right\rangle \rightarrow \left|\mathrm{X}^2\Sigma^+, v'', N=1\right\rangle$ transitions required for direct-fluorescence detection of AlH^+. All results are calculated using LEVEL (Ch. 7) and used as inputs for the rate equation simulations performed in Sec. 11.8.1.

v''	Energy (cm^{-1})	Energy (nm)	Γ
0	27708	360.91	$2\pi \times 2.6$ MHz
1	26203	381.64	$2\pi \times 69$ kHz
2	24845	402.49	$2\pi \times 1.4$ kHz
3	23636	423.08	$2\pi \times 49$ Hz

11.8. Prospect of CW molecular fluorescence

With rotational ground state preparation demonstrated, the possibility of a faster rotational cooling preparation timescale presented, and the hyperfine structure discussed, it is interesting to propose a non-destructive direct detection method for a molecular ion species, which has not yet been experimentally demonstrated. The parity barrier of AlH^+ provides a rotationally-closed but vibrationally-open cycling transition; if the AlH^+ ions could be prepared in the $\left|\mathrm{X}^2\Sigma^+, v=0, N=1\right\rangle$ state, driving the P(1) transition $\left|\mathrm{A}^2\Pi_{1/2}, v=0, N=0\right\rangle \leftarrow \left|\mathrm{X}^2\Sigma^+, v=0, N=1\right\rangle$ will result in electronic cycling until a vibrationally-exciting spontaneous emission event occurs. If the vibrational leak is repumped by driving the $\left|\mathrm{A}^2\Pi_{1/2}, v=0, N=0\right\rangle \leftarrow \left|\mathrm{X}^2\Sigma^+, v=1, N=1\right\rangle$ transition, substantially more photons will be scattered as the next vibrational leak, $\left|\mathrm{A}^2\Pi_{1/2}, v=0, N=0\right\rangle \rightarrow \left|\mathrm{X}^2\Sigma^+, v=2, N=1\right\rangle$, is additionally suppressed. Additional repumps driving the $\left|\mathrm{A}^2\Pi_{1/2}, v=0, N=0\right\rangle \leftarrow \left|\mathrm{X}^2\Sigma^+, v>0, N=1\right\rangle$ transitions are also considered. A level diagram of the proposed scheme is shown in Fig. 11.2 and required laser sources are listed in Table 11.3.

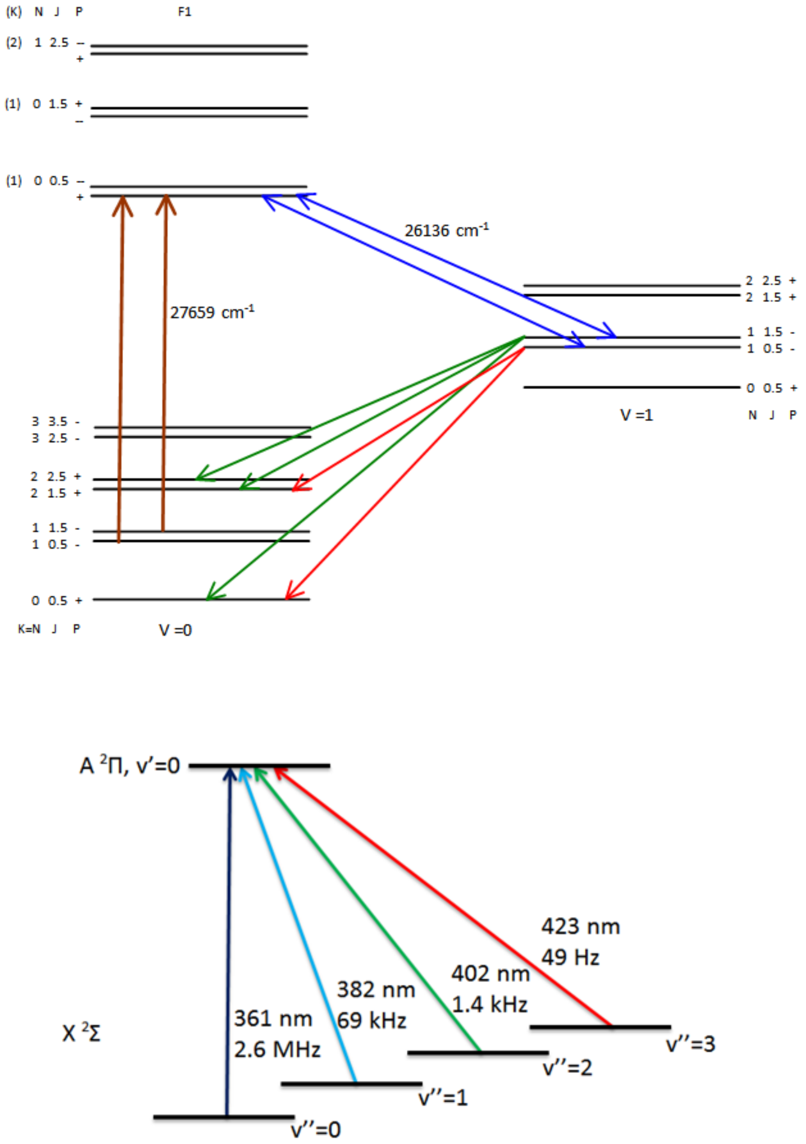


Figure 11.2. Transitions for CW fluorescence detection of AlH$^+$. (top) The main transition laser drives the parity-trapped $\left|A^2\Pi_{1/2}, v=0, N=0\right\rangle \leftarrow \left|X^2\Sigma^+, v=0, N=1\right\rangle$ transition until a vibrationally-exciting spontaneous emission occurs. (bottom) Vibrational repumping lasers considered.

Note that all vibrational laser frequencies are to the $\left|A^2\Pi_{1/2}, v=0, N=0\right\rangle$ state. Due to the small vibrational wave function overlap as Δv increases, these transitions become increasingly supressed. However, driving $\Delta v = -1$ transitions would be appreciably faster, but this is not considered in the rate equation simulations.

11.8.1. Rate Equation Simulations

Rate equation simulations are performed similarly to those in Sec. 7.4. Input files are located in App. D.2.2, and integrated total photon counts from a single AlH^+ is shown in Fig. 11.3, calculated by multiplying the rate-equation population in $\left|\mathrm{A}^2\Pi_{1/2}, v = 0, N = 0\right\rangle$ by $\frac{\Gamma}{2\pi}$.

From the simulation results, ∼2000 photons are scattered with one vibrational repump with AlH^+ population quickly falling into the dark $\left|\mathrm{X}^2\Sigma^+, v = 2, N = 1\right\rangle$ state after ∼1 ms. With a single vibrational repump laser driving the $\left|\mathrm{A}^2\Pi_{1/2}, v = 0, N = 0\right\rangle \leftarrow \left|\mathrm{X}^2\Sigma^+, v = 1, N = 1\right\rangle$ transition, direct fluorescence detection should be possible with a 10^{-4} photon collection efficiency with sufficient ion numbers for a desired signal-to-noise ratio.

Repumping the $\left|\mathrm{X}^2\Sigma^+, v = 2, N = 1\right\rangle$ state, with a total of two vibrational repumps, yields ∼10000 photons, which is more than sufficient for single-molecule fluorescence detection and direct Doppler cooling of AlH^+ [**112, 113**]. Adding a third vibrational repump does not significantly increase the total photon counts due to the transition Γ being small. However, driving $\Delta v = -1$ transitions is appreciably faster, but this is not considered in the rate equation simulation.

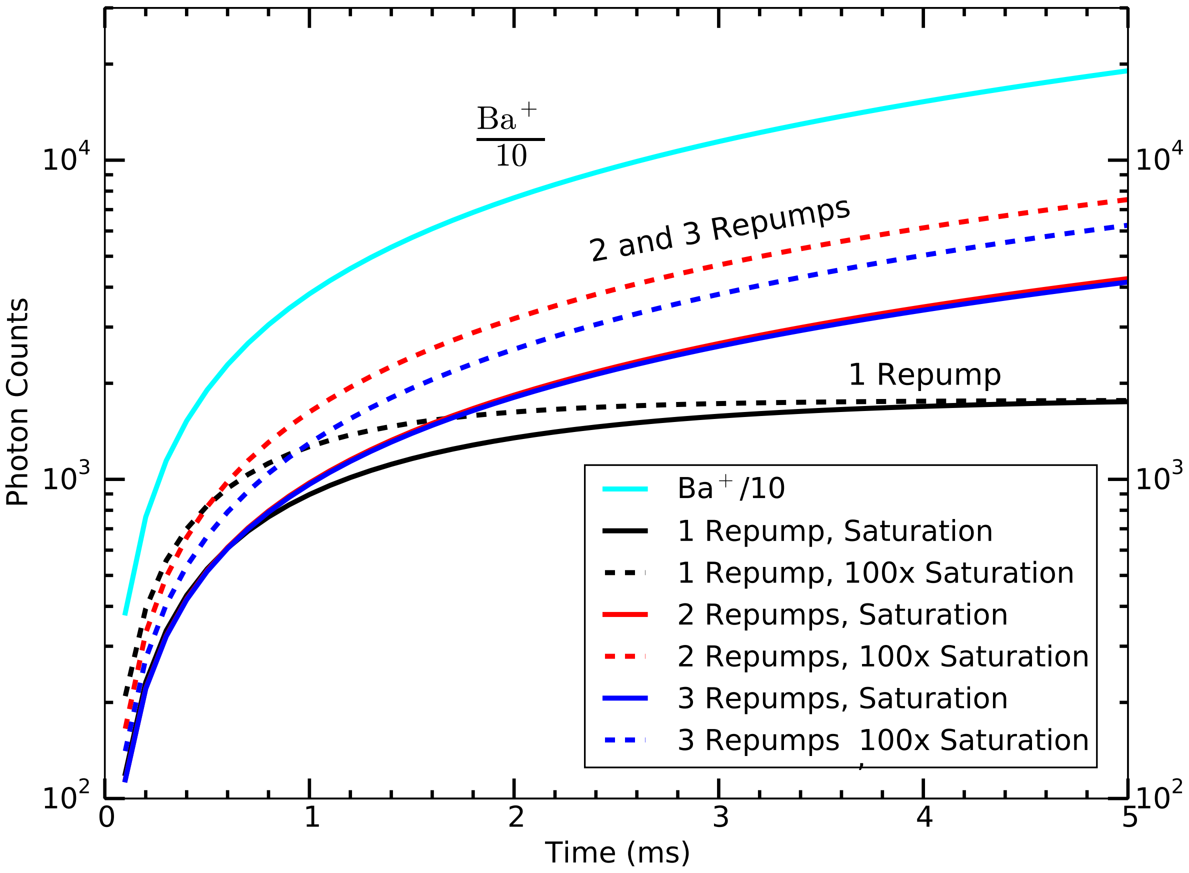


Figure 11.3. Total fluorescence photons emitted by a single AlH$^+$ versus time with multiple repumps and laser powers considered. For reference, Ba$^+$ fluorescence is also included.

11.8.2. Laser Sources and Requirements

The possible laser source requirements for AlH$^+$ fluorescence detection are listed in Table 11.3. All laser sources mentioned are commercially available, and higher-harmonic generation units could be either purchased or constructed. Note that the estimated ∼2 GHz modulation bandwidth required to address all hyperfine levels of the AlH$^+$ $X^2\Sigma^+$ state can be provided using AOM or EOM technology after higher-harmonic generation.

Table 11.3. Transition wavelengths, natural linewidths of, and possible laser sources for driving the $\left|A^2\Pi_{1/2}, v=0, N=0\right\rangle \leftarrow \left|X^2\Sigma^+, v'', N=1\right\rangle$ transitions required for direct-fluorescence detection of AlH^+. All laser sources here are assumed to be CW. Note that broadband lasers could also perform the vibrational repumping.

v''	Energy (nm)	Γ	Method	Laser Source
0	360.91	$2\pi \times 2.6$ MHz	SHG	722 nm ECDL, TA MOPA
			THG	1083 nm ECDL, Fiber Amplifier MOPA
1	381.64	$2\pi \times 69$ kHz	SHG	764 nm ECDL, TA MOPA
2	402.49	$2\pi \times 1.4$ kHz	Direct	402 nm ECDL
			SHG	804 nm ECDL, TA MOPA
3	423.08	$2\pi \times 49$ Hz	Direct	423 nm ECDL
			SHG	863 nm ECDL, TA MOPA

To lock the above laser systems, a transfer cavity, actively locked to a Rb reference and under vacuum, is proposed and is discussed in Sec. 3.1.

Part 3

Motional Cooling

With rotational ground state cooling of AlH^+ demonstrated and hyperfine state pumping discussed in Part 2, one of the two milestones/prerequisites for efficient mQLS has been realized. The second milestone prerequisite, motional ground state cooling (Fig. 1.3), has been previously demonstrated in many atomic ion species with a variety of cooling methods and is discussed in this part. Because the infrastructure had not been in place and group expertise not developed prior to this work, motional ground state cooling of a single $^{138}Ba^+$ ion is the logical initial demonstration.

Since QLS does not require the high gate fidelities usually associated with quantum computation and simulation, it is possible to make simplifying choices choosing logic ion isotope and quantum operation protocols at the expense of some state operation and/or measurement fidelity. In this part, we discuss sideband cooling and motional state detection protocols for $^{138}Ba^+$ of sufficient fidelity for QLS without an extremely narrowband laser or the use of a species with hyperfine structure, both of which would add considerable technical challenges and delay the effort towards the realization of efficient mQLS. We use the two $^2S_{1/2}$ Zeeman sublevels of $^{138}Ba^+$ to Raman sideband cool a single ion to the motional ground state. Because of the small Zeeman splitting, near-resonant Raman sideband cooling of $^{138}Ba^+$ requires only the Doppler cooling lasers and two additional AOMs, minimizing the complexity of the optical system. Observing the near-resonant Raman optical pumping fluorescence, we extract relevant experimental parameters and demonstrate a final $\bar{n} \ll 1$. We additionally employ a second, far-off-resonant laser driving Raman π pulses between the two $^2S_{1/2}$ Zeeman sublevels to provide motional state detection for QLS and to confirm the sideband cooling efficiency, measuring a final $\bar{n} = 0.15(6)$.

CHAPTER 12

Motional Ground State Cooling Techniques

Motional ground state cooling techniques typically use either a narrow, electric-dipole-forbidden single-photon resonance or a two-photon Raman resonance. For ground state cooling, the linewidth of such a transition must be narrower than the secular frequency of the motional mode being cooled; in this case, the motional sidebands are resolved, and sideband cooling can occur by cycling on the red sideband.

12.1. One-photon Resolved Sideband Cooling

Previously-used single-photon resonances include the first demonstration of resolved sideband cooling published in 1989 in which $^{198}\mathrm{Hg}^{+}$ was cooled to the motional ground state on the $^{2}\mathrm{S}_{1/2} \rightarrow {}^{2}\mathrm{D}_{5/2}$ transition [**363**]. The similar technique of driving an electric dipole-forbidden transition has been used to motionally cool many ion species with a non-exhaustive list in Table 12.1.

In CW sideband cooling, the ion is exposed to the quadrupole laser with frequency detuned to the red sideband. An additional repumping laser is applied to depopulate the excited state via a dipole-allowed transition. The shorter excited-state lifetime due to the repumping beam and the intentional power broadening of the quadrupole transition both assist in relaxing the laser frequency lock and narrow linewidth requirements from that of the natural linewidth to that of the order of the power-broadened linewidth. However, the power-broadened transition and quadrupole laser linewidths must still both

Table 12.1. A non-exhaustive list of species in which resolved sideband cooling to the motional ground state using a quadrupole transition has been demonstrated.

species	transition	wavelength	lifetime
Hg^+ [363]	$^2S_{1/2} \rightarrow {}^2D_{5/2}$	282 nm	~90 ms [364]
Ca^+ [365, 366]	$^2S_{1/2} \rightarrow {}^2D_{5/2}$	729 nm	~1.1 s [140]
In^+ [367]	$^1S_0 \rightarrow {}^3P_1$	231 nm	~2.8 μs [367]
Sr^+ [368]	$^2S_{1/2} \rightarrow {}^2D_{5/2}$	674 nm	~400 ms [369]
Ba^+ [138]	$^2S_{1/2} \rightarrow {}^2D_{5/2}$	1.76 μm	~35 s[134]
Yb^+	$^2S_{1/2} \rightarrow {}^2D_{5/2}$	411 nm	~7 ms [370]

be significantly less than the secular frequency of the motional mode being cooled such that the motional sidebands are still well-resolved.

In pulsed sideband cooling, a π pulse on the quadrupole transition red sideband decreases the motional quanta by one and is followed by an optical pumping pulse to place the ion back into the electronic ground state. This is typically performed after CW sideband cooling such that the ion is already in a low n so that the ion has a definite sideband Rabi frequency since the red sideband Rabi frequency scales as $\sqrt{n}\eta\Omega_0$ [85]. Repetition of these pulses assure that the ground state is reached with high probability. This process benefits from a high state transfer fidelity, which requires both a large Rabi frequency and a narrow linewidth laser to avoid both thermal and laser linewidth decoherence effects. The laser linewidth decoherence occurs from the multiple spectral components in the addressing laser each having a slightly different Rabi frequency to the desired power-broadened atomic line, and the interference of these results in the decoherence. Unfortunately, rapid adiabatic passage (RAP) techniques, often reducing demanding requirements on the driving laser system, only slightly relax this large Rabi frequency and narrow linewidth requirement [371].

Additionally, pulsed sideband cooling of Yb^+ to the motional ground state has been demonstrated using RF radiation and a static magnetic field gradient [372].

12.2. Two-photon Raman Sideband Cooling

Using a two-photon Raman transition was initially proposed by Marzoli in 1994 [141] and first demonstrated using the ground state hyperfine splitting in $^{9}\mathrm{Be}^+$ in 1995 [143]. This sideband cooling technique uses a *tailored* Raman transition between two quantum states, e.g. electronic, hyperfine, or Zeeman levels; a non-exhaustive list is shown in Table 12.2. In this context, tailored means that with appropriate choice of detuning and Rabi frequencies of the two laser sources coupling the involved quantum states, the three-level system can be described as a two-level system with a transition linewidth suitable for resolved sideband cooling [141].

Choice of a hyperfine interval for Raman sideband cooling has the advantage that the states can be chosen to be magnetically insensitive, allowing the same lasers to be used for cooling and high-fidelity coherent operations. On the other hand, Raman sideband cooling on a Zeeman splitting, previously demonstrated [147, 148, 149, 150, 151, 31], allows use of even isotopes, which are simpler to cool and manipulate due to their lack of hyperfine structure. Additionally, ground state hyperfine intervals for some ions are too large for easy access by AOM (8 GHz for $^{137}\mathrm{Ba}^+$) and would require e.g. either a phase-lock loop (PLL) with separate lasers or EOM modulation with optical filtering for sideband cooling.

Table 12.2. A non-exhaustive list of species and quantum states in which Raman sideband cooling to the motional ground state has been demonstrated. All experiments use the $^2P_{1/2}$ intermediate state except for those marked with a (*), which use the $^2P_{3/2}$ intermediate state. For $^{171}Yb^+$, the ultrafast source is detuned roughly half way between the $^2P_{1/2}$ and $^2P_{3/2}$ states, minimizing the effective light shift [373].

species	transition type	wavelengths (nm)	detuning	ref.
$^9Be^+$	hyperfine	313	-12 GHz	[143]
$^{137}Ba^+$	hyperfine	493	∼few GHz	[146, 374]
$^{111}Cd^+$	hyperfine*	215	-300 GHz	[144]
$^{25}Mg^+$	hyperfine*	280	+900 MHz	[145]
$^{171}Yb^+$	hyperfine*	355		[373]
$^{138}Ba^+$	electronic	493, 650	-500 GHz	[136, 142]
$^{40}Ca^+$	Zeeman	397	+75 MHz	[147]
$^{40}Ca^+$	Zeeman	397	-180 MHz	[150]
			+260 MHz	[150]
			-30 GHz	[150]
$^{40}Ca^+$	Zeeman	397	-130 MHz	[151]
			+30 GHz	[151]
$^{40}Ca^+$	Zeeman	397		[149, 152]
$^{172}Yb^+$	Zeeman	369	-60 MHz	[148]
			+60 MHz	[148]
$^{24}Mg^+$	Zeeman	280	+97 MHz	[31]

12.3. Sideband Cooling and Ba^+

For high-fidelity coherent operations or sideband cooling of Ba^+, a 1.76 μm laser is sometimes used to drive the quadrupole $S_{1/2} \rightarrow D_{5/2}$ transition ($D_{5/2}$ lifetime $\sim$ 35 s [134]) [135, 375, 137, 138, 139]. Compared with the Ca^+ quadrupole transition wavelength of 729 nm and $\sim$ 1.1 s [140] lifetime, the Ba^+ transition wavelength is a factor of two longer, and the upper state lifetime is a factor of thirty longer. The longer transition wavelength decreases the coupling of the quadrupole transition to the motional mode. Additionally, the longer transition wavelength and state lifetime both increase the laser power that must be applied to reach a similar power-broadened linewidth. For example, an NKT Photonics 1.76 μm fiber laser, similar to the one used in Ref. [375], is available in the laboratory. In Ref. [375], the laser system provides 3.5 mW at 1.76 μm after wavelength stabilization for use in the experiment with a $\sim$70 μm beam waist. This is sufficient to power-broaden the carrier transition to 28 kHz FWHM for a carrier Rabi frequency of $\sim$20 kHz. Assuming the best-case Lamb-Dicke parameter of $\sim$0.02 for this wavelength along the ion trap z-axis, the sideband Rabi frequency for motional state readout would be $\sim$400 Hz. If the decoherence rate is similar to the 7.5 kHz observed in Sec. 15.5, the detection contrast would be insufficient using the 1.76 μm source. Thus, due to the laser power requirements on and frequency stabilization challenges of the 1.76 μm laser source, Raman sideband cooling is an attractive alternative.

For Ba^+, both electronic [136, 142] and hyperfine levels [146] have been previously used for Raman sideband cooling. The 8 GHz hyperfine spacing in $^{137}\mathrm{Ba}^+$ is too large for easy access by AOM, and would require non-trivial laser and modulation sources such as using an EOM with spectral filtering provided by an actively-locked optical cavity, injection-locked diode laser system with additional SHG cavity, or offset phase lock between two ECDL systems. For these technically-challenging reasons, the Zeeman splitting of $^{138}\mathrm{Ba}^+$ is an attractive system for Raman sideband cooling for QLS, in which high quantum gate fidelities are not required. Here, each of the two Raman beams is red-detuned from the single-photon resonance to prevent Doppler heating during setup and diagnostic steps. First, the ability to collect off-resonant fluorescence without unwanted heating simplifies the tasks of beam alignment. Second, red-detuning enables collections of near-resonant Raman spectra (Ch. 14) in which there is clear evidence for efficient sideband cooling with proper two-photon detuning. Finally, we are able to employ a Fano-like fit function to the spectra to extract relevant parameters (Rabi frequencies of each near-resonant Raman beam, the secular motional frequency, and the Zeeman splitting) needed to set up far-off-resonant π pulses for quantitatively determining our final $\bar{n}$ value (Ch. 15). In Sec. 16.1.2, we discuss a potential upgrade to our apparatus that would allow switching from near-resonant red-detuning after the setup stages to near-resonant blue-detuning, thereby taking advantage of so-called EIT cooling [**147, 31**] to obtain a lower $\bar{n}$ by preventing off-resonant excitation of the carrier transition, which is currently our limiting heating mechanism.

CHAPTER 13

Experimental Apparatus, Optical Systems, and Magnetic Field

This chapter discusses the apparatus, laser optical systems, and magnetic field that the ground state cooling experiment relies upon.

13.1. The Single-ion Scale Trap

The motional ground state cooling experiment is conducted in a linear RF trap of single-ion-scale (Fig. 13.1) constructed and previously described by YWL in Refs. [**130, 235**] with vacuum chamber pressure $< 10^{-10}$ Torr as measured by a Granville-Phillips 370 Stabil-ion gauge and controller. The ion trap parameters are $r_0 = 1.26$ mm, $z_0 = 0.95$ mm, and $\Omega_{\mathrm{RF}} = 2\pi \times 23.420$ MHz with the RF applied to two opposing rod electrodes and the other two held at ground. All drawings can be found in Ref. [**235**].

A schematic of the trap RF drive system is shown in Fig. 13.2, and the resonator resonance and Q-factor are shown in Fig. 13.3. The RF power applied to (reflected from) the resonator is measured using the forward (reverse) directional coupler outputs connected to power detectors. The RF frequency is tuned to the helical resonator resonance by minimizing the RF power reflected from the resonator; when on-resonance, the resonator and amplifier output are impedance-matched at 50 Ω, and the reflected power is minimized. The RF voltage applied to the trap and associated radial secular frequencies are then calculated using the code in Sec. E.1.12. The RF voltage is controlled via PC (App. C.3.4) using the RF AM input on the function generator with the generator's maximum power

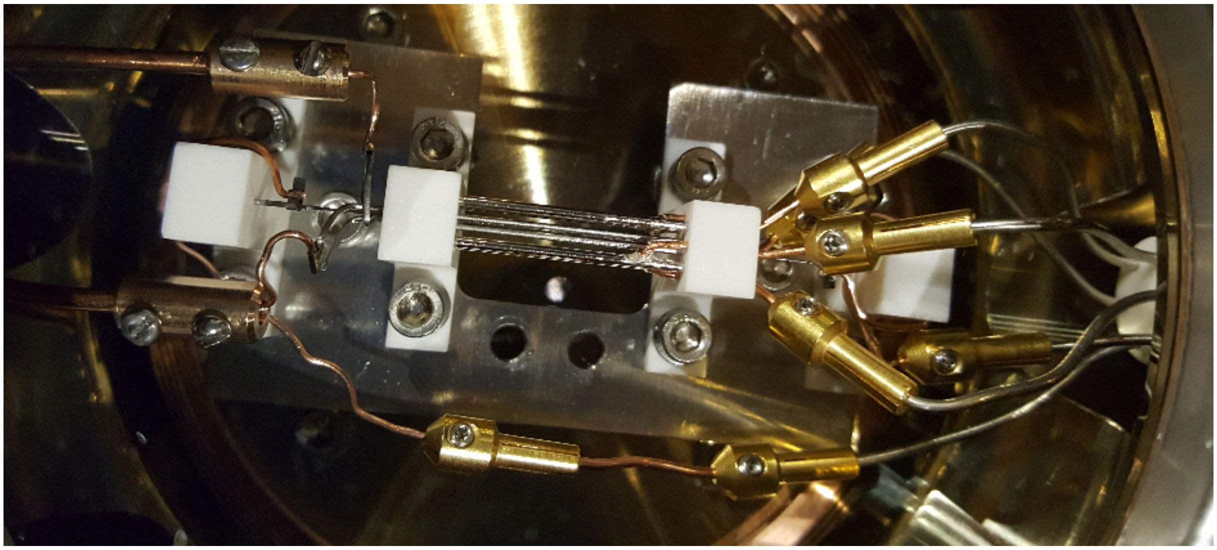


Figure 13.1. Single-ion trap picture. The RF electrical connections are on the left and the DC electrical connections are on the right. The four inner rods provide the quadrupole field for ion trapping while the outer four rods provide the DC patch potential compensation fields. Note that the barium oven and skimmer plates have been removed in this picture.

set to -5.00 dBm. The minimum RF voltage required for ${}^{138}\text{Ba}^{+}$ trapping stability as a function of endcap DC voltage is shown in Fig. 13.4.

The trap endcap rod electrode and trap compensation voltages are controlled using an analog amplifier circuit for each channel (App. A.3.3) at 400 V/10 V = 40x gain with analog voltages controlled via PC (App. C.3.4). An Acopian P01.5HA20MT power supply provides the HV for each circuit. Both endcap voltages are actively monitored via PC using 11.6 MΩ, 330 kΩ voltage dividers to analog input channels. All DC voltages are filtered before entering the vacuum chamber using CRC filters comprised of 1 μF, 1000 V capacitors and a 50 kΩ resistor to reduce RF pickup contamination on the DC supplies. The DC filter box and vacuum feedthru pin assignments are shown in Fig. 13.5.

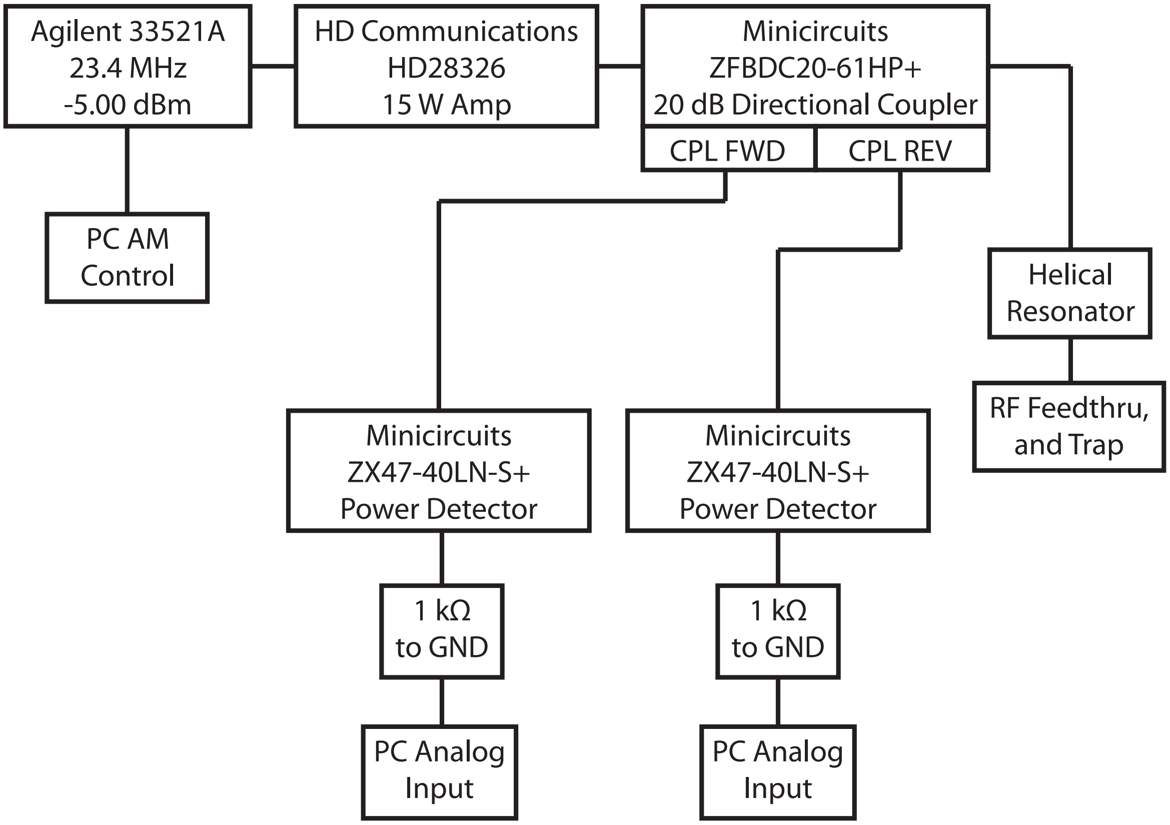


Figure 13.2. Single-ion trap RF drive block diagram. The power detectors monitor the RF power sent to (reflected from) the helical resonator. This allows the function generator to be tuned to resonance and the RF power and hence RF voltage on the trap RF rods to be monitored.

13.1.1. Operating Secular Frequencies

For an ion trap of this design, the axial secular frequency ω_z is given by

$$\omega_z^2 = \frac{2e\alpha_z V_z}{m z_0^2} \tag{13.1}$$

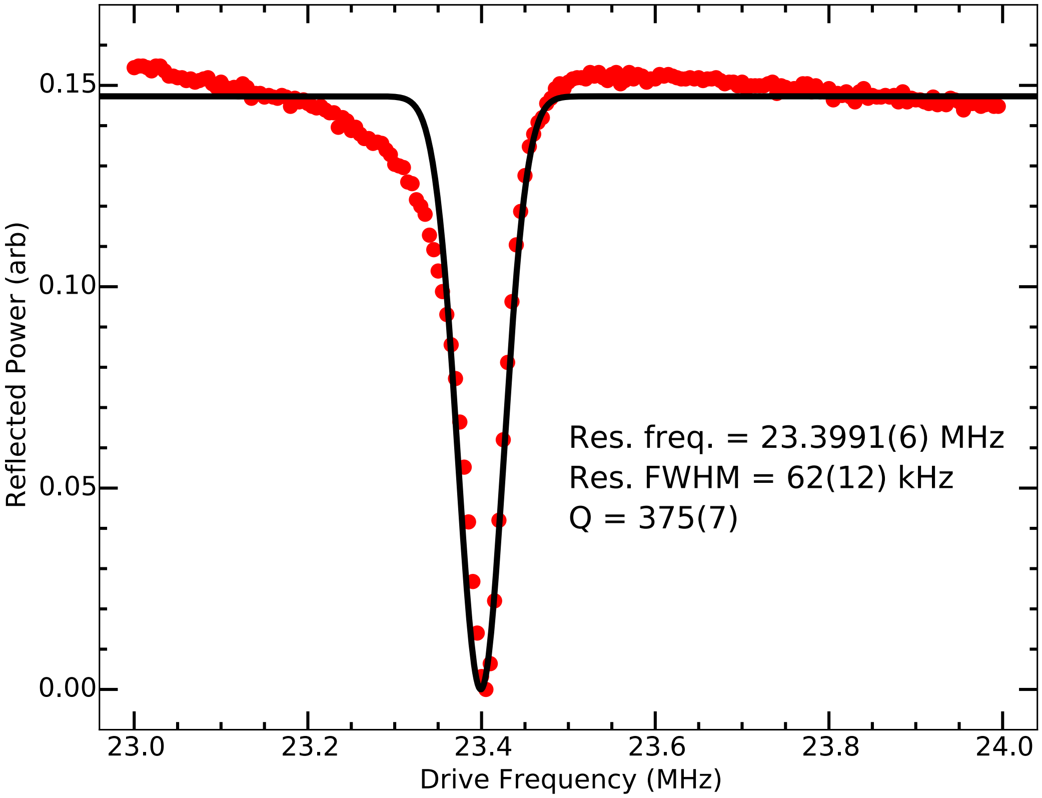


Figure 13.3. Helical resonator Q-factor measurement for the single-ion experiment. An inverted Gaussian fit function is used constrained to have a minimum at zero. The Q-factor is calculated as the resonance frequency divided by the resonator FWHM bandwidth. Note that this reanalyzed fit differs from YWL's previous result of Q = 414. However Q = 414 is used throughout this work, and α_{ac} compensates for this difference.

where e is the elementary charge, α_z is a geometric factor with typical value of $\lesssim$0.2, V_z is the DC voltage applied the endcaps, m is the mass of the ion, and $2z_0$ is the distance between endcaps. In the absence of the axial potential, the radial secular frequency ω_{r0}

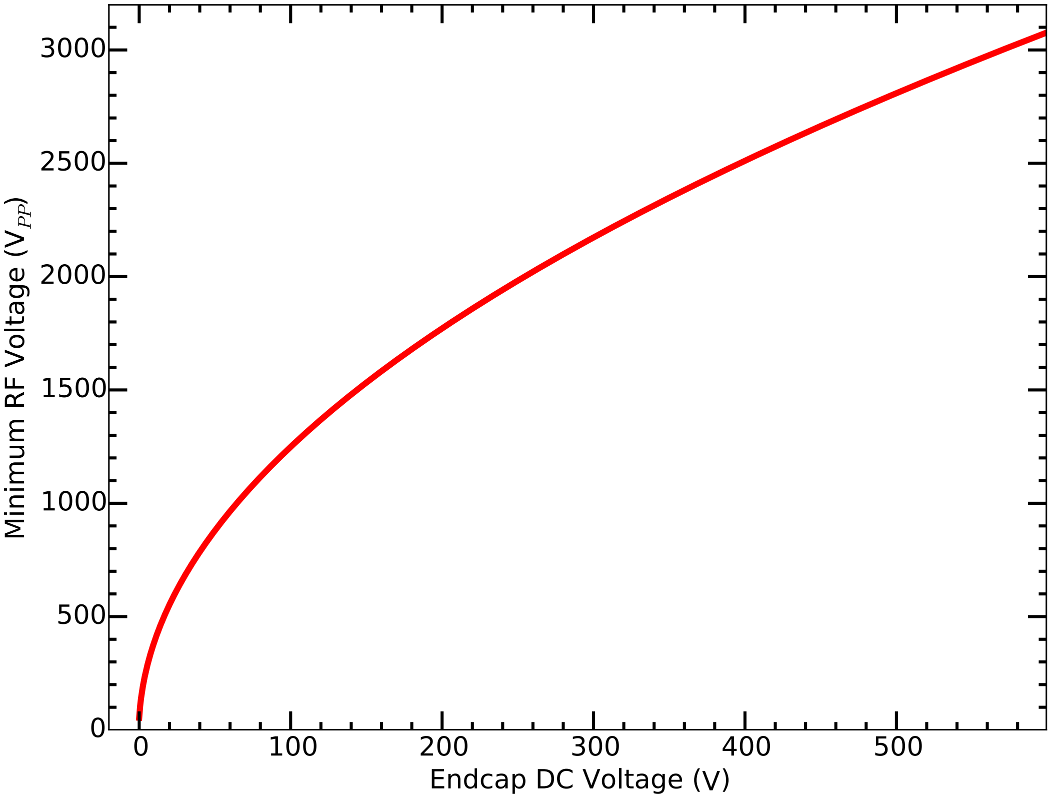


Figure 13.4. Plot of the minimum RF voltage required for stable trapping of ^{138}Ba$^+$ at a given endcap DC voltage.

is given by

$$\omega_{r0} = \frac{\alpha_{ac} e V_P}{\sqrt{2} m R^2 \Omega} \tag{13.2}$$

where α_{ac} is a geometric factor with typically value of ~1, V_P is the RF voltage amplitude at angular frequency Ω applied to the rods, and R is the distance from trap center to the surface of the RF rods. Due to the axial potential, the radial confinement is weakened

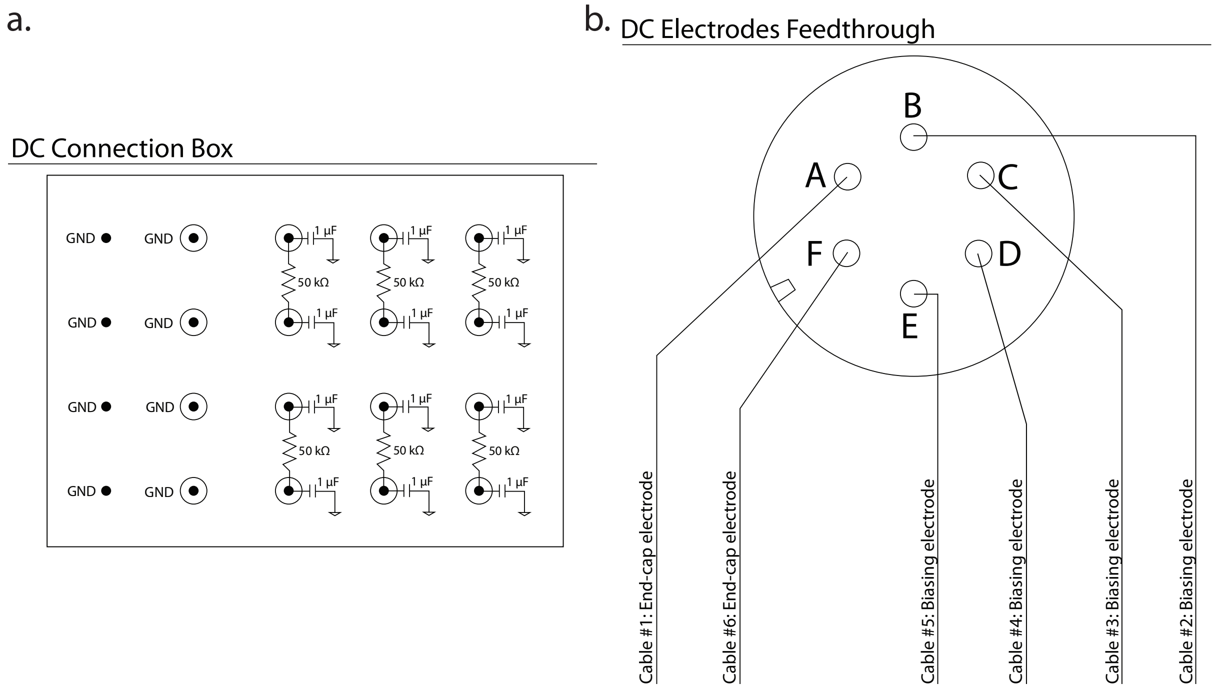

a.
DC Connection Box
GND
GND
GND
GND
GND
GND
GND
GND
1 µF
50 kΩ
b.
DC Electrodes Feedthrough
A
B
C
D
E
F
Cable #1: End-cap electrode
Cable #6: End-cap electrode
Cable #5: Biasing electrode
Cable #4: Biasing electrode
Cable #3: Biasing electrode
Cable #2: Biasing electrode

Figure 13.5. (a.) the DC voltage filter box pinout and (b.) vacuum chamber feedthru pin assignments. Note that in (a.), the lower row capacitor on the upper right filter is non-functional as measured via multimeter.

slightly with the compensated radial secular frequency ω_r given by

$$\omega_r^2 = \omega_{r0}^2 - \frac{\omega_z^2}{2}. \tag{13.3}$$

Using the far-off-resonant π pulse detection method described in Ch. 15, the radial secular sidebands are observed. By recording the sideband frequency as a function of RF voltage, the value $\alpha_{ac} = 1.16$ (Table 13.1) is measured. This is larger than the expected value of 1; however, this compensates the artificially-high Q-factor used for the calculations; dividing Q = 414 by $\alpha_{ac} = 1.16$ results in a new Q-factor of 357, which is comparable to the Q-factor of 375 in Fig. 13.3. Expected radial secular frequencies using $\alpha_{ac} = 1.16$ and corresponding observed radial secular frequencies are listed in Table 13.2.

Table 13.1. Experimental trap radial secular frequencies for various operating RF powers. Averaging values, $\alpha_{ac} = 1.16$.

RF AM	Peak Location (MHz)	Exp. f_{ac} (MHz)	Trap Voltage (V_{PP})	α_{ac}
-3.4	-10.36	0.680	2830	1.14
-3.2	-10.05	0.990	3210	1.21
-3.0	-9.89	1.15	3590	1.18
-2.8	-9.79	1.25	3970	1.13

Table 13.2. Expected trap radial secular frequencies for various operating RF powers using the value of α_{ac} from Table 13.1.

RF AM	Trap Voltage (V_{PP})	Calculated f_{ac} (MHz)	Experimental f_{ac} (MHz)
-3.4	2830	0.707	0.680
-3.2	3210	0.921	0.990
-3.0	3590	1.11	1.15
-2.8	3970	1.29	1.25
-2.6	4341	1.47	
-2.4	4680	1.62	
-2.0	5410	1.93	
-1.6	5970	2.17	1.88

Prior to these measurements, the operating radial secular frequency is 0.990 MHz. Due to peak confusion on the far-off-resonant frequency spectrum between radial and axial secular modes arising from the Raman beam $\Delta\vec{k}$ not being perfectly aligned along the axial direction of the trap, the operating RF power is changed to a typical observed radial secular frequency of 1.88 MHz.

The far-off-resonant π pulse detection method is used to measure the typical axial secular sideband frequency of 1.14 MHz with 200 V supplied to the endcaps, which determines $\alpha_z = 0.165$.

13.1.2. Loading ^{138}Ba$^+$

The trap is typically operated under two sets of trap voltages corresponding to (1) loading the trap, and (2) conducting the motional ground state cooling experiments. Relaxing the trap secular frequencies assists in loading and Doppler cooling ions in the trap. Typically, the RF AM setting is -4.0 corresponding to 1.75 kV$_{\mathrm{PP}}$ on the RF electrodes with the endcaps held at 15 V. For typical experimental operation, the trap RF AM setting is -1.6 corresponding to >6.2 kV$_{\mathrm{PP}}$ on the RF electrodes, and the trap endcap rod electrodes are held at 200 V. The observed secular frequencies are $\omega_z = 2\pi \times 1.14$ MHz and $\omega_r = 2\pi \times 1.9$ MHz. Ion lifetimes in the trap without Doppler cooling are >48 hours.

A single ^{138}Ba$^+$ ion is loaded into the trap via $(1 + 1')$ resonance-enhanced multiphoton ionization of neutral barium emanating from an oven under trap center of the same design described in Sec. 8.9.1 with current supplied by an Agilent E3615A supply passed through a filter before entering the chamber. Although the collimation method only works in the low-flux limit when neutral-neutral collisions are negligible, two skimmer plates sit above the oven with collimation holes aligned with trap center [309, 310]. The supply is controlled via PC using LabVIEW (Appendix C.3.2). Similarly, a 791 nm ECDL (laboratory-built laser system versus the Toptica DL-Pro used in Sec. 8.9.1) drives the ^{1}S$_0$ → ^{3}P$_1$ intercombination line for isotope-selective loading [221] while a 310 nm UV LED provides the second photon energetic enough to surpass the ionization threshold [222]. The UV LED is a SET Inc. UVTOP-310TO39BL mounted in a ThorLabs S1LEDM mount and driven by a laboratory-built diode laser current driver (App. A.1.3) with TTL/interlock circuit (App. A.1.10). The modulation and PZT sections of the current driver are removed, and the circuit is powered by the same supplies as the 455 nm laser system and 650 nm injection-locked laser system when used.

The flux of neutral barium from the oven could in the future be measured by laser-induced fluorescence. To perform this measurement, the 493 nm band-pass filter in the imaging system (Sec. 13.2) would be replaced with a 791 nm bandpass filter. Here, the 791 nm AOM would switch off with the PMT gated with a short delay. Since the intercombination line is long lived, any excited barium atoms would emit a single 791 nm photon on average long after the residual laser light scatter diminishes. This fluorescence, collected on the PMT, can then be plotted versus the 791 nm lock wavelength to find the center of the intercombination line, and the total counts at resonance can be used to estimate the total barium flux from the oven.

13.1.3. Micromotion Compensation

As discussed in Sec. 8.1.1, excess micromotion must be minimized. Since the ion position method (Sec. 8.1.1.1) is not sufficiently sensitive to minimize the micromotion in this tight trap, a different technique is used to quantify the amount of excess micromotion.

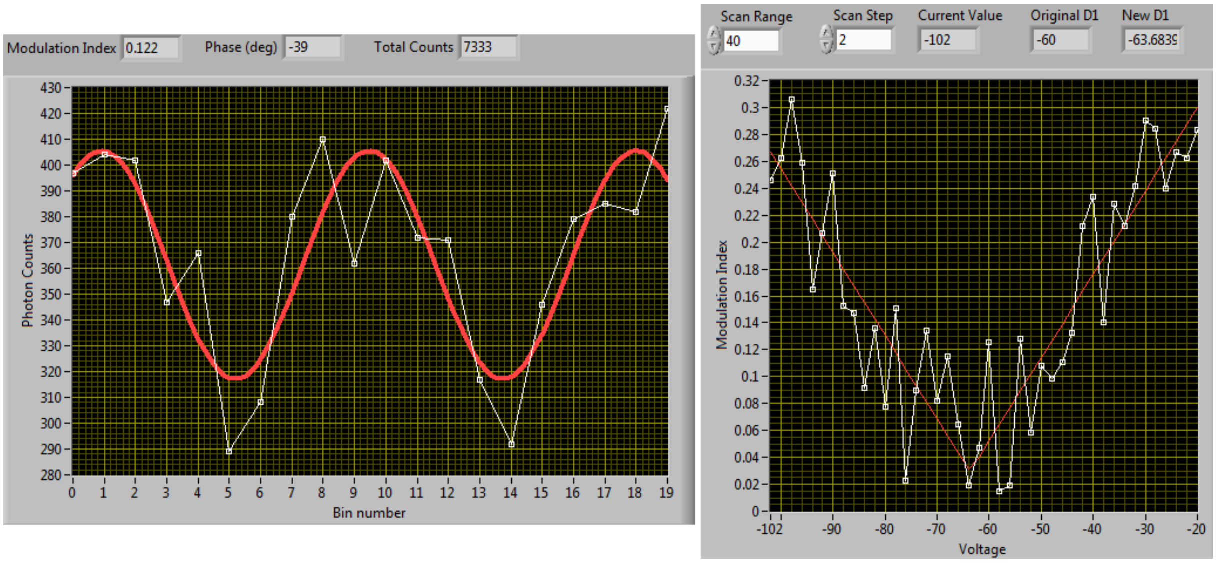
13.1.3.1. Sideband Amplitude Reduction. This method is applicable when the trap drive frequency $\Omega \gg \gamma$, where γ is the power-broadened linewidth of a narrow atomic resonance, i.e. a quadrupole transition or tailored Raman resonance. In this case, the excitation spectrum of the ion develops sidebands at integer multiples of Ω while the strength of the carrier decreases [303]. Patch potential compensation can be performed by minimizing the amplitude of the micromotion sidebands and maximizing the strength of the carrier. Additionally, the imparted AC Stark shift from the trap drive can also be used if the atomic resonance is sufficiently narrow, i.e. the 1.76 μm $^2\mathrm{S}_{1/2} \rightarrow {}^2\mathrm{D}_{5/2}$ quadrupole transition in Ba$^+$ [135].

13.1.3.2. RF-photon Correlation. An ion undergoing excess micromotion experiences a first-order Doppler shift $-\vec{k} \cdot \vec{V_\mu}$ where $\vec{k}$ is the wave vector of the laser used to excite the ion and $\vec{V_\mu}$ is the ion velocity due to excess micromotion. In the low intensity limit with a laser detuning of $\Gamma/2$ from resonance for the lowest Doppler-cooling temperature and under the condition of low micromotion amplitude, the detected fluorescence rate is [303]

$$\frac{R_d}{R_{\max}} \simeq \frac{1}{2} + \frac{\beta\Omega \sin(\Omega t + \delta)}{\Gamma} \tag{13.4}$$

where $\beta\Omega$ is the modulation amplitude or modulation index and δ is the phase of the fluorescence modulation signal due to excess micromotion. The first term in Eq. 13.4 is the fluorescence rate of the ion while being Doppler cooled. The second term is the modulation of the fluorescence rate due to the excess micromotion with amplitude $\beta\Omega$ that occurs at the trap drive frequency Ω. Therefore, if the fluorescence rate is monitored as a function of the phase of the trap drive, the value of $\beta\Omega$ can be extracted. Additionally, this amplitude can be minimized by scanning the ion trap DC compensation voltages.

Micromotion compensation is performed using this technique. A SRS SR400 gated photon counter is used with the RF-drive function generator sync signal (frequency divided by 32x) and PMT output. Because the SR400 does not gate incoming photon arrival times, it is instead programmed to scan a time bin after the sync pulse of the RF drive. A histogram is then built bin-by-bin. While significantly slower than timing each photon arrival relative to the RF phase, this method is sufficient to extract a micromotion amplitude from the data. The modulation index is recorded at each compensation



Modulation Index 0.122 Phase (deg) -39 Total Counts 7333

Scan Range 40 Scan Step 2 Current Value -102 Original D1 -60 New D1 -63.6839

Figure 13.6. A typical RF-fluorescence-correlation spectrum (left) and typical compensation voltage scan (right). For the voltage scan, a correlation spectrum is taken at each compensation voltage, modulation index recorded, and resulting data plotted with a fit yielding the minimum modulation index and optimum compensation voltage.

voltage as it is swept (two radial, one axial), and a fit function determines the minimum of the modulation index. The LabVIEW code for this is located in App. C.3.6, a typical RF-fluorescence correlation scan and typical voltage scan is shown in Fig. 13.6, and a table of compensation voltages is listed in Table 13.3.

13.2. Imaging System

The optical system for the experiment, assembled by YWL, is shown in Fig. 13.7. The objective, a +40 mm aspheric lens, sits below the top surface of the upper chamber viewport flange via a re-entrant window. A removable 150 μm pinhole after the first lens pair dramatically reduces stray background light that reaches both the camera and the PMT. After this pinhole is another lens followed by a 493 nm bandpass filter and

Table 13.3. Trap compensation voltages for various operating conditions. D1 and D2 are the radial compensation values and z is the axial compensation value. All units are in volts. The geometric axial compensation voltage is 7.0 V.

RF AM	Endcap (dc)	D1	D2	z	Notes
-4.0	15	-75.0	-100.0	7.0	Typical loading conditions.
-4.0	15	-75.0	-101.2	9.0	
-3.6	15	-67.5	-101.9	9.2	
-3.2	15	-75.6	-113.1	7.0	
-4.0	30	-71.2	-102.1	9.7	
-3.6	30	-72.4	-101.0	9.3	
-3.2	30	-72.3	-110.9	7.0	
-4.0	60	-65.7	-98.2	9.3	
-3.6	60	-62.9	-102.5	8.3	
-3.2	60	-61.6	-107.6	7.0	
-3.6	130	-39.7	-103.6	6.7	
-3.2	130	-38.5	-104.9	5.4	
-3.2	200	-6.1	-108.5	1.86	Typical operating conditions prior to Sec. 13.1.1 work.
-2.8	200	-10.3	-106.7	3.86	
-2.4	200	-17.4	-103.4	3.36	
-2.0	200	-21.2	-99.5	5.36	
-1.6	200	-58.3	-106.5	-4.79	Typical operating conditions.

removable/swappable NPBS (a 50:50 is used, but a 90:10 is readily swappable). This splits the optical system image to a Hamamatsu H8259-01 (S/N 57500415) photon counting head (PMT) and an Andor Luca S EMCCD. Each arm has separate imaging optics after the beamsplitter.

LabVEIW is used for EMCCD camera control (App. C.3.3). For a time, a TCP/IP data server within the program output the camera counts within the ROI, allowing for data transfer from LabVIEW to Python before all programs are modified to exclusively use the PMT. For experimental PMT gating, and AND gate is used with the camera trigger signal; the camera triggers on the rising TTL edge allowing for the PMT gate to

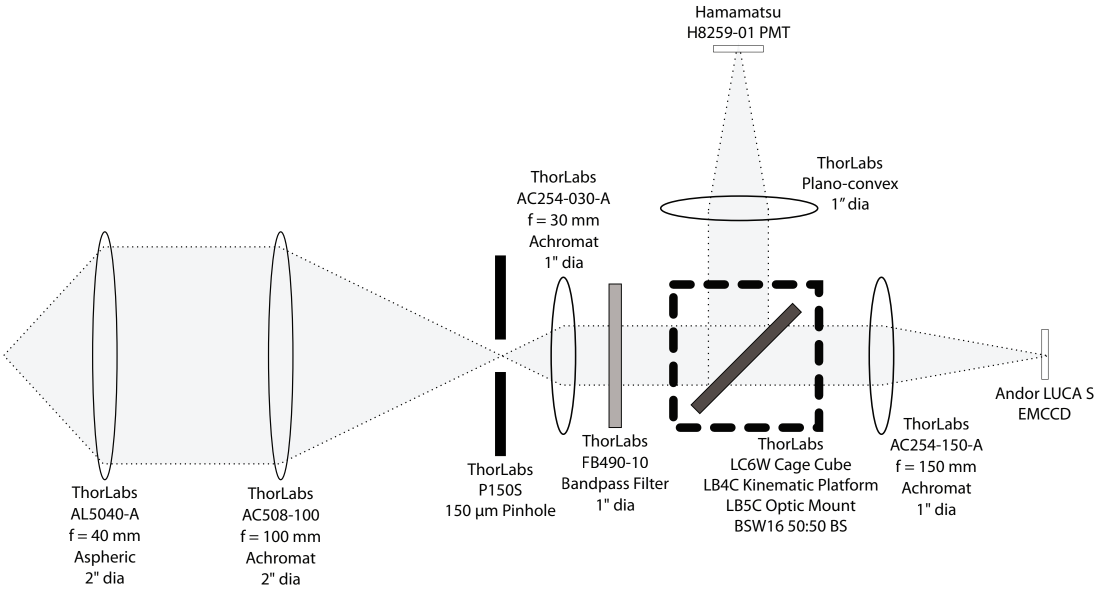


Figure 13.7. Single-ion experiment optical imaging system. The system is mounted vertically on an aluminum plate with L-brackets fixing the optical cage assembly between the first two achromats and immediately above the cage cube. The lens imaging the ion to the PMT is either a +100, +75, or +50 mm spherical singlet; because the active area of the PMT is large, the lens focal length and focusing are not critical. The entire optical system is mounted to a ThorLabs TP3 XYZ translation stage.

be arbitrarily long. To enable micromotion compensation, the PMT signal is also sent directly to the SR400.

13.3. Laser Optical Systems

A schematic of the laser geometry into the vacuum chamber is shown in Fig. 13.8. Each beam group excluding the UV LED source is combined on dichroics, 50:50 NPBSs and/or PBSs with individual position and pointing steering mirrors for each beam path. Each beamline is then focused into the vacuum system by a ThorLabs LA1986-A, $f = 125$

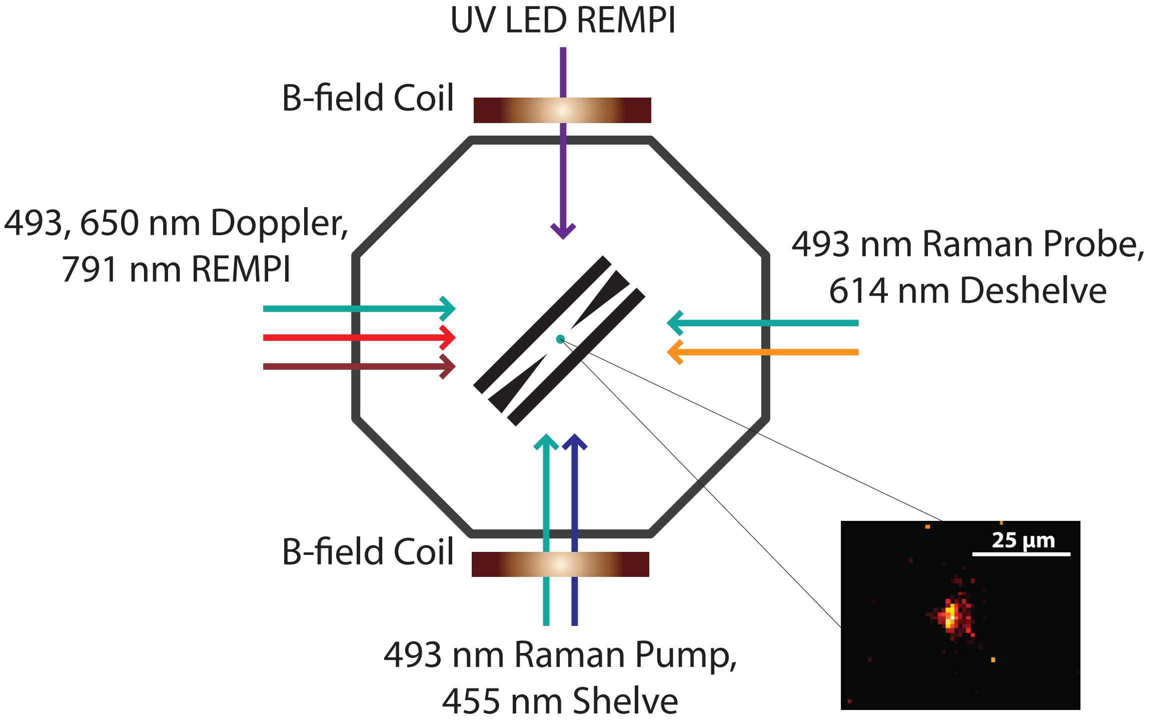


Figure 13.8. Apparatus and beam geometry. Note that the magnetic field is parallel to the σ^+-polarized Raman pump beam and perpendicular to the π-polarized Raman probe beam and that $\Delta\vec{k}$ lies along the trap z-axis. Inset: an EMCCD image of a single fluorescing ^{138}Ba$^+$ in the ion trap.

mm lens on a Newport 460A-XYZ three-dimensional linear translation stage using SM-13 micrometers. The Doppler and 791 nm beams are defocused from the ion position to aid in Doppler cooling unlocalized ions. Each Raman beamline is focused on the ion to both provide the maximum intensity at the ion position and improve beam position drift stability. The polarization of each Raman beamline is cleaned with a PBS. For the σ^+-polarized near-resonant and far-off-resonant Raman pump beams, a 493 nm QWP is immediately after the PBS. All waveplates on the apparatus table are zero-order from Tower Optical.

The 791 nm isotope-selective photoionization laser source enters the apparatus area via a ThorLabs 780HP SM fiber. A ThorLabs F240FC-B fixed-focal-length fiber collimator roughly collimates the output. The beam then passes through an SRS SR475 shutter and an IntraAction ATM-801A2 AOM in a (+1)-order single-pass configuration for switching driven by an IntraAction DE-802T driver. Both the shutter and AOM are triggered using a NI digital I/O line via LabVIEW. The beam then passes through an iris, is reflected by two steering mirrors, and is focused by a ThorLabs LA1484-B, $f = 300$ mm lens. This lens forms a telescope with the lens focusing the Doppler beams into the vacuum system, providing a roughly-collimated, large 791 nm beam through the center of the ion trap.

The 650 nm $^2\mathrm{D}_{3/2} \rightarrow {}^2\mathrm{P}_{1/2}$ Doppler cooling repump source enters the apparatus area via a ThorLabs PM630-HP PM SM fiber. A fiber-to-fiber coupling scheme is then employed using collimators, a HWP, and steering mirrors; this PM630-HP fiber patch cable ensures laser alignment through the subsequent optical arrangement regardless of the 650 nm laser source. A ThorLabs CFC-8X-B collimates the output of the on-table fiber. The beam polarization is then cleaned with a HWP and PBS, and an AR-coated, 5 mm thick window samples the beam to provide power stabilization. This is performed using a ThorLabs SM05PD1A photodiode behind a LA1540-A, $f = 15$ mm lens to focus both reflection spots on the detector, a 100 kΩ TIA (App. A.3.6), a PI servo loop (circuit in App. A.3.4, enclosure in App. F.3.4), and the double-pass AOM. This AOM, an IntraAction ATM-801A1, is in the (-1)-order double-pass beam-walk configuration (Fig. 8.6), which additionally provides FM via PC and is driven by an IntraAction DE-801M driver. The beam then passes through a (+1)-order single-pass IntraAction ATM-2001A1 AOM for switching set to 230 MHz providing a net $\sim$+80 MHz frequency shift followed by a

SR475 shutter. The 200 MHz AOM is driven by a DE-2001 driver with internal Mini-circuits ZYSWA-2-50DR TTL RF switch between the oscillator and amplifier to provide switching of the RF.

The 493 nm ${}^2\mathrm{S}_{1/2} \rightarrow {}^2\mathrm{P}_{1/2}$ Doppler cooling and near-resonant Raman source enters the apparatus area via a ThorLabs PM460-HP PM SM fiber. A ThorLabs CFC-5X-A collimates the output, which is then polarization-cleaned using a HWP and PBS that are part of the double-pass AOM setup in the polarization configuration (Fig. 8.7). This AOM is driven by a DE-801M. An on-table PM460-HP fiber then provides spatial filtering. A CFC-5X-A collimates the output, the polarization is cleaned, and the power is stabilized similarly to the 650 nm beamline. The power is then split to the Raman AOMs in the double-pass polarization configuration for FM and switching and to the single-pass Doppler AOM for frequency shifting and switching. The former are driven by the DDS AOM driver system described in Sec. 13.5, and the latter is driven by a ME-801T. All AOMs in this beamline are IntraAction ATM-801A1, and each beamline is shuttered by a SR475 shutter. A block diagram of the optical setup is shown in Fig. 13.9. The Raman beams are then directed to the appropriate area of the chamber with the near-resonant Raman probe beam using a periscope.

The far-off-resonant 493 nm Raman beam similarly enters the apparatus area via a ThorLabs PM460-HP PM SM fiber. A fiber-to-fiber coupling arrangement allows switching between the far-off-resonant light and near-resonant light using a flip mirror for alignment purposes. A single-pass AOM and servo loop are used for power control, and each double-pass AOM in the polarization configuration allows for FM and switching. The single-pass AOM is driven by a DE-801M and double-pass AOMs by the DDS AOM

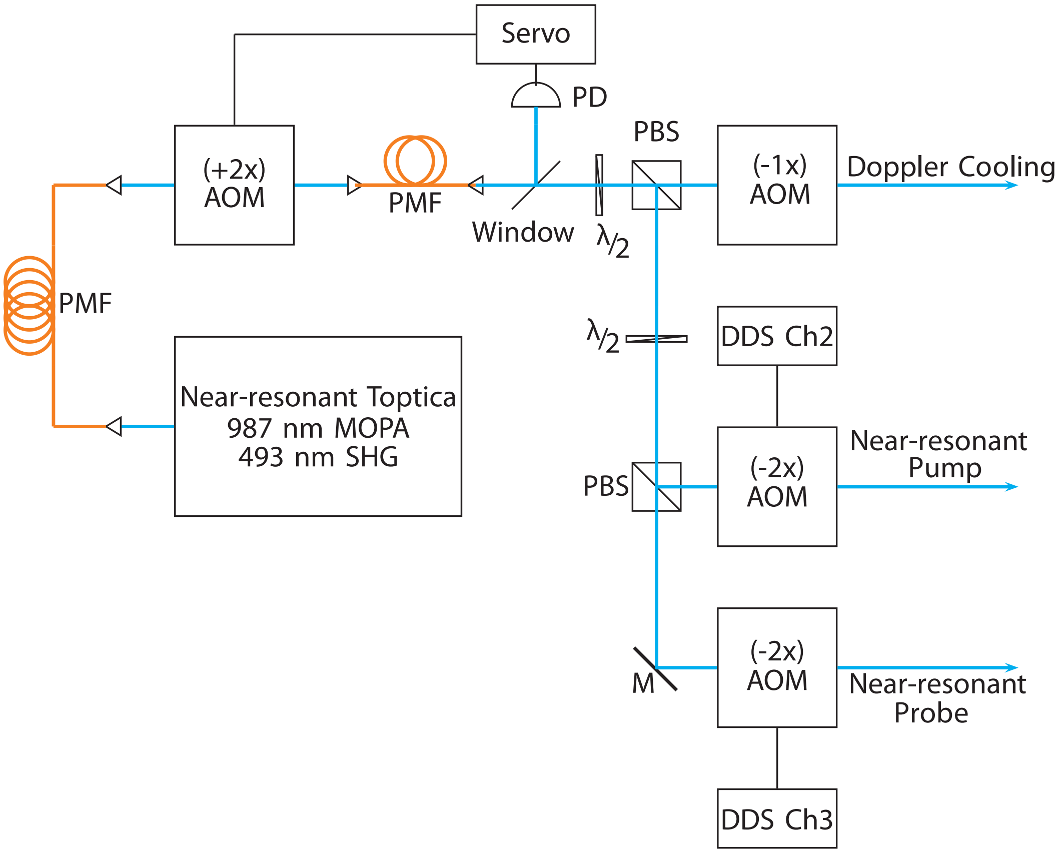


Figure 13.9. Simplified AOM setup for the near-resonant 493 nm beams. An on-table PM fiber (PMF) provides spatial filtering of the double-pass AOM output. An amplified photodiode (PD) and servo circuit control the laser power to the double-pass Raman AOMs. All AOM frequencies are ∼80 MHz.

system (Sec. 13.5). All AOMs in this beamline are IntraAction ATM-801A1, and the beamline is shuttered by a SR475 shutter. A block diagram of the optical setup is shown in Fig. 13.10.

The QWP that produces the circularly-polarized near-resonant and far-off-resonant Raman pump beams is aligned using a HWP and PBS; the QWP angle is optimized to minimize the effect of the HWP rotation angle on the power transmitted from the PBS;

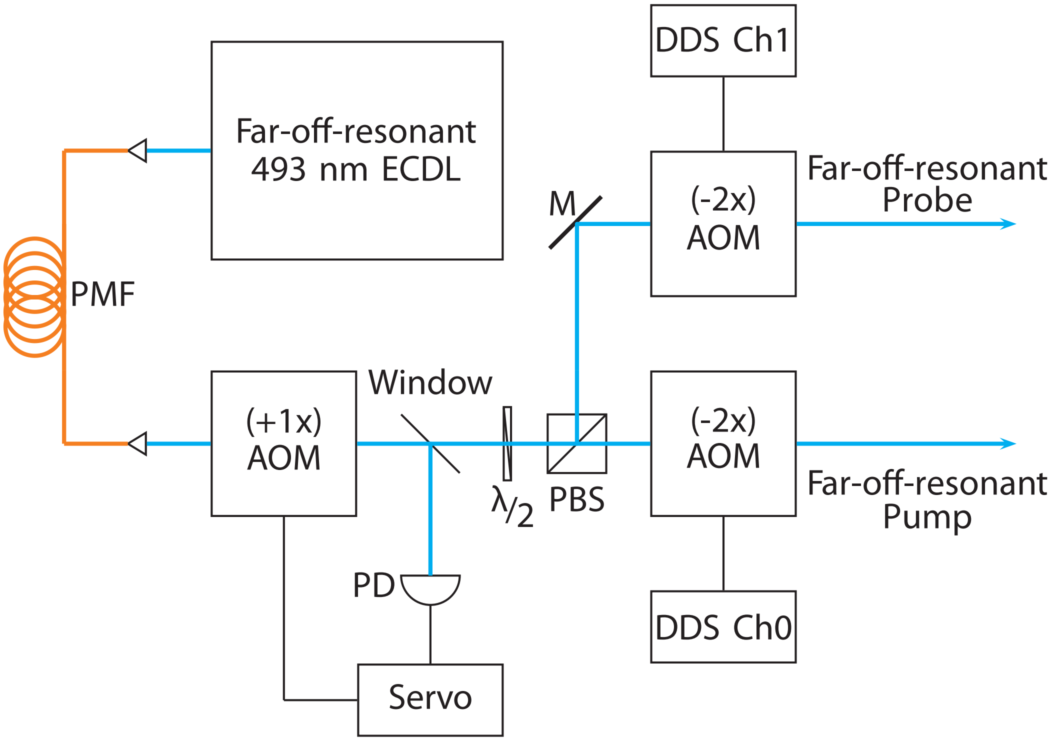


Figure 13.10. Simplified AOM setup for the far-off-resonant 493 nm beams. A free-space fiber-to-fiber coupling and set of flip mirrors (not shown) couple light from either the far-off-resonant source or before the Doppler (-1x) AOM in Fig. 13.9 to the PMF to the far-off-resonant (+1x) AOM. This allows alignment of the far-off-resonant beams using resonant light. An amplified PD and servo circuit control the laser power to the double-pass Raman AOMs. All AOM frequencies are ∼80 MHz.

when optimally circularly-polarized, the HWP rotation angle will not vary the transmitted power. Changing the polarization from right-handed to left-handed can be performed by simply reversing the QWP mount. Rotating the HPW through 2π only produces a variation of 0.8 μW out of a total incident power of 25.5 μW for a polarization extinction ratio of 30:1.

The 455 nm source is located on the apparatus optical table. The laser's output is power controlled with a PBS and HWP. The beam is then shuttered by a SR475 shutter and ATM-801A1 AOM in a (-1)-order single-pass configuration driven by a VCO channel on the DDS AOM system (Sec. 13.5) before being fiber coupled to the vacuum system via a SM400 SM fiber with CFC-5X-A output collimator. A HWP at the output assures that the transmitted power through the polarization cleaning PBS is at a maximum.

The 614 nm laser source enters the apparatus area via a ThorLabs SM600 SM fiber. A CFC-5X-A collimates the beam before a SR475 shutter and ATM-801A1 AOM in a (-1)-order single-pass configuration and driven by a VCO channel on the DDS AOM system (Sec. 13.5) are used for beam switching. A HWP at the output assures that the transmitted power through the polarization cleaning PBS is at a maximum.

Each beam group is first roughly aligned by overlapping the beams before and after the vacuum system. The translation stages are then adjusted to find the endcap rods, and the beams are overlapped optimally (excluding the 791 nm) by observing the transmitted light scatter. From the micrometer positions of each endcap, the trap center can be estimated. The 791 nm alignment is optimized by slightly adjusting its vertical and horizontal position to maximize the transmitted power through the vacuum system. Since the Doppler beams are defocused from trap center, an ion can be trapped, and the Doppler beam alignment optimized in situ using the translation stage and steering mirrors. For the Raman beam paths, the near-resonant Raman beams are first aligned to the ion using off-resonant fluorescence using the translation stages with an additional QWP before the vacuum system on the Raman pump beamline. The 493 nm far-off-resonant Raman beams are then aligned via steering mirrors using resonant light and optimizing ion fluorescence.

The 455 nm shelving and 614 nm deshelving beams are then assumed to be aligned with the ion and later verified by efficient shelving and deshelving.

13.4. IntraAction AOM Driver Modification

During the course of the experiments described in Chap. 15, leakage light through the 493 nm Doppler switching AOM is problematic when not using the shutters, causing the ion to undesirably scatter a few photons and optically pump to the $D_{3/2}$ state. Other channels have much smaller effects on the ion during experiments. This problem was temporarily solved by using the SR475 shutters; however, this greatly increases the experimental cycle time. The leakage light source is traced to poor RF drive extinction in the IntraAction T-option driver. The extinction provided by the Minicircuits ZYSWA-50-DR is 61 dB at 100 MHz while the ZASWA-2-50DR used by Webster [**150**] is 110 dB. Therefore, increasing the RF extinction should prevent the leakage light and undesired photon scattering.

To increase the RF extinction in the TTL-switched IntraAction drivers, two Minicircuits ZYSWA-50-DR switches are installed in series between the oscillator and amplifier (Fig. 13.11). Since the drivers are powered by a single-pole +24 V supply and the Minicircuits switches require ±5 V, a charge-pump circuit is used (App. A.3.5) to generate the negative polarity. This modification resolves the leakage light problems and permits experiments to collect data without using the SR475 shutters. Table 13.4 summarizes the observed RF power measurements before and after upgrading the RF systems with two Minicircuits ZYSWA-50-DR switches in series.

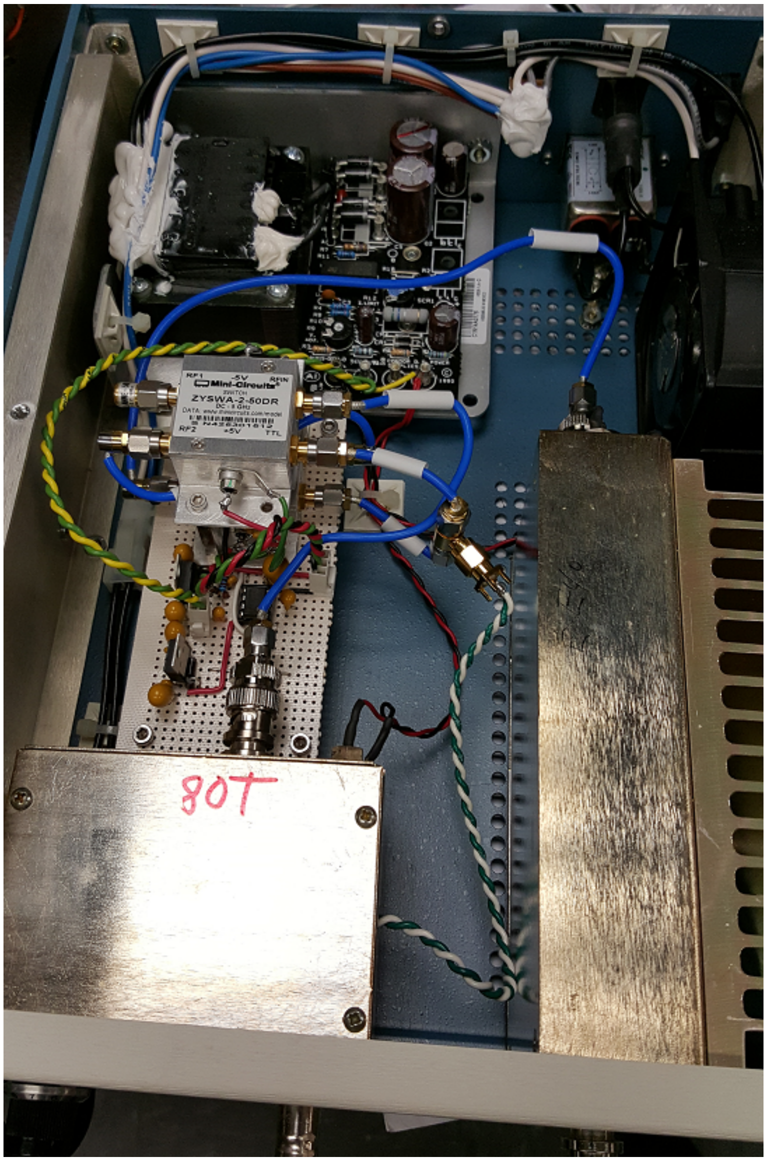


Figure 13.11. Picture of the modified IntraAction AOM driver. The two Minicircuits ZYSWA-50-DR switches (center) are installed in series between the oscillator (lower left) and amplifier (lower right). The charge pump circuit (App. A.3.5) provides the -5 V supply voltage from the +24 V internal supply (top left). Note that the oscillator must be opened to install the wiring to connect the TTL signal to the added switches.

Table 13.4. AOM driver RF extinction with TTL switching. All DDS output channels are similarly configured, so the RF extinction is similar. Note that the 650 nm Doppler uses a single Minicircuits ZYSWA-50-DR switch and is not upgraded. Measurements are taken using an Agilent N9320A spectrum analyzer.

AOM	Driver	Frequency (MHz)	Extinction (dB)	Upgraded (dB)
493 nm Doppler	DE-801T	80.00	50.0	100
650 nm Doppler	DE-2001	229.9	72.5	
493 nm Raman	DDS Ch. 0	74.46	86.8	120
455 nm Shelve	VCO Ch. 1	83.37	85.0	117
614 nm Deshelve	VCO Ch. 2	83.26	85.0	119

13.5. DDS AOM Driver System

To prevent decoherence in driving Raman transitions, each Raman beam pair must be phase coherent (fixed frequency difference between laser beams) by either implementing a phase lock loop between AOM drivers or using DDS technology. The former can be time-consuming to implement while the latter would provide more features from a single device (i.e. amplitude, frequency, and relative phase control of each channel), and has the added benefit of scaling by running multiple DDS cores on the same master clock. Because of the four Raman channels required, two for the near-resonant and two for the far-off-resonant Raman sources, and to minimize setup time, the four-channel Analog Devices AD9959 DDS evaluation board is chosen. Additionally, the evaluation board includes DACs for each channel, output RF filtering, onboard clock oscillator buffer and multiplier, and USB connectivity for PC interfacing.

Power for the system is provided by a custom Acopian power supply with ±5 V (support electronics and RF switches), +15 V (preamplifiers), and +24 V (amplifiers) outputs with a sufficient current overhead to ensure supply longevity. Each main component or subsystem of the circuit has individually switched power (support electronics, DDS board, RF switches, VCO, preamplifiers, and amplifiers). All enclosure drawings are in App. F.3.2.

13.5.1. Clocking the DDS Evaluation Board

In order to clock the DDS evaluation board, a crystal oscillator is chosen to avoid using an external clock source. In order to operate using a crystal oscillator, several modifications to the evaluation board are required; x2 (25 MHz, TCX Corporation 0C-25.000MAAJ-T), C66 and C67 (39 pF, 0805 package), and R52 and R53 (0 Ω shorts) are added to the board since they are omitted from the factory. Due to the added capacitive coupling of the clock lines to C21 and C22, these capacitors must be pulled up on the side facing the DDS chip (clear when looking at the evaluation board). The DDS evaluation board modifications are shown in Fig. 13.12. A simplified schematic of the evaluation board is in App. A.3.2. After switching the appropriate jumper to specify the crystal oscillator clock be used, the board should connect to the GUI programming software over USB if the modifications were successful.

13.5.2. RF Amplification, Switching, and Support Electronics

The output of each DDS DAC 200 MHz LPF filter output is first preamplified by a Minicircuits ZFL-500+ wideband amplifier before being switched by two Minicircuits ZYSWA-50-DR switches in series for >100 dB extinction. The signal is then amplified up to a maximum of 32 dBm by a Minicircuits ZHL-3A+ amplifier. Fixed-value SMA attenuators are placed throughout the system to assure that no amplifier operates in a

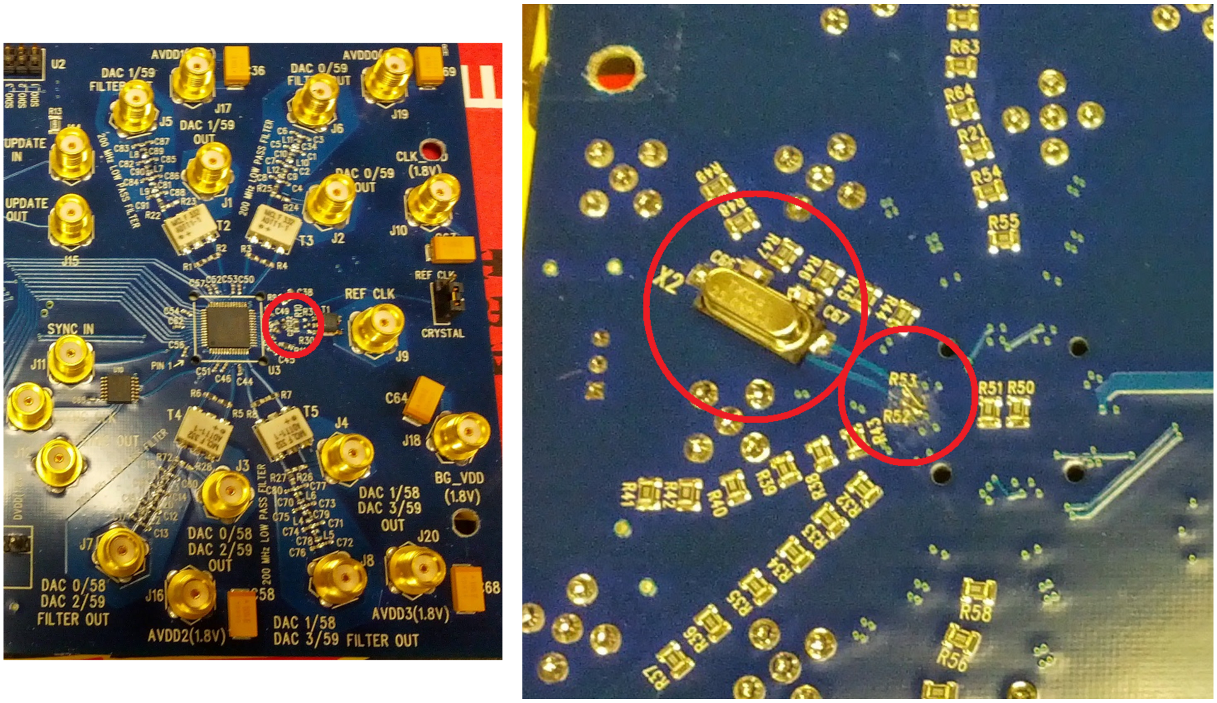


Figure 13.12. Picture of the modifications necessary to clock the DDS evaluation board with a crystal oscillator. The capacitors C21 and C22 must be pulled up on the side facing the DDS chip (red circle, left). X2, C66, C67, R52, and R53 must be installed on the underside of the board as well (red circles, right).

saturated regime and to limit the maximum output power of each channel to ≤32 dBm. Amplitude control is performed via the DDS. Pictures of the assembled AOM system are in Fig. 13.13. All preamplifiers and amplifiers use a 22 μF tantalum and a 100 nF ceramic capacitor in parallel for power supply filtering. All switches use only a single 100 nF ceramic capacitor per supply voltage for the same purpose.

In order to provide two AOM RF sources for switching the 455 nm shelve and 614 nm deshelve sources, the output of a Minicircuits ZX95-100+ VCO is split with a Minicircuits 1:2 splitter. Each output is then switched using two ZYSWA-50-DR in series before being amplified by a ZHL-3A+ amplifier. Without preamplifiers, the maximum output power

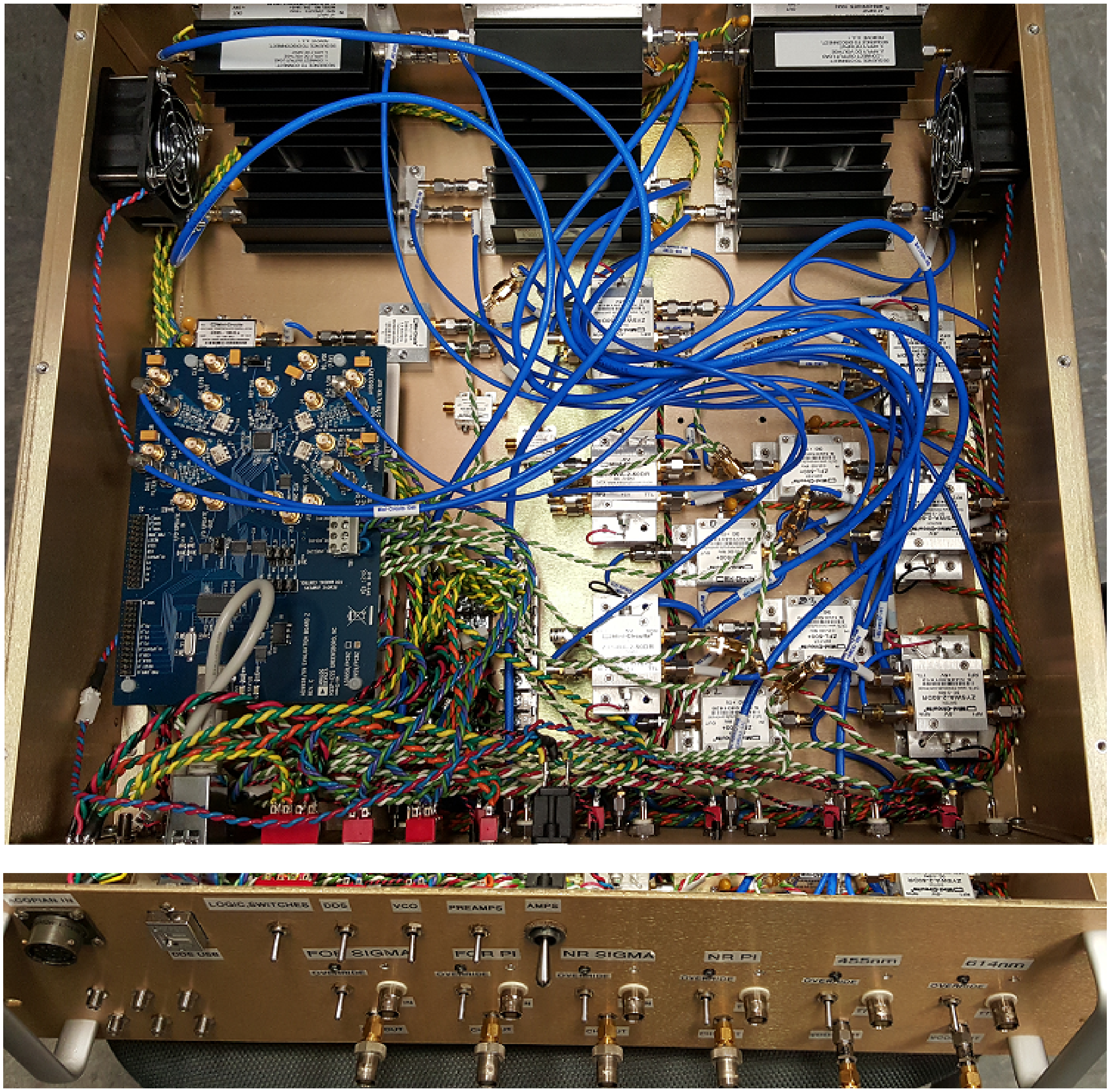


Figure 13.13. Pictures of the assembled DDS system. The DDS support circuit is located under the DDS evaluation board (left), all preamplifiers and RF switches are mounted directly to the case for heatsinking (right), all high-output amplifiers are positioned in the airflow of the push-pull fan pair (top), and the front panel arrangement is also shown (bottom).

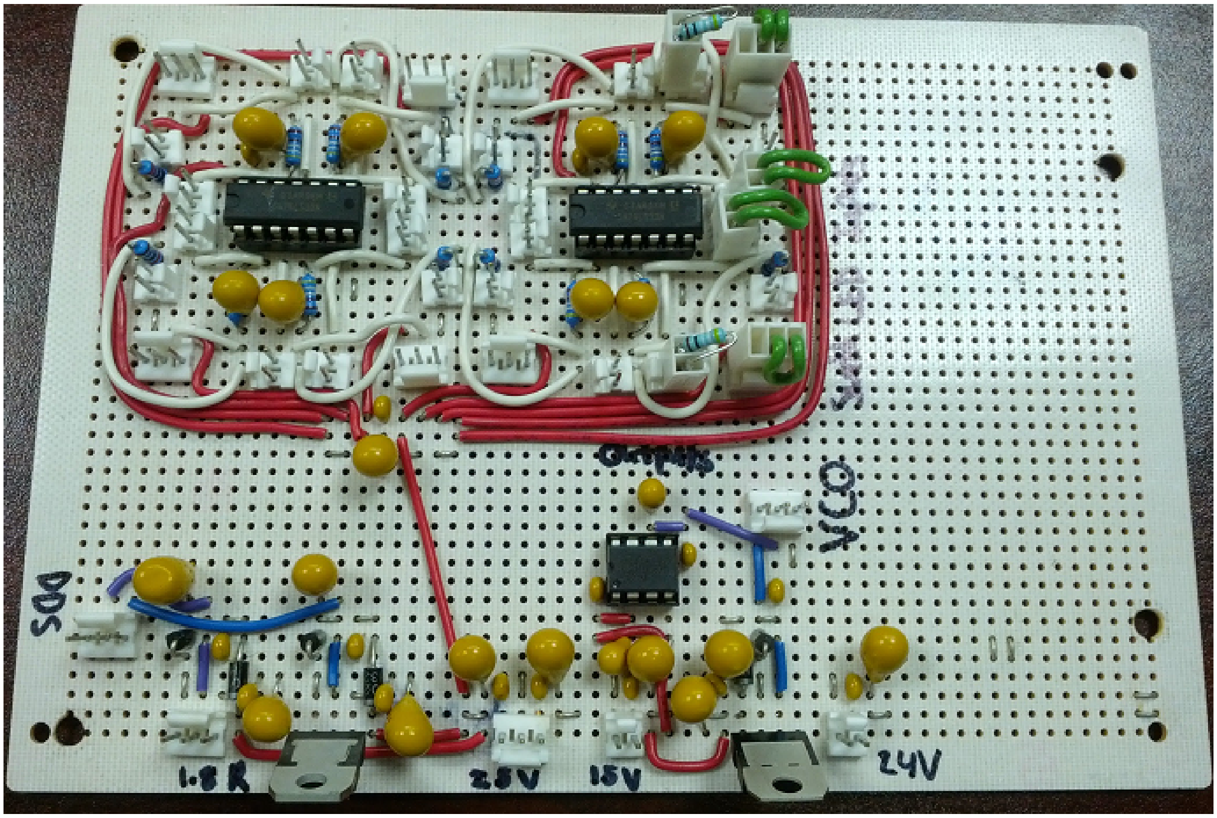


Figure 13.14. Picture of the DDS board support circuit. The DDS power supply regulators are in the lower left, input power is on the lower center, VCO supply regulator and control voltage are on the lower right, and all TTL logic channels are on the upper portion of the picture.

for each channel is ≈29 dBm. Power control for optimum AOM diffraction efficiency is provided by fixed value attenuators between the system enclosure and AOMs.

To allow operation of each channel without computer control, 2-channel OR gates are employed with an override switch on the system front panel as long as the associated TTL BNC input channel is shorted. The board additionally regulates the +5 V supply to +1.8 and +3.3 V for the DDS evaluation board, regulates the +15 V to +12 V for the VCO supply, and uses a REF01 +10 V reference for the VCO control voltage. A picture of this support circuit is in Fig. 13.14 with circuit diagram in App. A.3.1.

13.5.3. Configuration for STIRAP

The DDS system can be configured for outputting STIRAP pulses (attempts as a state readout technique are discussed in Sec. 16.3.3). In this configuration, the filtered output of each DDS channel is again preamplified. One channel then connects to the LO port of a Minicircuits ZX05-1LHW+ frequency mixer, and the RF output to a ZYSWA-50-DR to switch between either the mixer output or a single frequency tone from another DDS channel. The mixer's IF port is then connected to a SMA bulkhead on the front panel to receive a function generator waveform to generate the correct RF amplitude envelope. A second RF switch then switches the RF output on or off before the signal is amplified by a ZHL-3A+. In this case, the DDS only drives two ATM-801A1 AOMs rather than the four used with the far-off-resonant detection technique.

The function generators used to generate the waveforms are SRS DS345 with the following settings: 40 MHz frequency, 1 V_{PP} amplitude, positive input trigger source, arbitrary output waveform, burst sweep modulation, and sweep on. The pulse sequence is programmed using the LabVIEW code found in App. C.3.7 with pulse sequence triggered via PC. Fig. 13.15 shows the resultant optical pulses before the vacuum system measured using ThorLabs PDA10A amplified photodiodes.

13.6. PC control and Timing

All PC control is performed using the NI-DAQmx hardware platform with a PXI-1042 chassis and interface cards listed in Table 13.5. All PXI-6602 cards use a breakout box located in App. F.3.1.

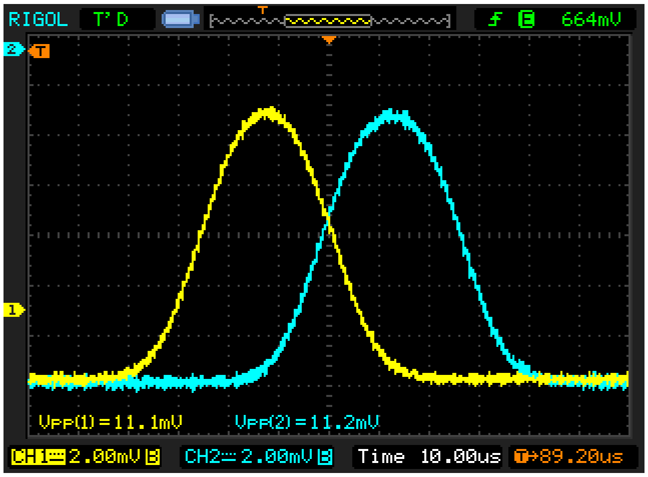


Figure 13.15. Near-resonant 493 nm STIRAP laser pulses as measured by amplified photodiodes immediately before the vacuum system. Yellow (blue) corresponds to the near-resonant Raman pump (probe) light. Here, the pulse amplitudes are comparable.

Table 13.5. National Instruments PXI I/O cards used for apparatus control

PXI Slot	Card	Function
1	PXI-8360	MXI-express for remote PCI control of PXI
2	PXI-6602	8-channel counter/timer with digital I/O
3	PXI-8430	High-performance, 2-port RS232 serial interface
4	PXI-6602	8-channel counter/timer with digital I/O
5	PXI-6722	Static and waveform analog output, 13-bit, 8 channels
6	PXI-6602	8-channel counter/timer with digital I/O
7	PXI-6704	Static analog output, 16-bit
8	PXI-6251	Analog input, 16-bit, 1 MS/s (multichannel), 16 channels

The LabVIEW environment is used for AOM and shutter switching while not collecting data and UV LED enable control (App. C.3.1); ramping the barium oven current (App. C.3.2); operating the EMCCD camera (App. C.3.3); setting the trap RF AM voltage (RF drive power) and DC voltages (App. C.3.4); setting operating laser conditions, i.e. laser power via PI servo set points and AOM driver FM voltage, and magnetic field current control voltages (App. C.3.5); micromotion compensation voltage scans along the two radial and axial axis (App. C.3.6); and STIRAP pulse timing and amplitude programming (App. C.3.7).

The Python environment (version 2.7) is used for collecting experimental data. Interfacing with the NI hardware is performed by generating a class to define the task, i.e. 6602 pulse sequence timing (App. E.2.1) and plotting (App. E.2.2), AI channels (App. E.2.3), AO channels (App. E.2.4), a data server for importing data from LabVIEW to Python (App. E.2.5), DDS control over USB (App. E.2.6), PMT pulse counting and timing with one or more 6602 channels (App. E.2.7), and a master trigger control using a 6602 digital I/O channel (App. E.2.8). Experimental timing for AOM pulses is performed using the 6602 counter channels, and AOM AM/FM shifting for fluorescence detection (Sec. 15.1) is performed with 6722 waveform analog outputs with the master trigger for all timing being a 6602 digital I/O channel.

13.7. Fixing Laser Parameters

For the experiments performed with this apparatus, six laser frequencies are set with five of the six actively locked. A ^{138}Ba^{+} level diagram (Fig. 13.16) shows the relevant transitions and laser wavelengths. Wavelength stabilization of all ECDLs is performed as described in Sec. 2.6.

13.7.1. 791 nm Photoionization

The 791 nm $^{1}S_{0} \rightarrow {}^{3}P_{1}$ intercombination line photoionization wavelength is previously measured using a Doppler-broadened absorption line in a Hamamatsu L2783-56NE-BA barium laser galvatron using optogalvanic spectroscopy [**200**, **201**, **202**, **204**, **205**, **119**,

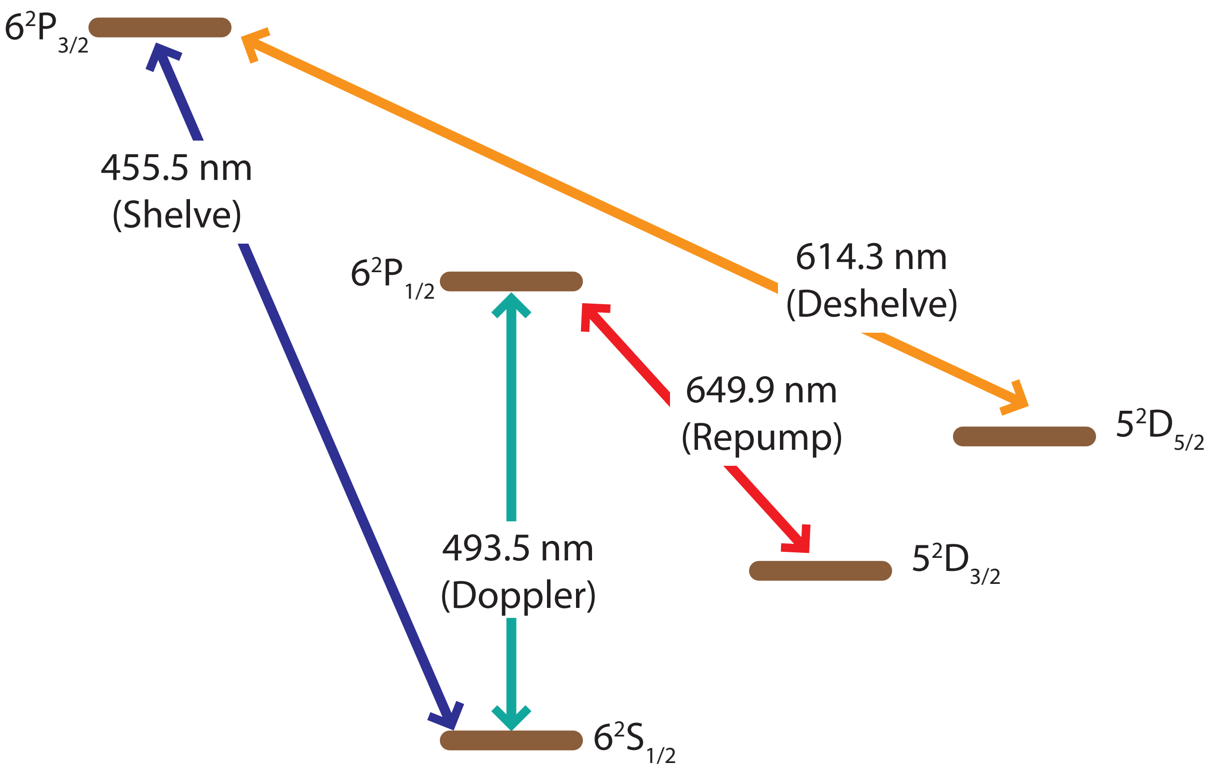


Figure 13.16. Relevant transitions in $^{138}\mathrm{Ba}^+$. All laser sources except the 455 nm free-running laser diode are ECDL systems locked by wavelength meter.

206]. The wavelength is later fine-tuned using the apparatus described in Part 2. The wavelength is further optimized using the apparatus discussed here by maximizing the loading rate and minimizing the number of loaded dark ions. Note however that trap loading is performed using >10 mW of 791 nm power, so the transition is substantially power broadened.

13.7.2. 493 and 650 nm Doppler Cooling

From other experiments in the laboratory, the 493 nm $^2S_{1/2} \rightarrow {}^2P_{1/2}$ and 650 nm $^2D_{3/2} \rightarrow {}^2P_{1/2}$ Doppler cooling wavelengths are known to within ∼10 MHz by $^{138}Ba^+$ fluorescence. Because this apparatus is significantly more sensitive to the Doppler cooling laser frequencies and powers, each is finely tuned using the fluorescence of a single $^{138}Ba^+$ ion collected on the PMT. The code that performs this (App. E.3.4) operates at two frequencies–first a set frequency for efficient Doppler cooling for 100 ms, the data point frequency for 100 ms to reach equilibrium, the PMT exposure for 100 ms begins while not changing the frequency, and the frequency then returns to the set Doppler cooling frequency. Each experimental frequency is repeated 10 times and each data point error is given as the standard deviation of the mean. Later Doppler fluorescence curves are taken with 10 ms exposures at each frequency (App. E.3.5) with data collection randomized and interleaved. Data analysis is performed via the code in App. E.1.1.

Prior to using the 650 nm repump light directly from the Toptica 650 nm DL100, the Sacher 650 nm injection-locked system is used with ∼10 μW of seed light (Sec. 2.5.6.3). Using this injection-locked source, the 493 nm laser lock point is set to place the Doppler fluorescence edge at 100 MHz and the 650 nm laser lock point is set to place the only visible CPT dip [**376, 377, 378**] at 90 MHz. To ensure that the Doppler laser powers do not contribute a significant amount a power broadening to the transition, increasing the temperature, the power of the 493 nm is swept with the 650 nm repump power held at 30 μW (Fig. 13.17). From the normalized data, power broadening is clearly a concern for powers of $\gtrsim$8 μW. The 493 nm Doppler power is therefore set at 5.0 μW.

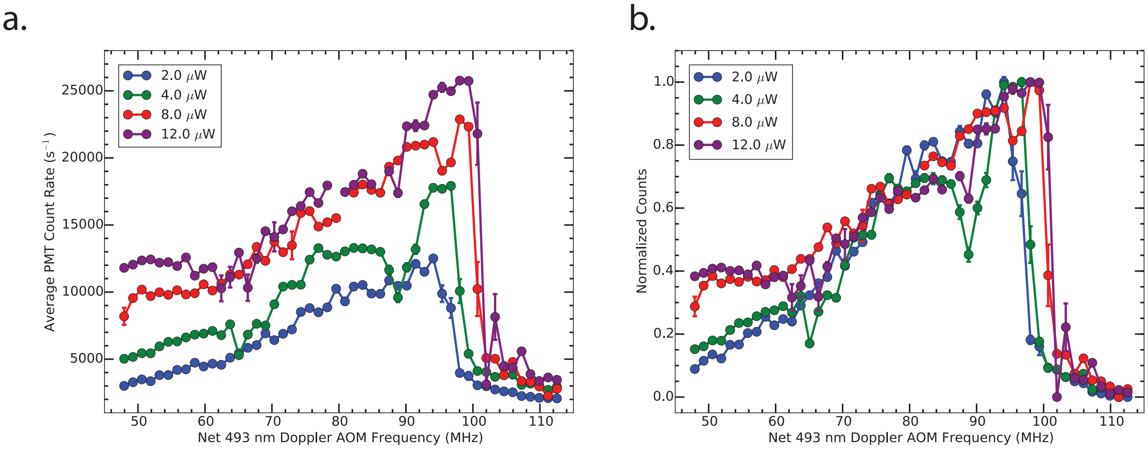


Figure 13.17. Determining the optimal 493 nm Doppler laser power and lock wavelength to address the ^{138}Ba$^+$ ^{2}S$_{1/2}$ $\rightarrow$ ^{2}P$_{1/2}$ transition. (a.) Average PMT count rate and (b.) normalized and background-subtracted count rate. These curves are used regularly to fix the 493 nm and 650 nm Doppler laser lock points to place the 493 Doppler fluorescence edge at 100 MHz and 650 nm CPT dip at 90 MHz. From the data, power broadening becomes a concern with 493 nm Doppler powers of $\gtrsim$8 μW.

Concerned with the possibility that the ASE in the injection-locked Sacher 650 nm laser increases our Doppler cooling temperature, the injection-locked system is replaced with light directly from the Toptica DL100 master. This improvement significantly increases the overall fluorescence and the observed number and contrast of the CPT dips (Fig. 13.18) due to the improved coherence between the Doppler cooling beams [**376**, **377**, **378**].

The same procedure as setting the 493 nm Doppler laser power is performed with the 650 nm power with the 493 nm Doppler laser power fixed at 5.0 μW (Fig. 13.19). From the normalized data, power broadening is clearly a concern for powers of $\gtrsim$13 μW. The 650 nm Doppler power is therefore set at 11.0 μW.

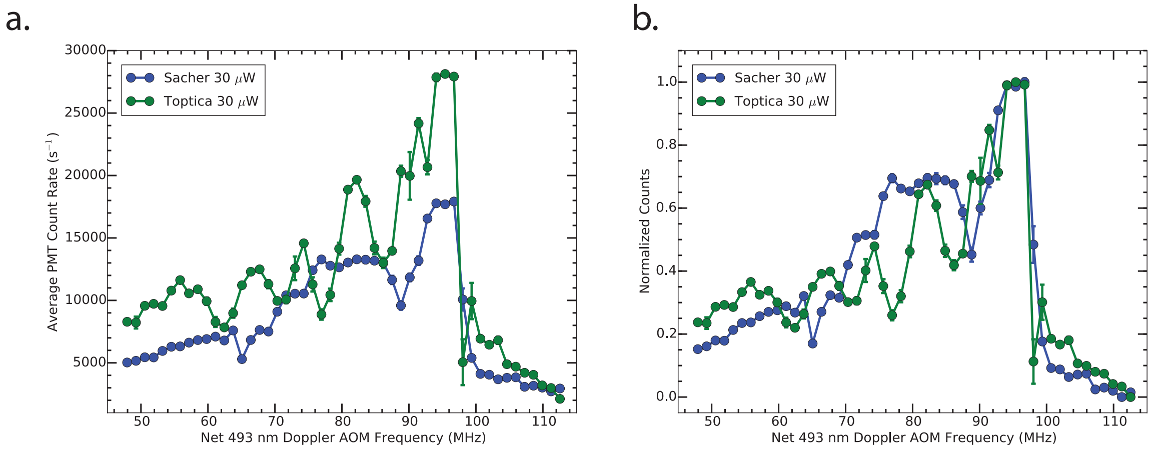


Figure 13.18. Comparing the 650 nm Doppler laser source to address the ^{138}Ba^{+} $^{2}D_{3/2} \rightarrow {}^{2}P_{1/2}$ transition. (a.) Average PMT count rate and (b.) normalized and background-subtracted count rate. Note that the significantly reduced ASE using the Toptica 650 nm laser's output directly allows multiple CPT dips to be resolved due to the improved coherence between the Doppler cooling beams. Both sets of data use 5.0 μW of 493 nm Doppler laser power.

13.7.3. Far-off-resonant 493 nm

Used as a tool to diagnose the near-resonant Raman sideband cooling (Chap. 14), the far-off-resonant 493 nm wavelength is set to -59 GHz from the $^{2}S_{1/2} \rightarrow {}^{2}P_{1/2}$ transition, limited by the small island of stability of this ECDL system. Although closer detuning ($\sim$30 GHz) would be beneficial for the experiments conducted here, no stable modes with closer detuning could be found. Although the large detuning permits the laser to operate unlocked (drifts of $\pm$1 GHz are fractionally small), the laser's small island of stability necessitates that this laser system be actively locked; a drift of the laser system by $\lesssim$1 GHz results in the desired laser mode crashing and laser source becoming broadband.

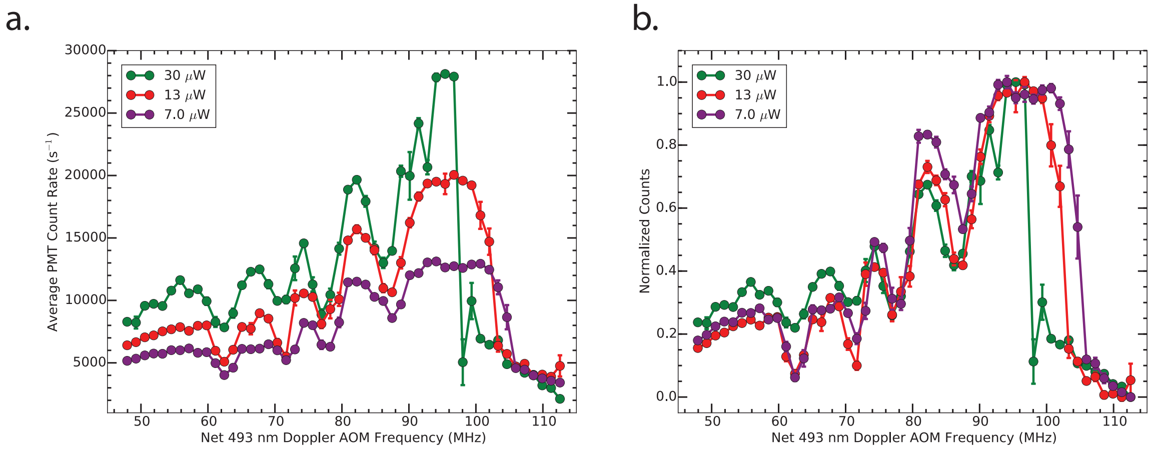


Figure 13.19. Determining the optimal 650 nm Doppler laser power and lock wavelength to address the $^{138}\mathrm{Ba}^+$ $^2D_{3/2} \rightarrow {}^2P_{1/2}$ transition. (a.) Average PMT count rate and (b.) normalized and background-subtracted count rate. These curves are used regularly to fix the 493 nm and 650 nm Doppler laser lock points to place the 493 Doppler fluorescence edge at 100 MHz and overlay the 650 nm CPT dips as in the figure. From the data, power broadening becomes a concern for 650 nm Doppler powers of $\gtrsim$13 μW.

13.7.4. 455 nm Shelving

The free-running 455 nm diode laser is tuned to be on resonance with the $^2S_{1/2} \rightarrow {}^2P_{3/2}$ transition using the drive current with the measured tuning of 26(12) GHz/mA (Sec. 2.5.2). Using low 455 nm power at a fixed 455 nm shelving exposure time with a single $^{138}\mathrm{Ba}^+$ ion (Sec. 15.1), sweeping the drive current of the laser while measuring the shelving probability (Fig. 13.20) provides the optimal drive current (76.83 mA), and therefore output wavelength (455.5310 nm [**199**]), for the laser system. Using a Gaussian fit to the data and assuming that the transition linewidth is infinitely narrow compared with the laser linewidth, the 455 nm laser linewidth FWHM is 51(2) GHz.

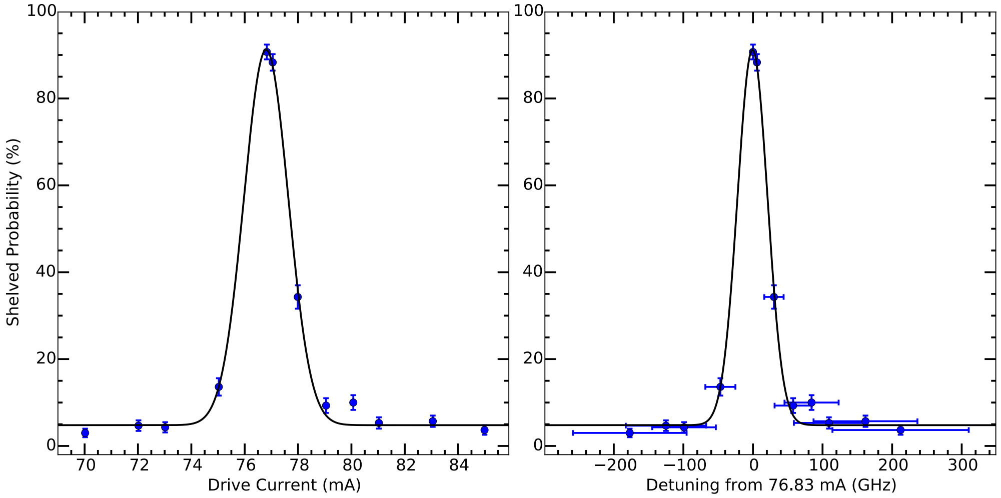


Figure 13.20. Determining the 455 nm free running laser diode operating current (output wavelength) to address the $^{138}\mathrm{Ba}^{+}$ $^{2}\mathrm{S}_{1/2} \rightarrow {}^{2}\mathrm{P}_{3/2}$ transition. The black line is a Gaussian fit to the experimental data (blue circles). From the fit, the 455 nm laser linewidth FWHM is 51(2) GHz.

13.7.5. 614 nm Deshelving

The 1228 → 614 nm ECDL is tuned to be on resonance with the $^{138}\mathrm{Ba}^{+}$ $^{2}\mathrm{D}_{5/2} \rightarrow {}^{2}\mathrm{P}_{3/2}$ transition using the method described above for the 455 nm shelving wavelength only by tuning the 614 nm wavelength lock point via LabVIEW nearby the expected 614.3413 nm [**199**]. Due to the 1228 nm ECDL multimoding (Sec. 2.5.5), a single, ∼10-MHz-wide resonance is not observed. Additionally, the deshelving efficiency is regularly checked to assure that the wavelength meter lock reads the correct laser mode, which is adjusted via the wavelength locking optics 614 nm zero-order HWP from Tower Optical.

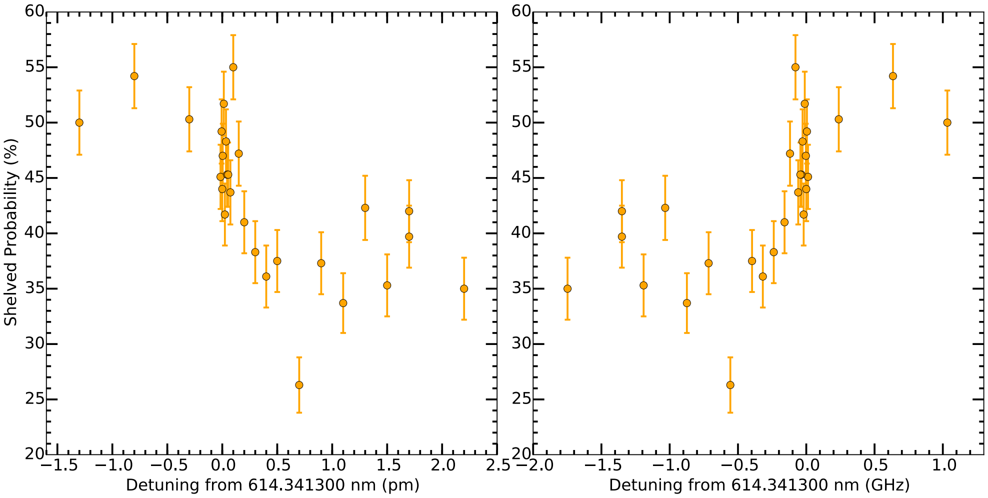


Figure 13.21. Determining the 614 nm lock wavelength to address the ^{138}Ba^{+} ^{2}D$_{5/2}$ → ^{2}P$_{3/2}$ transition. A single optimal lock wavelength is not observed due to the multimode behavior of the 1228 nm laser output (Sec. 2.5.5).

13.8. Magnetic Field and Alignment

The magnetic field that produces the Zeeman splitting in the experiment is aligned along the y-axis and is generated using a Helmholtz coil pair each with 120 turns and 4.5 inch mean diameter (Fig. 13.8). This field is trimmed in the x- direction with a single coil placed around the viewport in which the Doppler beams enter the vacuum system with a mean diameter of 3 inches and 75 turns wound on a form (App. F.3.3). The field is trimmed in the z- direction with a Helmholtz coil pair each with 80 turns and 7 inch mean diameter wound on a form (App. F.3.3) on either side of the vacuum system octagon. Acopian YL010MC400B158C67E23F1H3K37 10 V, 4 A supplies drive each axis and are computer controlled via external modulation (App. C.3.5).

To align the main magnetic field (y-axis) to be parallel to the circularly-polarized Raman pump beam, the off-resonant fluorescence of a single trapped ion while exposed to the near-resonant Raman pump beam is recorded as a function of the x- and z-axis trimming coil currents. When the near-resonant Raman pump beam and magnetic field are aligned, the beam will be maximally σ^+-polarized, and the spontaneous scattering rate will be minimized. To implement this trimming, the Python code in App. E.3.7 is used in which the trim fields are scanned one quadrant at a time. Polarity changes for each axis are performed by hand at the coil breakout box using the convention of positive polarity being red-to-red and black-to-black connections. For negative polarity, these are reversed.

At each trimming x- and z-axis current set point, the 650 nm repump beam is first shut off to collect 493 nm background photon counts using the PMT since the ion is optically pumped to the $\mathrm{D}_{3/2}$ state and is dark to the 493 nm near-resonant Raman pump beam. The 650 nm repump beam is then turned on, the 493 nm near-resonant Raman pump photon counts are collected, and the background counts subtracted to obtain the fluorescence counts of the ion. All PMT exposures are 500 ms, and the resulting data point uncertainty is calculated as the standard deviation of the mean of 10 repetitions.

A typical magnetic field trimming contour plot is shown in Fig. 13.22 and heatmap in Fig. 13.23 with the analysis and plotting code in App. E.1.6. From this data, the ideal control voltages to the x- and z-axis supplies are 3.0 and 0.2 V, respectively.

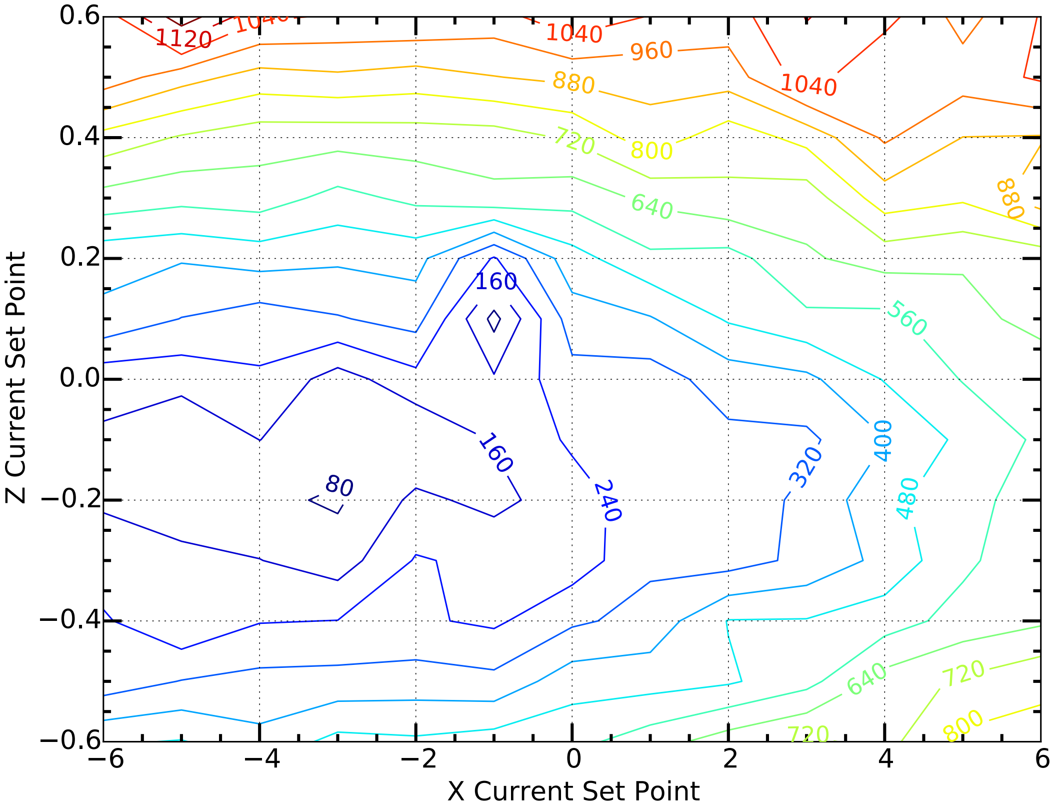


Figure 13.22. Contour plot of a single $^{138}\mathrm{Ba}^{+}$ ion fluorescence photon counts (contour-axis) over 500 ms of exposure from the near-resonant Raman pump beam. Note that the near-resonant Raman pump beam is maximally σ^{+}-polarized at x- and z-axis current set voltages of 3.0 and 0.2 V, respectively.

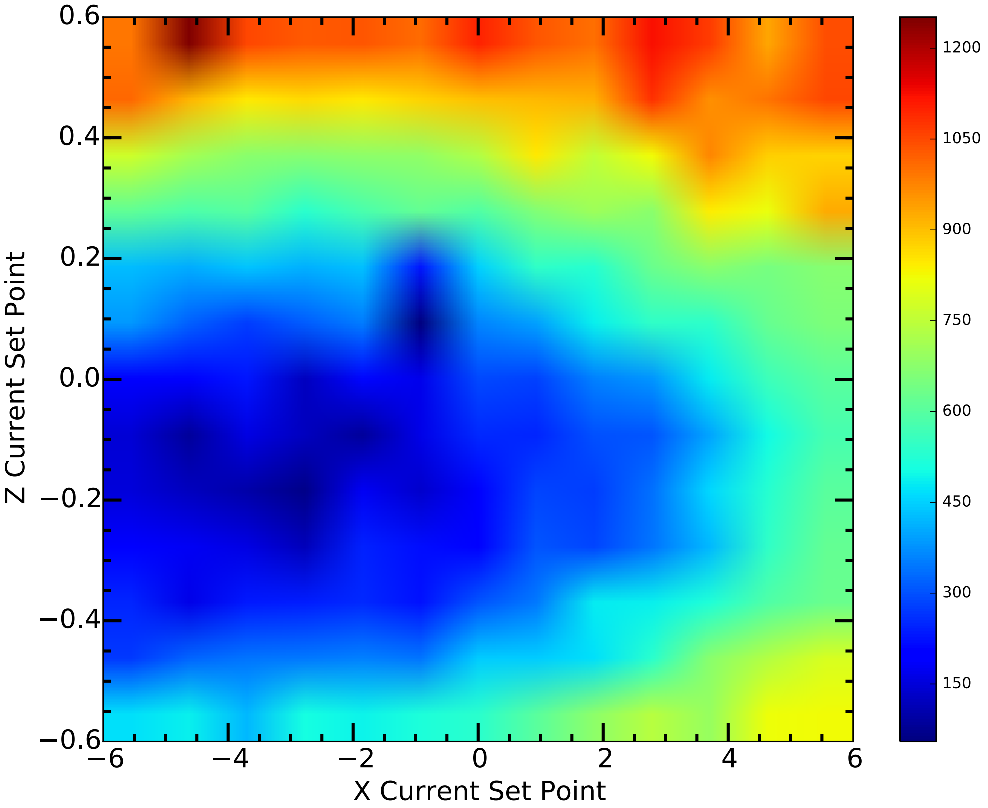


Figure 13.23. Heatmap of a single ^{138}Ba^{+} ion fluorescence photon counts (contour-axis) over 500 ms of exposure from the near-resonant Raman pump beam. Note that the near-resonant Raman pump beam is maximally σ^{+}-polarized at x- and z-axis current set voltages of 3.0 and 0.2 V, respectively.

CHAPTER 14

Continuous, Near-resonant Raman Sideband Cooling

This chapter describes the near-resonant Raman sideband cooling, which uses a pair of near-detuned lasers to cool the ion to the motional ground state of the ion trap with high probability. It begins with a theoretical discussion on near-resonant Raman sideband cooling and the connection of the cooling technique to Fano profiles. Using an expression for the steady-state ${}^2\mathrm{P}_{1/2}$ population, the theoretical minimum $\bar{n}$ and the cooling rate, $\frac{\mathrm{d}}{\mathrm{d}t}\bar{n}$ are calculated neglecting the ion trap heating rate. The near-resonant Raman sideband cooling is then experimentally confirmed, and the relevant experimental parameters are determined using the Fano profile function discussed in Sec. 14.1.2.

14.1. Raman Sideband Cooling Theory

This section discusses the theory behind the near-resonant two-photon Raman transitions used for sideband cooling and the connection between near-resonant two-photon Raman transitions and Fano profiles. Additionally, these derivations provide a method to calculate the minimum equilibrium mean motional occupation number $\bar{n}_{\mathrm{min}}$ and the cooling rate $\frac{\mathrm{d}}{\mathrm{d}t}\bar{n}$ as a function of Rabi frequencies and detuning.

14.1.1. Two-photon Raman Transitions

Consider a three-level atomic Λ system (Fig. 14.1) with two applied laser fields connecting the lower two levels to the upper state. In the interaction picture and under the rotating

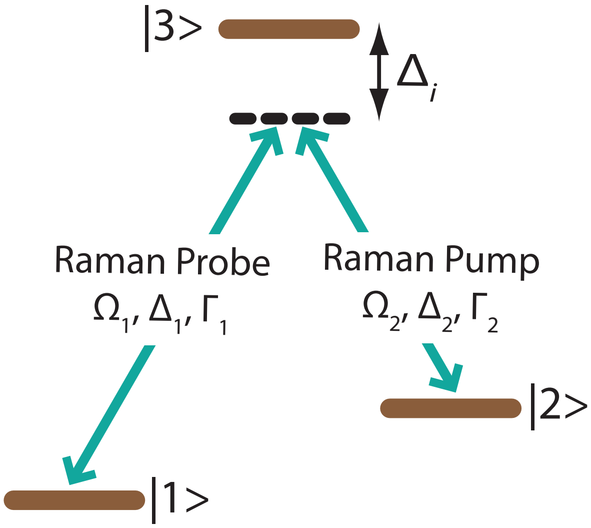


Figure 14.1. Levels relevant for the theoretical discussion of Raman transitions. $|1\rangle$ ($|2\rangle$) is coupled to $|3\rangle$ with Rabi frequency Ω_1 (Ω_2) with detuning Δ_1 (Δ_2) from resonance. $|3\rangle$ has a decay rate of $\Gamma = \Gamma_1 + \Gamma_2$. Note that the case shown is for $\Delta_{1,2} < 0$, but $\Delta_{1,2} \geq 0$ are other possibilities.

wave approximation, the optical Bloch equations of the three-level system with the two applied laser fields including decoherence rates are [379]

$$\dot{\rho}_{33} = -\Gamma\rho_{33} - \frac{i}{2}(\rho_{13} - \rho_{31})\Omega_1 - \frac{i}{2}(\rho_{23} - \rho_{32})\Omega_2, \tag{14.1}$$

$$\dot{\rho}_{11} = -\Gamma_1\rho_{33} + \frac{i}{2}(\rho_{13} - \rho_{31})\Omega_1, \tag{14.2}$$

$$\dot{\rho}_{22} = -\Gamma_2\rho_{33} + \frac{i}{2}(\rho_{23} - \rho_{32})\Omega_2, \tag{14.3}$$

$$\dot{\rho}_{13} = (-\Gamma_{13} - i\Delta_1)\rho_{13} - \frac{i}{2}(\rho_{33} - \rho_{11})\Omega_1 + \frac{i}{2}\rho_{12}\Omega_2, \tag{14.4}$$

$$\dot{\rho}_{23} = (-\Gamma_{23} - i\Delta_2)\rho_{23} - \frac{i}{2}(\rho_{33} - \rho_{22})\Omega_2 + \frac{i}{2}\rho_{21}\Omega_1, \tag{14.5}$$

$$\dot{\rho}_{12} = i(\Delta_2 - \Delta_1)\rho_{12} + \frac{i}{2}\rho_{13}\Omega_2 - \frac{i}{2}\rho_{32}\Omega_1 - \gamma\rho_{12} \tag{14.6}$$

where Γ is the decay rate of $|3\rangle$, Γ_1 (Γ_2) is the decay rate of $|3\rangle$ to $|1\rangle$ ($|2\rangle$), and Ω_1 (Ω_2) is the Rabi frequency of the driven transition between states $|1\rangle$ ($|2\rangle$) and $|3\rangle$ due to the "probe" ("pump") laser excitation. Γ_{13}, Γ_{23}, and $\gamma \equiv \Gamma_{12}$ are the decay rates of the coherences between the labeled states. If the coherence decay is only due to Γ and laser linewidths γ_1 and γ_2, then

$$\Gamma_{13} = \frac{1}{2}(\Gamma + \gamma_1), \tag{14.7}$$

$$\Gamma_{23} = \frac{1}{2}(\Gamma + \gamma_2), \tag{14.8}$$

$$\gamma = \frac{1}{2}(\gamma_1 + \gamma_2) \tag{14.9}$$

where the last equation is for two laser beams with independent dephasing. If the frequency difference is imposed by frequency shifters, i.e. AOMs, then γ is the rate of dephasing of the imparted frequency difference. Since we will ultimately use the same laser source and AOMs driven from similar RF sources, the linewidths of the two laser beams should be roughly equal and small compared to Γ such that

$$\Gamma_{13} = \Gamma_{23} = \frac{1}{2}\alpha\Gamma \tag{14.10}$$

or

$$\alpha = 2\frac{\Gamma_{13}}{\Gamma}. \tag{14.11}$$

Note that if $\gamma \ll \Gamma$, then $\alpha \simeq 1$.

Under the normalization condition

$$\rho_{11} + \rho_{22} + \rho_{33} = 1, \tag{14.12}$$

the steady-state population in $|3\rangle$ can be solved, and is given by

$$\rho_{33} = 2\Omega_1^2\Omega_2^2 \frac{2\alpha\Gamma(\delta^2+\gamma^2)+(\Omega_1^2+\Omega_2^2)\gamma}{c_0+c_1\gamma+c_2\gamma^2} \tag{14.13}$$

where $\delta = \Delta_1 - \Delta_2$ is the two-photon detuning from the dark resonance (when $\Delta_1 = \Delta_2$), and the coefficients in the denominator are given by

$$\begin{aligned} c_0 = 16\Omega_2^2\Gamma_1\Delta_1^2\left(\delta-\frac{\Omega_2^2}{4\Delta_1}\right)^2 + 16\Omega_1^2\Gamma_2\Delta_2^2\left(\delta-\frac{\Omega_1^2}{4\Delta_2}\right)^2 + \\ 16\delta^2\Gamma_{13}^2 Y + 4\delta^2\Omega_1^2\Omega_2^2[6\Gamma_{13}-(\Gamma_1+\Gamma_2)] + \\ \Omega_1^2\Omega_2^2[Y+(\Omega_1^2+\Omega_2^2)(\Gamma_1+\Gamma_2)], \end{aligned} \tag{14.14}$$

$$c_1 = 2(\Omega_1^2+\Omega_2^2)(4\Gamma_{13}Y+3\Omega_1^2\Omega_2^2)+4\frac{\Omega_1^2\Omega_2^2}{\Gamma_{13}}[\Gamma_1\Delta_1^2+\Gamma_2\Delta_2^2+(\Gamma_1+\Gamma_2)\Delta_1\Delta_2], \tag{14.15}$$

$$c_2 = 8[2\Gamma_{13}^2 Y + 3\Gamma_{13}\Omega_1^2\Omega_2^2 + 2(\Delta_2^2\Gamma_2\Omega_1^2+\Delta_1^2\Gamma_1\Omega_2^2)], \tag{14.16}$$

and where Y is given by

$$Y = \Gamma_2\Omega_1^2 + \Gamma_1\Omega_2^2. \tag{14.17}$$

Because Eq. 14.13 is too cumbersome to provide insights on the system, the limiting case of large detuning and weak probe will be further investigated. The second limiting case, resonant lasers, is discussed in Ref. [**379**].

14.1.2. Raman Transitions and Fano Profiles

The weak probe and large detuning limit with non-zero laser linewidth is discussed in Ref. [**379**]. Assuming zero laser linewidth ($\gamma = 0$ and $\alpha = 0$),

$$\Omega_1^2 \ll \frac{\Gamma_1}{\Gamma_2}\Omega_2^2, \tag{14.18}$$

$$\Delta_1^2 \gg \delta^2. \tag{14.19}$$

Then Eq. 14.13 can be simplified to [**380, 381, 382, 85, 379**]

$$\rho_{33} = \frac{\Omega_1^2\delta^2\frac{\Gamma}{\Gamma_1}}{4\Delta_1^2(\delta-\Delta')^2 + \delta^2\Gamma^2 + 4\delta^2\Delta_1^2\frac{\Omega_1^2}{\Omega_2^2}\frac{\Gamma_2}{\Gamma_1} + \frac{\Omega_1^2\Omega_2^2}{4}\left(\frac{\Gamma_2}{\Gamma_1}+2\right)} \tag{14.20}$$

where

$$\Delta' = \frac{\Omega_2^2}{4\Delta_1} \tag{14.21}$$

is the light shift of the two-photon bright resonance due to the strong Raman pump beam. From Eq. 14.20, there is a zero at the dark resonance ($\delta = 0$) and a maximum at the bright resonance ($\delta \simeq \Delta'$), where the precise peak location is discussed in Ref. [**381**].

Eq. 14.20 can further be approximated by replacing δ^2 with Δ'^2 while keeping the $(\delta - \Delta')$ term. Since we are only concerned about the behavior of ρ_{33} around the neighboring bright and dark resonances, this approximation is appropriate since it is accurate when $\delta = \Delta'$, and the first term in the denominator dominates under the large detuning limit. Eq. 14.20 can then be written as [**379**]

$$\rho_{33} \simeq \frac{\Omega_{\text{eff}}^2\left(\frac{\delta}{\Delta'}\right)^2\left(\frac{R}{4\Gamma_1}\right)}{(\delta-\Delta')^2 + \frac{R^2}{4} + \frac{\Omega_{\text{eff}}^2\Gamma}{2\Gamma_1}} \tag{14.22}$$

where

$$\Omega_{\text{eff}} = \frac{\Omega_1 \Omega_2}{2\Delta_1} \tag{14.23}$$

is the effective Raman Rabi frequency for oscillations on the two-photon Raman resonance and

$$R = \frac{\Omega_2^2}{4\Delta_1^2}\Gamma \tag{14.24}$$

is the off-resonant scattering rate on the strongly driven Raman pump transition. Fig. 14.2 plots Eq. 14.22 as a function of δ/Γ with $\Omega_1 = 0.05\Gamma$, $\Omega_2 = \Gamma$, $\Gamma = \Gamma_1 + \Gamma_2 = \frac{1}{3}\Gamma + \frac{2}{3}\Gamma$, and $\Delta_1 = -5\Gamma$. Note the dark resonance at $\delta/\Gamma = 0$ and the bright resonance at $\delta/\Gamma \simeq \Delta'$. Additionally, the FWHM of the peak f under the condition $f \ll \Delta'$ is given by

$$f = \left(R^2 + \Omega_{\text{eff}}^2 \frac{2\Gamma}{\Gamma_1} \right)^{1/2}. \tag{14.25}$$

Eq. 14.22 is a canonical "Fano" profile [**383**] and is plotted in Fig. 14.2. In the weak probe and large detuning limit, the excited state $|3\rangle$ appears as a continuum. The two interfering paths that lead to the Fano profile are (1) the single-photon direct absorption from $|2\rangle$ to $|3\rangle$ and (2) the two-photon Raman process from $|1\rangle$ to $|2\rangle$ followed by the one-photon absorption process from $|2\rangle$ to $|3\rangle$ [path (1)]. Path (2) passes through a narrow state via the Raman process and interferes with path (1) to the "continuum" and such interference leads to Fano profiles [**380**, **381**, **382**, **85**, **379**, **376**, **377**, **378**]. Note that if Δ_1 is sufficiently far-detuned, the 1-photon off-resonant scattering rate [path (1)] will be negligible and the Fano profile is no longer observed.

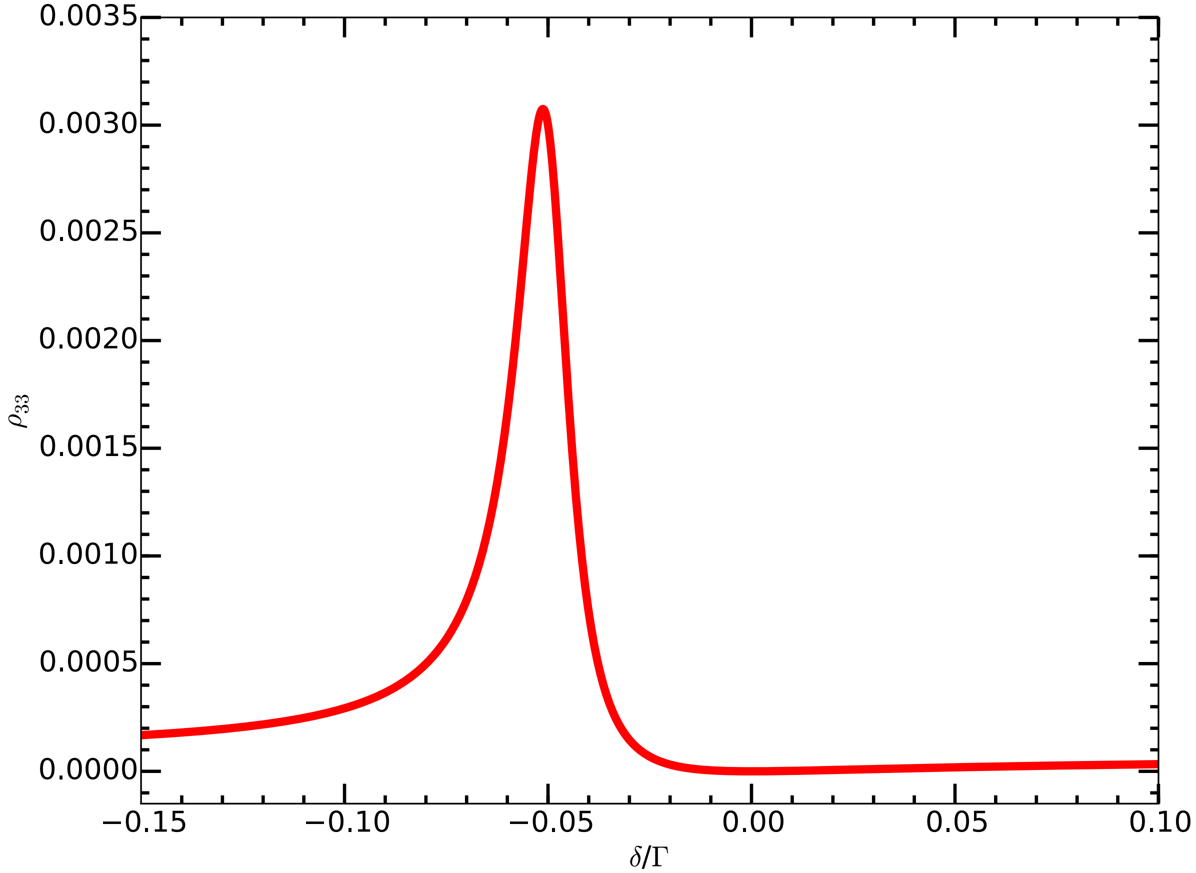


Figure 14.2. Fano profile lineshape given by Eq. 14.22 using $\Omega_1 = 0.05\Gamma$, $\Omega_2 = \Gamma$, $\Gamma = \Gamma_1 + \Gamma_2 = \frac{1}{3}\Gamma + \frac{2}{3}\Gamma$, and $\Delta_1 = -5\Gamma$. Note the dark resonance at $\delta/\Gamma = 0$ and the bright resonance at $\delta/\Gamma \simeq \Delta'$.

14.1.3. Raman Sideband Cooling

From Eq. 14.25, the two-photon Fano profile resonance can be tailored such that the transition FWHM is less than the secular frequency of an ion confined in an ion trap [**141**]. Therefore, with appropriate choices of detuning and Rabi frequencies, this resonance profile can be used for sideband cooling. Much of the discussion below uses the formalism used in Ref. [**85**].

Consider an ion trapped in a ion trap with a thermal distribution at least close to the Lamb-Dicke regime (obtainable by Doppler cooling) with $\eta^2\bar{n} \leq 1$, where $\bar{n}$ is the mean motional occupation number of the ion in the harmonic well with frequency ω and η is the Lamb-Dicke parameter, which is given by

$$\eta = \sqrt{\frac{\hbar|\Delta\vec{k}|^2}{2m\omega}}. \tag{14.26}$$

Exposing the ion to the Raman beams with wave vector difference $\Delta\vec{k}$ will induce photon scattering at a rate of $\rho_{33}(\delta)\Gamma$. In the Lamb-Dicke regime, the photon scattering rate will be dominated by the carrier followed by the first-order red and blue sidebands with transition probabilities being proportional to Ω^2, $\eta^2\Omega^2 n$, and $\eta^2\Omega^2(n+1)$, respectively. From these probabilities, the red and blue sideband transitions are suppressed by $\eta^2 n$ and $\eta^2(n+1)$, respectively. The scattering rates on the red (R^n_{n-1}) and blue (R^n_{n-1}) sidebands are

$$R^n_{n-1} = \rho_{33}(\delta)\Gamma\eta^2 n + \rho_{33}(\delta+\omega)\Gamma\eta^2 n, \tag{14.27}$$

$$R^n_{n+1} = \rho_{33}(\delta)\Gamma\eta^2(n+1) + \rho_{33}(\delta-\omega)\Gamma\eta^2(n+1). \tag{14.28}$$

The rate equations for the motional-level populations are then

$$\frac{\mathrm{d}}{\mathrm{d}t}P(n) = R^{n+1}_n P_{n+1} + R^{n-1}_n P_{n-1} - (R^n_{n-1} - R^n_{n+1})P_n, \tag{14.29}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}P(n) = A_-[P_{n+1}(n+1) - P_n n] + A_+[P_{n-1}n - P_n(n+1)] \tag{14.30}$$

where the $A_\pm$ coefficients are independent of n and are given by

$$A_\pm = \eta^2[\rho_{33}(\delta)\Gamma + \rho_{33}(\delta \mp \omega)\Gamma]. \tag{14.31}$$

In terms of $\bar{n}$,

$$\frac{\mathrm{d}}{\mathrm{d}t}\bar{n} = \frac{\mathrm{d}}{\mathrm{d}t}\sum_{n=1}^{\infty} nP(n) = \sum_{n=1}^{\infty} n\frac{\mathrm{d}}{\mathrm{d}t}P_n = -(A_- - A_+)\bar{n} + A_+. \tag{14.32}$$

Therefore, the value of $\bar{n}$ exponentially decays at a rate of

$$\frac{\mathrm{d}}{\mathrm{d}t}\bar{n} \simeq -(A_- - A_+) = \eta^2[\rho_{33}(\delta+\omega)\Gamma - \rho_{33}(\delta-\omega)\Gamma] \tag{14.33}$$

Under the condition of cooling, $-(A_- - A_+) < 0$, $\bar{n}_f$, the equilibrium value of $\bar{n}$, is given by

$$\bar{n}_f = \frac{A_+}{A_- + A_+} = \frac{\rho_{33}(\delta)\Gamma + \rho_{33}(\delta-\omega)\Gamma}{\rho_{33}(\delta+\omega)\Gamma - \rho_{33}(\delta-\omega)\Gamma}. \tag{14.34}$$

14.1.3.1. EIT cooling. In the equilibrium value of $\bar{n}$ above (Eq. 14.34), off-resonant excitation of the carrier occurs while driving the red sideband. Resulting carrier excitation and following spontaneous emission on the blue sideband creates a heating mechanism that balances with the cooling provided by driving the red sideband. If instead both the Raman pump and probe beams are *blue-detuned* ($\Delta_1, \Delta_2 > 0$) and with appropriate choice of Ω_2 such that $\Delta' \simeq \omega$, the red sideband is placed at the dark resonance of the carrier ($\delta = 0$), and off-resonant excitation of the carrier is completely suppressed, eliminating the heating mechanism. Raman sideband cooling under this condition is known as electromagnetically induced transparency (EIT) cooling [**382**, **147**, **85**, **31**]. In this condition, $\delta = 0$ and the first term in the numerator of Eq. 14.34 vanishes. Then

$\bar{n}_f^{\mathrm{EIT}}$, the equilibrium value of $\bar{n}$ under this EIT condition, is then given by [85]

$$\bar{n}_f^{\mathrm{EIT}} = \frac{\rho_{33}(-\omega)\Gamma}{\rho_{33}(+\omega)\Gamma - \rho_{33}(-\omega)\Gamma} = \left(\frac{\Gamma}{4\Delta_1}\right)^2 . \tag{14.35}$$

Additionally, EIT cooling permits cooling of multiple motional modes in a two-ion crystal simultaneously as the carrier excitation rate is low for a large range of δ around the dark resonance [31, 384].

14.1.4. Raman Sideband Cooling Expectations

Following the discussion of Sec. 14.1, the near-resonant Raman sideband cooling is modeled for a single $^{138}\mathrm{Ba}^+$ ion in our trap. Here, $\Gamma = 2\pi \times 15.2$ MHz, $\Gamma_1 = \frac{2\pi}{3} \times 15.2$ MHz, $\Gamma_2 = \frac{4\pi}{3} \times 15.2$ MHz, $k = \frac{2\pi}{493}$ nm^{-1}, $\Delta_1 = -2\pi \times 80$ MHz, and $\omega_z = 2\pi \times 1.15$ MHz. Both $\bar{n}_f$ (Eq. 14.34) and $\frac{\mathrm{d}}{\mathrm{d}t}\bar{n}$ (Eq. 14.33) are calculated as a function of two-photon detuning δ and shown in Fig. 14.3. Note that the Zeeman spliting is not included. Modeling code is located in App. E.1.9.

From the results, the minimum of $\bar{n}_f$ is $\bar{n}_{f,\,\mathrm{min}} = 0.11$, and the cooling rate is $\frac{\mathrm{d}}{\mathrm{d}t}\bar{n} = 890$ s^{-1} with a two-photon detuning of $\delta/(2\pi) = -1.88$ MHz. Therefore, we expect to reach the motional ground state with high probability as long as our operating parameters are comparable with the input parameters above.

The sensitivity of the model results to various experimentally-expected parameter drifts is also explored. Ω_1 can vary significantly (up to ∼50%), while Ω_2 must be kept within ∼5%, or the set two-photon detuning will result in slow cooling and a much-higher $\bar{n}_f$. Since Ω scales as $\sqrt{I}$ where I is the laser intensity, the laser power will need to be stabilized to within a few percent, but this requirement is not unrealistic. Additionally,

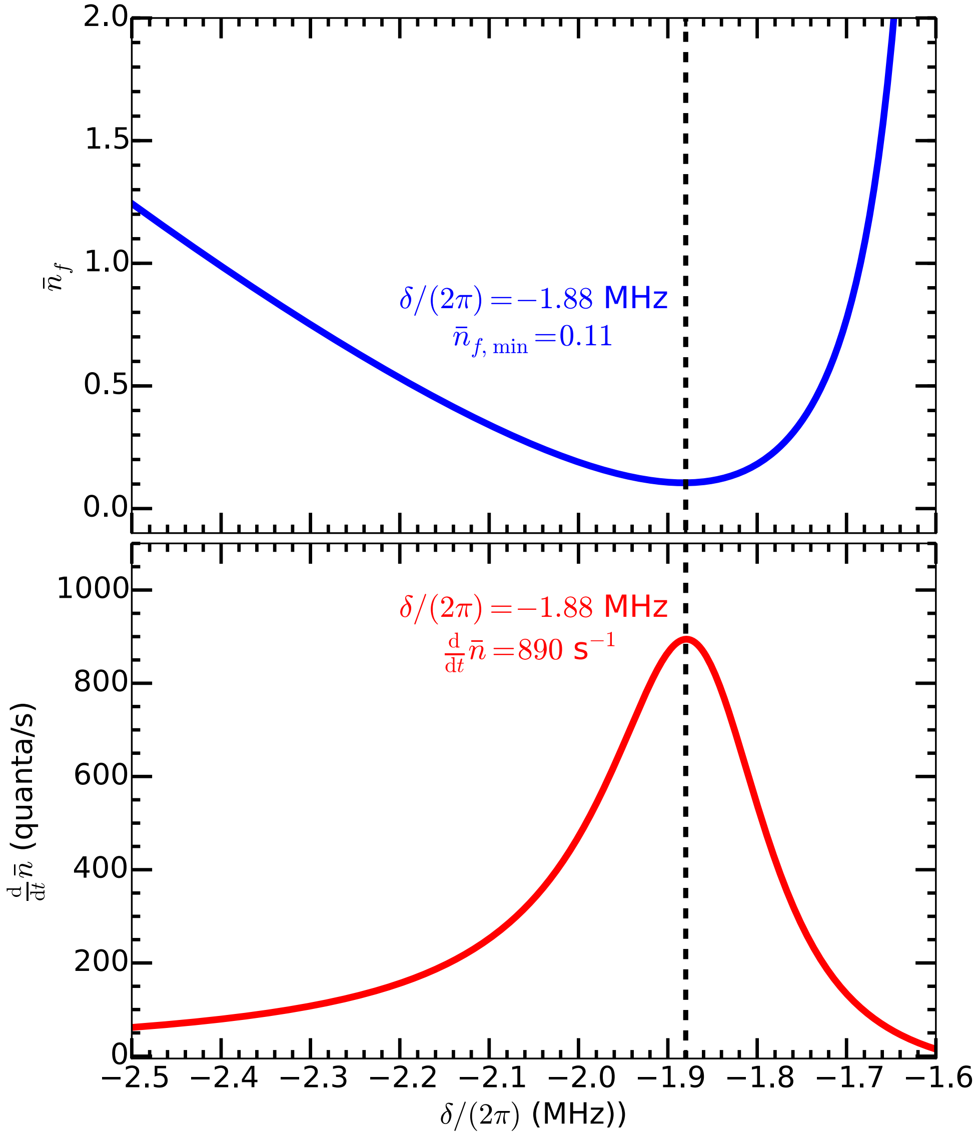


Figure 14.3. Calculations of $\bar{n}_f$ (top) and $\frac{\mathrm{d}}{\mathrm{d}t}\bar{n}$ (bottom) as a function of two-photon detuning δ. Model input values are given in the text. From the results, we expect the minimum $\bar{n}_f$ value to be $\bar{n}_{f,\,\mathrm{min}} = 0.11$ and the cooling rate to be $\frac{\mathrm{d}}{\mathrm{d}t}\bar{n} = 890\ \mathrm{s}^{-1}$ with a two-photon detuning of $\delta/(2\pi) = -1.88$ MHz.

as Δ_1 goes further red, $\bar{n}_{f,\,\mathrm{min}}$ decreases and the cooling rate slows significantly; this parameter should be kept within ∼10%, which is the maximum expected wavelength lock drift using the wavelength meter. Therefore, the proposed near-resonant Raman sideband cooling should work with our anticipated experimental drifts.

After the experimental results are quantified using the Fano-like fit function, the model is updated with the experimental parameters in Table 14.1. The updated results (Fig. 14.4) indicate a slightly higher value for $\bar{n}_{f,\,\mathrm{min}} = 0.17$, but a faster cooling rate with $\frac{\mathrm{d}}{\mathrm{d}t}\bar{n} = 1070\ \mathrm{s}^{-1}$, both with a two-photon detuning of $\delta/(2\pi) = -1.87$ MHz, which is similar to the previous result.

14.2. Near-resonant Raman Sideband Cooling Experiment

The near-resonant Raman sideband cooling laboratory setup, AOM setup, and process are shown in Figs. 13.8, 13.9, and 14.5, respectively. The near-resonant Raman pump (probe) beams enter the trap parallel (perpendicular) to an applied magnetic field of 3.919(7) Gauss, which splits the $^2\mathrm{S}_{1/2}$ ground state Zeeman sublevels by 10.97(2) MHz. The magnetic field, oriented 45° from the z-axis of the trap, is aligned to the σ^+-polarized near-resonant Raman pump beam using three additional coils, minimizing the spontaneous emission rate from this beam (Sec. 13.8). Entering the trap 90° from the near-resonant pump beam, the π-polarized Raman probe beam provides a maximum value of $\Delta\vec{k}$ along the trap z-axis with our polarizations and geometry.

The two Zeeman $^2\mathrm{S}_{1/2}$ sublevels serve as our quantum states for sideband cooling and motional state detection. For Raman sideband cooling (Fig. 14.5), the σ^+-polarized Raman pump beam with Rabi frequency $\Omega_{\mathrm{pump}} = 2\pi \times 14.9(3)$ MHz optically pumps the ion to the $m_j = +\frac{1}{2}$ state via off-resonant spontaneous scattering and imparts a light shift of 620(20) kHz. With the near-resonant Raman probe beam sufficiently weak, the

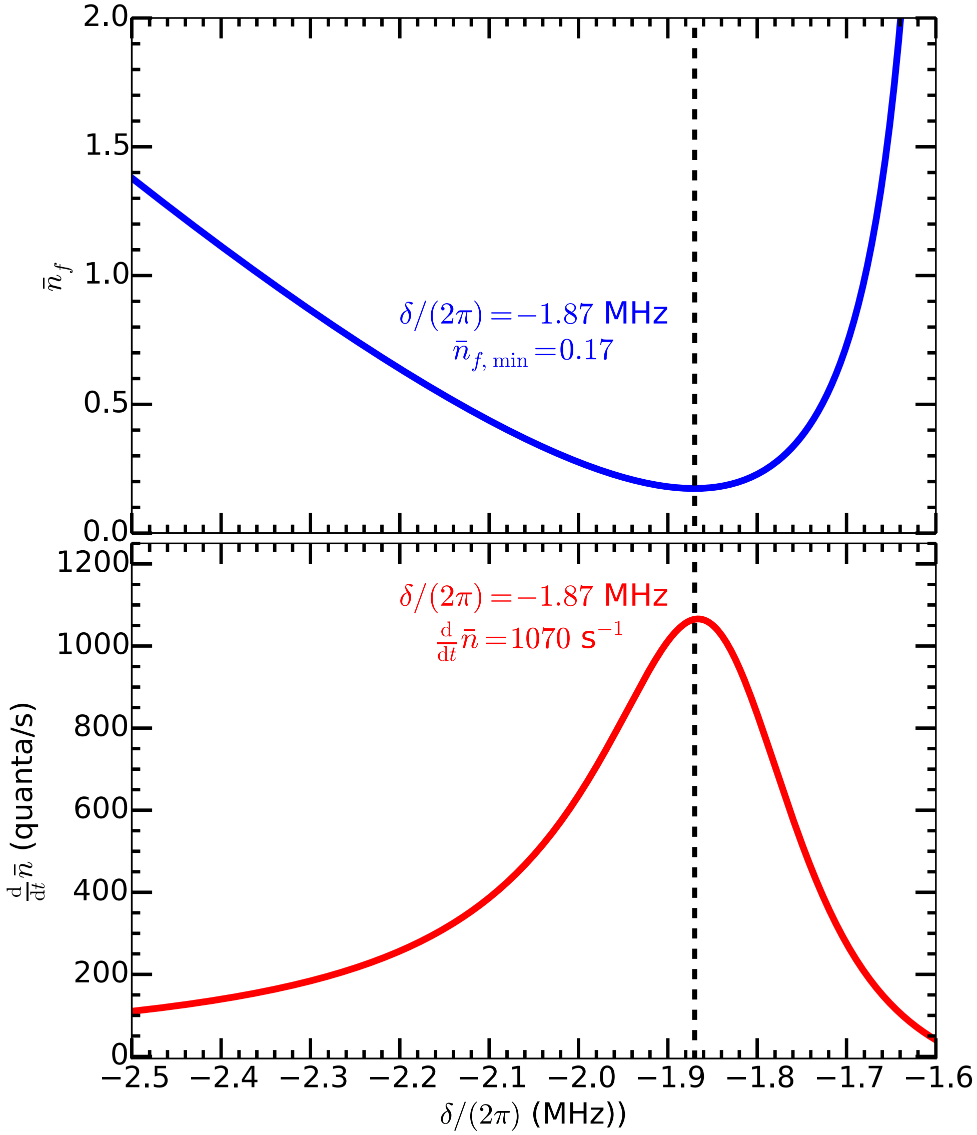


Figure 14.4. Calculations of $\bar{n}_f$ (top) and $\frac{\mathrm{d}}{\mathrm{d}t}\bar{n}$ (bottom) as a function of two-photon detuning δ. Model input values are updated with those in Table 14.1. From the results, we expect the minimum $\bar{n}_f$ value to be $\bar{n}_{f,\,\mathrm{min}} = 0.17$ and the cooling rate to be $\frac{\mathrm{d}}{\mathrm{d}t}\bar{n} = 1070\ \mathrm{s}^{-1}$ with a two-photon detuning of $\delta/(2\pi) = -1.87$ MHz.

width of the carrier transition decreases below the axial secular frequency, and the axial secular sidebands are resolved. In this regime, the near-resonant Raman probe beam with

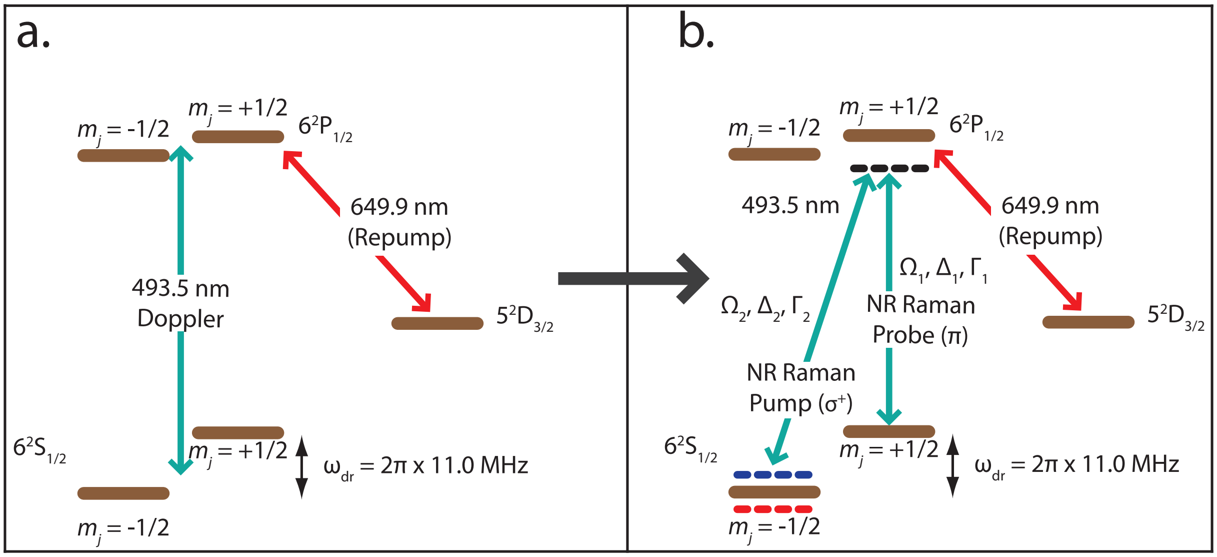


Figure 14.5. Stages of ^{138}Ba$^+$ cooling. Solid lines show applied laser drives. Horizontal dashed lines show the red and blue motional sidebands of the driven Raman transition when relevant and the virtual state of the Raman transition (black), detuned by ≈ 80 MHz. (a) Doppler cooling lasts 20 ms. (b) near-resonant Raman sideband cooling lasts either 1, 10, or 100 ms. The PMT is gated with the near-resonant Raman beams.

Rabi frequency $\Omega_{\text{probe}} = 2\pi \times 1.07(2)$ MHz provides an effective Raman Rabi frequency of $\Omega_{\text{eff}} = 2\pi \times 89(2)$ kHz. When the frequency difference between the two Raman beams is near the light-shifted two-photon Raman bright resonance or on either the red sideband or blue sideband, Raman transitions occur from the $m_j = +\frac{1}{2}$ state to the $m_j = -\frac{1}{2}$ state. The ion is then quickly optically pumped to the $m_j = +\frac{1}{2}$ state by the σ^+-polarized near-resonant Raman pump beam. With the 650 nm $\mathrm{D}_{3/2} \rightarrow \mathrm{P}_{1/2}$ repump laser on, this process results in the emission of 493 nm photons at a low rate similar to that of the optical pumping, which are collected on a PMT (Figs. 14.7, 14.6).

The photon collection rate as a function of two-photon detuning is shown in Figs. 14.6 and 14.7 for three PMT exposure times with the near-resonant Raman pump frequency

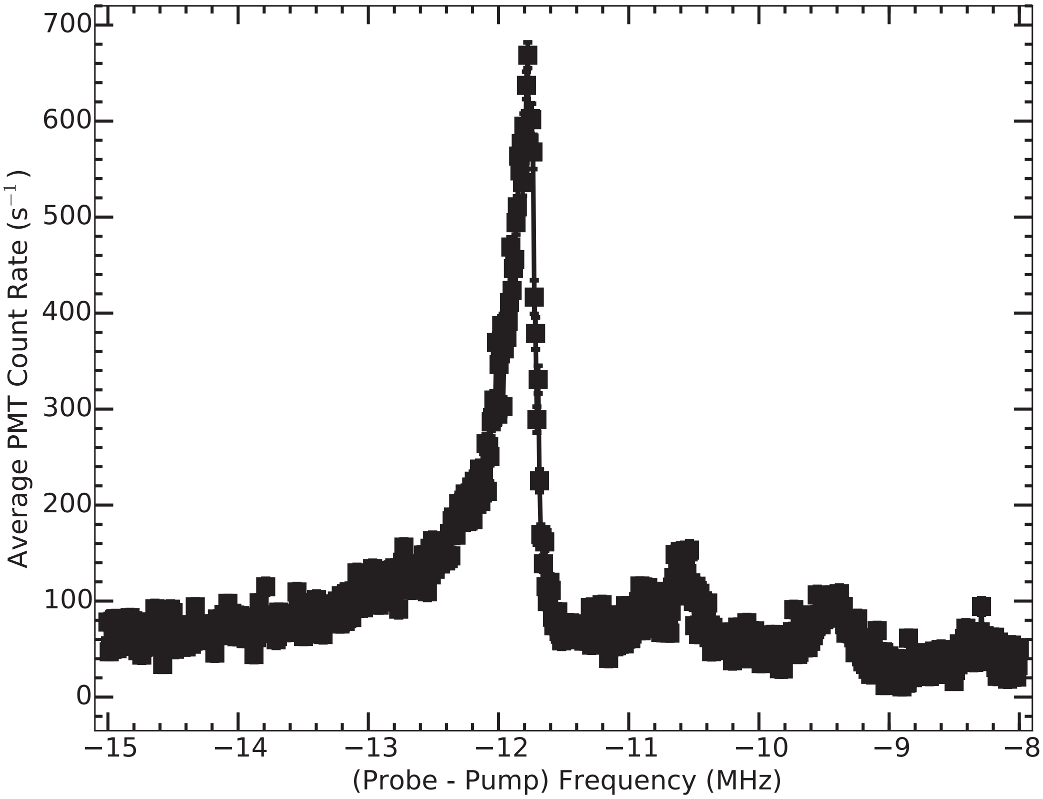


Figure 14.6. Near-resonant Raman 493 nm fluorescence spectrum with the PMT exposure over the first 100 ms of near-resonant Raman cooling. Error bars are given by the standard deviation of the mean of each data point, and are comparable to the marker size. The solid line connects data points and is meant to guide the eye.

fixed while the near-resonant Raman probe frequency is scanned over the Raman resonance. All data collection is performed using the code in App. E.3.1 with frequencies randomized and interleaved with a total exposure time of 20 s per data point with 20 ms of Doppler cooling between experimental repetitions. Data analysis is performed using the code in App. E.1.10. In 1 ms of exposure, the red sideband is well resolved. In 10

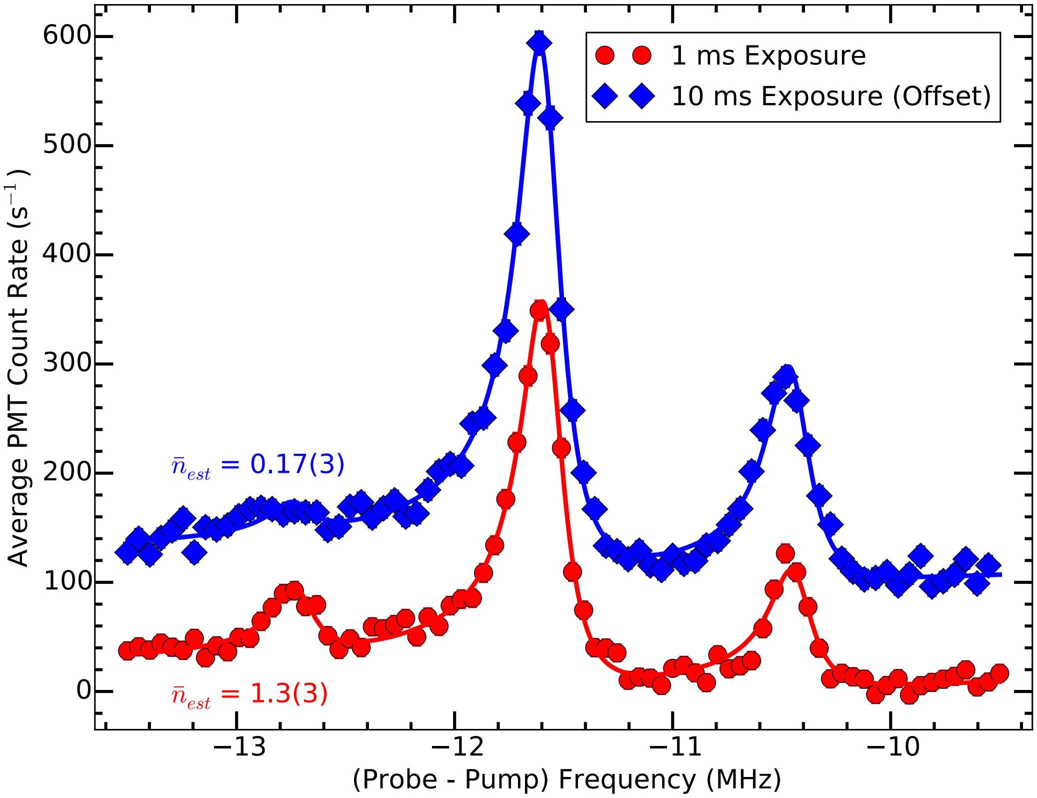


Figure 14.7. Near-resonant Raman 493 nm fluorescence spectrum with red and blue motional sidebands resolved. The red circles (blue diamonds) are data corresponding to PMT exposure for the first 1 (10) ms of near-resonant Raman exposure. Solid lines are fits that can be used to estimate equilibrium-equivalent occupation numbers, $\bar{n}_{est}$. Error bars are given by the standard deviation of the mean of each data point, and are comparable to the marker size.

ms of exposure, the red sideband amplitude averages down to a low value since the ion cools to the ground state. At this detuning and elsewhere red of the carrier, the carrier transition is still excited at a low rate, contributing a constant photon background. With

100 ms of near-resonant Raman exposure blue of the carrier, the ion heats sufficiently such that multiple blue sidebands are observed.

A Fano-like fit function discussed in Sec. 14.1.2 is used to extract relevant parameters from the near-resonant Raman spectra with 1 and 10 ms of exposure. For our experiment, $\Gamma = \Gamma_1 + \Gamma_2 = \frac{1}{3}\Gamma + \frac{2}{3}\Gamma$ is the $\mathrm{S}_{1/2} \to \mathrm{P}_{1/2}$ natural linewidth and branching ratios for the Raman probe and pump transitions, $\Delta_1 = \Delta_2 + \delta$, and $\Delta_2 = 2\pi \times -79$ MHz so the 80-MHz-center-frequency near-resonant Raman probe AOM operates at its peak efficiency near the carrier bright resonance. The full fit function for the near-resonant Raman spectrum is then the summation of ρ_{33} for the carrier and each motional sideband with independent amplitudes:

$$A_{\mathrm{C}}\rho_{33}\left(\delta - \omega_{\mathrm{dr}}\right) + A_{\mathrm{RSB}}\rho_{33}\left(\delta - \omega_{\mathrm{dr}} + \omega_z\right) + A_{\mathrm{BSB}}\rho_{33}\left(\delta - \omega_{\mathrm{dr}} - \omega_z\right) + D \tag{14.36}$$

where A_{C} is the carrier amplitude, A_{RSB} (A_{BSB}) is the red (blue) sideband amplitude, ω_{dr} is the location of the dark resonance (equal to the Zeeman splitting of the ${}^2\mathrm{S}_{1/2}$ state), and ω_z is the axial secular frequency (both fit parameters). Although all backgrounds are subtracted from the data, a constant background term (D) is also included to account for any small background due to changes in the 493 nm light scatter over the data collection time. The additional fit parameters are Ω_1, Ω_2, and ω_{dr}.

Although the data presented in Fig. 14.7 represent time averages over timescales where cooling (heating) is occurring while driving the red sideband (blue sideband), it is very useful to extract from the fit the near-resonant Raman pump and probe Rabi frequencies, Zeeman splitting, and axial secular frequency. Fit values from each data set are summarized in Table 14.1 and averaged fit values extracted from the two data sets are given

Table 14.1. Fano profile fit results to Eqn. 14.36 for the data presented in Fig. 14.7. All frequencies have a factor of 2π removed and are in MHz. Amplitudes are in counts/s.

	1 ms RSC	10 ms RSC	Average
Ω_1	1.11(4)	1.05(3)	1.07(2)
Ω_2	14.4(4)	15.3(4)	14.9(3)
ω_{dr}	-11.00(3)	-10.95(3)	-10.97(2)
ω_z	1.148(5)	1.150(4)	1.149(3)
A_{C}	211(9)	272(10)	
A_{BSB}	64(4)	109(5)	
A_{RSB}	36(3)	16(3)	
D	-0.2(3)	-0.4(3)	
$\bar{n}_{est}$	1.3(3)	0.17(3)	

in the text above. The fit also provides the location of the red sideband, which should correspond to the optimal Raman sideband cooling frequency for our system, which is verified in Sec. 15.7.

Despite the spectra representing an integration over a nonequilibrium history, we conclude that after 10 ms of cooling $\bar{n} \ll 1$ by the absence of a significant red sideband amplitude compared with that of the blue sideband [85]. This suggests that the system reaches equilibrium in $\lesssim$10 ms. If we assume that the data presented in Fig. 14.7 are equilibrium spectra and provide an estimate of $\bar{n}$ ($\bar{n}_{est}$), $\bar{n}_{est} = 1.3(3)$ after 1 ms of Raman sideband cooling and $\bar{n}_{est} = 0.17(3)$ after 10 ms of red sideband, which is in agreement with our final $\bar{n}$ result from near-resonant Raman sideband cooling modeling (Sec. 14.1.4) and our measurement described in Sec. 15.6.

In principle, this detection method with sufficiently short exposure times–when on average only a single Raman transition and subsequent optical pumping occurs–could also provide an accurate measure of the final $\bar{n}$. An advantage of that approach would be to allow a final $\bar{n}$ measurement with no further lasers required. However, since we are ultimately interested in quantum logic operations which do require additional lasers, we use a far-off-resonant laser system to conduct temperature diagnostics to confirm our final $\bar{n}$ value (Sec. 15.6) and verify that the fitted red sideband location provides the best cooling efficiency (Sec. 15.7).

CHAPTER 15

State Readout and Motional State Detection

The near-resonant Raman sideband cooling technique discussed in the previous chapter (Ch. 14) allows for an estimation of $\bar{n}$. However, since the spontaneous emission rate is too high, it does not provide a mechanism to drive the π pulses needed for single-shot measurement of n, as would be required for QLS. Far-off-resonant Raman excitation, achieved in this experiment using a single dedicated ECDL, allows the coherent Rabi flopping required for measurements of $\bar{n}$. This chapter discusses the far-off-resonant motional state detection using a spin-dependent shelving technique.

15.1. Shelving and Deshelving

Electron shelving detection is a common state-readout technique for Group II ions [**136**, **385**, **386**, **387**, **149**, **388**, **79**]. The technique presented here uses dipole-allowed transitions, previously demonstrated in Ba$^+$ [**386**, **387**], to avoid the 1.76 μm quadrupole transition. The relevant levels and transitions in ^{138}Ba$^+$ are shown in Fig. 15.1.

The general idea of the detection method is to place the ion into the ^{2}D$_{3/2}$ state conditional of the property being measured with the ion remaining in the ^{2}S$_{1/2}$ state if the condition is not met. The ion is then exposed to the 455 nm laser and cycles on the ^{2}S$_{1/2}$ → ^{2}P$_{3/2}$ transition until the ion is optically pumped to the desired ^{2}D$_{5/2}$ state. The Doppler cooling lasers are then turned on, and the ion will (will not) fluoresce if it is in the ^{2}D$_{3/2}$ (^{2}D$_{5/2}$) state, resulting in a bright (dark) ion. This process however is not

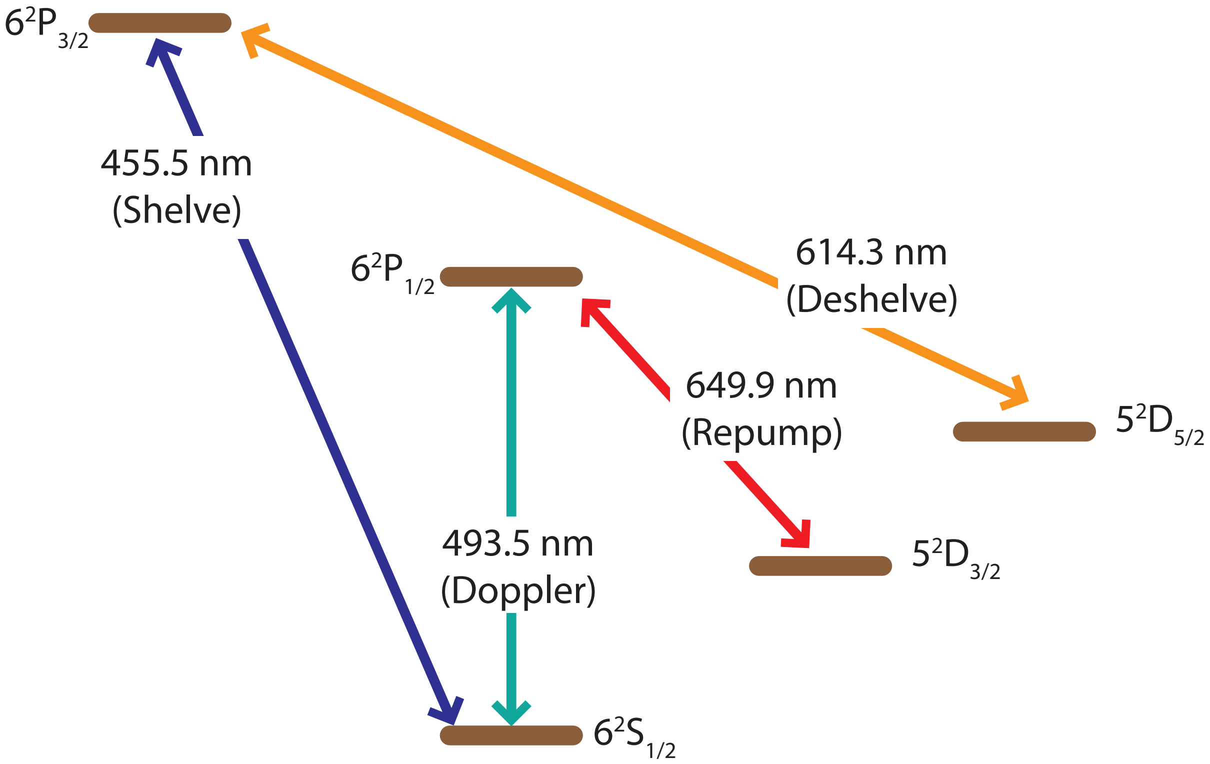


Figure 15.1. Relevant transitions in ^{138}Ba^{+} for shelving and resulting detection discussion. The ion will (will not) fluoresce if it is in the ^{2}D$_{3/2}$ (^{2}D$_{5/2}$) state, resulting in a bright (dark) ion when exposed to Doppler cooling light.

perfect as the branching ratio from the ^{2}P$_{3/2}$ to the ^{2}D$_{5/2}$ state is 88% with the remaining 12% decaying to the ^{2}D$_{3/2}$ state. After detection, the ion is then exposed to the 614 nm deshelving laser to optically pump the ion to either the ^{2}S$_{1/2}$ or ^{2}D$_{3/2}$ states in which Doppler cooling will again occur.

Experimentally, fluorescence detection is performed by increasing the 493 nm and 650 nm Doppler laser powers and tuning the 493 nm Doppler laser to resonance. This ensures maximum ion fluorescence during the 5 ms of detection exposure providing a maximum separation in fluorescence rate when the ion is bright or dark. Analyzing the data, a discriminator level is set to distinguish a bright or dark ion. A sample histogram of 83000

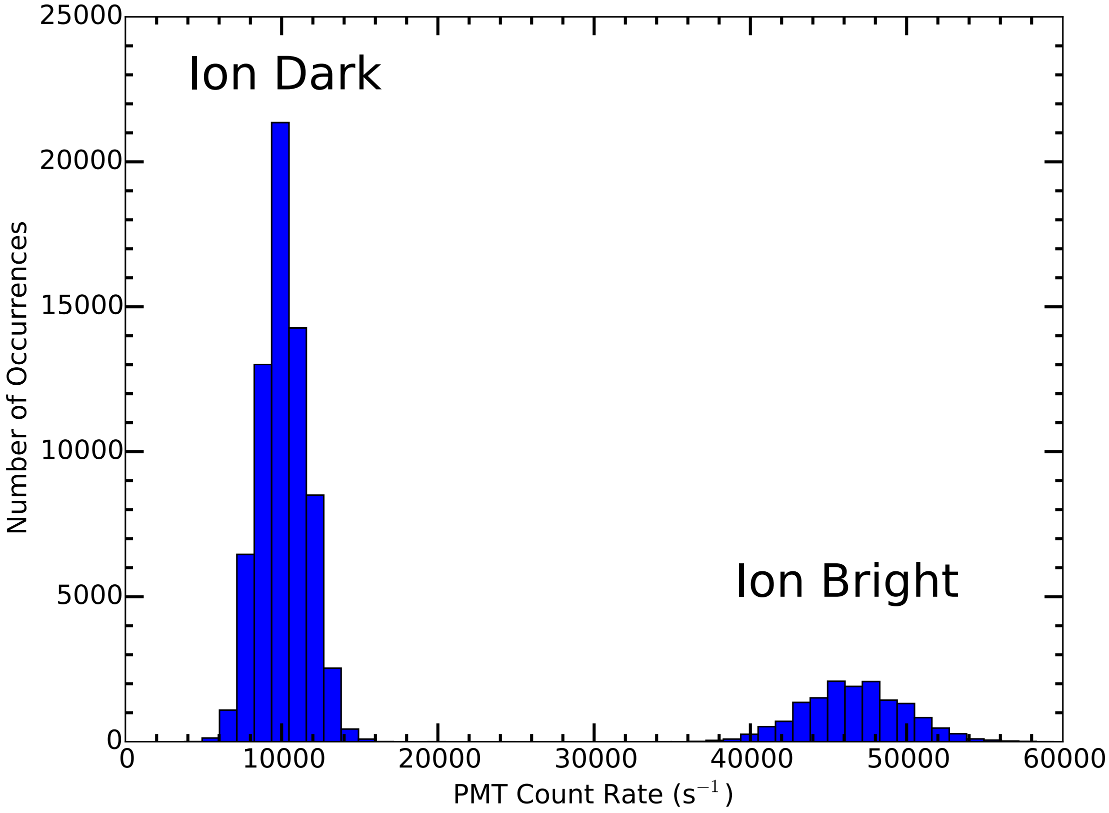


Figure 15.2. Sample shelving histogram for 5 ms of fluorescence collection on the PMT for the data shown in Fig. 15.10 with no Raman sideband cooling. The two populations (ion bright and ion dark) are clearly well separated. The lower discriminator level is set at 20000 counts/s and the upper discriminator level is set to 60000 counts/s.

experimental cycles is shown in Fig. 15.2, which uses the no Raman sideband cooling data from Fig. 15.10. Since the ion is either bright or dark for each repetition, binomial statistics are used for statistical analysis.

To verify efficient shelving, a test program (App. E.3.8) initializes the ion in the $^2S_{1/2}$ state, turns on the 455 nm light, performs fluorescence detection, deshelves the ion with a long 614 nm exposure time, and repeats. The resulting shelving probability, calculated with the code in App. E.1.5, should be 88%. Similarly to verify efficient deshelving, a test program (App. E.3.3) runs the same sequence only with a long shelving time and a short deshelving time. The resulting deshelving probability, calculated with the same analysis code, should be 100%. If either shelving or deshelving probability deviates significantly from the expected, the problem is investigated.

15.2. Spin-state Measurement Using Shelving

The 493 nm far-off-resonant Raman laser source, detuned -59 GHz from the $^2\mathrm{S}_{1/2} \rightarrow {}^2\mathrm{P}_{1/2}$ transition, is used to coherently transfer population between the $m_j = +\frac{1}{2}$ and $m_j = -\frac{1}{2}$ Zeeman sublevels. After the far-off-resonant Raman pulse, the population in each m_j state must be measured. This is performed using the shelving technique described above in a spin-dependent method previously demonstrated in Ba$^+$ [**386, 387**].

The experimental procedure, preceded by near-resonant Raman sideband cooling steps (a) and (b) in Fig. 14.5, is outlined in Fig. 15.3 and proceeds as follows: (c) the ion is optically pumped to the $m_j = +\frac{1}{2}$ state; (d) the far-off-resonant Raman pulse is applied to transfer population, conditional on the motional state; (e) the near-resonant Raman pump "protects" population driven to the $m_j = -\frac{1}{2}$ state by transferring it to the $^2\mathrm{D}_{3/2}$ manifold; (f) all remaining population in $^2\mathrm{S}_{1/2}$ is shelved to the long-lived $^2\mathrm{D}_{5/2}$ manifold, (g) the Doppler lasers detect whether the ion is in the $^2\mathrm{D}_{3/2}$ protected state (ion fluoresces) or the $^2\mathrm{D}_{5/2}$ shelved state (ion dark); and (h) any shelved ions are deshelved and the ion is Doppler cooled.

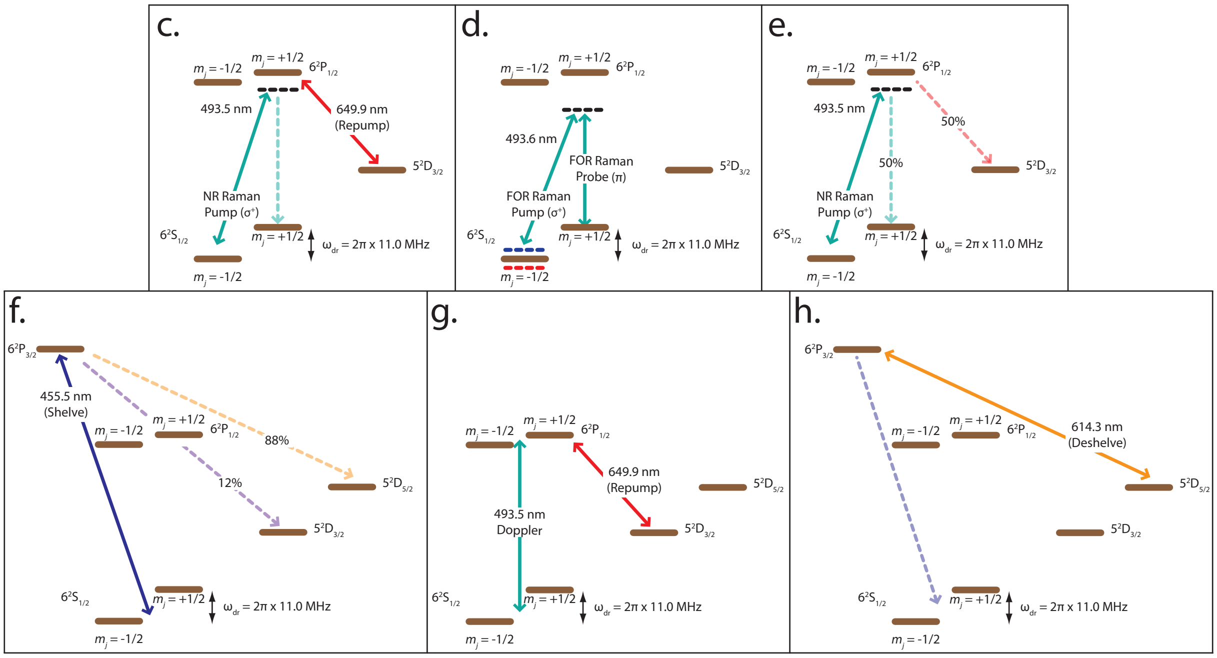


Figure 15.3. Stages of ^{138}Ba^{+} far-off-resonant state detection that occur after near-resonant Raman cooling steps (a) and (b) in Fig. 14.5. Solid arrows show applied laser drives, and dashed arrows indicate dominant spontaneous emission channels with corresponding branching fractions. Horizontal dashed lines show the red and blue motional sidebands of the driven Raman transition when relevant and the virtual state of the Raman transition (black), detuned by $\approx$ 80 MHz ($\approx$ 59 GHz) for the near-resonant (far-off-resonant) Raman beams. (c) Optically pump to the $m_j = +\frac{1}{2}$ state. (d) far-off-resonant Raman π pulse, conditional on motional state. (e) Protect population in $m_j = -\frac{1}{2}$ by pumping to ^{2}D$_{3/2}$. (f) Shelve any population remaining in ^{2}S$_{1/2}$ to ^{2}D$_{5/2}$. (g) Determination of ^{2}D$_{3/2}$ versus ^{2}D$_{5/2}$ ion state by observing fluorescence with the PMT. (h) Deshelve any population in ^{2}D$_{5/2}$ and additional Doppler cooling (not shown).

The net result of the above procedure is that a successful (unsuccessful) far-off-resonant Raman transition yields a bright (dark) ion when exposed to Doppler cooling light. There are two critical times involved in the procedure above, the optical pumping time and the protect time, which are experimentally determined in Sec. 15.3 and Sec. 15.4, respectively. Three contrast-reduction mechanisms occur in this spin-dependent shelving detection scheme: (1) an imperfect π pulse on the motional sidebands (discussed in Sec. 15.5), (2) an imperfect protect stage (discussed in Sec. 15.4), and (3) an imperfect shelving stage. For the latter, the branching ratio from ${}^2\mathrm{P}_{3/2} \rightarrow {}^2\mathrm{D}_{5/2}$ is only 88% with the remaining 12% branching to the protected ${}^2\mathrm{D}_{3/2}$ state.

15.3. Determining the Shortest Optical Pumping Time

After near-resonant Raman sideband cooling, the population is optically pumped to the $m_j = +\frac{1}{2}$ state using the near-resonant Raman pump beam with the 650 nm repump beam on. The repump beam ensures that population in the target $m_j = +\frac{1}{2}$ state is not pumped to the ${}^2\mathrm{D}_{3/2}$ manifold by polarization impurities and magnetic field imperfections, allowing the optical pumping time to be arbitrarily long. However, to minimize the effects of ion heating after the Raman sideband cooling on the final $\bar{n}$ measurement, this time should only be long enough to allow for maximum detection contrast.

To determine the optimum optical pumping time, the procedure described in Fig. 15.3 is performed using the programs in App. E.3.6 and App. E.3.6.6 with data in each set taken randomly and interleaved with 1000 trials per point. Using the optimal protect time discussed in Sec. 15.4, the optical pumping is varied. Both experiments allow measurement of the optical pumping time, but they also allow us to understand different state preparation fidelities in the experiment. To prevent complications from the near-resonant Raman sideband cooling, this is left off for these data taken below. Data analysis is performed using the code in App. E.1.11.

Without the far-off-resonant π pulse, we want to find the shortest optical pumping time that minimizes the bright fraction. At long optical pumping times, we expect that the population will always be in $m_j = +\frac{1}{2}$ with a resulting 12% bright fraction due to the branching ratio from ${}^2\mathrm{P}_{3/2}$ to ${}^2\mathrm{D}_{3/2}$. At short times, the population should have equal probability of being in either Zeeman sublevel. The protect stage then places the population starting in $m_j = -\frac{1}{2}$ to the protected ${}^2\mathrm{D}_{3/2}$ state with 50% efficiency ($\Gamma_{650}/\Gamma_{493} \approx \frac{1}{3}$); after the protect stage, 75% of the population is in the ${}^2\mathrm{S}_{1/2}$, $m_j = +\frac{1}{2}$ state with 25% in the protected ${}^2\mathrm{D}_{3/2}$ state. After shelving and accounting for the 12% branching ratio from ${}^2\mathrm{P}_{3/2} \to {}^2\mathrm{D}_{3/2}$, the ion should be bright with probability $(25\% + 12\% \times 75\%) = 34\%$ [386, 387]. Experimentally at long optical pumping time, we observe the population saturate at the expected bright fraction of 12% with 10 μs of optical pumping. At very short optical pumping times, we observe a higher-than-expected initial bright fraction of ≈42% (red diamonds in Fig. 15.4).

With the far-off-resonant π pulse, we want to verify the optical pumping time determined above by finding the shortest optical pumping time that maximizes the bright fraction. At long optical pumping times, the ion is optically pumped to $m_j = +\frac{1}{2}$ before the π pulse, and is maximally bright. For this experiment, the maximum bright fraction of (44%+12%) = 56% is not reached due to fast decoherence of the far-off-resonant Raman carrier without Raman sideband cooling. However, the maximum bright fraction occurs after 10 μs of optical pumping. At very short optical pumping times, we expect that the population would be independent of optical pumping time, and the population should be 34% bright. We note a decrease from this expected value by ≈5% (blue circles

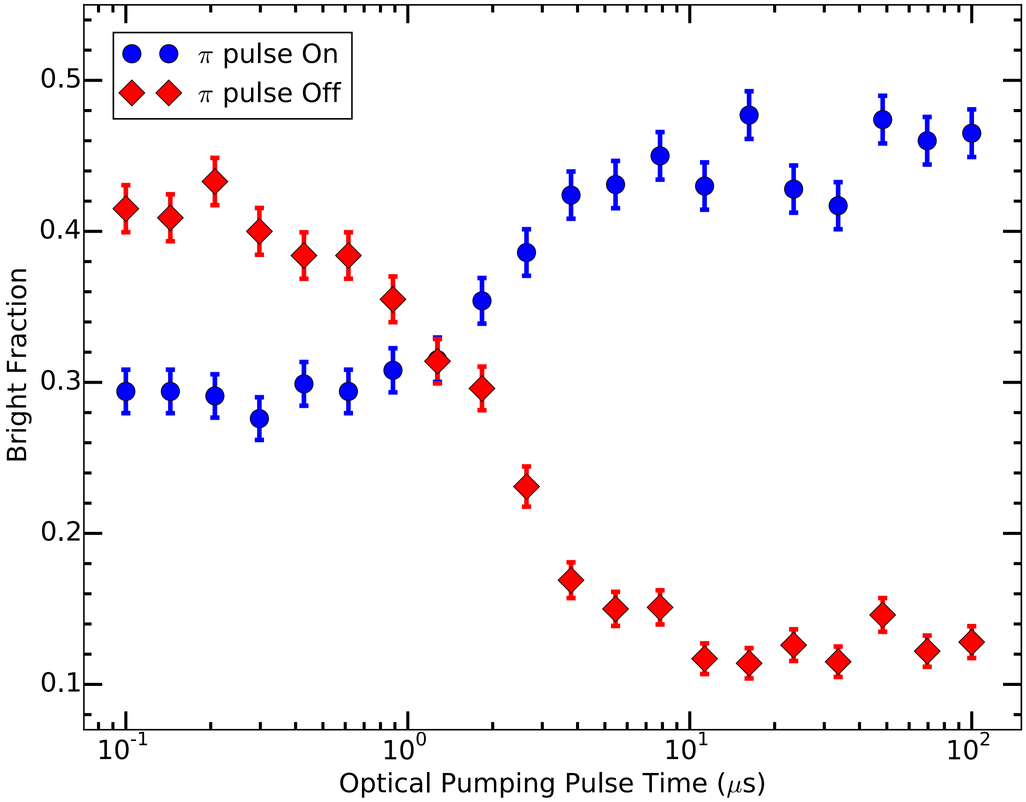


Figure 15.4. Determination of the shortest optical pumping time for the step shown in Fig. 15.3c with far-off-resonant π pulse off (red diamonds) and far-off-resonant π pulse on (blue circles). We see that the minimum optical pumping time for minimum (maximum) bright fraction with the far-off-resonant π pulse off (on) is ≈10 μs. Data points in each data set are taken randomly and interleaved. Error bars are given by binomial statistics.

in Fig. 15.4), which correlates to the bright fraction excess without the far-off-resonant π pulse above.

With our Zeeman ${}^2\mathrm{S}_{1/2}$ splitting of $\omega_{\mathrm{dr}} \sim \frac{1}{2}\Gamma$ and the smaller Zeeman splitting of the ${}^2\mathrm{P}_{1/2}$ state, we expect some degree of optical pumping by Doppler cooling. When the π-polarized Doppler cooling frequency is fixed at $\approx\Gamma/2$ based on ion fluorescence, the $m_j =$

$+\frac{1}{2}$ transition is closer to resonance than the $m_j = -\frac{1}{2}$ transition. In a steady-state regime, this results in a population excess in $m_j = -\frac{1}{2}$, which we observe in the procedure without the far-off-resonant π pulse. With the far-off-resonant Raman π pulse, the populations in each $^2\mathrm{S}_{1/2}$ Zeeman sublevel exchange, resulting in the observed population deficiency using this procedure.

From both data set results, the shortest optical pumping time that allows for the maximum detection contrast is 10 μs.

15.4. Determining the Optimal Protect Time

After the far-off-resonant pulse, the near-resonant Raman pump beam is turned on for a short duration ("protect" time) with the 650 nm repump beam off. Since this process is also limited by the rate of off-resonantly scattered photons, we expect this protect time to be similar to the optical pumping time of 10 μs as discussed in Sec. 15.3.

If this time is too short, insufficient photon scatters will result in reduced optical pumping. If this time is too long, polarization impurities in the near-resonant Raman pump beam and magnetic field imperfections will cause undesired scattering events from the $m_f = +\frac{1}{2}$ state to the $^2\mathrm{D}_{3/2}$ manifold. The efficiency of this step is limited to 50% by the branching ratios of the $^2\mathrm{P}_{1/2}$ state to each $^2\mathrm{S}_{1/2}$ Zeeman sublevel and the $^2\mathrm{D}_{3/2}$ manifold. Combined with the shelving efficiency of 88%, the maximum spin-state detection contrast is 44%.

To determine the optimal protect time, the procedure described in Fig. 15.3 is performed as the protect time is scanned without and with the far-off-resonant π pulse (Fig. 15.5) using the programs in App. E.3.6 and App. E.3.6.7 with data in each set taken randomly and interleaved with 1000 trials per point. Data analysis is performed using the code in App. E.1.11. Without the far-off-resonant π pulse, the resultant bright ion fraction is only from polarization impurities and magnetic field imperfections (red diamonds). With the far-off-resonant Raman π pulse, the resultant bright ion fraction is from both a successful Rabi π pulse and background effects (blue circles). Subtracting the two signals yields the detection contrast, which is used to obtain the optical protect time of 10 μs as expected from the optimal optical pumping time. The maximum experimental detection contrast is 43(2)% (Black squares), consistent with the maximum theoretical of 44%.

15.5. Rabi Oscillation Experiments

Estimating the far-off-resonant Raman carrier frequency and protect time using the results of the near-resonant Raman spectra fit results (Sec. 14.2) with long optical pumping time, the first Rabi oscillations observed in the group are collected. This group milestone data is presented in Fig. 15.6.

Using the optimal optical pumping and protect times described in Sec. 15.3 and Sec. 15.4, respectively, the far-off-resonant pulse time is swept while observing the bright ion fraction on the carrier as shown in Fig. 15.7. The programs in App. E.3.6 and App. E.3.6.9 are used for data collection with data taken randomly and interleaved with between 100 and 1000 (650 mean) trials per point. Data analysis is performed using the code in App. E.1.3. Note that the minimum and maximum bright fraction is as expected using the spin-dependent shelving technique used here. The sinusoidal fit function used is given below where A is the amplitude around the long-time average D, τ is the amplitude decay time (coherence time), and P is the period of oscillation.

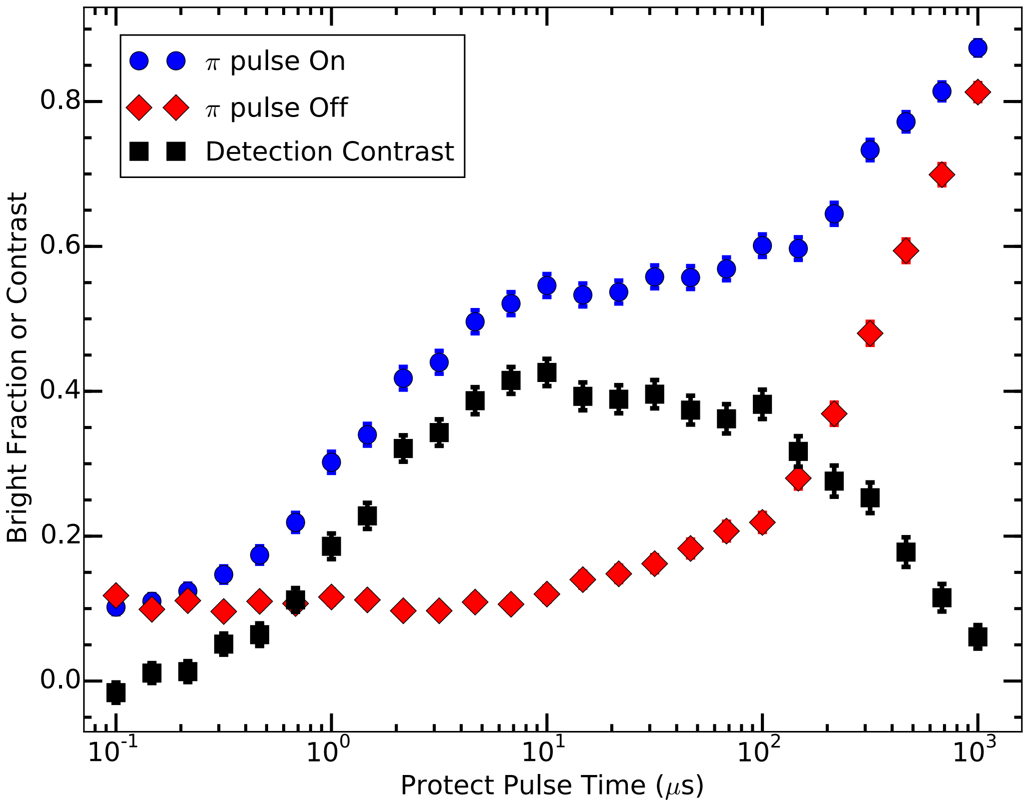


Figure 15.5. Determination of the optimal protect pulse time for the step shown in Fig. 15.3e. The detection contrast (black squares) is calculated as the difference between the experiment without the far-off-resonant Raman π pulse (red diamonds) and with the far-off-resonant π pulse (blue circles). The detection contrast is maximum at a protect pulse time of ≈10 μs as expected from the minimum optical pumping time determined in Sec. 15.3. Data points in each data set are taken randomly and interleaved. Error bars are given by binomial statistics.

$$\frac{A}{2}\mathrm{e}^{-\frac{t}{\tau}}\sin\left(2\pi\frac{t}{P}-\frac{\pi}{2}\right)+D \tag{15.1}$$

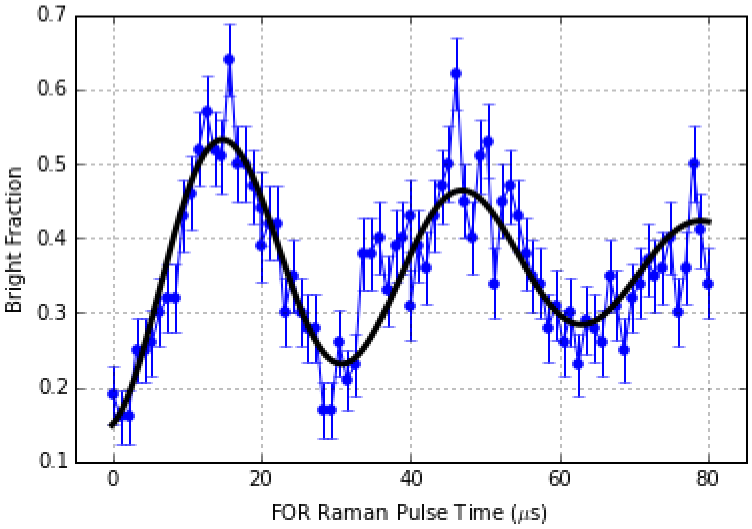


Figure 15.6. Far-off-resonant Raman excitation Rabi oscillations on the carrier with parameters estimated from the near-resonant Raman spectra fit results. This data is the first observation of Rabi oscillations in the group. Points are experimental data and the solid line is a fit to the data using Eqn. 15.1.

From the fit to the multiple Rabi oscillations, the far-off-resonant $\Omega_{\text{eff, carrier}} = 2\pi \times 50.92(3)$ kHz, which is fast compared with the extracted decoherence rate of $2\pi \times 7.7(3)$ kHz. Raman laser intensity fluctuations, the residual thermal spread in the two radial motional states [**129**], and non-zero $\bar{n}$ value in the cooled axial motional state (Debye-Waller factors [**93**]) are all possible sources of the observed decoherence. Consistent with the latter possibility, we observe the decoherence rate increase by a factor of 4 when not applying Raman sideband cooling.

Fig. 15.8 shows the same experiment performed on the blue sideband with 1000 trials per point. Due to the weak coupling to the motional mode, the blue sideband Rabi frequency is an order of magnitude lower, $\Omega_{\text{eff, BSB}} = 2\pi \times 6.02(8)$ kHz, which is similar to the decoherence rate of $2\pi \times 7.5(2)$ kHz. Thus, the amplitude of the Rabi oscillation

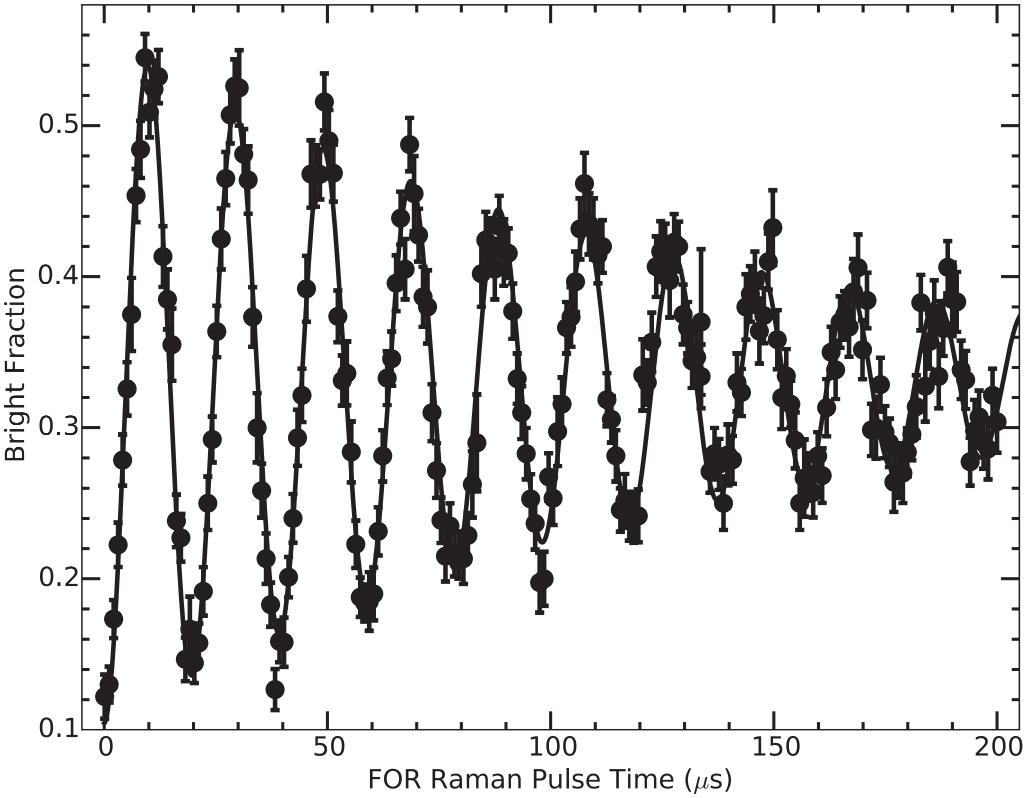


Figure 15.7. Far-off-resonant Raman excitation Rabi oscillations on the carrier. Multiple Rabi oscillations are observed on the carrier with $\Omega_{\text{eff, carrier}} = 2\pi \times 50.92(3)$ kHz with a coherence time of 130(5) μs. Data points are taken randomly and interleaved and the solid line is a fit to the data using Eqn. 15.1.

at the π pulse time is significantly decreased. Assuming that the ion is in the motional ground state for the Rabi oscillation experiments, the ratio of Rabi frequencies yields the Lamb-Dicke parameter $\eta = 0.118(2)$, comparable to the theoretical 0.102 for ^{138}Ba^{+} using our ion trap parameters.

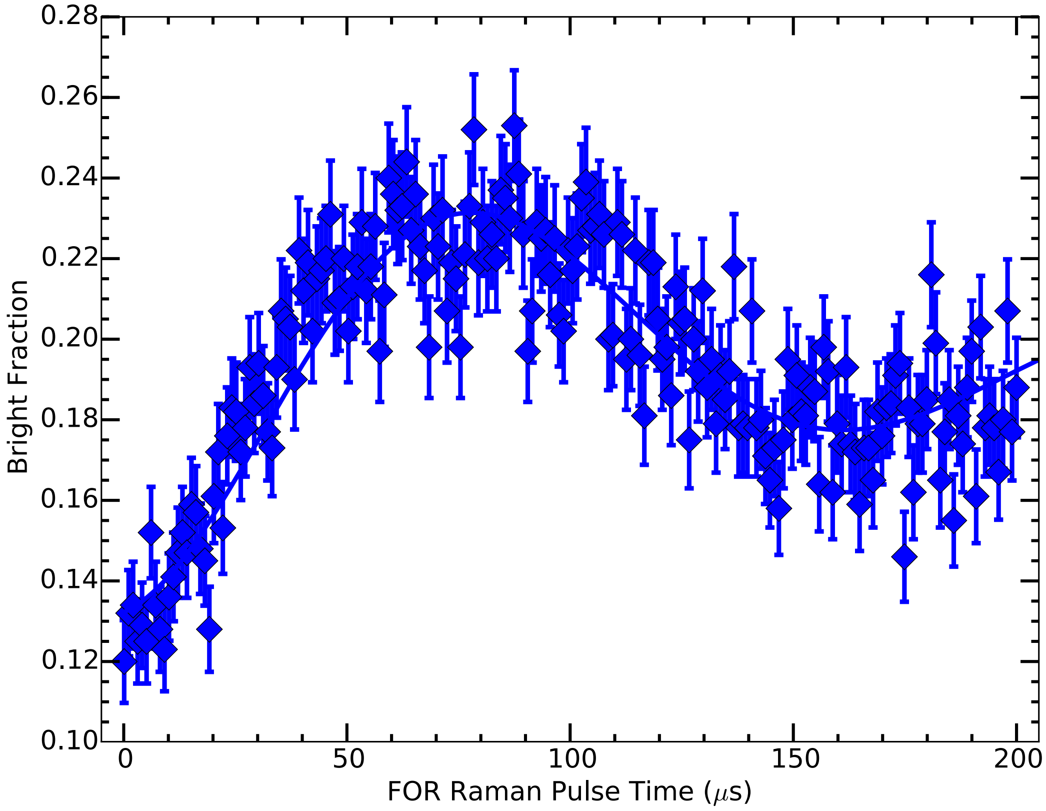


Figure 15.8. Far-off-resonant Raman excitation Rabi oscillations on the blue sideband. An efficient π pulse on the blue sideband cannot be performed due to the coherence time of 134(11) μs being shorter than the π pulse time with $\Omega_{\text{eff, BSB}} = 2\pi \times 6.02(8)$ kHz. Data points are taken randomly and interleaved and solid line is a fit to the data using Eqn. 15.1.

15.6. Far-off-resonant Frequency Spectrum and Temperature Measurement

To measure $\bar{n}$ using the far-off-resonant Raman technique, the far-off-resonant two-photon detuning is swept while driving π pulses on the carrier, red sideband, and blue sideband as shown in Fig. 15.9. Five frequency ranges are collected: the full carrier with 500, red sideband with 2000, blue sideband with 1000, and a region nearby each sideband to constrain the background with 1000 trials. The programs in App. E.3.6 and App. E.3.6.8 are used for data collection with in each data set taken randomly and interleaved. Data analysis is performed using the code in App. E.1.2.

The data are globally fit to the summation of three Lorentzian lineshapes as a function of the two-photon detuning δ using the equation below.

$$\frac{A_{\text{carrier}}\left(\frac{1}{2c_w}\right)^2}{(\delta-c_l)^2+\left(\frac{1}{2c_w}\right)^2}+\frac{A_{\text{BSB}}\left(\frac{1}{2sb_w}\right)^2}{\left(\delta-\left(c_l+\frac{\omega_z}{2\pi}\right)\right)^2+\left(\frac{1}{2sb_w}\right)^2}+\frac{A_{\text{RSB}}\left(\frac{1}{2sb_w}\right)^2}{\left(\delta-\left(c_l-\frac{\omega_z}{2\pi}\right)\right)^2+\left(\frac{1}{2sb_w}\right)^2}+D \quad (15.2)$$

where A_{carrier} is the carrier amplitude, c_w is the carrier width, and c_l is the carrier location. The sidebands are constrained to share the same width sb_w and $\frac{\omega_z}{2\pi}$ displacement from the carrier but with independent amplitudes A_{BSB} and A_{RSB} for the blue sideband and red sideband, respectively. A constant background term D is also included to account for the background 12% bright fraction due to imperfect shelving to the ${}^2\text{D}_{5/2}$ manifold.

When no near-resonant Raman sideband cooling is applied, equal-amplitude red and blue sidebands are observed, indicating $\bar{n} \gg 1$. With 10 ms of Raman sideband cooling, the red sideband amplitude decreases to a near-zero value of 0.017(5) (Fig. 15.10), and comparing the sideband amplitudes yields $\bar{n} = 0.15(6)$. This value is within agreement to the expectations from Sec. 14.1.4 and Sec. 14.2. All fit parameters are given in Table 15.1.

15.7. Cooling Efficiency

To verify that the lowest axial motional temperature is achieved while performing near-resonant Raman sideband cooling on the red sideband as given by the near-resonant Raman spectra, the near-resonant two-photon detuning is swept while measuring the final

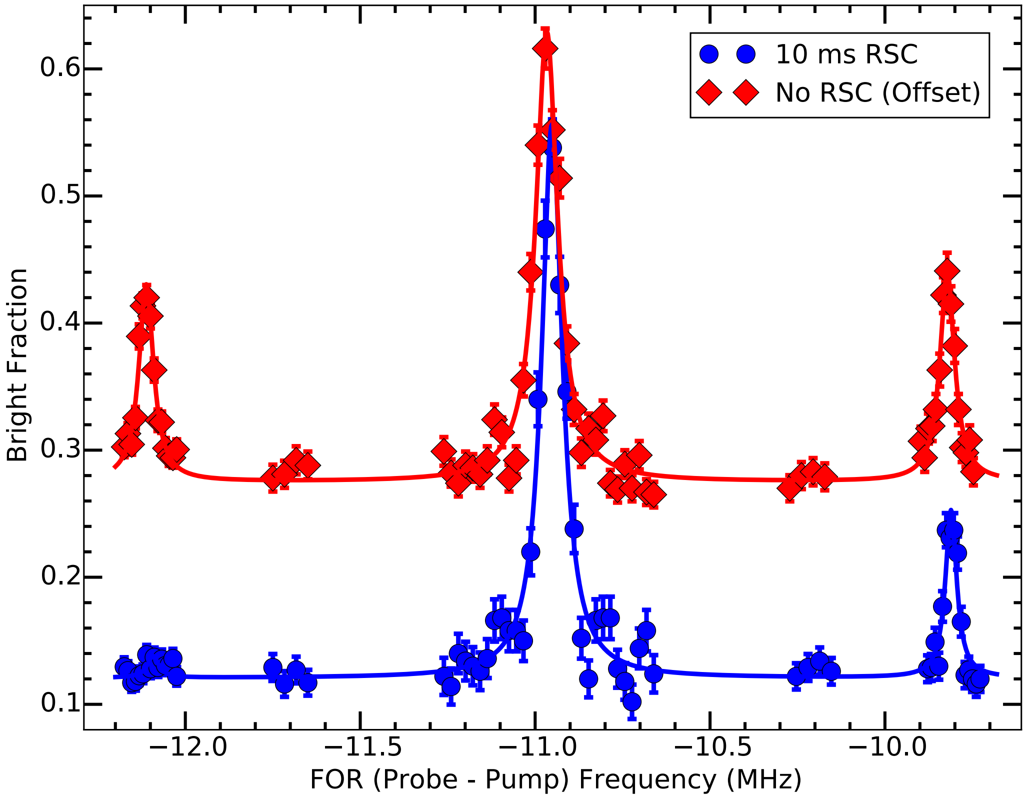


Figure 15.9. Far-off-resonant Raman frequency spectrum showing the full spectrum. A π pulse is used on the carrier. The longer sideband π pulse is used on the red sideband and blue sideband as well as the background points that help the fit establish the background shelving level. Near equal blue and red sideband amplitudes are observed if no near-resonant Raman cooling is performed (red diamonds, y-axis offset for display). The blue circles correspond to 10 ms of near-resonant Raman cooling on the red sideband with an observed $\bar{n} = 0.15(6)$. Data points in each data set are taken randomly and interleaved. Error bars are given by binomial statistics. The slight shift in carrier frequency between the traces is due to magnetic field drift changing the Zeeman splitting of the ${}^2\mathrm{S}_{1/2}$ state.

axial red sideband to blue sideband amplitude ratio after 10 ms of cooling as shown in Fig. 15.11. The programs in App. E.3.6 and App. E.3.6.1 are used for data collection

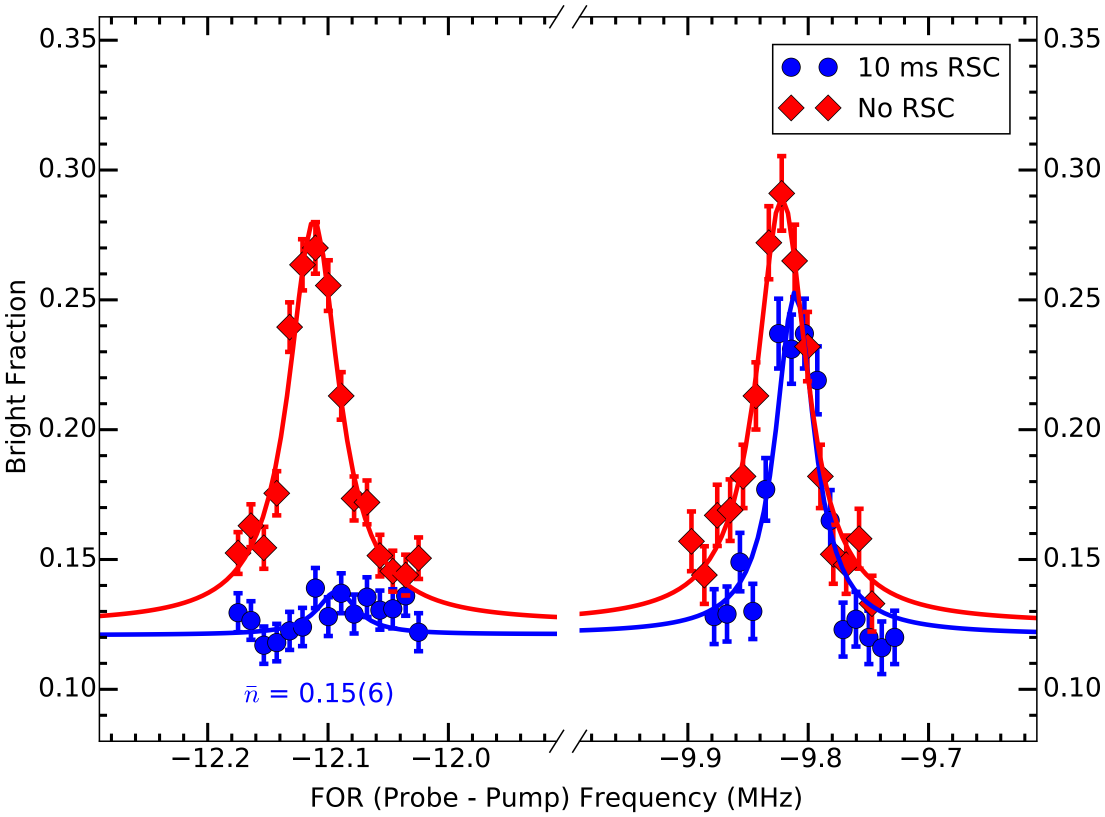


Figure 15.10. Far-off-resonant Raman frequency spectrum showing a zoom-in of the motional sidebands from Fig. 15.9. Near equal blue and red sideband amplitudes are observed if no near-resonant Raman cooling is performed (red diamonds). The blue circles correspond to 10 ms of near-resonant Raman cooling on the red sideband with an observed $\bar{n} = 0.15(6)$. Data points in each data set are taken randomly and interleaved. Error bars are given by binomial statistics. The slight shift in between the two data sets is due to magnetic field drift changing the Zeeman splitting of the $^2S_{1/2}$ state.

with data taken randomly and interleaved. Data analysis is performed using the code in App. E.1.7.

Table 15.1. Far-off-resonant Raman frequency spectrum fit results using Eqn. 15.2 and the data presented in Fig. 15.9 and Fig. 15.10.

	10 ms RSC	No RSC
A_{carrier}	0.44(2)	0.355(13)
c_l (MHz)	-10.9517(14)	-10.9668(8)
c_w (kHz)	71(4)	76(4)
$\frac{\omega_z}{2\pi}$ (MHz)	1.141(2)	1.1450(9)
A_{BSB}	0.132(11)	0.164(8)
sb_w (kHz)	39(5)	51(3)
A_{RSB}	0.017(5)	0.155(7)
D	0.121(2)	0.125(2)

This data is collected using a relatively fast four-point measurement in which the measured frequencies are the red sideband and blue sideband peaks and corresponding nearby background points. Each data point contains 1500 four-point cycles with data collected randomly and interleaved. The corresponding background point value is subtracted from the sideband peak value before the sideband amplitude ratio is calculated. Uncertainties are calculated by binomial statistics for all four points and propagated through to the amplitude ratio. If the propogated uncertainty is less than 0.05, the uncertainty is replaced with this value to account for some degree of magnetic field drift since this four-point measurement is very sensitive to the sideband peak locations. From the data, the lowest temperature is achieved close to the peak of the near-resonant red sideband location, as expected.

15.8. Cooling Timescale

To quantify the timescale at which the near-resonant Raman sideband cooling cools the ion to the motional ground state, the near-resonant Raman sideband cooling exposure time is varied while measureing the final red sideband to blue sideband amplitude ratio

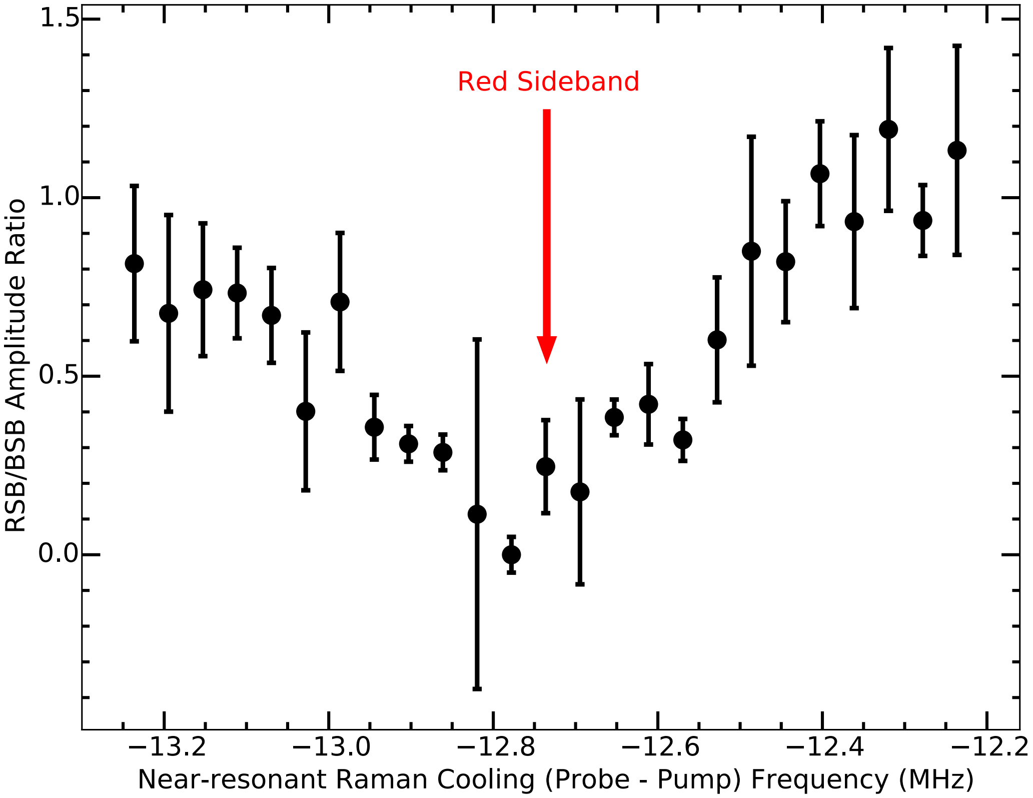


Figure 15.11. Near-resonant Raman frequency scan around the red sideband while measuring the RSB/BSB amplitude ratio (ion temperature) with 10 ms of sideband cooling. The minimum amplitude ratio is reached while sitting on the peak of the red sideband as measured by the near-resonant Raman fluorescence. Data points are taken randomly and interleaved. Error bars are given by the propagation of the binomial uncertainties for the red and blue sideband amplitudes.

as shown in Fig. 15.12 using the relatively fast four-point temperature measurement. The programs in App. E.3.6 and App. E.3.6.2 are used for data collection with data taken randomly and interleaved with each data point containing 1500 four-point cycles. Data analysis is performed using the code in App. E.1.8 and uncertainties are calculated by binomial statistics for all four points and propagated through to the amplitude ratio. Due to the magnetic field noise, data points with red sideband to blue sideband amplitude ratio uncertainty larger than 1 and $\bar{n}$ uncertainty larger than 0.5 are omitted.

The cooling rate is ∼1 ms which is reasonable considering the 1 ms near-resonant Raman PMT spectrum in Fig. 14.7. The spectrum in Fig. 14.7 is the integration over a time-varying signal; as the initial $\bar{n} \gg 1$ cools, the PMT collects photons due to red sideband Raman transitions. After ∼1 ms, fluorescence from the red sideband ceases, but the off-resonant excitation of the carrier has not yet masked the red sideband signal unlike after 10 ms of exposure.

15.9. Ion Heating Rate

A critical parameter for quantum information processing and QLS is the ion trap heating rate; if the system must remain in the ground state for the duration of an experiment, the heating rate will limit the length of time an individual experimental cycle has to complete. Previous work has shown that this anomalous heating rate scales typically as the size of the trap $\sim R^{-5}$ [**93**] and does not arise from circuit resistance thermal noise [**93**] or fluctuating patch potentials on the electrode surfaces [**389**] as these scale as R^{-2} and $\sim R^{-4}$, respectively. Since our trap is relatively large compared to surface traps, the anomalous heating rate is expected to be small and should be similar to the value 0.02(3) quanta/ms = 20(30) quanta/s measured by Webster [**150**].

To quantify the motional heating rate, the ion is cooled to the ground state using near-resonant Raman sideband cooling. After optical pumping, a time delay is scanned before the temperature measurement is performed using the relatively fast four-point

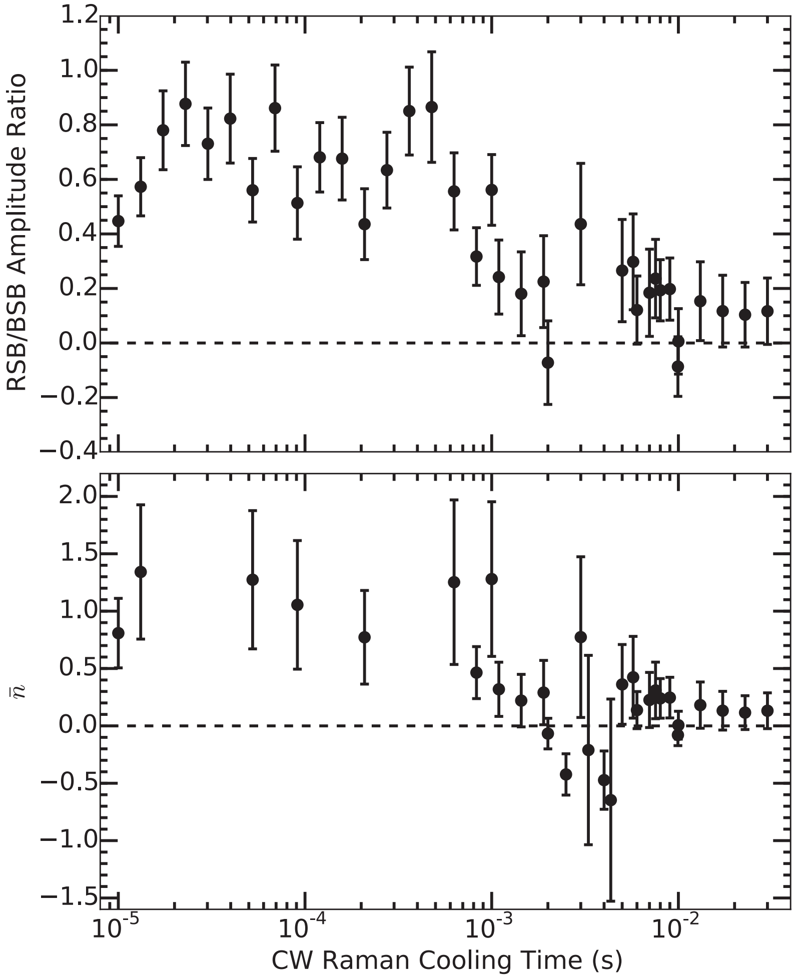


Figure 15.12. Near-resonant Raman cooling time scan while measuring the RSB/BSB amplitude ratio ($\bar{n}$). The near-resonant Raman sideband cooling time appears to be ∼1 ms based on both the amplitude ratio and $\bar{n}$ approaching zero. Horizontal dashed lines are at zero and are meant to guide the eye. Note that the approximation used to calculate $\bar{n}$ only holds for small RSB/BSB amplitude ratios, which explains why $\bar{n}$ saturates at ∼1.25 at short times.

Table 15.2. Linear Least-squares fit results to the data in Fig. 15.13.

Label	$\bar{n}(t=0)$ Constraint	Heating Rate (quanta/ms)	Heating Rate (quanta/s)	Intercept (quanta)
Unconstrained	Free	0.024(11)	24(11)	-0.08(11)
Constrained	0.0	0.016(6)	16(6)	0.0
Measured $\bar{n}$	0.15	0.002(6)	2(6)	0.15

temperature measurement. The programs in App. E.3.6 and App. E.3.6.5 are used for data collection with data taken randomly and interleaved with each data point containing between 500 and 1000 four-point cycles. Data analysis is performed using the code in App. E.1.4, and uncertainties are calculated by binomial statistics for all four points and propagated through to the amplitude ratio. Unfortunately, the four-point data collection measurement technique failed for this application with the data being too inconsistent.

Instead, the temperature measurement technique described in 15.6 is used with a variable delay added after the optical pumping step to allow for motional mode heating (Fig. 15.13). Each data point is a single measurement with 2000 cycles on the carrier and blue sideband and 4000 cycles on the red sideband with data collected randomly and interleaved. Linear least-squares fits are performed with various constraints placed on the vertical-axis intercept: unconstrained, constrained to zero, and constrained to the value measured in Sec. 15.6. Fit results, including heating rates, are listed in Table 15.2. All heating rates are sufficiently low that a single experimental cycle time can exceed ~10 ms without additional cooling stages, which should be more than sufficient time to perform mQLS.

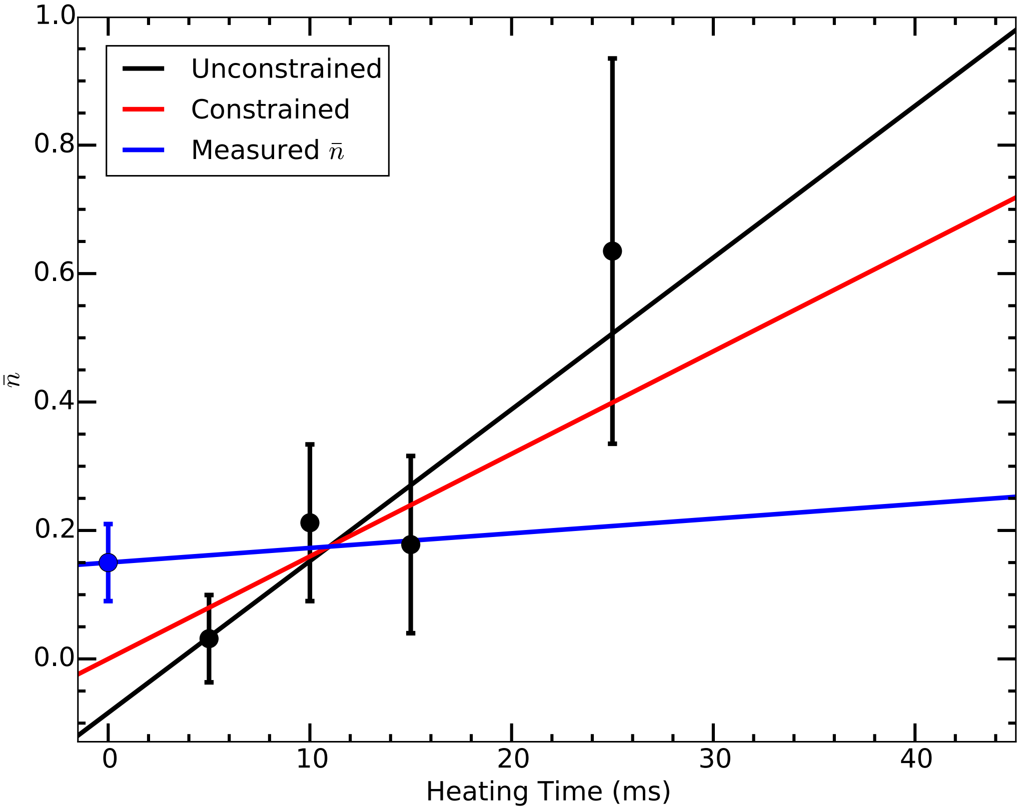


Figure 15.13. Ion trap heating rate measured using the technique described in 15.6. Data points represent experimental results and lines represent linear least-squares fits to the data under various constraints placed on the vertical-axis intercept: unconstrained (black), constrained to zero (red), and constrained to the value measured in Sec. 15.6 with associated data point added (blue). The slope of these lines, given in Table 15.2, yields the ion trap heating rate. Error bars are given by the propagation of the least squares curve fit errors for the red and blue sideband amplitudes.

CHAPTER 16

Apparatus Improvements

Although this work demonstrates sufficient experimental motional state preparation and detection contrast for QLS, a few simple improvements will decrease the final $\bar{n}$ value: pulsed Raman sideband cooling and EIT cooling; increase apparatus stability: the magnetic field, beam alignment, and the ECDL laser frequencies; and increase the far-off-resonant detection contrast: decrease the far-off-resonant Raman decoherence rate, increase the far-off-resonant sideband Rabi frequency, and improve the protect and shelving stage efficiencies. Additionally, the experimental cycle time may be reduced allowing faster data collection via a new trigger system, and the 650 nm Doppler and 791 nm photoionization beamlines could be improved.

16.1. Lowering $\bar{n}$

The measured final $\bar{n}$ in this work is sufficient for QLS. However, further decreasing this value will decrease the number of experimental cycles in a mQLS experiment required to reach a given statistical uncertainty. Pulsed Raman sideband cooling would be relatively simple to implement with the current optical arrangement. On the other hand, implementing EIT cooling would require additional optical infrastructure.

16.1.1. Pulsed Raman Sideband Cooling

To implement pulsed Raman sideband cooling, the far-off-resonant timing sequence in Fig. 15.3 would be modified; steps (c) and (d) on the red sideband would be repeated multiple times before step (d) interrogating the motional mode and following steps. In doing this, the ion, if not already in the motional ground state, would undergo a $n \rightarrow n-1$ transition and be again optically pumped back to the $m_j = +\frac{1}{2}$ state before motional state measurement. Repeating steps (c) and (d) ensure that the ion has multiple opportunities to decrease in n.

Experimentally, this would require (1) changing the timing sequence and (2) enabling the frequency shift keying (FSK) feature on the DDS. For (1), the Python backend controlling the counter cards would need to be modified to output multiple pulses, or an additional 6602 channel used to gate and/or trigger the added pulses. For (2), the far-off-resonant DDS channels would need to switch between the red sideband frequency and the measurement frequency quickly. The frequency shift keying feature of the DDS chip would perform this; the feature would need to be added to the Python backend code along with adding a trigger line and logic-level switch to trigger the frequency shift keying.

16.1.2. EIT Cooling

The ion could be cooled to lower $\bar{n}$ using EIT cooling [**382**] as discussed in Sec. 14.1.3.1. In this method, both the near-resonant Raman pump and probe beams are *blue-detuned* from the ${}^2\mathrm{S}_{1/2} \rightarrow {}^2\mathrm{P}_{1/2}$ transition. By adjusting the strong near-resonant Raman pump Rabi frequency and hence imparted light shift, the red sideband can be placed at the dark resonance of the carrier, preventing off-resonant carrier excitation. A comparison of

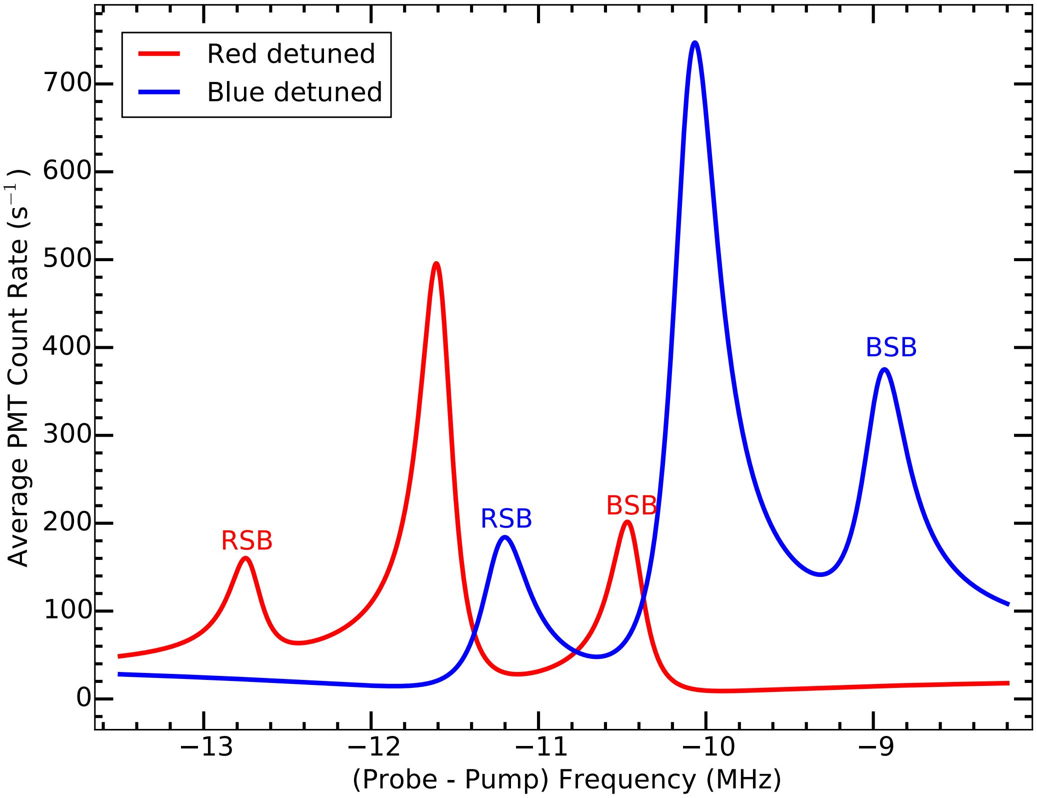


Figure 16.1. Expected near-resonant Raman spectra comparing red detuning and blue detuning (EIT case). The curves use the fit function described by Eqn. 14.36. The red curve, representing the red detuning case, uses the fit results for the 1 ms data in Fig. 14.7 with the RSB amplitude increased by a factor of four. The blue curve, representing the blue detuning (EIT) case, uses the same parameters as the red curve with the Raman pump Rabi frequency increased to provide the correct lightshift to place the red sideband over the dark resonance.

the expected spectra is shown in Fig. 16.1. Additionally, EIT cooling permits cooling of multiple motional modes simultaneously in a multi-ion crystal simultaneously permitted that they are not widely separated [**31, 384**]. However, if this is the case, frequency shift keying can be used to cool each mode sequentially.

Experimentally, this could be implemented by sending the near-resonant 493 nm light from before the near-resonant Raman pump and probe AOMs in Fig. 13.9 to an optical setup similar to the far-off-resonant Raman arrangement (Fig. 13.10). This added setup would first contain a single-pass AOM for power control followed by a switching element such as flip mirror or polarization optics, possibly PC controlled for automation, that would direct the light either directly to a single-mode, polarization-maintaining fiber patch cable or second AOM setup whose output would couple into the same fiber using a 50:50 NPBS. The output of this fiber would then proceed to the two near-resonant Raman AOMs after passing through a HWP, PBS, and window. The two added AOMs would result in the red-detuned near-resonant Raman detuning shifting further red (from the power control AOM) while the second would be chosen to reverse the sign of the final detuning.

16.2. Stability Improvements

Temperature measurements using the fast, four point temperature measurement are problematic with the sideband frequencies plagued by drifts, complicating data collection for measuring the near-resonant Raman sideband cooling efficiency (Sec. 15.7), cooling timescale (Sec. 15.8), and ion trap heating rate (Sec. 15.9). These drifts shift the sideband frequencies up to ~30 kHz, which is near the fitted ~50 kHz sideband width shown in Fig. 15.10.

This drift is likely be caused by (1) axial secular frequency drift, (2) significant (decoupled to zero-order) radial secular frequency drift, and/or (3) magnetic field (Zeeman

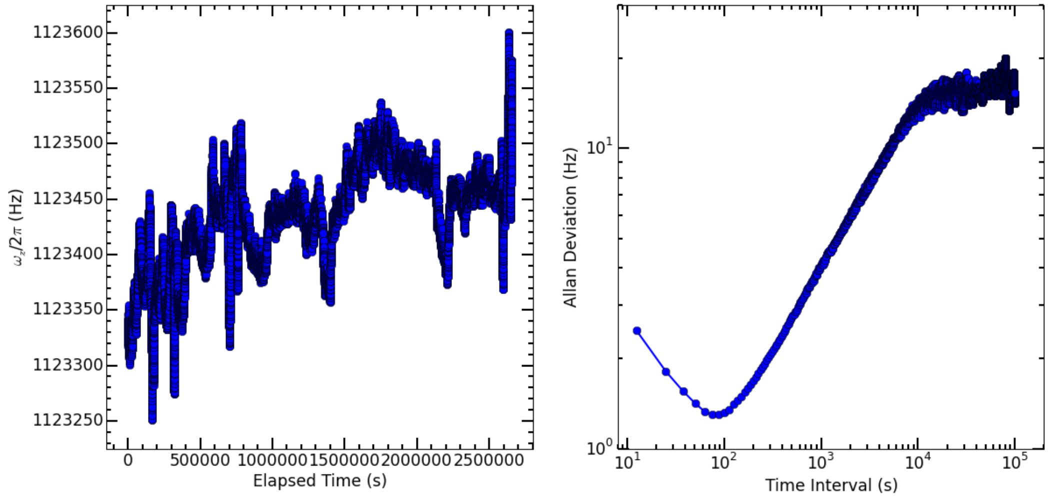


Figure 16.2. Axial secular frequency over time (left) and Allan deviation of the same data (right). From the data, the axial secular frequency is stable to within 350 Hz over 750 hours. From the Allan deviation, the axial secular frequency is stable for 100 s to ∼1 Hz.

splitting) drift. To investigate these possible causes, voltages are monitored using the code in App. E.3.10 with data analysis performed by the code in App. E.1.15.

For (1), the endcap voltages applied to the DC filter box are monitored each using a pair of voltage dividers comprised of 11.6 MΩ and 330 kΩ resistors. From this data, the resulting axial secular frequency is calculated (Fig. 16.2). From the data, the axial secular frequency drifts $\lesssim$350 Hz over >750 hours. Short-term stability is to the ∼1 Hz level over 100 s before drifts begin to occur. Over a typical experiment timescale of 1 hour, the axial secular frequency is stable to $\lesssim$100 Hz. Therefore, drift in the axial secular frequency is not a cause of the observed drift.

For (2), the directional coupler CPL FWD and CPL REV power detector voltages are monitored, and the resulting radial secular frequency is calculated (Fig. 16.3). From the

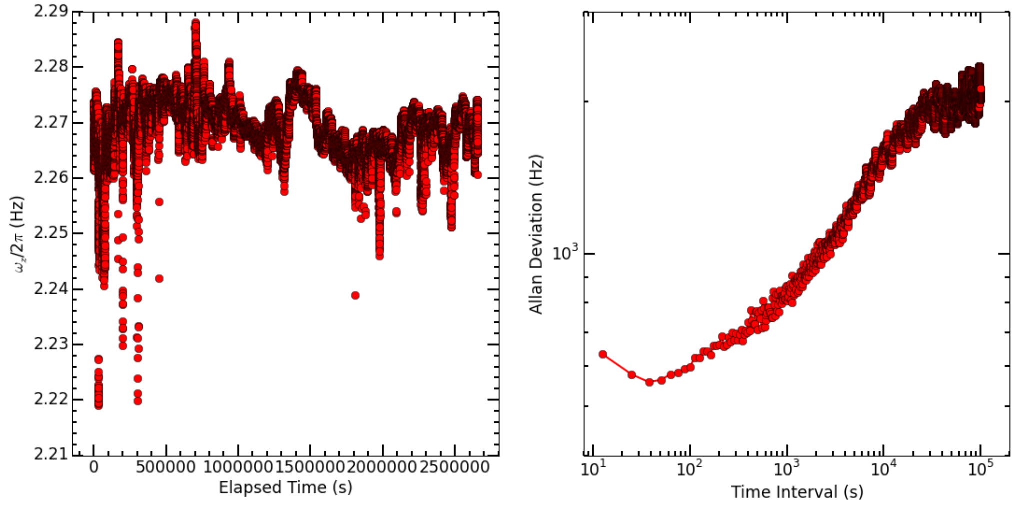


Figure 16.3. Radial secular frequency over time (left) and Allan deviation of the same data (right). From the data, the radial secular frequency is stable to within 70 kHz over 750 hours. From the Allan deviation, the radial secular frequency is stable for 50 s to ~1 kHz.

data, the radial secular frequency drifts $\lesssim$70 kHz over >750 hours. Short-term stability is to the ~600 Hz level over 50 s before drifts begin to occur. Over a typical experiment timescale of 1 hour, the radial secular frequency is stable to $\lesssim$1 kHz. Therefore, drift in the radial secular frequency is likely not a cause of the observed drift.

For (3), a Stefan Mayer FL1-500 fluxgate magnetometer with sensitivity of 2 mV/mG using an LT1010 buffer on the output is placed ~ 12 cm from the y-axis coil along the Raman pump beamline. The resulting Zeeman splitting is calculated by scaling the measured magnetic field (670 mG average) by the DC field at trap center (3.92 G) as measured by the ion (Fig. 16.4). From the data, the Zeeman splitting drifts $\lesssim$50 kHz over >750 hours. Short-term stability is to the ~400 Hz level over 60 s before drifts begin to occur. Over a typical experiment timescale of 1 hour, the axial secular frequency is stable

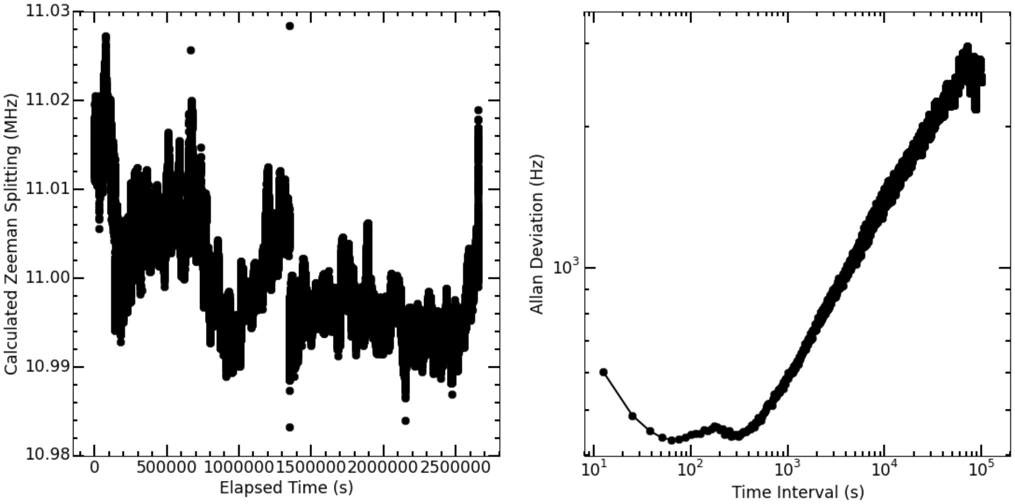


Figure 16.4. Zeeman splitting secular frequency over time (left) and Allan deviation of the same data (right). From the data, the Zeeman splitting is stable to within 50 kHz over 750 hours. Over a few-hour timescale (a few experiments), this can be as large as ∼30 kHz, which is our observed drift. From the Allan deviation, the Zeeman splitting is stable for 60 s to ∼400 Hz.

to $\lesssim$1 kHz. However, a few experiments in series can experience a drift of ∼30 kHz, which is our observed drift. Additionally, this drift can occur on fast timescales, which is noted during some experimental runs. Therefore, drift in the magnetic field (Zeeman splitting) is the likely cause of the observed drift.

Over the course of this work, we additionally note that the apparatus reacts poorly to temperature fluctuations of $\lesssim$2 °C and associated humidity fluctuations. These fluctuations occur regularly due to poor building HVAC, partially mitigated by tenting the experimental area and using a humidifier, dehumidifier, and spot chiller (Fig. 16.5). Temperature and humidity are monitored using a ThorLabs TSP01 with two external temperature probes; the first is suspended above the middle of the optical table and the second is taped to the helical RF resonator shield. Monitoring is performed via the logger application, although it is possible to include this into the voltages monitor code (App. E.3.10) using the monitor's programming references, and plotted using the code in App. E.1.14. Laser power fluctuations on the apparatus optical table are small. However, the Raman and shelve/deshelve lasers all required regular realignment via translation stages. Fortunately, the co-propagating beams rarely become misaligned from each other. The near-resonant Raman spectra visually identify which translation stage requires realignment using off-resonant ion fluorescence. If the lightshift of the carrier is lower than expected, the near-resonant Raman pump beamline is misaligned. If the lightshift is acceptable but the fluorescence rate low, the near-resonant Raman probe beam is misaligned.

16.2.1. Magnetic field Stabilization

From measurements above, magnetic field drift is a significant challenge for the apparatus, especially for the fast, four-point temperature measurements. The magnetic field that provides the Zeeman-splitting in the apparatus is produced by a Helmholtz coil pair driven by a constant-current power supply. Stabilizing this field can be done by using a PI servo loop with a 3-axis magnetometer (GMW Mag-03MCTP250) to modulate the analog voltage controlling the output current of the driving power supply. A heavily-updated PID servo circuit (App. A.4.1) will be used for this application.

Additionally, synchronizing the experimental trigger with the power line phase could improve fast magnetic field fluctuation effects. JCM built a circuit, powered by the wall, that outputs a TTL pulse near the top of the 60-Hz line phase. A J/K flip-flop is added to the circuit so the experimental start trigger causes only a single trigger pulse to output

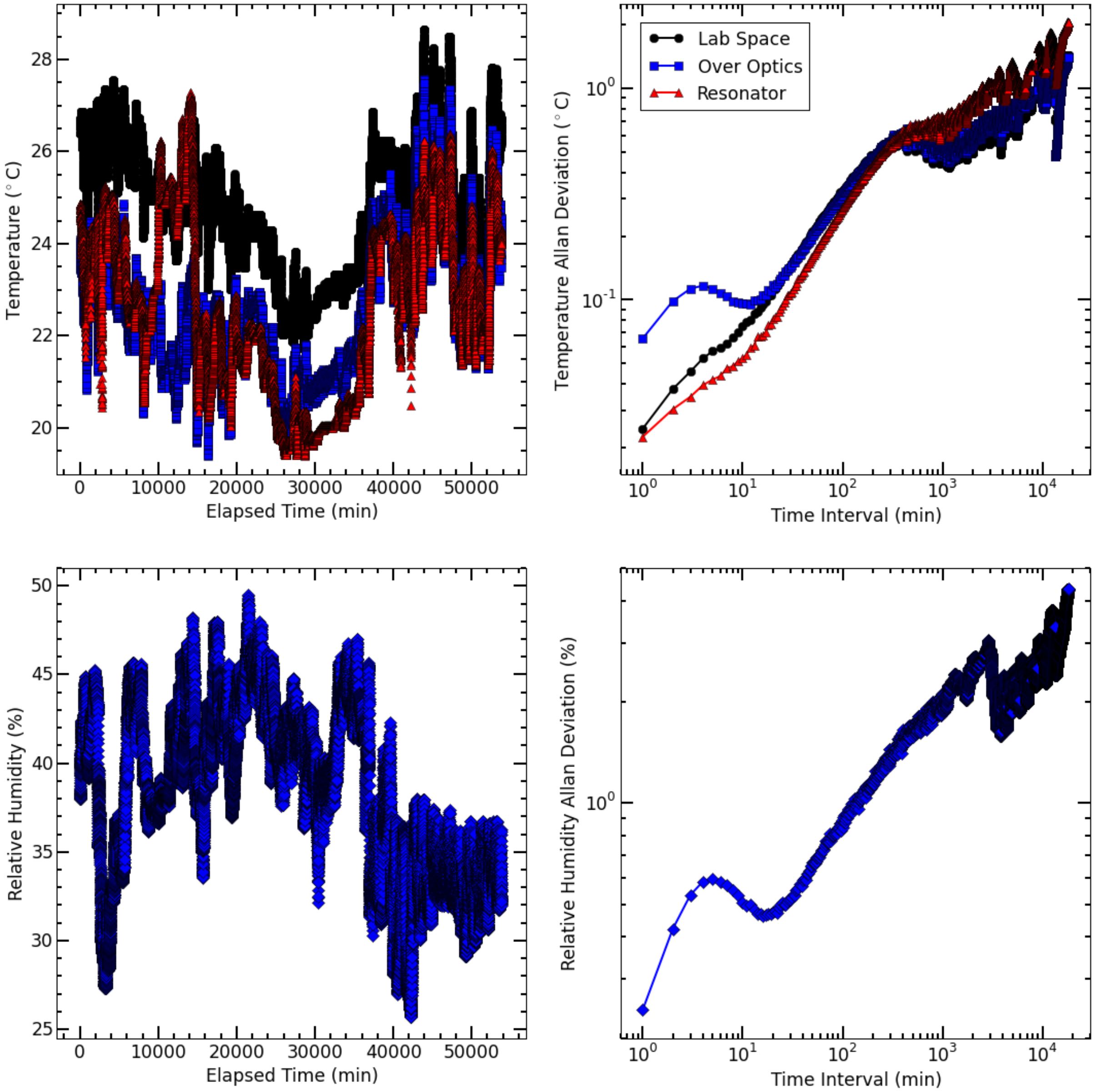


Figure 16.5. Temperature (upper) and humidity (lower) over time (left) and Allan deviation of the same data (right). From the data, the temperature and humidity are unstable on timescales less than one minute to over 1000 hours.

to the NI hardware. However, line triggering did not make an observable difference in the far-off-resonant Raman experiments and significantly increased the experimental cycle time.

After long-term magnetic field stability is achieved, further suppression of magnetic field noise may be required. This can be performed by observing the 60-Hz line and higher-harmonics noise on the magnetic field, and using a function generator triggered on line to feed-forward a correcting current. This would remove line noise from the magnetic field, removing the possible necessity of triggering on line.

16.2.2. Beam Alignment Automation

Many other ion trapping laboratories suffer much smaller temperature and humidity fluctuations compared to the space in which this apparatus resides. These laboratories, where gate fidelity is an upmost concern, utilize motorized and/or PZT-actuated translation stages and/or steering mirrors for their critical beams. This could be implemented on our apparatus in order to have routine automated alignment via PC with these calibrations and alignments interleaved during experimental data collection. This would mitigate temperature and humidity fluctuations changing the Raman and shelving/deshelving beam alignments, making the apparatus much more robust.

16.2.3. Additional Laser Stabilization

Another stability improvement, currently underway by PRS [**216**], is significantly improved laser frequency locks using temperature stabilized cavities (App. F.4.4). In this system, a Zerodur block holds six optical cavity mirror sets and resides in a temperature-stabilized, evacuated vacuum system (Fig. 16.6). The Pound-Drever-Hall technique will be used to actively lock laser systems to the cavities with up to $\sim$ 100 kHz lock bandwidth, which is a 5-order-of-magnitude improvement over the wavelength meter lock. Additionally, this system is anticipated to drift on a much slower timescale compared with the wavelength meter lock. Both of these improvements will decrease the timescale of drift and reduce acoustic vibration noise of the laser frequencies. Experimentally, this will result in a consistent initial ion temperature during Doppler cooling, perhaps lower than the Doppler limit using the sharper slope of one of the CPT dips. Additionally, the CPT dips can be used as an additional temperature measurement technique when $\bar{n} \gg 1$ [378].

Additionally, if the 455 nm laser source is converted to an ECDL and locked to the cavity, a Te_2 line lying $\sim$80 MHz from the $^{138}Ba^+$ transition [209] could provide a real reference to monitor the cavity drift. All lasers locked to the cavity could then be actively corrected for this temperature-induced frequency drift.

16.3. Contrast improvements

Increasing the experimental state detection contrast, which would decrease the number of experimental cycles in a mQLS experiment required to reach a given statistical uncertainty, can be achieved by improving two contrast reduction mechanisms in the experiment: (1) the far-off-resonant pulse decoherence rate and/or (2) the protect and shelving stages. Additionally, prospects of using STIRAP are discussed.

16.3.1. Far-off-resonant Laser Source and Decoherence Rate

The sideband Rabi frequency can be increased with additional far-off-resonant laser power and less detuning from the $^2S_{1/2} \rightarrow {}^2P_{1/2}$ transition, each providing a linear increase in

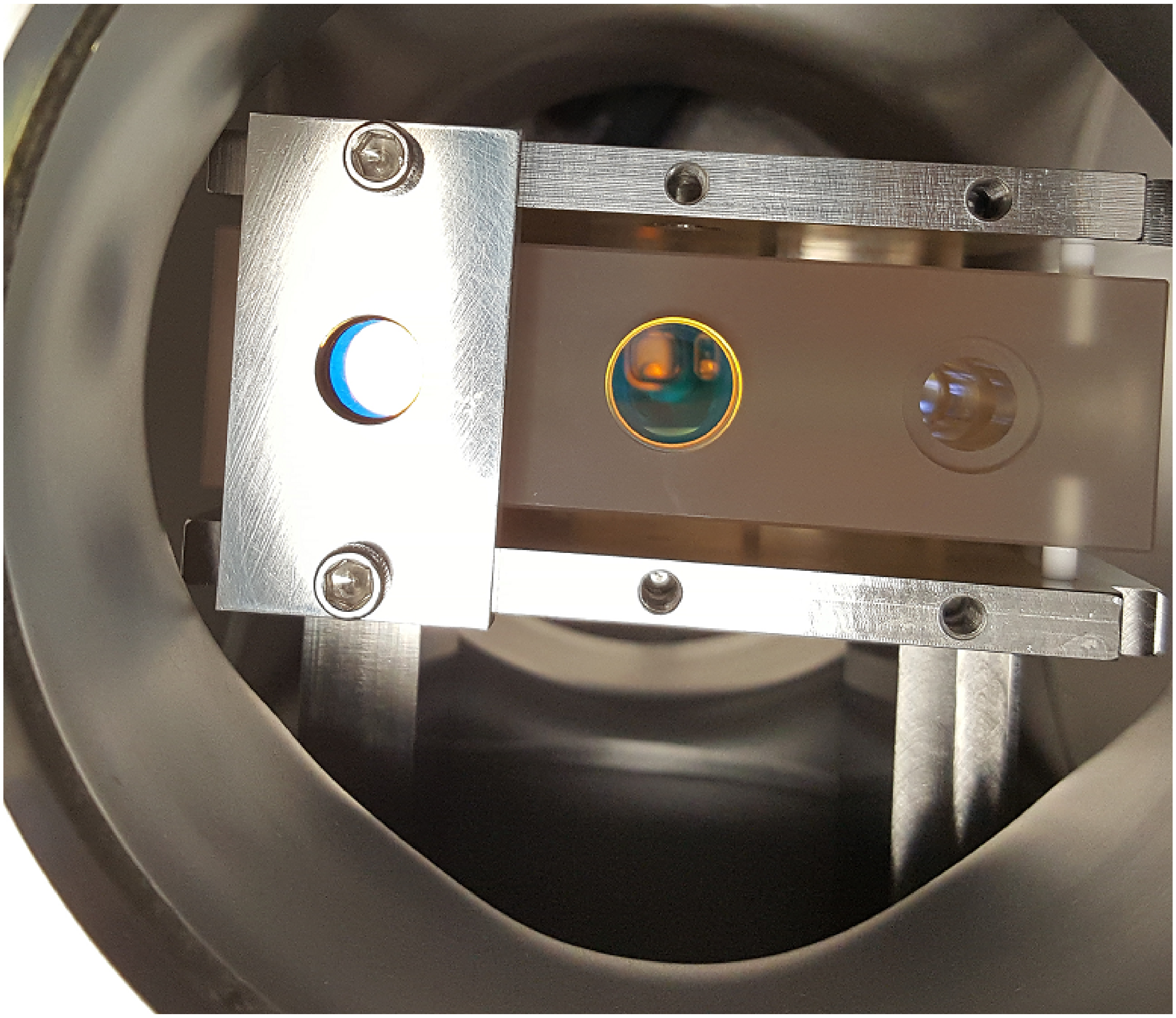


Figure 16.6. Picture of the partially-assembled temperature-stabilized Zerodur cavity block. From this angle, three of the six cavities can be seen. The mirrors are held in place by o-rings with pressure applied by the stainless steel mounting plates. The Zerodur cavity block sits in place aligned by pins such that the block is not over-constrained. Additionally note that the six inch six-way cross is temperature controlled and will be under vacuum.

the Rabi frequency of the Raman transition. Both are currently limited by the small island of stability of the far-off-resonant 493 nm laser source, resulting in a poor output spatial mode and large frequency detuning. Replacing the laser diode in the ECDL with a Toptica LD-0488-0060-2 laser diode, which is a hand-selected Nichia NDS4116 laser diode with a higher power rating and a center wavelength closer to 493 nm (492.0 nm), will allow both higher power and improved tunability. By driving an efficient π pulse on the motional sidebands, the number of experimental cycles required to reach the same statistical uncertainty would decrease by a factor of $\sim$2.

Furthermore, the new laser system may contain significantly less ASE, a decoherence mechanism, by running the laser at a much lower fractional output power than the diode's maximum. If ASE is determined to be a problem, a diffraction grating and long beamline before fiber coupling could be used to remove a significant amount of ASE.

Additionally, the current far-off-resonant Raman decoherence rate is likely dominated by residual thermal motion in the radial modes. The decoherence rate could be decreased by actively stabilizing the radial secular frequencies and applying Raman sideband cooling to the radial modes as well. This would reduce decoherence of the far-off-resonant Raman pulses, increasing detection contrast. Note that this would provide an additional contrast increase when combined with a higher sideband Rabi frequency.

Experimentally, this would involve adding another Raman beamline, phase coherent with the near-resonant Raman beams, to the viewport opposite the current near-resonant Raman pump beamline. This would provide a $\Delta\vec{k}$ along the radial direction. To stabilize the radial secular frequencies, a pickup wire could be attached to the helical resonator before the vacuum chamber RF feedthru, a small-value capacitor voltage divider employed followed by a rectifier bridge and LPF to ground. This would provide an accurate measurement of the RF voltage applied. A servo loop could then be implemented feeding back to the trap RF function generator RF AM or to a variable attenuator between function generator and amplifier similarly to that performed in Ref. [390].

16.3.2. Protect and Shelving Stages

To improve the protect stage, a Raman technique could again be used with the Raman pump beam and the 650 nm repump laser to selectively transfer population from the $^2\mathrm{S}_{1/2}$ $m_j = +\frac{1}{2}$ Zeeman sublevel to the $^2\mathrm{D}_{3/2}$ protected manifold, which has been previously demonstrated. This would increase the fidelity of the protect stage from 50% to ∼90% [142].

To improve the shelving stage, a Raman technique could similarly be used with the 455 nm and 614 nm laser sources. This has previously been demonstrated [391], increasing the shelving contrast from (88%-12%) = 76% to ∼95%. Experimentally, the latter improvement would require the 455 nm laser source be converted to an ECDL, and both of these improvements would require additional phase-coherent DDS AOM systems. The necessary steps to implement each improvement have been demonstrated and are well-described in this work.

Although the mentioned improvements would require at most one additional far-off-resonant 650 nm laser to couple between the $^2\mathrm{P}_{1/2}$ and $^2\mathrm{D}_{3/2}$ manifolds, the laser linewidth and locking requirements to address these dipole-allowed transitions are significantly more relaxed than those required for the 1.76 μm quadrupole laser. Combining all contrast-increasing solutions, we expect a spin-state detection contrast of >90%, a factor of 2 improvement over the current contrast.

16.3.3. STIRAP

For a time, we attempted to use STIRAP [392, 393, 394, 149, 395, 377, 132] using the near-resonant Raman beams in order to perform motional state detection. STIRAP is advantageous here due to the suppression of spontaneous scattering events, and would have allowed motional state measurements without using the additional 493 nm far-off-resonant Raman laser source. The hardware for this is described in Sec. 13.5, the experimental code is located in App. E.3.9, and the analysis code is located in App. E.1.13.

STIRAP between the two ${}^2\mathrm{S}_{1/2}$ Zeeman sublevels using either the Doppler, near-resonant, or far-off-resonant Raman beams, rather than the far-off-resonant 493 nm Raman laser, does not work due to the additional coupling of the π-polarized Raman probe beam to the ${}^2\mathrm{P}_{1/2}$, $m_j = -\frac{1}{2}$ Zeeman sublevel, confirmed by MGK by integrating the optical Bloch equations. Under further study, this additional coupling prevents a suitable adiabatic transfer state in the dressed atom picture [396].

The additional coupling of the π-polarized Raman probe beam to the ${}^2\mathrm{P}_{1/2}$, $m_j = -\frac{1}{2}$ Zeeman sublevel limitation can be avoided by using a species with hyperfine structure since the ground state hyperfine splitting is sufficiently large that the additional coupling is small, or by using a target state with higher multiplicity than the initial. For example, the ion could be prepared in the ${}^2\mathrm{S}_{1/2}$, $m_j = -\frac{1}{2}$ Zeeman sublevel. The two STIRAP beams (493 nm and 650 nm) would then both be σ^+ polarized and couple the initial state to the ${}^2\mathrm{D}_{3/2}$, $m_j = +\frac{3}{2}$ Zeeman sublevel. This would avoid a π-polarized beam that would couple multiple levels. In ${}^{138}\mathrm{Ba}^+$, this has been previously demonstrated [374].

STIRAP could be used to selectively transfer population from the ${}^2\mathrm{S}_{1/2}$ $m_j = -\frac{1}{2}$ Zeeman sublevel to the ${}^2\mathrm{D}_{3/2}$ protected manifold, which has been previously demonstrated in a similar system [374, 149]. STIRAP between the ${}^2S_{1/2}$ and ${}^2D_{5/2}$ states could also be implemented. These techniques could also serve both as the motional state detection and following state readout technique, replacing the far-off-resonant Raman source and state readout technique described in this work.

16.4. Speed improvements

The experimental cycle time is currently limited by the Python control software; software wait times and triggering only operate on the ∼10 ms timescale without substantial programming effort. Since $\sim 10^3$ Doppler cooling photon scatters are required to heat the ion back to the Doppler temperature, and the Doppler scattering rate is $\sim 10^8$ s^{-1}, time between successive experimental cycles can be $\ll 1$ ms.

To decrease this time, the trigger system could be moved to a 6602 channel programmed to output a pulse sequence at a specific rate for a given amount of time. The PMT would then programmed to buffer input counts for each experimental cycle for a set number of cycles before reading out the buffered PMT data. This could decrease the time between experimental cycles by one or two orders of magnitude, which is a significant improvement.

16.5. Beamline and Source Improvements

The next stage of the experiment, to be performed by MGK, is to transition to ablation loading for Ba^+ and AlH^+. For both cases, the ablation laser will use low power to produce a plume of neutrals that will then be photoionized via REMPI. The Ba source will be a piece of metallic barium, and photoionization sources the same as used previously in the apparatus. The precursor for AlH^+ is lithium aluminum hydride ($LiAlH_4$) [397].

Unfortunately, the precursor becomes unstable at vacuum baking temperatures and decays to a different compound.

Thus, a linear translation stage will be used to introduce the ablation targets under trap center only during loading. This section, valved off using a gate valve, will be baked to reach a vacuum pressure in the high 10^{-11} or low 10^{-10} Torr range without the ablation targets. The section will then be vented with dry nitrogen, ablation targets added, and pumped down.

The neutral AlH flux will then be ionized by a 448 nm (2+1) REMPI technique described in [**311**]. However, a significant fraction of the resulting ions are Al^+. The far-off-resonant frequency spectrum will be used to perform mass spectrometry [**398, 399**], and if the ion is Al^+, the trap parameters will be changed to selectively untrap lighter masses and keep the $^{138}Ba^+$ ion.

To make room for the linear translator and other new vacuum components, the 650 nm Doppler and 791 nm photoionization beamlines were moved.

16.5.1. 650 nm Doppler Beamline

The 650 nm Doppler beamline is changed to use the double-pass polarization arrangement (Sec. 8.7) identical to that of the 493 nm Doppler beamline (Sec. 13.3 and Fig. 13.9). The on-table fiber acts as a spatial filter, assisting in having a spot similar in size at the ion comparable to the 493 nm Doppler beam. Additionally, the new setup uses all ATM-801A1, 80 MHz AOMs with the switching AOM using (2) Minicircuits RF switches in series for high extinction as described in Sec. 13.4. Furthermore, the new arrangement eliminates higher-order AOM diffraction orders from being included in the beam that were previously included, decreasing the measured laser power required for Doppler cooling. Because of the lower power now required, a 330 kΩ TIA is used in the power control servo loop for improved power control.

16.5.2. 791 nm Photoionization Beamline

The 791 nm photoionization beamline is moved, AOM and shutter switching performed, and light coupled into the apparatus using a 780HP SM fiber patch cable similar to that of the 455 nm shelving arrangement. The output of this fiber is collimated, passes through a ThorLabs LA1484-B $f = 300$ mm lens as previously, and is reflected by two steering mirrors before combining with the Doppler beams. All three collimators are ThorLabs F230FC-780; however, the beam diverges significantly, so the on-table fiber coupling is not efficient. Regardless, >2 mW of power reaches the trap, which also helps reduce power broadening and isotope selective loading.

Conclusion

CHAPTER 17

Summary and Outlook

17.1. External Cavity Diode Laser Systems

An ever-increasing workhorse for tunable, CW laser applications, diode laser systems continue to cover a broader wavelength range. The in-house ECDL systems developed for Ba^+ applications and Rb spectroscopy provide a low-cost, customizable tunable laser solution for the laboratory. Future improvements can be made to the EDCL optomechanics, enclosure, and current driver modulation electronics to decrease the output frequency noise of one of these ECDL systems and decrease the short- and long-timescale linewidth of the output light.

17.2. Molecular Rotational Ground State Cooling

Rotational state analysis is an essential ingredient for a wide range of applications using trapped molecular ions, including precision spectroscopy, quantum information processing, ultracold chemistry, and coherent control. REMPD provides a destructive but powerful rotational state readout technique which can be relatively straightforward to implement in the laboratory. Our collaborators make necessary *ab initio* structure calculations for AlH^+, and we calculate the dissociation cross section for a convenient REMPD pathway from the ground vibrational state. The predicted cross section is large enough to be readily accessible in the laboratory, and we perform the first experimental demonstration of $(1 + 1')$ REMPD of this species. For all future experiments with this species, REMPD/time-of-flight mass spectrometry analysis will either serve as the implemented state readout technique or as an essential starting point for developing non-destructive alternatives.

We then use REMPD/time-of-flight mass spectrometry to analyze the rotational distribution of trapped AlH^+ to test broadband rotational optical cooling of AlH^+ with a spectrally filtered pulsed laser [114]. We realize rotational ground state cooling in AlH^+ using vibrationally-assisted broadband rotational optical cooling and demonstrate rotational cooling from room temperature to $3.8^{+0.9}_{-0.3}$ K with the ground state population increasing from $\sim$3% to $95.4^{+1.3}_{-2.1}$% on the 140(20) ms timescale.

In analogy with the numerous applications arising from translational laser cooling of only a handful of atomic species, we envision many new possibilities coming from rotational laser cooling of select molecular species with diagonal FCFs. Additionally, the combination of broadband rotational optical cooling with broadband vibrational cooling [105] could make possible optical rotational cooling of a still larger class of molecules. Building upon the current work, AlH^+ is a good candidate for many currently unrealized goals for trapped molecules, including molecular coherent control (alignment and orientation) and ultracold and quantum chemistry [1, 2, 3, 4, 5], quantum computing and simulation as well as hybrid quantum devices [6, 7, 8, 9, 10, 11], precision spectroscopy of importance to fundamental physics research [12, 13, 14, 15, 16, 17, 10, 11, 18, 19, 20], all-optical cooling of all internal and external molecular degrees of freedom, and single-molecule fluorescence detection.

17.3. Motional Ground State Cooling

We realize motional ground state cooling in $^{138}\mathrm{Ba}^+$ using Raman sideband cooling with the comparatively simple Zeeman structure of the $\mathrm{S}_{1/2}$ state rather than a barium isotope with hyperfine structure. Our near-resonant Raman sideband cooling method requires no additional lasers to those required for Doppler cooling, and we determine our relevant experimental parameters using a fit to the optical pumping fluorescence resulting from near-resonant Raman transitions. Using these spectra alone, it is apparent that sideband cooling yields a final $\bar{n} \ll 1$.

Employing a second, far-off-resonant laser driving Raman π pulses between the two Zeeman sublevels and using a spin-dependent shelving technique, we verify our final motional occupation number of $\bar{n} = 0.15(6)$, and that the optimal near-resonant Raman sideband cooling frequency corresponds to the red sideband. Along with broadband rotational optical cooling of AlH^+ also discussed in this work, these simple techniques for $^{138}\mathrm{Ba}^+$ will allow for motional ground state cooling of a two-ion crystal [79] and efficient mQLS to be performed. mQLS represents an important step forward for molecular quantum technology and will enable ultrahigh precision spectroscopy for fundamental studies, such as the search for time-variation of the proton-to-electron mass ratio.

17.4. New AlH^+ Sources

Two new sources for AlH^+ have been recently developed by MGK. The first, described in Ref. [311], produces a high flux of neutral AlH molecules in a technique similar to Ref. [400], which are then photoionized by $(2+1)$ REMPI at 448 nm. Unfortunately, the 448 nm REMPI process produces a significant Al^+ ion fraction due to predissociation.

Current efforts focus on investigating other REMPI transitions that may not have the same predissociation problem and sample purification in the ion trap by destabilizing the trap for Al^+ while keeping AlH^+ trapped.

The second source, ablation of a $LiAlH_4$ target, is described in Sec. 16.5. As noted in Ref. [397], this method produces a large flux of AlH neutrals, which then will be photoioinized. Direct loading of an ion can similarly be performed [75]. In the single-ion-scale trap, the mass of the dark ion can be measured by the axial secular frequencies of the two-ion crystal [398, 399]. If the resulting dark ion is not AlH^+, the trap parameters can be changed to selectively eject the dark ion and AlH^+ loading can be repeated.

17.5. Outlook

With customizable in-house laser systems developed, rotational ground state cooling of AlH^+ demonstrated, and motional ground state cooling of a $^{138}Ba^+$ ion performed, we are in a unique position to perform the first QLS on a *molecule*. The challenge of a reliable AlH^+ source has been overcome with two new sources for the molecular ion. All elements are in the process of being combined in the single-ion-scale apparatus with anticipated co-trapping and ground motional state cooling of a $^{138}Ba^+$ and AlH^+ ion pair anticipated over the next several months. Molecular ion ground state cooling, i.e. state preparation, will then be performed in this apparatus. The laboratory will then proceed with mQLS upon the foundation developed and extensively described in this work, searching for time-variation of the proton to electron mass ratio $\dot{\mu}$ in the hopes of uncovering interesting and new physics beyond the Standard Model.

References

[1] A. T. Grier, M. Cetina, F. Oručević, and V. Vuletić, "Observation of Cold Collisions between Trapped Ions and Trapped Atoms," *Phys. Rev. Lett.*, vol. 102, p. 223201, Jun 2009. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.102.223201

[2] S. Schmid, A. Härter, and J. H. Denschlag, "Dynamics of a Cold Trapped Ion in a Bose-Einstein Condensate," *Phys. Rev. Lett.*, vol. 105, p. 133202, Sep 2010. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.105.133202

[3] C. Zipkes, S. Palzer, L. Ratschbacher, C. Sias, and M. Köhl, "Cold Heteronuclear Atom-Ion Collisions," *Phys. Rev. Lett.*, vol. 105, p. 133201, Sep 2010. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.105.133201

[4] F. H. Hall, M. Aymar, M. Raoult, O. Dulieu, and S. Willitsch, "Light-assisted cold chemical reactions of barium ions with rubidium atoms," *Molecular Physics*, vol. 111, no. 12-13, pp. 1683–1690, 2013. [Online]. Available: http://dx.doi.org/10.1080/00268976.2013.770930

[5] M. Lemeshko, R. V. Krems, J. M. Doyle, and S. Kais, "Manipulation of molecules with electromagnetic fields," *Molecular Physics*, vol. 111, no. 12-13, pp. 1648–1682, 2013. [Online]. Available: http://dx.doi.org/10.1080/00268976.2013.813595

[6] D. DeMille, "Quantum Computation with Trapped Polar Molecules," *Phys. Rev. Lett.*, vol. 88, p. 067901, Jan 2002. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.88.067901

[7] A. André, D. DeMille, J. M. Doyle, M. D. Lukin, S. E. Maxwell, P. Rabl, R. J. Schoelkopf, and P. Zoller, "A coherent all-electrical interface between polar molecules and mesoscopic superconducting resonators," *Nature Physics*, vol. 2, no. 9, pp. 636–642, 2006. [Online]. Available: http://dx.doi.org/10.1038/nphys386

[8] L. D. Carr, D. DeMille, R. V. Krems, and J. Ye, "Cold and ultracold molecules: science, technology and applications," *New Journal of Physics*, vol. 11, no. 5, p. 055049, 2009. [Online]. Available: http://dx.doi.org/10.1088/1367-2630/11/5/055049

[9] D. I. Schuster, L. S. Bishop, I. L. Chuang, D. DeMille, and R. J. Schoelkopf, "Cavity QED in a molecular ion trap," *Phys. Rev. A*, vol. 83, p. 012311, Jan 2011. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.83.012311

[10] J.-P. Karr, A. Douillet, and L. Hilico, "Photodissociation of trapped H_2^+ ions for REMPD spectroscopy," *Applied Physics B*, vol. 107, no. 4, pp. 1043–1052, 2012. [Online]. Available: http://dx.doi.org/10.1007/s00340-011-4757-z

[11] V. Q. Tran, J.-P. Karr, A. Douillet, J. C. J. Koelemeij, and L. Hilico, "Two-photon spectroscopy of trapped HD^+ ions in the Lamb-Dicke regime," *Phys. Rev. A*, vol. 88, p. 033421, Sep 2013. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.88.033421

[12] S. Schiller and V. Korobov, "Tests of time independence of the electron and nuclear masses with ultracold molecules," *Phys. Rev. A*, vol. 71, p. 032505, Mar 2005. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.71.032505

[13] D. DeMille, S. B. Cahn, D. Murphree, D. A. Rahmlow, and M. G. Kozlov, "Using Molecules to Measure Nuclear Spin-Dependent Parity Violation," *Phys. Rev. Lett.*, vol. 100, p. 023003, Jan 2008. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.100.023003

[14] D. DeMille, S. Sainis, J. Sage, T. Bergeman, S. Kotochigova, and E. Tiesinga, "Enhanced Sensitivity to Variation of m_e/m_p in Molecular Spectra," *Phys. Rev. Lett.*, vol. 100, p. 043202, Jan 2008. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.100.043202

[15] A. Shelkovnikov, R. J. Butcher, C. Chardonnet, and A. Amy-Klein, "Stability of the Proton-to-Electron Mass Ratio," *Phys. Rev. Lett.*, vol. 100, p. 150801, Apr 2008. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.100.150801

[16] M. Kajita, M. Abe, M. Hada, and Y. Moriwaki, "Estimated accuracies of pure XH^+ (X: even isotopes of group II atoms) vibrational transition frequencies: towards the test of the variance in m_p/m_e," *Journal of Physics B: Atomic, Molecular and Optical Physics*, vol. 44, no. 2, p. 025402, 2011. [Online]. Available: http://dx.doi.org/10.1088/0953-4075/44/2/025402

[17] A. E. Leanhardt, J. L. Bohn, H. Loh, P. Maletinsky, E. R. Meyer, L. C. Sinclair, R. P. Stutz, and E. A. Cornell, "High-resolution spectroscopy on trapped molecular ions in rotating electric fields: A new approach for measuring the electron electric dipole moment," *Journal of Molecular Spectroscopy*, vol. 270, no. 1, pp. 1–25, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.jms.2011.06.007

[18] J. Baron, W. C. Campbell, D. DeMille, J. M. Doyle, G. Gabrielse, Y. V. Gurevich, P. W. Hess, N. R. Hutzler, E. Kirilov, I. Kozyryev *et al.*, "Order of magnitude smaller limit on the electric dipole moment of the electron," *Science*, vol. 343, no. 6168, pp. 269–272, 2014. [Online]. Available: http://dx.doi.org/10.1126/science.1248213

[19] J. Bagdonaite, W. Ubachs, M. T. Murphy, and J. B. Whitmore, "Constraint on a Varying Proton-Electron Mass Ratio 1.5 Billion Years after the Big Bang," *Phys. Rev. Lett.*, vol. 114, p. 071301, Feb 2015. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.114.071301

[20] W. Ubachs, J. Bagdonaite, E. J. Salumbides, M. T. Murphy, and L. Kaper, "*Colloquium*: Search for a drifting proton-electron mass ratio from H_2," *Rev. Mod. Phys.*, vol. 88, p. 021003, May 2016. [Online]. Available: http://dx.doi.org/10.1103/RevModPhys.88.021003

[21] R. M. Godun, P. B. R. Nisbet-Jones, J. M. Jones, S. A. King, L. A. M. Johnson, H. S. Margolis, K. Szymaniec, S. N. Lea, K. Bongs, and P. Gill, "Frequency Ratio of Two Optical Clock Transitions in ^{171}Yb^{+} and Constraints on the Time Variation of Fundamental Constants," *Phys. Rev. Lett.*, vol. 113, p. 210801, Nov 2014. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.113.210801

[22] T. W. Hänsch and A. L. Schawlow, "Cooling of gases by laser radiation," *Optics Communications*, vol. 13, no. 1, pp. 68–69, 1975. [Online]. Available: http://dx.doi.org/10.1016/0030-4018(75)90159-5

[23] W. Neuhauser, M. Hohenstatt, P. Toschek, and H. Dehmelt, "Optical-Sideband Cooling of Visible Atom Cloud Confined in Parabolic Well," *Phys. Rev. Lett.*, vol. 41, pp. 233–236, Jul 1978. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.41.233

[24] D. J. Wineland, R. E. Drullinger, and F. L. Walls, "Radiation-Pressure Cooling of Bound Resonant Absorbers," *Phys. Rev. Lett.*, vol. 40, pp. 1639–1642, Jun 1978. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.40.1639

[25] E. L. Raab, M. Prentiss, A. Cable, S. Chu, and D. E. Pritchard, "Trapping of Neutral Sodium Atoms with Radiation Pressure," *Phys. Rev. Lett.*, vol. 59, pp. 2631–2634, Dec 1987. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.59.2631

[26] W. Paul, "Electromagnetic traps for charged and neutral particles," *Rev. Mod. Phys.*, vol. 62, pp. 531–540, Jul 1990. [Online]. Available: http://dx.doi.org/10.1103/RevModPhys.62.531

[27] P. Bowe, L. Hornekær, C. Brodersen, M. Drewsen, J. S. Hangst, and J. P. Schiffer, "Sympathetic Crystallization of Trapped Ions," *Phys. Rev. Lett.*, vol. 82, pp. 2071–2074, Mar 1999. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.82.2071

[28] K. Mølhave and M. Drewsen, "Formation of translationally cold MgH^+ and MgD^+ molecules in an ion trap," *Physical Review A*, vol. 62, p. 011401, June 2000. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.62.011401

[29] B. Roth, A. Ostendorf, H. Wenz, and S. Schiller, "Production of large molecular ion crystals via sympathetic cooling by laser-cooled Ba^+," *Journal of Physics B: Atomic, Molecular and Optical Physics*, vol. 38, no. 20, pp. 3673–3685, October 2005. [Online]. Available: http://dx.doi.org/10.1088/0953-4075/38/20/004

[30] X. Tong, A. Winney, and S. Willitsch, "Sympathetic Cooling of Molecular Ions in Selected Rotational and Vibrational States Produced by Threshold Photoionization," *Physical Review Letters*, vol. 105, p. 143001, September 2010. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.105.143001

[31] Y. Lin, J. P. Gaebler, T. R. Tan, R. Bowler, J. D. Jost, D. Leibfried, and D. J. Wineland, "Sympathetic Electromagnetically-Induced-Transparency Laser Cooling of Motional Modes in an Ion Chain," *Phys. Rev. Lett.*, vol. 110, p. 153002, Apr 2013. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.110.153002

[32] K. B. Davis, M. O. Mewes, M. R. Andrews, N. J. van Druten, D. S. Durfee, D. M. Kurn, and W. Ketterle, "Bose-Einstein Condensation in a Gas of Sodium Atoms," *Phys. Rev. Lett.*, vol. 75, pp. 3969–3973, Nov 1995. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.75.3969

[33] J. R. Ensher, D. S. Jin, M. R. Matthews, C. E. Wieman, and E. A. Cornell, "Bose-Einstein Condensation in a Dilute Gas: Measurement of Energy and Ground-State Occupation," *Phys. Rev. Lett.*, vol. 77, pp. 4984–4987, Dec 1996. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.77.4984

[34] B. C. Young, "A Measurement of the Fine-structure Constant Using Atom Interferometry," Ph.D. dissertation, Stanford University, January 1997.

[35] K. Levin, A. Fetter, and D. Stamper-Kurn, *Ultracold Bosonic and Fermionic Gases*. Elsevier, 2012, vol. 5.

[36] A. Peters, K. Y. Chung, and S. Chu, "High-precision gravity measurements using atom interferometry," *Metrologia*, vol. 38, no. 1, p. 25, 2001. [Online]. Available: http://dx.doi.org/10.1088/0026-1394/38/1/4

[37] M. Fattori, C. D'Errico, G. Roati, M. Zaccanti, M. Jona-Lasinio, M. Modugno, M. Inguscio, and G. Modugno, "Atom Interferometry with a Weakly Interacting Bose-Einstein Condensate," *Phys. Rev. Lett.*, vol. 100, p. 080405, Feb 2008. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.100.080405

[38] S. M. Dickerson, J. M. Hogan, A. Sugarbaker, D. M. S. Johnson, and M. A. Kasevich, "Multiaxis Inertial Sensing with Long-Time Point Source Atom Interferometry," *Phys. Rev. Lett.*, vol. 111, p. 083001, Aug 2013. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.111.083001

[39] M. Greiner, C. A. Regal, and D. S. Jin, "Emergence of a molecular Bose–Einstein condensate from a Fermi gas," *Nature*, vol. 426, no. 6966, pp. 537–540, 2003. [Online]. Available: http://dx.doi.org/10.1038/nature02199

[40] I. Kominis, T. Kornack, J. Allred, and M. Romalis, "A subfemtotesla multichannel atomic magnetometer," *Nature*, vol. 422, no. 6932, pp. 596–599, 2003. [Online]. Available: http://dx.doi.org/10.1038/nature01484

[41] T. Rosenband, P. O. Schmidt, D. B. Hume, W. M. Itano, T. M. Fortier, J. E. Stalnaker, K. Kim, S. A. Diddams, J. C. J. Koelemeij, J. C. Bergquist, and D. J. Wineland, "Observation of the $^1S_0 \rightarrow ^3 P_0$ Clock Transition in ^{27}Al$^+$," *Phys. Rev. Lett.*, vol. 98, p. 220801, May 2007. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.98.220801

[42] B. Bloom, T. Nicholson, J. Williams, S. Campbell, M. Bishof, X. Zhang, W. Zhang, S. Bromley, and J. Ye, "An optical lattice clock with accuracy and stability at the 10^-18 level," *Nature*, vol. 506, no. 7486, pp. 71–75, 2014. [Online]. Available: http://dx.doi.org/10.1038/nature12941

[43] A. D. Ludlow, M. M. Boyd, J. Ye, E. Peik, and P. O. Schmidt, "Optical atomic clocks," *Rev. Mod. Phys.*, vol. 87, pp. 637–701, Jun 2015. [Online]. Available: http://dx.doi.org/10.1103/RevModPhys.87.637

[44] P. Kok, W. J. Munro, K. Nemoto, T. C. Ralph, J. P. Dowling, and G. J. Milburn, "Linear optical quantum computing with photonic qubits," *Rev. Mod. Phys.*, vol. 79, pp. 135–174, Jan 2007. [Online]. Available: http://dx.doi.org/10.1103/RevModPhys.79.135

[45] N. N. Oza, Y.-P. Huang, and P. Kumar, "Entanglement-Preserving Photonic Switching: Full Cross-Bar Operation With Quantum Data Streams," *IEEE Photonics Technology Letters*, vol. 26, no. 4, pp. 356–359, 2014. [Online]. Available: http://dx.doi.org/10.1109/LPT.2013.2293953

[46] R. Mann, *An Introduction to Particle Physics and the Standard Model*. CRC Press, 2011.

[47] C. collaboration *et al.*, "Evidence for the direct decay of the 125 GeV Higgs boson to fermions," *Nature Physics*, vol. 10, no. 8, pp. 557–560, 2014. [Online]. Available: http://dx.doi.org/10.1038/nphys3005

[48] SNO Collaboration, "Measurement of the Rate of $\nu_e + d \rightarrow p + p + e^-$ Interactions Produced by ^8B Solar Neutrinos at the Sudbury Neutrino Observatory," *Phys. Rev. Lett.*, vol. 87, p. 071301, Jul 2001. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.87.071301

[49] V. Barger, D. Marfatia, and K. L. Whisnant, *The Physics of Neutrinos*. Princeton University Press, 2012.

[50] T. Mannel, "Theory and Phenomenology of CP Violation," *Nuclear Physics B-Proceedings Supplements*, vol. 167, pp. 170–174, 2007. [Online]. Available: http://dx.doi.org/10.1016/j.nuclphysbps.2006.12.083

[51] U. Sarkar, *Particle and Astroparticle Physics*. CRC Press, 2007.

[52] F. Jegerlehner and A. Nyffeler, "The muon $g - 2$," *Physics Reports*, vol. 477, no. 1, pp. 1–110, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.physrep.2009.04.003

[53] M. G. Kozlov and L. N. Labzowsky, "Parity violation effects in diatomics," *Journal of Physics B: Atomic, Molecular and Optical Physics*, vol. 28, no. 10, p. 1933, 1995. [Online]. Available: http://dx.doi.org/10.1088/0953-4075/28/10/008

[54] J. J. Hudson, B. E. Sauer, M. R. Tarbutt, and E. A. Hinds, "Measurement of the Electron Electric Dipole Moment Using YbF Molecules," *Phys. Rev. Lett.*, vol. 89, p. 023003, Jun 2002. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.89.023003

[55] J. J. Hudson, D. M. Kara, I. Smallman, B. E. Sauer, M. R. Tarbutt, and E. A. Hinds, "Improved measurement of the shape of the electron," *Nature*, vol. 473, no. 7348, pp. 493–496, 2011. [Online]. Available: http://dx.doi.org/10.1038/nature10104

[56] J. Barrow, "Varying constants," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 363, no. 1834, pp. 2139–2153, 2005. [Online]. Available: http://dx.doi.org/10.1098/rsta.2005.1634

[57] F. Wilczek, “Fundamental constants,” *arXiv:0708.4361*, 2007. [Online]. Available: https://arxiv.org/abs/0708.4361

[58] J.-P. Uzan, “Varying constants, gravitation and cosmology,” *Living Rev. Relativity*, vol. 14, no. 2, 2011. [Online]. Available: http://dx.doi.org/10.12942/lrr-2011-2

[59] B. N. Spaun, “A Ten-Fold Improvement to the Limit of the Electron Electric Dipole Moment,” Ph.D. dissertation, Harvard University, May 2014.

[60] C. Chin, V. Flambaum, and M. Kozlov, “Ultracold molecules: new probes on the variation of fundamental constants,” *New Journal of Physics*, vol. 11, no. 5, p. 055048, 2009. [Online]. Available: http://dx.doi.org/10.1088/1367-2630/11/5/055048

[61] M. G. Kozlov and S. A. Levshakov, “Microwave and submillimeter molecular transitions and their dependence on fundamental constants,” *Annalen der Physik*, vol. 525, no. 7, pp. 452–471, 2013. [Online]. Available: http://dx.doi.org/10.1002/andp.201300010

[62] P. Jansen, H. L. Bethlem, and W. Ubachs, “Perspective: Tipping the scales: Search for drifting constants from molecular spectra,” *The Journal of chemical physics*, vol. 140, no. 1, p. 010901, 2014. [Online]. Available: http://dx.doi.org/10.1063/1.4853735

[63] L. F. Pašteka, A. Borschevsky, V. V. Flambaum, and P. Schwerdtfeger, “Search for the variation of fundamental constants: Strong enhancements in $X\ ^2\Pi$ cations of dihalogens and hydrogen halides,” *Phys. Rev. A*, vol. 92, p. 012103, Jul 2015. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.92.012103

[64] V. A. Dzuba, V. V. Flambaum, and J. K. Webb, “Space-Time Variation of Physical Constants and Relativistic Corrections in Atoms,” *Phys. Rev. Lett.*, vol. 82, pp. 888–891, Feb 1999. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.82.888

[65] S. Levshakov, P. Molaro, A. Lapinov, D. Reimers, C. Henkel, and T. Sakai, “Searching for chameleon-like scalar fields with the ammonia method,” *Astronomy & Astrophysics*, vol. 512, p. A44, 2010. [Online]. Available: http://dx.doi.org/10.1051/0004-6361/201015332

[66] J. K. Webb, J. A. King, M. T. Murphy, V. V. Flambaum, R. F. Carswell, and M. B. Bainbridge, “Indications of a Spatial Variation of the Fine Structure Constant,” *Phys. Rev. Lett.*, vol. 107, p. 191101, Oct 2011. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.107.191101

[67] J. Bagdonaite, P. Jansen, C. Henkel, H. L. Bethlem, K. M. Menten, and W. Ubachs, "A stringent limit on a drifting proton-to-electron mass ratio from alcohol in the early universe," *Science*, vol. 339, no. 6115, pp. 46–48, 2013. [Online]. Available: http://dx.doi.org/10.1126/science.1224898

[68] V. Flambaum and V. Dzuba, "Search for variation of the fundamental constants in atomic, molecular, and nuclear spectra," *Canadian Journal of Physics*, vol. 87, no. 1, pp. 25–33, 2009. [Online]. Available: http://dx.doi.org/10.1139/p08-072

[69] N. Huntemann, B. Lipphardt, C. Tamm, V. Gerginov, S. Weyers, and E. Peik, "Improved Limit on a Temporal Variation of m_p/m_e from Comparisons of Yb^+ and Cs Atomic Clocks," *Phys. Rev. Lett.*, vol. 113, p. 210802, Nov 2014. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.113.210802

[70] V. V. Flambaum and A. F. Tedesco, "Dependence of nuclear magnetic moments on quark masses and limits on temporal variation of fundamental constants from atomic clock experiments," *Phys. Rev. C*, vol. 73, p. 055501, May 2006. [Online]. Available: http://dx.doi.org/10.1103/PhysRevC.73.055501

[71] T. H. Dinh, A. Dunning, V. A. Dzuba, and V. V. Flambaum, "Sensitivity of hyperfine structure to nuclear radius and quark mass variation," *Phys. Rev. A*, vol. 79, p. 054102, May 2009. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.79.054102

[72] P. O. Schmidt, T. Rosenband, C. Langer, W. M. Itano, J. C. Bergquist, and D. J. Wineland, "Spectroscopy Using Quantum Logic," *Science*, vol. 309, no. 5735, pp. 749–752, 2005. [Online]. Available: http://dx.doi.org/10.1126/science.1114375

[73] C. W. Chou, D. B. Hume, J. C. J. Koelemeij, D. J. Wineland, and T. Rosenband, "Frequency Comparison of Two High-Accuracy Al^+ Optical Clocks," *Phys. Rev. Lett.*, vol. 104, p. 070802, Feb 2010. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.104.070802

[74] B. Roth, P. Blythe, and S. Schiller, "Motional resonance coupling in cold multispecies Coulomb crystals," *Phys. Rev. A*, vol. 75, p. 023402, Feb 2007. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.75.023402

[75] K. Chen, S. J. Schowalter, S. Kotochigova, A. Petrov, W. G. Rellergert, S. T. Sullivan, and E. R. Hudson, "Molecular-ion trap-depletion spectroscopy of BaCl^+," *Phys. Rev. A*, vol. 83, p. 030501, Mar 2011. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.83.030501

[76] A. K. Hansen, M. A. Sørensen, P. F. Staanum, and M. Drewsen, “Single-ion recycling reactions,” *Angewandte Chemie International Edition*, vol. 51, no. 32, pp. 7960–7962, 2012. [Online]. Available: http://dx.doi.org/10.1002/anie.201203550

[77] M. V. DePalatis and M. S. Chapman, “Production of translationally cold barium monohalide ions,” *Phys. Rev. A*, vol. 88, p. 023403, Aug 2013. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.88.023403

[78] V. Rajagopal, “Trapped Ion Thermometry and Mass Determination through Imaging,” Ph.D. dissertation, Northwestern University, June 2014.

[79] R. Rugango, J. E. Goeders, T. H. Dixon, J. M. Gray, N. Khanyile, G. Shu, R. J. Clark, and K. R. Brown, “Sympathetic cooling of molecular ion motion to the ground state,” *New Journal of Physics*, vol. 17, no. 3, p. 035009, 2015. [Online]. Available: http://dx.doi.org/10.1088/1367-2630/17/3/035009

[80] J. I. Cirac and P. Zoller, “Quantum Computations with Cold Trapped Ions,” *Phys. Rev. Lett.*, vol. 74, pp. 4091–4094, May 1995. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.74.4091

[81] F. Schmidt-Kaler, H. Häffner, M. Riebe, S. Gulde, G. P. Lancaster, T. Deuschle, C. Becher, C. F. Roos, J. Eschner, and R. Blatt, “Realization of the Cirac–Zoller controlled-NOT quantum gate,” *Nature*, vol. 422, no. 6930, pp. 408–411, 2003. [Online]. Available: http://dx.doi.org/10.1038/nature01494

[82] F. Wolf, Y. Wan, J. C. Heip, F. Gebert, C. Shi, and P. O. Schmidt, “Non-destructive state detection for quantum logic spectroscopy of molecular ions,” *Nature*, vol. 530, no. 7591, pp. 457–460, 2016. [Online]. Available: http://dx.doi.org/10.1038/nature16513

[83] P. K. Ghosh, *Ion Traps*. Clarendon Press, 1995.

[84] M. Drewsen and A. Brøner, “Harmonic linear Paul trap: Stability diagram and effective potentials,” *Phys. Rev. A*, vol. 62, p. 045401, Sep 2000. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.62.045401

[85] D. Leibfried, R. Blatt, C. Monroe, and D. Wineland, “Quantum dynamics of single trapped ions,” *Rev. Mod. Phys.*, vol. 75, pp. 281–324, Mar 2003. [Online]. Available: http://dx.doi.org/10.1103/RevModPhys.75.281

[86] C. Champenois, “Trapping and cooling of ions,” 2008 Les Houches predoc-school, p. 28, october 2008.

[87] D. Tabor, V. Rajagopal, Y.-W. Lin, and B. Odom, "Suitability of linear quadrupole ion traps for large Coulomb crystals," *Applied Physics B*, vol. 107, no. 4, pp. 1097–1104, 2012. [Online]. Available: http://dx.doi.org/10.1007/s00340-012-5058-x

[88] D. Tabor, "Toward Rotational Cooling of Trapped SiO$^+$ by Optical Pumping," Ph.D. dissertation, Northwestern University, June 2014.

[89] D. J. Wineland and W. M. Itano, "Laser cooling of atoms," *Phys. Rev. A*, vol. 20, pp. 1521–1540, Oct 1979. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.20.1521

[90] J. I. Cirac, A. S. Parkins, R. Blatt, and P. Zoller, ""Dark" squeezed states of the motion of a trapped ion," *Phys. Rev. Lett.*, vol. 70, pp. 556–559, Feb 1993. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.70.556

[91] J. I. Cirac, L. J. Garay, R. Blatt, A. S. Parkins, and P. Zoller, "Laser cooling of trapped ions: The influence of micromotion," *Phys. Rev. A*, vol. 49, pp. 421–432, Jan 1994. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.49.421

[92] A. Steane, "The ion trap quantum information processor," *Applied Physics B: Lasers and Optics*, vol. 64, no. 6, pp. 623–643, 1997. [Online]. Available: http://dx.doi.org/10.1007%2Fs003400050225

[93] D. J. Wineland, C. Monroe, W. M. Itano, D. Leibfried, B. E. King, and D. M. Meekhoff, "Experimental Issues in Coherent Quantum-State Manipulation of Trapped Atomic Ions," *J. Res. Natl. Inst. Stand. Technol.*, vol. 103, pp. 259–328, 1998. [Online]. Available: http://dx.doi.org/10.6028/jres.103.019

[94] D. Budker, D. F. Kimball, and D. P. DeMille, *Atomic Physics: An Exploration Through Problems and Solutions.* Oxford University Press, USA, 2004.

[95] C. J. Foot, *Atomic Physics.* Oxford University Press, 2005, vol. 7.

[96] S. Removille, R. Dubessy, B. Dubost, Q. Glorieux, T. Coudreau, S. Guibal, J.-P. Likforman, and L. Guidoni, "Trapping and cooling of Sr$^+$ ions: strings and large clouds," *Journal of Physics B: Atomic, Molecular and Optical Physics*, vol. 42, no. 15, p. 154014, August 2009. [Online]. Available: http://dx.doi.org/10.1088/0953-4075/42/15/154014

[97] C. E. Wieman and L. Hollberg, "Using diode lasers for atomic physics," *Review of Scientific Instruments*, vol. 62, no. 1, pp. 1–20, 1991. [Online]. Available: http://dx.doi.org/10.1063/1.1142305

[98] R. W. Fox, L. W. Hollberg, and A. S. Zibrov, “Semiconductor Diode Lasers,” in *Atomic, Molecular, and Optical Physics: Electromagnetic Radiation*, F. B. Dunning and R. G. Hulet, Eds. Elsevier, 1997, vol. 29C, ch. 4, pp. 77–102. [Online]. Available: http://dx.doi.org/10.1016/S0076-695X(08)60613-4

[99] H. Nasim and Y. Jamil, “Recent advancements in spectroscopy using tunable diode lasers,” *Laser Physics Letters*, vol. 10, no. 4, p. 043001, 2013. [Online]. Available: http://dx.doi.org/10.1088/1612-2011/10/4/043001

[100] E. S. Shuman, J. F. Barry, and D. Demille, “Laser cooling of a diatomic molecule,” *Nature*, vol. 467, no. 7317, pp. 820–823, September 2010. [Online]. Available: http://dx.doi.org/10.1038/nature09443

[101] M. T. Hummon, M. Yeo, B. K. Stuhl, A. L. Collopy, Y. Xia, and J. Ye, “2D Magneto-Optical Trapping of Diatomic Molecules,” *Phys. Rev. Lett.*, vol. 110, p. 143001, Apr 2013. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.110.143001

[102] V. Zhelyazkova, A. Cournol, T. E. Wall, A. Matsushima, J. J. Hudson, E. A. Hinds, M. R. Tarbutt, and B. E. Sauer, “Laser cooling and slowing of CaF molecules,” *Phys. Rev. A*, vol. 89, p. 053416, May 2014. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.89.053416

[103] M. Zeppenfeld, B. G. Englert, R. Glöckner, A. Prehn, M. Mielenz, C. Sommer, L. D. van Buuren, M. Motsch, and G. Rempe, “Sisyphus cooling of electrically trapped polyatomic molecules,” *Nature*, vol. 491, no. 7425, pp. 570–573, 2012. [Online]. Available: http://dx.doi.org/10.1038/nature11595

[104] A. Prehn, M. Ibrügger, R. Glöckner, G. Rempe, and M. Zeppenfeld, “Optoelectrical Cooling of Polar Molecules to Submillikelvin Temperatures,” *Phys. Rev. Lett.*, vol. 116, p. 063005, Feb 2016. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.116.063005

[105] M. Viteau, A. Chotia, M. Allegrini, N. Bouloufa, O. Dulieu, D. Comparat, and P. Pillet, “Optical pumping and vibrational cooling of molecules,” *Science*, vol. 321, no. 5886, pp. 232–234, 2008. [Online]. Available: http://dx.doi.org/10.1126/science.1159496

[106] D. Sofikitis, S. Weber, a. Fioretti, R. Horchani, M. Allegrini, B. Chatel, D. Comparat, and P. Pillet, “Molecular vibrational cooling by optical pumping with shaped femtosecond pulses,” *New Journal of Physics*, vol. 11, no. 5, p. 055037, May 2009. [Online]. Available: http://dx.doi.org/10.1088/1367-2630/11/5/055037

[107] A. Wakim, P. Zabawa, M. Haruza, and N. Bigelow, "Luminorefrigeration: vibrational cooling of NaCs," *Optics express*, vol. 20, no. 14, pp. 16 083–16 091, 2012. [Online]. Available: http://dx.doi.org/10.1364/OE.20.016083

[108] A. K. Hansen, O. Versolato, S. B. Kristensen, A. Gingell, M. Schwarz, A. Windberger, J. Ullrich, J. C. López-Urrutia, M. Drewsen *et al.*, "Efficient rotational cooling of Coulomb-crystallized molecular ions by a helium buffer gas," *Nature*, vol. 508, no. 7494, pp. 76–79, 2014. [Online]. Available: http://dx.doi.org/10.1038/nature12996

[109] I. Manai, R. Horchani, H. Lignier, P. Pillet, D. Comparat, A. Fioretti, and M. Allegrini, "Rovibrational Cooling of Molecules by Optical Pumping," *Phys. Rev. Lett.*, vol. 109, p. 183001, Oct 2012. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.109.183001

[110] P. F. Staanum, K. Højbjerre, P. S. Skyt, A. K. Hansen, and M. Drewsen, "Rotational laser cooling of vibrationally and translationally cold molecular ions," *Nature Physics*, vol. 6, no. 4, pp. 271–274, March 2010. [Online]. Available: http://dx.doi.org/10.1038/nphys1604

[111] T. Schneider, B. Roth, H. Duncker, I. Ernsting, and S. Schiller, "All-optical preparation of molecular ions in the rovibrational ground state," *Nature Physics*, vol. 6, no. 4, pp. 275–278, March 2010. [Online]. Available: http://dx.doi.org/10.1038/nphys1605

[112] J. H. V. Nguyen and B. Odom, "Prospects for Doppler cooling of three-electronic-level molecules," *Phys. Rev. A*, vol. 83, p. 053404, May 2011. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.83.053404

[113] J. H. Nguyen, C. R. Viteri, E. G. Hohenstein, C. D. Sherrill, K. R. Brown, and B. Odom, "Challenges of laser-cooling molecular ions," *New Journal of Physics*, vol. 13, no. 6, p. 063023, 2011. [Online]. Available: http://dx.doi.org/10.1088/1367-2630/13/6/063023

[114] C.-Y. Lien, S. R. Williams, and B. Odom, "Optical pulse-shaping for internal cooling of molecules," *Physical Chemistry Chemical Physics*, vol. 13, no. 42, pp. 18 825–18 829, 2011. [Online]. Available: http://dx.doi.org/10.1103/10.1039/C1CP21201J

[115] W. Shi and S. Malinovskaya, "Implementation of a single femtosecond optical frequency comb for rovibrational cooling," *Physical Review A*, vol. 82, no. 1, p. 013407, July 2010. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.82.013407

[116] G. M. Almy and M. C. Watson, “The Band Spectrum of Ionized Aluminum Hydride,” *Phys. Rev.*, vol. 45, pp. 871–876, Jun 1934. [Online]. Available: http://dx.doi.org/10.1103/PhysRev.45.871

[117] B. Müller and C. Ottinger, “Chemiluminescent reactions of second-row atomic ions. I. $Al^+ + H_2 \to AlH^+$ ($A^2\Pi$, $B^2\Sigma^+$)+ H,” *The Journal of chemical physics*, vol. 85, no. 1, pp. 232–242, 1986. [Online]. Available: http://dx.doi.org/10.1063/1.451642

[118] ——, “The Spectroscopic Constants of the $A^2\Pi$ and $X^2\Sigma$ States of AlH^+,” pp. 1007–1008, 1988.

[119] W. Szajna and M. Zachwieja, “High-resolution emission spectroscopy of the $A^2\Pi$–$X^2\Sigma^+$ system of AlH^+,” *Journal of Molecular Spectroscopy*, vol. 269, no. 1, pp. 56–60, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.jms.2011.05.002

[120] R. Klein, P. Rosmus, and H. Werner, “*Ab initio* calculations of low lying states of the BH^+ and AlH^+ ions,” *The Journal of Chemical Physics*, vol. 77, no. 7, pp. 3559–3570, 1982. [Online]. Available: http://dx.doi.org/10.1063/1.444257

[121] A. Bertelsen, S. Jørgensen, and M. Drewsen, “The rotational temperature of polar molecular ions in Coulomb crystals,” *Journal of Physics B: Atomic, Molecular and Optical Physics*, vol. 39, no. 5, p. L83, 2006. [Online]. Available: http://dx.doi.org/10.1088/0953-4075/39/5/L02

[122] B. Roth, J. C. J. Koelemeij, H. Daerr, and S. Schiller, “Rovibrational spectroscopy of trapped molecular hydrogen ions at millikelvin temperatures,” *Phys. Rev. A*, vol. 74, p. 040501, Oct 2006. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.74.040501

[123] K.-K. Ni, H. Loh, M. Grau, K. C. Cossel, J. Ye, and E. A. Cornell, “State-specific detection of trapped HfF^+ by photodissociation,” *Journal of Molecular Spectroscopy*, vol. 300, pp. 12–15, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.jms.2014.02.001

[124] A. M. Steane, “Overhead and noise threshold of fault-tolerant quantum error correction,” *Physical Review A*, vol. 68, no. 4, p. 042322, 2003. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.68.042322

[125] P. Aliferis and A. W. Cross, “Subsystem Fault Tolerance with the Bacon-Shor Code,” *Phys. Rev. Lett.*, vol. 98, p. 220502, May 2007. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.98.220502

[126] A. M. Steane, "How to build a 300 bit, 1 Giga-operation quantum computer," *Quantum Information and Computation*, vol. 7, no. 3, pp. 171–183, 2007. [Online]. Available: http://arxiv.org/abs/quant-ph/0412165

[127] R. Blatt and C. Roos, "Quantum simulations with trapped ions," *Nature Physics*, vol. 8, no. 4, pp. 277–284, 2012. [Online]. Available: http://dx.doi.org/10.1038/nphys2252

[128] Q. A. Turchette, C. S. Wood, B. E. King, C. J. Myatt, D. Leibfried, W. M. Itano, C. Monroe, and D. J. Wineland, "Deterministic Entanglement of Two Trapped Ions," *Phys. Rev. Lett.*, vol. 81, pp. 3631–3634, Oct 1998. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.81.3631

[129] K. Brown, C. Ospelkaus, Y. Colombe, A. Wilson, D. Leibfried, and D. Wineland, "Coupled quantized mechanical oscillators," *Nature*, vol. 471, no. 7337, pp. 196–199, 2011. [Online]. Available: http://dx.doi.org/10.1038/nature09721

[130] Y.-W. Lin, S. Williams, and B. C. Odom, "Resonant few-photon excitation of a single-ion oscillator," *Phys. Rev. A*, vol. 87, p. 011402, Jan 2013. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.87.011402

[131] Y. Wan, F. Gebert, J. B. Wübbena, N. Scharnhorst, S. Amairi, I. D. Leroux, B. Hemmerling, N. Lörch, K. Hammerer, and P. O. Schmidt, "Precision spectroscopy by photon-recoil signal amplification," *Nature communications*, vol. 5, 2014. [Online]. Available: http://dx.doi.org/10.1038/ncomms4096

[132] F. Gebert, "Precision measurement of the isotopic shift in calcium ions using photon recoil spectroscopy," Ph.D. dissertation, Gottfried Wilhelm Leibniz Universität Hannover, January 2015.

[133] C. Auchter, C.-K. Chou, T. W. Noel, and B. B. Blinov, "Ion–photon entanglement and Bell inequality violation with $^{138}\mathrm{Ba}^{+}$," *J. Opt. Soc. Am. B*, vol. 31, no. 7, pp. 1568–1572, Jul 2014. [Online]. Available: http://dx.doi.org/10.1364/JOSAB.31.001568

[134] J. Gurell, E. Biémont, K. Blagoev, V. Fivet, P. Lundin, S. Mannervik, L.-O. Norlin, P. Quinet, D. Rostohar, P. Royen, and P. Schef, "Laser-probing measurements and calculations of lifetimes of the $5d\,^2D_{32}$ and $5d\,^2D_{52}$ metastable levels in Ba II," *Phys. Rev. A*, vol. 75, p. 052506, May 2007. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.75.052506

[135] N. Yu, X. Zhao, H. Dehmelt, and W. Nagourney, "Stark shift of a single barium ion and potential application to zero-point confinement in a rf

trap," *Phys. Rev. A*, vol. 50, pp. 2738–2741, Sep 1994. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.50.2738

[136] B. Appasamy, I. Siemers, Y. Stalgies, J. Eschner, R. Blatt, W. Neuhauser, and P. Toschek, "Quantized infrared-optical triple resonance on a single cold Barium ion," *Applied Physics B*, vol. 60, no. 5, pp. 473–477, 1995. [Online]. Available: http://dx.doi.org/10.1007/BF01081329

[137] M. R. Dietrich, N. Kurz, T. Noel, G. Shu, and B. B. Blinov, "Hyperfine and optical barium ion qubits," *Phys. Rev. A*, vol. 81, p. 052328, May 2010. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.81.052328

[138] L. Slodička, G. Hétet, N. Röck, S. Gerber, P. Schindler, M. Kumph, M. Hennrich, and R. Blatt, "Interferometric thermometry of a single sub-Doppler-cooled atom," *Phys. Rev. A*, vol. 85, p. 043401, Apr 2012. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.85.043401

[139] M. R. Hoffman, T. W. Noel, C. Auchter, A. Jayakumar, S. R. Williams, B. B. Blinov, and E. N. Fortson, "Radio-frequency-spectroscopy measurement of the Landé g_J factor of the $5D_{5/2}$ state of Ba^+ with a single trapped ion," *Phys. Rev. A*, vol. 88, p. 025401, Aug 2013. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.88.025401

[140] P. A. Barton, C. J. S. Donald, D. M. Lucas, D. A. Stevens, A. M. Steane, and D. N. Stacey, "Measurement of the lifetime of the $3d^2D_{5/2}$ state in $^{40}\mathrm{Ca}^+$," *Phys. Rev. A*, vol. 62, p. 032503, Aug 2000. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.62.032503

[141] I. Marzoli, J. I. Cirac, R. Blatt, and P. Zoller, "Laser cooling of trapped three-level ions: Designing two-level systems for sideband cooling," *Phys. Rev. A*, vol. 49, pp. 2771–2779, Apr 1994. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.49.2771

[142] B. L. Chuah, N. C. Lewty, R. Cazan, and M. D. Barrett, "Sub-Doppler cavity cooling beyond the Lamb-Dicke regime," *Phys. Rev. A*, vol. 87, p. 043420, Apr 2013. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.87.043420

[143] C. Monroe, D. M. Meekhof, B. E. King, S. R. Jefferts, W. M. Itano, D. J. Wineland, and P. Gould, "Resolved-Sideband Raman Cooling of a Bound Atom to the 3D Zero-Point Energy," *Phys. Rev. Lett.*, vol. 75, pp. 4011–4014, Nov 1995. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.75.4011

[144] L. Deslauriers, P. C. Haljan, P. J. Lee, K.-A. Brickman, B. B. Blinov, M. J. Madsen, and C. Monroe, "Zero-point cooling and low heating of trapped ^{111}Cd$^+$ ions," *Phys. Rev. A*, vol. 70, p. 043408, Oct 2004. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.70.043408

[145] R. J. Epstein, S. Seidelin, D. Leibfried, J. H. Wesenberg, J. J. Bollinger, J. M. Amini, R. B. Blakestad, J. Britton, J. P. Home, W. M. Itano, J. D. Jost, E. Knill, C. Langer, R. Ozeri, N. Shiga, and D. J. Wineland, "Simplified motional heating rate measurements of trapped ions," *Phys. Rev. A*, vol. 76, p. 033411, Sep 2007. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.76.033411

[146] R. DeVoe and C. Kurtsiefer, "Observation of long decoherence times in a quantum logic gate using trapped Ba137 ions," in *Quantum Electronics and Laser Science Conference, 2001. QELS'01. Technical Digest. Summaries of Papers Presented at the.* IEEE, 2001, pp. 240–241. [Online]. Available: http://dx.doi.org/10.1109/QELS.2001.962184

[147] C. F. Roos, D. Leibfried, A. Mundt, F. Schmidt-Kaler, J. Eschner, and R. Blatt, "Experimental Demonstration of Ground State Laser Cooling with Electromagnetically Induced Transparency," *Phys. Rev. Lett.*, vol. 85, pp. 5547–5550, Dec 2000. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.85.5547

[148] P. J. Blythe, "Optical Frequency Measurement and Ground State Cooling of Single Trapped Yb$^+$ ions," Ph.D. dissertation, University of London, March 2004.

[149] P. Staanum, "Quantum Optics with Trapped Calcium Ions," Ph.D. dissertation, University of Aarhus, February 2004.

[150] S. Webster, "Raman Sideband Cooling and Coherent Manipulation of Trapped Ions," Ph.D. dissertation, St. John's College, Oxford, 2005.

[151] J. P. Home, M. J. McDonnell, D. J. Szwer, B. C. Keitch, D. M. Lucas, D. N. Stacey, and A. M. Steane, "Memory coherence of a sympathetically cooled trapped-ion qubit," *Phys. Rev. A*, vol. 79, p. 050305, May 2009. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.79.050305

[152] P. Schindler, D. Nigg, T. Monz, J. T. Barreiro, E. Martinez, S. X. Wang, S. Quint, M. F. Brandl, V. Nebendahl, C. F. Roos *et al.*, "A quantum information processor with trapped ions," *New Journal of Physics*, vol. 15, no. 12, p. 123012, 2013. [Online]. Available: http://dx.doi.org/10.1088/1367-2630/15/12/123012

[153] A. E. Siegman, *Lasers.* University Science Books, 1986.

[154] F. L. Pedrotti, L. S. Pedrotti, and L. M. Pedrotti, *Introduction to Optics*, 3rd ed. Pearson Prentice Hall, 2007.

[155] W. T. Silfvast, *Laser Fundamentals*, 2nd ed. Cambridge, 2004.

[156] A. L. Schawlow and C. H. Townes, "Infrared and Optical Masers," *Phys. Rev.*, vol. 112, pp. 1940–1949, Dec 1958. [Online]. Available: http://dx.doi.org/10.1103/PhysRev.112.1940

[157] C. Henry, "Theory of the linewidth of semiconductor lasers," *IEEE Journal of Quantum Electronics*, vol. 18, no. 2, pp. 259–264, 1982. [Online]. Available: http://dx.doi.org/10.1109/JQE.1982.1071522

[158] Y. Cunyun, *Tunable External Cavity Diode Lasers.* World Scientific, 2004.

[159] R. Paschotta, H. R. Telle, and U. Keller, "Noise of Solid State Lasers," in *Solid-State Lasers and Applications*, A. Sennaroglu, Ed. CRC Press, 2007, ch. 12, pp. 473–510.

[160] E. C. Cook, P. J. Martin, T. L. Brown-Heft, J. C. Garman, and D. A. Steck, "High passive-stability diode-laser design for use in atomic-physics experiments," *Review of Scientific Instruments*, vol. 83, no. 4, p. 043101, 2012. [Online]. Available: http://dx.doi.org/10.1063/1.3698003

[161] R. Paschotta, "Derivation of the Schawlow-Townes Linewidth of Lasers," RP Photonics Consulting, June 2013.

[162] R. J. Lang and A. Yariv, "Local-field rate equations for coupled optical resonators," *Phys. Rev. A*, vol. 34, pp. 2038–2043, Sep 1986. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.34.2038

[163] L. Ricci, M. Weidemüller, T. Esslinger, A. Hemmerich, C. Zimmermann, V. Vuletic, W. König, and T. W. Hänsch, "A compact grating-stabilized diode laser system for atomic physics," *Optics Communications*, vol. 117, no. 5, pp. 541–549, 1995. [Online]. Available: http://dx.doi.org/10.1016/0030-4018(95)00146-Y

[164] J. Hult, I. S. Burns, and C. F. Kaminski, "Wide-bandwidth mode-hop-free tuning of extended-cavity GaN diode lasers," *Applied optics*, vol. 44, no. 18, pp. 3675–3685, 2005. [Online]. Available: http://dx.doi.org/10.1364/AO.44.003675

[165] T. Führer and T. Walther, "Extension of the mode-hop-free tuning range of an external cavity diode laser based on a model of the mode-hop

dynamics," *Optics letters*, vol. 33, no. 4, pp. 372–374, 2008. [Online]. Available: http://dx.doi.org/10.1364/OL.33.000372

[166] H. Loh, Y.-J. Lin, I. Teper, M. Cetina, J. Simon, J. K. Thompson, and V. Vuletić, "Influence of grating parameters on the linewidths of external-cavity diode lasers," *Applied optics*, vol. 45, no. 36, pp. 9191–9197, 2006. [Online]. Available: http://dx.doi.org/10.1364/AO.45.009191

[167] M. Fleming and A. Mooradian, "Spectral characteristics of external-cavity controlled semiconductor lasers," *IEEE Journal of Quantum Electronics*, vol. 17, no. 1, pp. 44–59, 1981. [Online]. Available: http://dx.doi.org/10.1109/JQE.1981.1070634

[168] J.-P. Taché, "Ray matrices for tilted interfaces in laser resonators," *Applied optics*, vol. 26, no. 3, pp. 427–429, 1987. [Online]. Available: http://dx.doi.org/10.1364/AO.26.000427

[169] K. MacAdam, A. Steinbach, and C. Wieman, "A narrow-band tunable diode laser system with grating feedback, and a saturated absorption spectrometer for Cs and Rb," *American Journal of Physics*, vol. 60, pp. 1098–1098, 1992. [Online]. Available: http://dx.doi.org/10.1119/1.16955

[170] A. Andalkar, S. Lamoreaux, and R. Warrington, "Improved external cavity design for cesium D1 (894 nm) diode laser," *Review of Scientific Instruments*, vol. 71, no. 11, pp. 4029–4031, 2000. [Online]. Available: http://dx.doi.org/10.1063/1.1319860

[171] K. Liu and M. G. Littman, "Novel geometry for single-mode scanning of tunable lasers," *Optics Letters*, vol. 6, no. 3, pp. 117–118, 1981. [Online]. Available: http://dx.doi.org/10.1364/OL.6.000117

[172] K. Harvey and C. Myatt, "External-cavity diode laser using a grazing-incidence diffraction grating," *Optics Letters*, vol. 16, no. 12, pp. 910–912, 1991. [Online]. Available: http://dx.doi.org/10.1364/OL.16.000910

[173] M. Gilowski, C. Schubert, M. Zaiser, W. Herr, T. Wübbena, T. Wendrich, T. Müller, E. Rasel, and W. Ertmer, "Narrow bandwidth interference filter-stabilized diode laser systems for the manipulation of neutral atoms," *Optics Communications*, vol. 280, no. 2, pp. 443–447, 2007. [Online]. Available: http://dx.doi.org/10.1016/j.optcom.2007.08.043

[174] K. Hayasaka, "Frequency stabilization of an extended-cavity violet diode laser by resonant optical feedback," *Optics communications*, vol. 206, no. 4, pp. 401–409, 2002. [Online]. Available: http://dx.doi.org/10.1016/S0030-4018(02)01446-3

[175] A. Wicht, M. Rudolf, P. Huke, R.-H. Rinkleff, and K. Danzmann, "Grating enhanced external cavity diode laser," *Applied Physics B*, vol. 78, no. 2, pp. 137–144, 2004. [Online]. Available: http://dx.doi.org/10.1007/s00340-003-1304-6

[176] A. Wicht, P. Huke, R.-H. Rinkleff, and K. Danzmann, "Advancing the optical feed back concept: Grating enhanced external cavity diode laser," *Physica Scripta*, vol. 2005, no. T118, p. 82, 2005. [Online]. Available: http://dx.doi.org/10.1238/Physica.Topical.118a00082

[177] J. Sacher, D. Baums, P. Panknin, W. Elsässer, and E. O. Göbel, "Intensity instabilities of semiconductor lasers under current modulation, external light injection, and delayed feedback," *Phys. Rev. A*, vol. 45, pp. 1893–1905, Feb 1992. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.45.1893

[178] V. Gerginov, Y. Dancheva, M. Taslakov, and S. Cartaleva, "Frequency tunable monomode diode laser at 670 nm for high resolution spectroscopy," *Optics communications*, vol. 149, no. 1, pp. 162–169, 1998. [Online]. Available: http://dx.doi.org/10.1016/S0030-4018(97)00686-X

[179] C. Petridis, I. Lindsay, D. Stothard, and M. Ebrahimzadeh, "Mode-hop-free tuning over 80 GHz of an extended cavity diode laser without antireflection coating," *Review of Scientific Instruments*, vol. 72, no. 10, pp. 3811–3815, 2001. [Online]. Available: http://dx.doi.org/10.1063/1.1405783

[180] L. Hildebrandt, R. Knispel, S. Stry, J. R. Sacher, and F. Schael, "Antireflection-coated blue GaN laser diodes in an external cavity and Doppler-free indium absorption spectroscopy," *Applied optics*, vol. 42, no. 12, pp. 2110–2118, 2003. [Online]. Available: http://dx.doi.org/10.1364/AO.42.002110

[181] M. De Labachelerie and G. Passedat, "Mode-hop suppression of Littrow grating-tuned lasers," *Applied optics*, vol. 32, no. 3, pp. 269–274, 1993. [Online]. Available: http://dx.doi.org/10.1364/AO.32.000269

[182] M. De Labachelerie, H. Sasada, and G. Passedat, "Mode-hop suppression of Littrow grating-tuned lasers: erratum," *Applied optics*, vol. 33, no. 18, pp. 3817–3819, 1994. [Online]. Available: http://dx.doi.org/10.1364/AO.33.003817

[183] T. Hof, D. Fick, and H. Jänsch, "Application of diode lasers as a spectroscopic tool at 670 nm," *Optics communications*, vol. 124, no. 3, pp. 283–286, 1996. [Online]. Available: http://dx.doi.org/10.1016/0030-4018(95)00698-2

[184] L. Nilse, H. J. Davies, and C. S. Adams, "Synchronous tuning of extended cavity diode lasers: the case for an optimum pivot point," *Applied optics*, vol. 38, no. 3, pp. 548–553, 1999. [Online]. Available: http://dx.doi.org/10.1364/AO.38.000548

[185] S. Dutta, D. Elliott, and Y. P. Chen, "Mode-hop-free tuning over 135 GHz of external cavity diode lasers without antireflection coating," *Applied Physics B*, vol. 106, no. 3, pp. 629–633, 2012. [Online]. Available: http://dx.doi.org/10.1007/s00340-011-4841-4

[186] K. Christandl, "A Compact, Grating-stabilized Diode Laser for Atomic Spectroscopy," Master's thesis, The Ohio State University, 2000.

[187] ——, "Advancing Neutral Atom Quantum Computing: Studies of One-dimensional and Two-dimensional Optical Lattices on a Chip," Ph.D. dissertation, The Ohio State University, 2005.

[188] S. D. Saliba and R. E. Scholten, "Linewidths below 100 kHz with external cavity diode lasers," *Applied optics*, vol. 48, no. 36, pp. 6961–6966, 2009. [Online]. Available: http://dx.doi.org/10.1364/AO.48.006961

[189] R. Lang, "Injection locking properties of a semiconductor laser," *IEEE Journal of Quantum Electronics*, vol. 18, no. 6, pp. 976–983, 1982. [Online]. Available: http://dx.doi.org/10.1109/JQE.1982.1071632

[190] P. Bouyer, T. Gustavson, K. Haritos, and M. Kasevich, "Microwave signal generation with optical injection locking," *Optics letters*, vol. 21, no. 18, pp. 1502–1504, 1996. [Online]. Available: http://dx.doi.org/10.1364/OL.21.001502

[191] C. Liekhus-Schmaltz, R. Mantifel, M. Torabifard, I. Burgess, and J. Martin, "Injection-locked diode laser current modulation for Pound-Drever-Hall frequency stabilization using transfer cavities," *JOSA B*, vol. 29, no. 6, pp. 1394–1398, 2012. [Online]. Available: http://dx.doi.org/10.1364/JOSAB.29.001394

[192] K. Libbrecht and J. L. Hall, "A low-noise high-speed diode laser current controller," *Review of Scientific Instruments*, vol. 64, no. 8, pp. 2133–2135, 1993. [Online]. Available: http://dx.doi.org/10.1063/1.1143949

[193] C. J. Erickson, M. Van Zijll, G. Doermann, and D. S. Durfee, "An ultrahigh stability, low-noise laser current driver with digital control," *Review*

of Scientific Instruments, vol. 79, no. 7, p. 073107, 2008. [Online]. Available: http://dx.doi.org/10.1063/1.2953597

[194] D. L. Troxel, C. J. Erickson, and D. S. Durfee, "Note: Updates to an ultra-low noise laser current driver," *Review of Scientific Instruments*, vol. 82, no. 9, p. 096101, 2011. [Online]. Available: http://dx.doi.org/10.1063/1.3630950

[195] M. S. Taubman, "Low-noise high-performance current controllers for quantum cascade lasers," *Review of Scientific Instruments*, vol. 82, no. 6, p. 064704, 2011. [Online]. Available: http://dx.doi.org/10.1063/1.3600602

[196] D. J. Armstrong and J. Cooper, "Isotope-selective photoionization spectroscopy of barium," *Phys. Rev. A*, vol. 47, pp. R2446–R2449, Apr 1993. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.47.R2446

[197] D. Rotter, "Quantum feedback and quantum correlation measurements with a single Barium ion," Ph.D. dissertation, University of Innsbruck, February 2008.

[198] G. Leschhorn, T. Hasegawa, and T. Schaetz, "Efficient photo-ionization for barium ion trapping using a dipole-allowed resonant two-photon transition," *Applied Physics B*, vol. 108, no. 1, pp. 159–165, 2012. [Online]. Available: http://dx.doi.org/10.1007/s00340-012-5101-y

[199] A. Kramida, Yu. Ralchenko, J. Reader, and and NIST ASD Team, NIST Atomic Spectra Database (ver. 5.3). National Institute of Standards and Technology, Gaithersburg, MD., 2015. [Online]. Available: http://physics.nist.gov/asd

[200] R. Bacis, "A new source for the search and study of electronic molecular spectra: the composite wall hollow cathode (CWHC)," *Journal of Physics E: Scientific Instruments*, vol. 9, no. 12, p. 1081, 1976. [Online]. Available: http://dx.doi.org/10.1088/0022-3735/9/12/021

[201] B. Barbieri, N. Beverini, and A. Sasso, "Optogalvanic spectroscopy," *Rev. Mod. Phys.*, vol. 62, pp. 603–644, Jul 1990. [Online]. Available: http://dx.doi.org/10.1103/RevModPhys.62.603

[202] L. Gianfrani and G. Tino, "High resolution spectroscopy of iridium in a hollow cathode discharge," *Zeitschrift für Physik D Atoms, Molecules and Clusters*, vol. 25, no. 2, pp. 113–116, 1993. [Online]. Available: http://dx.doi.org/10.1007/BF01450162

[203] G. Rao, M. Reddy, and E. Hecht, "Atomic hyperfine structure studies using temperature/current tuning of diode lasers: An undergraduate experiment,"

American Journal of Physics, vol. 66, no. 8, pp. 702–712, 1998. [Online]. Available: http://dx.doi.org/10.1119/1.18936

[204] U. Dammalapati, I. Norris, and E. Riis, “Saturated absorption spectroscopy of calcium in a hollow-cathode lamp,” *Journal of Physics B: Atomic, Molecular and Optical Physics*, vol. 42, no. 16, p. 165001, 2009. [Online]. Available: http://dx.doi.org/10.1088/0953-4075/42/16/165001

[205] K. Brodie and S. Neate, “Features and Operation of Hollow Cathode Lamps and Deuterium Lamps,” Agilent Application Note, November 2010.

[206] Y. Shimada, Y. Chida, N. Ohtsubo, T. Aoki, M. Takeuchi, T. Kuga, and Y. Torii, “A simplified 461-nm laser system using blue laser diodes and a hollow cathode lamp for laser cooling of Sr,” *Review of Scientific Instruments*, vol. 84, no. 6, p. 063101, 2013. [Online]. Available: http://dx.doi.org/10.1063/1.4808246

[207] B. Harvey and C. Siu, “Surge Damage Prevention Manual,” Opnext Technical Note, September 2008.

[208] J. Sacher, November 2011, private communication.

[209] T. Dutta, D. De Munshi, and M. Mukherjee, “Absolute Te_2 reference for barium ion at 455.4 nm,” *JOSA B*, vol. 33, no. 6, pp. 1177–1181, 2016. [Online]. Available: http://dx.doi.org/10.1364/JOSAB.33.001177

[210] J. C. Kangara, A. J. Hachtel, M. C. Gillette, J. T. Barkeloo, E. R. Clements, S. Bali, B. E. Unks, N. A. Proite, D. D. Yavuz, P. J. Martin *et al.*, “Design and construction of cost-effective tapered amplifier systems for laser cooling and trapping experiments,” *American Journal of Physics*, vol. 82, no. 8, pp. 805–817, 2014. [Online]. Available: http://dx.doi.org/10.1119/1.4867376

[211] A. Truscott, N. Heckenberg, and H. Rubinsztein-Dunlop, “Frequency stabilised grating feedback laser diode for atom cooling applications,” *Optical and quantum electronics*, vol. 31, no. 5-7, pp. 417–430, 1999. [Online]. Available: http://dx.doi.org/10.1023/A:1006962911229

[212] W.-K. Lee, C. Y. Park, D.-H. Yu, S. E. Park, S.-B. Lee, and T. Y. Kwon, “Generation of 578-nm yellow light over 10 mW by second harmonic generation of an 1156-nm external-cavity diode laser,” *Optics express*, vol. 19, no. 18, pp. 17 453–17 461, 2011. [Online]. Available: http://dx.doi.org/10.1364/OE.19.017453

[213] J. Sacher, May 2015, private communication.

[214] E. Power, November 2015, private communication.

[215] A. Jechow, *Tailoring the Emission of Stripe Array Diode Lasers with External Cavities to Enable Nonlinear Frequency Conversion.* Universitätsverlag Potsdam, 2009.

[216] P. Shen, "Stabilizing Diode Lasers to High-finesse Cavities," Master's thesis, Northwestern University, August 2016.

[217] S. Gerstenkorn and P. Luc, "Atlas du spectre d'absorption de la molecule d'iode 14800-20000 cm^{-1}," *Paris: Editions du Centre National de la Recherche Scientifique (CNRS), 1978*, vol. 1, 1978.

[218] C. J. Sansonetti, "Precise measurements of hyperfine components in the spectrum of molecular iodine," *JOSA B*, vol. 14, no. 8, pp. 1913–1920, 1997. [Online]. Available: http://dx.doi.org/10.1364/JOSAB.14.001913

[219] H. Salami and A. J. Ross, "A molecular iodine atlas in ascii format," *Journal of Molecular Spectroscopy*, vol. 233, no. 1, pp. 157–159, 2005. [Online]. Available: http://dx.doi.org/10.1016/j.jms.2005.06.002

[220] D. A. Steck, "Rubidium 85 D Line Data," September 2012.

[221] A. V. Steele, L. R. Churchill, P. F. Griffin, and M. S. Chapman, "Photoionization and photoelectric loading of barium ion traps," *Phys. Rev. A*, vol. 75, p. 053404, May 2007. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.75.053404

[222] B. Wang, J. W. Zhang, C. Gao, and L. J. Wang, "Highly efficient and isotope selective photo-ionization of barium atoms using diode laser and LED light," *Opt. Express*, vol. 19, no. 17, pp. 16 438–16 447, Aug 2011. [Online]. Available: http://dx.doi.org/10.1364/OE.19.016438

[223] P. Herskind, "Cavity Quantum Electrodynamics with Ion Coulomb Crystals," Ph.D. dissertation, The University of Aarhus, September 2008.

[224] J. Nguyen, "The Linewidth and Hyperfine A Constant of the $^2\mathrm{P}_{1/2}$ State of a Magnesium Ion Confined in a Linear Paul Trap," Ph.D. dissertation, McMaster University, 2009.

[225] R. Cazan, C. Geppert, W. Nörtershäuser, and R. Sánchez, "Towards sympathetic cooling of trapped ions with laser-cooled Mg^+ ions for mass spectrometry and laser spectroscopy," *Hyperfine Interactions*, vol. 196, no. 1-3, pp. 177–189, 2010. [Online]. Available: http://dx.doi.org/10.1007/s10751-009-0144-0

[226] A. Friedenauer, F. Markert, H. Schmitz, L. Petersen, S. Kahra, M. Herrmann, T. Udem, T. Hänsch, and T. Schätz, "High power all solid state laser system near 280 nm," *Applied Physics B*, vol. 84, no. 3, pp. 371–373, 2006. [Online]. Available: http://dx.doi.org/10.1007/s00340-006-2274-2

[227] J.-I. Kim and D. Meschede, "Continuous-wave coherent ultraviolet source at 326 nm based on frequency trippling of fiber amplifiers," *Optics express*, vol. 16, no. 14, pp. 10 803–10 808, 2008. [Online]. Available: http://dx.doi.org/10.1364/OE.16.010803

[228] S. Vasilyev, A. Nevsky, I. Ernsting, M. Hansen, J. Shen, and S. Schiller, "Compact all-solid-state continuous-wave single-frequency UV source with frequency stabilization for laser cooling of Be^+ ions," *Applied Physics B*, vol. 103, no. 1, pp. 27–33, 2011. [Online]. Available: http://dx.doi.org/10.1007/s00340-011-4435-1

[229] K. Devi, S. C. Kumar, and M. Ebrahim-Zadeh, "High-power, continuous-wave, single-frequency, all-periodically-poled, near-infrared source," *Optics letters*, vol. 37, no. 24, pp. 5049–5051, 2012. [Online]. Available: http://dx.doi.org/10.1364/OL.37.005049

[230] G. Boyd and D. Kleinman, "Parametric interaction of focused Gaussian light beams," *Journal of Applied Physics*, vol. 39, no. 8, pp. 3597–3639, 1968. [Online]. Available: http://dx.doi.org/10.1063/1.1656831

[231] M. Brieger, H. Büsener, A. Hese, F. v. Moers, and A. Renn, "Enhancement of single frequency SGH in a passive ring resonator," *Optics Communications*, vol. 38, no. 5, pp. 423–426, 1981. [Online]. Available: http://dx.doi.org/10.1016/0030-4018(81)90081-X

[232] A.-P. A. V. Smith, "SNLO Nonlinear Optics Code," Albuquerque, NM, 2014. [Online]. Available: http://www.as-photonics.com/snlo

[233] L. Busse, L. Goldberg, M. Surette, and G. Mizell, "Absorption losses in MgO-doped and undoped potassium niobate," *Journal of applied physics*, vol. 75, no. 2, pp. 1102–1110, 1994. [Online]. Available: http://dx.doi.org/10.1063/1.356493

[234] T. Mogi, Y. Fukuyama, T. Kobayashi, I. Tanihata, K. Uehara, and Y. Matsuo, "Translational motion of SiO^+ produced by laser ablation," *Applied Surface Science*, vol. 197-198, pp. 202–206, September 2002. [Online]. Available: http://dx.doi.org/10.1016/S0169-4332(02)00360-4

[235] Y.-W. Lin, Ph.D. dissertation, Northwestern University, 2016.

[236] C. Raab, J. Bolle, H. Oberst, J. Eschner, F. Schmidt-Kaler, and R. Blatt, "Diode laser spectrometer at 493 nm for single trapped Ba+ ions," *Applied Physics B*, vol. 67, no. 6, pp. 683–688, 1998. [Online]. Available: http://dx.doi.org/10.1007/s003400050566

[237] P. Bohlouli-Zanjani, K. Afrousheh, and J. Martin, "Optical transfer cavity stabilization using current-modulated injection-locked diode lasers," *Review of scientific instruments*, vol. 77, no. 9, p. 093105, 2006. [Online]. Available: http://dx.doi.org/10.1063/1.2337094

[238] N. Seymour-Smith, P. Blythe, M. Keller, and W. Lange, "Fast scanning cavity offset lock for laser frequency drift stabilization," *Review of Scientific Instruments*, vol. 81, no. 7, p. 075109, 2010. [Online]. Available: http://dx.doi.org/10.1063/1.3455830

[239] S. C. Doret, "Transfer Cavity/Stabilized HeNe based Laser Frequency Stabilization," January 2011.

[240] J. Biesheuvel, D. Noom, E. Salumbides, K. Sheridan, W. Ubachs, and J. Koelemeij, "Widely tunable laser frequency offset lock with 30 GHz range and 5 THz offset," *Optics express*, vol. 21, no. 12, pp. 14 008–14 016, 2013. [Online]. Available: http://dx.doi.org/10.1364/OE.21.014008

[241] E. D. Black, "An introduction to Pound–Drever–Hall laser frequency stabilization," *American Journal of Physics*, vol. 69, no. 1, pp. 79–87, 2001. [Online]. Available: http://dx.doi.org/10.1119/1.1286663

[242] M. Nickerson, "A review of Pound-Drever-Hall laser frequency locking," December 2008.

[243] F. Rohde, M. Almendros, C. Schuck, J. Huwer, M. Hennrich, and J. Eschner, "A diode laser stabilization scheme for ^{40}Ca^{+} single-ion spectroscopy," *Journal of Physics B: Atomic, Molecular and Optical Physics*, vol. 43, no. 11, p. 115401, 2010. [Online]. Available: http://dx.doi.org/10.1088/0953-4075/43/11/115401

[244] C. Wieman and T. W. Hänsch, "Doppler-Free Laser Polarization Spectroscopy," *Phys. Rev. Lett.*, vol. 36, pp. 1170–1173, May 1976. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.36.1170

[245] C. Pearman, C. Adams, S. Cox, P. Griffin, D. Smith, and I. Hughes, "Polarization spectroscopy of a closed atomic transition: applications to laser frequency locking," *Journal of Physics B: Atomic, Molecular and Optical Physics*, vol. 35, no. 24, p. 5141, 2002. [Online]. Available: http://dx.doi.org/10.1088/0953-4075/35/24/315

[246] M. Pachchigar, "Design Considerations for a Transimpedance Amplifiers," Analog Application Note AN-1803, 2008.

[247] X. Ramus, "Transimpedance Considerations for High-Speed Amplifiers," Texax Instruments Application Report SBOA122, November 2009.

[248] D. A. Steck, "Rubidium 87 D Line Data," December 2010.

[249] K. J. Aström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton university press, 2010.

[250] O. Bishop, *Electronics-Circuits and Systems*. Routledge, 2011.

[251] W. S. Levine, Ed., *Control System Applications*. CRC Press, 1999.

[252] H. Koivo and M. Elmusrati, *Systems Engineering in Wireless Communications*. John Wiley & Sons, November 2009. [Online]. Available: http://dx.doi.org/10.1002/9780470021804

[253] J. L. Hall, M. S. Taubman, and J. Ye, "Laser Stabilization," in *Optics Handbook*. Optical Society of America, 2000, ch. 27, pp. 27.1–27.24.

[254] R. W. Fox, C. W. Oates, and L. W. Hollberg, "Stabilizing diode lasers to high-finesse cavities," in *Cavity-Enhanced Spectroscopies*, R. D. van Zee and J. P. Looney, Eds. Elsevier, 2003, vol. 40, ch. 1, pp. 1–46. [Online]. Available: http://dx.doi.org/10.1016/S1079-4042(03)80017-6

[255] P. Horowitz and W. Hill, *The Art of Electronics*, 3rd ed. Cambridge, 2016.

[256] D. A. Steck, "Noise Analysis: Libbrecht-Hall Circuit," April 2014. [Online]. Available: http://atomoptics-nas.uoregon.edu/odom_current/libbrecht-hall-noiseanalysis.pdf

[257] E. H. Dietz, "Understanding and Reducing Noise Voltage on 3-Terminal Voltage Regulators," in *Troubleshooting Analog Circuits*, R. A. Pease, Ed. Butterworth-Heinemann, 1991, ch. Appendix C, p. 191.

[258] B. Harvey and C. Siu, "Driving Reactive Loads with High-Frequency Op-Amps," Intersil Application Note AN1092, 1998.

[259] K. Blake, "Driving Capacitive Loads with Op-Amps," Microchip Technology Inc. Application Note AN844, 2008.

[260] J. A. Sauer, "Cold Atom Manipulation for Quantum Computing and Control," Ph.D. dissertation, Georgia Institute of Technology, August 2004.

[261] J. Ardizzoni, "Protecting Off-Amps," Analog Dialogue, vol. 42, 2008. [Online]. Available: http://www.analog.com/library/analogDialogue/archives/42-10/off_amps.html

[262] D. A. Steck, "MIT Rv03 Circuit Analysis," August 2014. [Online]. Available: http://atomoptics-nas.uoregon.edu/odom_current/

[263] C.-Y. Lien, C. M. Seck, Y.-W. Lin, J. H. Nguyen, D. A. Tabor, and B. C. Odom, "Broadband optical cooling of molecular rotors from room temperature to the ground state," *Nature Communications*, vol. 5, 2014. [Online]. Available: http://dx.doi.org/10.1038/ncomms5783

[264] D. Sofikitis, R. Horchani, X. Li, M. Pichler, M. Allegrini, A. Fioretti, D. Comparat, and P. Pillet, "Vibrational cooling of cesium molecules using noncoherent broadband light," *Physical Review A*, vol. 80, no. 5, pp. 1–4, November 2009. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.80.051401

[265] W. G. Rellergert, S. T. Sullivan, S. J. Schowalter, S. Kotochigova, K. Chen, and E. R. Hudson, "Evidence for sympathetic vibrational cooling of translationally cold molecules," *Nature*, vol. 495, no. 7442, pp. 490–494, 2013. [Online]. Available: http://dx.doi.org/10.1038/nature11937

[266] R. deCarvalho, J. Doyle, B. Friedrich, T. Guillet, J. Kim, D. Patterson, and J. Weinstein, "Buffer-gas loaded magnetic traps for atoms and molecules: A primer," *The European Physical Journal D-Atomic, Molecular, Optical and Plasma Physics*, vol. 7, no. 3, pp. 289–309, 1999. [Online]. Available: http://dx.doi.org/10.1007/s100530050572

[267] D. Herschbach, "Molecular collisions, from warm to ultracold," *Faraday Discussions*, vol. 142, p. 9, 2009. [Online]. Available: http://dx.doi.org/10.1039/b910118g

[268] E. Hudson, "Method for producing ultracold molecular ions," *Physical Review A*, vol. 79, no. 3, p. 032716, March 2009. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.79.032716

[269] D. Patterson, J. Rasmussen, and J. M. Doyle, "Intense atomic and molecular beams via neon buffer-gas cooling," *New Journal of Physics*, vol. 11, no. 5, p. 055018, 2009. [Online]. Available: http://dx.doi.org/10.1088/1367-2630/11/5/055018

[270] X. Tong, D. Wild, and S. Willitsch, "Collisional and radiative effects in the state-selective preparation of translationally cold molecular ions in ion traps," *Physical Review A*, vol. 83, no. 2, p. 023415, February 2011. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.83.023415

[271] K.-K. Ni, S. Ospelkaus, M. De Miranda, A. Pe'er, B. Neyenhuis, J. Zirbel, S. Kotochigova, P. Julienne, D. Jin, and J. Ye, "A high phase-space-density gas of polar molecules," *science*, vol. 322, no. 5899, pp. 231–235, 2008. [Online]. Available: http://dx.doi.org/10.1126/science.1163861

[272] B. McGuyer, M. McDonald, G. Iwata, M. Tarallo, A. Grier, F. Apfelbeck, and T. Zelevinsky, "High-precision spectroscopy of ultracold molecules in an optical lattice," *New Journal of Physics*, vol. 17, no. 5, p. 055004, 2015. [Online]. Available: http://dx.doi.org/10.1088/1367-2630/17/5/055004

[273] N. Wells and I. C. Lane, "Electronic states and spin-forbidden cooling transitions of AlH and AlF," *Physical Chemistry Chemical Physics*, vol. 13, no. 42, pp. 19 018–19 025, 2011. [Online]. Available: http://dx.doi.org/10.1039/C1CP21313J

[274] Y. Zhang and M. Stuke, "Laser mass spectrometric detection of AlH molecules as collision-free excimer laser photoproducts from aluminum alkyls," *Chemical physics letters*, vol. 149, no. 3, pp. 310–315, 1988. [Online]. Available: http://dx.doi.org/10.1016/0009-2614(88)85032-2

[275] ——, "UV excimer laser photochemistry of gaseous and surface adsorbed organometallics," *Journal of Crystal Growth*, vol. 93, no. 1, pp. 143–150, 1988. [Online]. Available: http://dx.doi.org/10.1016/0022-0248(88)90519-2

[276] Y. F. Zhu, R. Shehadeh, and E. R. Grant, "Laser spectroscopy in a pulsed jet of AlH: Ionization-detected ultraviolet absorption spectra of the transitions $C^1\Sigma^+$–$X^1\Sigma^+$ and $b^3\Sigma^-$–$X^1\Sigma^+$," *The Journal of chemical physics*, vol. 97, no. 2, pp. 883–893, 1992. [Online]. Available: http://dx.doi.org/10.1063/1.463192

[277] G. V. Chertihin and L. Andrews, "Reactions of pulsed-laser ablated aluminum atoms with hydrogen: infrared spectra of aluminum hydride (AlH, AlH_2, AlH_3, and Al_2H_2) species," *The Journal of Physical Chemistry*, vol. 97, no. 40, pp. 10 295–10 300, 1993. [Online]. Available: http://dx.doi.org/10.1021/j100142a007

[278] D. Halfen and L. M. Ziurys, "Precise Rest Frequencies for AlH and AlD ($X^1\Sigma^+$): A Reassessment of the Aluminum Hyperfine Structure," *The Astrophysical Journal Letters*, vol. 607, no. 1, p. L63, 2004. [Online]. Available: http://dx.doi.org/10.1086/421083

[279] S. Kaur and K. L. Baluja, "Electron-impact study of AlH using the *R*-matrix method," *Phys. Rev. A*, vol. 80, p. 042701, Oct 2009. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.80.042701

[280] W. Szajna and M. Zachwieja, "Emission spectroscopy of the $A^1\Pi$–$X^1\Sigma^+$ system of AlH," *The European Physical Journal D*, vol. 55, no. 3, pp. 549–555, 2009. [Online]. Available: http://dx.doi.org/10.1140/epjd/e2009-00253-y

[281] N. H. Sabelli, M. Kantor, R. Benedek, and T. L. Gilbert, "SCF potential curves for AlH and AlH^+ in the attractive and repulsive regions," *The Journal of Chemical Physics*, vol. 68, no. 6, p. 2767, 1978. [Online]. Available: http://dx.doi.org/10.1063/1.436068

[282] M. F. Guest and D. M. Hirst, "Potential energy curves for the ground and excited states of AlH^+," *Chemical Physics Letters*, vol. 84, no. 1, p. 167, 1981. [Online]. Available: http://dx.doi.org/10.1016/0009-2614(81)85393-6

[283] R. J. L. Roy and G. T. Kraemer, "BCONT, Version 2.2. A Computer Program for Calculating Bound → Continuum Transition Intensities for Diatomic Molecules," Waterloo, Ontario, Canada, 2004. [Online]. Available: http://leroy.uwaterloo.ca/

[284] R. J. L. Roy, "LEVEL, Version 8.0. Computer Program for Solving the Radial Schrödinger Equation for Bound and Quasibound Levels," Waterloo, Ontario, Canada, 2007. [Online]. Available: http://leroy.uwaterloo.ca/

[285] A. Hansson and J. K. Watson, "A comment on Hönl-London factors," *Journal of molecular spectroscopy*, vol. 233, no. 2, pp. 169–173, 2005. [Online]. Available: http://dx.doi.org/10.1016/j.jms.2005.06.009

[286] J. K. Watson, "Hönl–London factors for multiplet transitions in Hunds case a or b," *Journal of Molecular Spectroscopy*, vol. 252, no. 1, pp. 5–8, 2008. [Online]. Available: http://dx.doi.org/10.1016/j.jms.2008.04.014

[287] I. Vogelius, L. Madsen, and M. Drewsen, "Rotational cooling of heteronuclear molecular ions with ^{1}Σ, ^{2}Σ, ^{3}Σ, and ^{2}Π electronic ground states," *Physical Review A*, vol. 70, no. 5, pp. 1–14, November 2004. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.70.053412

[288] M. Kokish, V. Rajagopal, J. Marler, and B. Odom, "Note: High density pulsed molecular beam for cold ion chemistry," *Review of Scientific Instruments*, vol. 85, no. 8, p. 086111, 2014. [Online]. Available: http://dx.doi.org/10.1063/1.4894262

[289] C. Helms, L. Hanko, K. Healey, G. Hager, and G. Perram, "Generation of high-number densities of iodine monofluoride in a supersonic flow," *Journal of Applied Physics*, vol. 66, no. 12, pp. 6093–6097, 1989. [Online]. Available: http://dx.doi.org/10.1063/1.343590

[290] H. S. Müller and M. C. Gerry, "Hyperfine constants of bromine and iodine monofluoride," *The Journal of chemical physics*, vol. 103, no. 2, pp. 577–583, 1995. [Online]. Available: http://dx.doi.org/10.1063/1.470092

[291] R. Martínez, J. Terrón, I. Merelas, M. S. Rayo, and F. Castaño, "Analysis of the optical emission following electron impact on CF_3I," *Journal of Physics B: Atomic, Molecular and Optical Physics*, vol. 31, no. 8, p. 1793, 1998. [Online]. Available: http://dx.doi.org/10.1088/0953-4075/31/8/028

[292] D. Khalil, N. Billy, G. Gouédard, and J. Vigué, "Crossed-beam studies of some chemiluminescent reactions producing if," *Physical Chemistry Chemical Physics*, vol. 2, no. 4, pp. 729–736, 2000. [Online]. Available: http://dx.doi.org/10.1039/A907836C

[293] M. Chiappero, G. Argüello, P. Garcia, H. Pernice, H. Willner, H. Oberhammer, K. Peterson, and J. Francisco, "Molecular structure, spectroscopy and matrix photochemistry of fluorocarbonyl iodide, FC(O)I," *Chemistry–A European Journal*, vol. 10, no. 4, pp. 917–924, 2004. [Online]. Available: http://dx.doi.org/10.1002/chem.200305506

[294] F. Innocenti, M. Eypper, S. Beccaceci, A. Morris, S. Stranges, J. B. West, G. C. King, and J. M. Dyke, "A study of the reactive intermediate IF and I atoms with photoelectron spectroscopy," *The Journal of Physical Chemistry A*, vol. 112, no. 30, pp. 6939–6949, 2008. [Online]. Available: http://dx.doi.org/10.1021/jp803141c

[295] O. Cabrol, B. Girard, F. Spiegelmann, and C. Teichteil, "Relativistic calculation of the electronic structure of the IF molecule," *The Journal of chemical physics*, vol. 105, no. 15, pp. 6426–6438, 1996. [Online]. Available: http://dx.doi.org/10.1063/1.472996

[296] I. H. Krouse and P. G. Wenthold, "Bond dissociation energy and Lewis acidity of the xenon fluoride cation," *Inorganic chemistry*, vol. 42, no. 14, pp. 4293–4298, 2003. [Online]. Available: http://dx.doi.org/10.1021/ic034301w

[297] D. Shen, J. Sahu, and W. Clarkson, "High-power widely tunable Tm: fibre lasers pumped by an Er, Yb co-doped fibre laser at 1.6 μm," *Optics express*, vol. 14, no. 13, pp. 6084–6090, 2006. [Online]. Available: http://dx.doi.org/10.1364/OE.14.006084

[298] A. J. Yencha, M. C. A. Lopes, and G. C. King, "Threshold photoelectron spectroscopy of iodine monochloride," *Chemical Physics Letters*, vol. 325, no. 5, pp. 559–567, 2000. [Online]. Available: http://dx.doi.org/10.1016/S0009-2614(00)00719-3

[299] K. Balasubramanian, "Relativistic configuration-interaction calculations of the low-lying states of ICl and ICl^+," *Chemical physics*, vol. 95, no. 2, pp. 225–234, 1985. [Online]. Available: http://dx.doi.org/10.1016/0301-0104(85)80074-4

[300] C. M. Seck, E. G. Hohenstein, C.-Y. Lien, P. R. Stollenwerk, and B. C. Odom, "Rotational state analysis of AlH^+ by two-photon dissociation," *Journal of Molecular Spectroscopy*, vol. 300, pp. 108–111, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.jms.2014.03.023

[301] C.-Y. Lien, "Broadband Optical Cooling of Molecular Rotors," Ph.D. dissertation, Northwestern University, August 2014.

[302] J. Pedregosa, C. Champenois, M. Houssin, and M. Knoop, "Anharmonic contributions in real RF linear quadrupole traps," *International Journal of Mass Spectrometry*, vol. 290, no. 2, pp. 100–105, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.ijms.2009.12.009

[303] D. J. Berkeland, J. D. Miller, J. C. Bergquist, W. M. Itano, and D. J. Wineland, "Minimization of ion micromotion in a Paul trap," *Journal of Applied Physics*, vol. 83, no. 10, pp. 5025–5033, 1998. [Online]. Available: http://dx.doi.org/10.1063/1.367318

[304] Y. Ibaraki, U. Tanaka, and S. Urabe, "Detection of parametric resonance of trapped ions for micromotion compensation," *Applied Physics B*, vol. 105, no. 2, pp. 219–223, 2011. [Online]. Available: http://dx.doi.org/10.1007/s00340-011-4463-x

[305] T. Huber, A. Lambrecht, J. Schmidt, L. Karpa, and T. Schaetz, "A far-off-resonance optical trap for a Ba^+ ion," *Nature communications*, vol. 5, 2014. [Online]. Available: http://dx.doi.org/10.1038/ncomms6587

[306] A. Korpel, *Acousto-optics*. CRC Press, 1996, vol. 57.

[307] E. Donley, T. Heavner, F. Levi, M. Tataw, and S. Jefferts, "Double-pass acousto-optic modulator system," *Review of Scientific Instruments*, vol. 76, no. 6, p. 063112, 2005. [Online]. Available: http://dx.doi.org/10.1063/1.1930095

[308] J. Snyder, “Paraxial ray analysis of a cat’s-eye retroreflector,” *Applied optics*, vol. 14, no. 8, pp. 1825–1828, 1975. [Online]. Available: http://dx.doi.org/10.1364/AO.14.001825

[309] N. Ramsey, *Molecular Beams.* Oxford University Press, 1985.

[310] H. Pauly, *Atom, Molecule, and Cluster Beams I: Basic Theory, Production and Detection of Thermal Energy Beams.* Springer Science & Business Media, 2000. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-04213-7

[311] M. G. Kokish, M. R. Dietrich, and B. C. Odom, “Simple and compact nozzle design for laser vaporization sources,” *Journal of Physics B: Atomic, Molecular and Optical Physics*, vol. 49, no. 3, p. 035301, 2016. [Online]. Available: http://dx.doi.org/10.1088/0953-4075/49/3/035301

[312] K. Luria, W. Christen, and U. Even, “Generation and propagation of intense supersonic beams,” *The Journal of Physical Chemistry A*, vol. 115, no. 25, pp. 7362–7367, 2011. [Online]. Available: http://dx.doi.org/10.1021/jp201342u

[313] Y. Segev, N. Bibelnik, N. Akerman, Y. Shagam, A. Luski, M. Karpov, J. Narevicius, and E. Narevicius, “Molecular beam brightening by shock-wave suppression,” *arXiv preprint arXiv:1607.04844*, 2016. [Online]. Available: http://arxiv.org/abs/1607.04844

[314] W. E. Harbaugh, “Tungsten, Thoriated-Tungsten, and Thoria Emitters,” in *Electron Tube Design*, R. S. Burnap, Ed. Radio Corporation of America, 1962, pp. 90–98.

[315] M. N. Kishore and P. K. Ghosh, “Trapping of ion injected from an external source into a three-dimensional rf quadrupole field,” *International Journal of Mass Spectrometry and Ion Physics*, vol. 29, no. 4, pp. 345–350, 1979. [Online]. Available: http://dx.doi.org/10.1016/0020-7381(79)80004-2

[316] J. Todd, D. Freer, and R. Waldren, “The quadrupole ion store (QUISTOR). Part xii. The trapping of ions injected from an external source: A description in terms of phase-space dynamics,” *International Journal of Mass Spectrometry and Ion Physics*, vol. 36, no. 3, pp. 371–386, 1980. [Online]. Available: http://dx.doi.org/10.1016/0020-7381(80)85070-4

[317] O. Chun-Sing and H. A. Schuessler, “Confinement of pulse-injected external ions in a radiofrequency quadruple ion trap,” *International Journal of Mass Spectrometry and Ion Physics*, vol. 40, no. 1, pp. 53–66, 1981. [Online]. Available: http://dx.doi.org/10.1016/0020-7381(81)85006-1

[318] ——, “Confinement of ions created externally in a radio-frequency ion trap,” *Journal of Applied Physics*, vol. 52, no. 3, pp. 1157–1166, 1981. [Online]. Available: http://dx.doi.org/10.1063/1.329731

[319] ——, “Confinement of ions injected into a radiofrequency quadrupole ion trap: Pulsed ion beams of different energies,” *International Journal of Mass Spectrometry and Ion Physics*, vol. 40, no. 1, pp. 67–75, 1981. [Online]. Available: http://dx.doi.org/10.1016/0020-7381(81)85007-3

[320] B. d. Darwent, “Bond dissociation energies in simple molecules,” *NIST Spec. Publ*, vol. 1, no. 1, 1970.

[321] S. Corrigan, “Dissociation of molecular hydrogen by electron impact,” *The Journal of Chemical Physics*, vol. 43, no. 12, pp. 4381–4386, 1965. [Online]. Available: http://dx.doi.org/10.1063/1.1696701

[322] S. Chung, C. C. Lin, and E. T. P. Lee, “Dissociation of the hydrogen molecule by electron impact,” *Phys. Rev. A*, vol. 12, pp. 1340–1349, Oct 1975. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.12.1340

[323] G. A. Khayrallah, “Electron impact dissociation of molecular hydrogen and deuterium: Production of atomic hydrogen and deuterium α, β, and γ Balmer lines,” *Phys. Rev. A*, vol. 13, pp. 1989–2003, Jun 1976. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.13.1989

[324] A. M. Weiner, “Femtosecond pulse shaping using spatial light modulators,” *Review of Scientific Instruments*, vol. 71, no. 5, pp. 1929–1960, 2000. [Online]. Available: http://dx.doi.org/10.1063/1.1150614

[325] D. Elliott and S. Smith, “Experimental synthesis of phase-diffusing optical fields,” *JOSA B*, vol. 5, no. 9, pp. 1927–1939, 1988. [Online]. Available: http://dx.doi.org/10.1364/JOSAB.5.001927

[326] S. J. Schowalter, K. Chen, W. G. Rellergert, S. T. Sullivan, and E. R. Hudson, “An integrated ion trap and time-of-flight mass spectrometer for chemical and photo-reaction dynamics studies,” *Review of Scientific Instruments*, vol. 83, no. 4, p. 043103, 2012. [Online]. Available: http://dx.doi.org/10.1063/1.3700216

[327] U. Hechtfischer, C. J. Williams, M. Lange, J. Linkemann, D. Schwalm, R. Wester, A. Wolf, and D. Zajfman, “Photodissociation spectroscopy of stored CH^+ ions: Detection, assignment, and close-coupled modeling of near-threshold Feshbach resonances,” *The Journal of Chemical Physics*, vol. 117, no. 19, pp. 8754–8777, 2002. [Online]. Available: http://dx.doi.org/10.1063/1.1513459

[328] H.-J. Werner, P. J. K. andG. Knizia, F. R. Manby, M. Schütz *et al.*, "MOLPRO, Version 2012.1: A Package of *ab initio* Programs," Cardiff, UK, 2012. [Online]. Available: http://www.molpro.net

[329] K. Ruedenberg, L. Cheung, and S. Elbert, "MCSCF optimization through combined use of natural orbitals and the Brillouin–Levy–Berthier theorem," *International Journal of Quantum Chemistry*, vol. 16, no. 5, pp. 1069–1101, 1979. [Online]. Available: http://dx.doi.org/10.1002/qua.560160511

[330] B. O. Roos, P. R. Taylor, P. E. Si *et al.*, "A complete active space SCF method (CASSCF) using a density matrix formulated super-CI approach," *Chemical Physics*, vol. 48, no. 2, pp. 157–173, 1980. [Online]. Available: http://dx.doi.org/10.1016/0301-0104(80)80045-0

[331] T. H. Dunning Jr, "Gaussian basis sets for use in correlated molecular calculations. I. The atoms boron through neon and hydrogen," *The Journal of chemical physics*, vol. 90, no. 2, pp. 1007–1023, 1989. [Online]. Available: http://dx.doi.org/10.1063/1.456153

[332] D. E. Woon and T. H. Dunning Jr, "Gaussian basis sets for use in correlated molecular calculations. III. The atoms aluminum through argon," *The Journal of chemical physics*, vol. 98, no. 2, pp. 1358–1371, 1993. [Online]. Available: http://dx.doi.org/10.1063/1.464303

[333] O. Christiansen, H. Koch, P. Jo *et al.*, "Response functions in the CC3 iterative triple excitation model," *The Journal of chemical physics*, vol. 103, no. 17, pp. 7429–7441, 1995. [Online]. Available: http://dx.doi.org/10.1063/1.470315

[334] H. Koch, O. Christiansen, P. Jørgensen, and J. Olsen, "Excitation energies of BH, CH_2 and Ne in full configuration interaction and the hierarchy CCS, CC2, CCSD and CC3 of coupled cluster models," *Chemical physics letters*, vol. 244, no. 1, pp. 75–82, 1995. [Online]. Available: http://dx.doi.org/10.1016/0009-2614(95)00914-P

[335] W. C. Martin and R. Zalubas, "Energy levels of aluminum, Al I through Al XIII," *Journal of Physical and Chemical Reference Data*, vol. 8, no. 3, pp. 817–864, 1979. [Online]. Available: http://dx.doi.org/10.1063/1.555608

[336] O. O. Versolato, M. Schwarz, A. K. Hansen, A. D. Gingell, A. Windberger, L. Kłosowski, J. Ullrich, F. Jensen, J. R. Crespo López-Urrutia, and M. Drewsen, "Decay Rate Measurement of the First Vibrationally Excited State of MgH^+ in a Cryogenic Paul Trap," *Phys. Rev. Lett.*, vol. 111, p. 053002, Jul 2013. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.111.053002

[337] L. D. Brown, T. T. Cai, and A. DasGupta, "Interval estimation for a binomial proportion," *Statistical science*, pp. 101–117, 2001. [Online]. Available: http://dx.doi.org/10.1214/ss/1009213286

[338] R. Barlow, "A calculator for confidence intervals," *Computer Physics Communications*, vol. 149, pp. 97–102, 2002. [Online]. Available: http://dx.doi.org/10.1016/S0010-4655(02)00588-X

[339] M. Di Rosa, "Laser-cooling molecules," *The European Physical Journal D-Atomic, Molecular, Optical and Plasma Physics*, vol. 31, no. 2, pp. 395–402, 2004. [Online]. Available: http://dx.doi.org/10.1140/epjd/e2004-00167-2

[340] H.-Y. Chen, C.-Y. Lien, W.-Y. Lin, Y. T. Lee, and J. J. Lin, "UV absorption cross sections of ClOOCl are consistent with ozone degradation models," *Science*, vol. 324, no. 5928, pp. 781–784, 2009. [Online]. Available: http://dx.doi.org/10.1126/science.1171305

[341] U. Bressel, A. Borodin, J. Shen, M. Hansen, I. Ernsting, and S. Schiller, "Manipulation of Individual Hyperfine States in Cold Trapped Molecular Ions and Application to HD^+ Frequency Metrology," *Phys. Rev. Lett.*, vol. 108, p. 183003, May 2012. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.108.183003

[342] G. Herzberg, *Molecular Spectra and Molecular Structure*, 2nd ed. D. Van Nostrand Company, 1950, vol. 1.

[343] C. Edwards, C. Maclean, and P. Sarre, "Hyperfine structure in electronic spectra of NH^+, PH^+, and SH^+ resolved by laser-ion beam spectroscopy," *The Journal of Chemical Physics*, vol. 76, no. 7, pp. 3829–3831, 1982. [Online]. Available: http://dx.doi.org/10.1063/1.443425

[344] L. B. Knight Jr and J. Steadman, "ESR investigations of H_2O^+, HDO^+, D_2O^+, and $H_2^{17}O^+$ isolated in neon matrices at 4 K," *The Journal of Chemical Physics*, vol. 78, no. 10, pp. 5940–5945, 1983. [Online]. Available: http://dx.doi.org/10.1063/1.4767932

[345] J. V. Coe, J. Owrutsky, E. Keim, N. Agman, D. Hovde, and R. Saykally, "Sub-Doppler direct infrared laser absorption spectroscopy in fast ion beams: The fluorine hyperfine structure of HF^+," *The Journal of Chemical Physics*, vol. 90, no. 8, pp. 3893–3902, 1989. [Online]. Available: http://dx.doi.org/10.1063/1.455800

[346] D. C. Hovde, E. R. Keim, and R. J. Saykally, "Velocity modulation laser spectroscopy of molecular ions: The hyperfine-resolved rovibrational spectrum of

HF^+," *Molecular Physics*, vol. 68, no. 3, pp. 599–607, 1989. [Online]. Available: http://dx.doi.org/10.1080/00268978900102381

[347] J. Kändler, T. Martell, and W. Ernst, "Electric dipole moments and hyperfine structure of SrF A 2 Π and B 2 Σ+," *Chemical physics letters*, vol. 155, no. 4, pp. 470–474, 1989. [Online]. Available: http://dx.doi.org/10.1016/0009-2614(89)87188-X

[348] W. Weltner, *Magnetic Atoms and Molecules.* Courier Corporation, 1989.

[349] A. Chanda, W. Ho, F. Dalby, and I. Ozier, "Fine and Hyperfine Structure in the Vibrational Spectrum of the $X^2\Pi$ State of HBr^+," *Journal of Molecular Spectroscopy*, vol. 169, no. 1, pp. 108–147, 1995. [Online]. Available: http://dx.doi.org/10.1006/jmsp.1995.1009

[350] B. E. Sauer, J. Wang, and E. A. Hinds, "Anomalous Spin-Rotation Coupling in the $X^2\Sigma^+$ State of YbF," *Phys. Rev. Lett.*, vol. 74, pp. 1554–1557, Feb 1995. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.74.1554

[351] J. Brown and A. Carrington, *Rotational Spectroscopy of Diatomic Molecules*, 1st ed. Cambridge, 2003.

[352] P. J. Bruna and F. Grein, "Hyperfine coupling constants, electron-spin g-factors and vertical spectra of the $X^2\Sigma^+$ radicals BeH, MgH, CaH and BZ^+, AlZ^+, GaZ^+ (Z = H, Li, Na, K). A theoretical study," *Physical Chemistry Chemical Physics*, vol. 5, no. 15, p. 3140, 2003. [Online]. Available: http://dx.doi.org/10.1039/b303698g

[353] D. Bakalov, V. I. Korobov, and S. Schiller, "High-Precision Calculation of the Hyperfine Structure of the HD^+ Ion," *Phys. Rev. Lett.*, vol. 97, p. 243001, Dec 2006. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.97.243001

[354] J. C. J. Koelemeij, B. Roth, A. Wicht, I. Ernsting, and S. Schiller, "Vibrational Spectroscopy of HD^+ with 2-ppb Accuracy," *Phys. Rev. Lett.*, vol. 98, p. 173002, Apr 2007. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.98.173002

[355] M. D. Correnti, K. P. Dickert, M. A. Pittman, J. W. Felmly, J. J. Banisaukas III, and L. B. Knight Jr, "Electron spin resonance investigation of H_2^+, HD^+, and D_2^+ isolated in neon matrices at 2 K," *The Journal of chemical physics*, vol. 137, no. 20, p. 204308, 2012. [Online]. Available: http://dx.doi.org/10.1063/1.4767932

[356] L. R. Hunter, S. K. Peck, A. S. Greenspon, S. S. Alam, and D. DeMille, "Prospects for laser cooling TlF," *Phys. Rev. A*, vol. 85, p. 012511, Jan 2012. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.85.012511

[357] J. van Veldhoven, J. Küpper, H. L. Bethlem, B. Sartakov, A. J. van Roij, and G. Meijer, “Decelerated molecular beams for high-resolution spectroscopy,” *The European Physical Journal D-Atomic, Molecular, Optical and Plasma Physics*, vol. 31, no. 2, pp. 337–349, 2004. [Online]. Available: http://dx.doi.org/10.1140/epjd/e2004-00160-9

[358] L. B. Knight Jr, S. T. Cobranchi, B. W. Gregory, and E. Earl, “Reinvestigation of the aluminum hydride (AlH^+ and AlD^+) cation radicals by ESR in argon matrices at 4 K: Generation by reactive laser sputtering,” *The Journal of chemical physics*, vol. 86, no. 6, pp. 3143–3150, 1987. [Online]. Available: http://dx.doi.org/10.1063/1.452025

[359] R. A. Frosch and H. M. Foley, “Magnetic Hyperfine Structure in Diatomic Molecules,” *Phys. Rev.*, vol. 88, pp. 1337–1349, Dec 1952. [Online]. Available: http://dx.doi.org/10.1103/PhysRev.88.1337

[360] J. Mes, E. J. van Duijn, R. Zinkstok, S. Witte, and W. Hogervorst, “Third-harmonic generation of a continuous-wave Ti:Sapphire laser in external resonant cavities,” *Applied Physics Letters*, vol. 82, no. 25, pp. 4423–4425, 2003. [Online]. Available: http://dx.doi.org/10.1063/1.1584515

[361] B. McCall, July 2013, private communication.

[362] F. J. Duarte, *Tunable Lasers Handbook*. Academic Press, 1995.

[363] F. Diedrich, J. C. Bergquist, W. M. Itano, and D. J. Wineland, “Laser Cooling to the Zero-Point Energy of Motion,” *Phys. Rev. Lett.*, vol. 62, pp. 403–406, Jan 1989. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.62.403

[364] J. C. Bergquist, D. J. Wineland, W. M. Itano, H. Hemmati, H. U. Daniel, and G. Leuchs, “Energy and Radiative Lifetime of the $5d^96s^2$ $^2D_{\frac{5}{2}}$ State in Hg II by Doppler-Free Two-Photon Laser Spectroscopy,” *Phys. Rev. Lett.*, vol. 55, pp. 1567–1570, October 1985. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.55.1567

[365] C. Roos, T. Zeiger, H. Rohde, H. C. Nägerl, J. Eschner, D. Leibfried, F. Schmidt-Kaler, and R. Blatt, “Quantum State Engineering on an Optical Transition and Decoherence in a Paul Trap,” *Phys. Rev. Lett.*, vol. 83, pp. 4713–4716, Dec 1999. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.83.4713

[366] F. Schmidt-Kaler, C. Roos, H. C. Ngerl, H. Rohde, S. Gulde, A. Mundt, M. Lederbauer, G. Thalhammer, T. Zeiger, P. Barton, L. Hornekaer, G. Reymond,

D. Leibfried, J. Eschner, and R. Blatt, "Ground state cooling, quantum state engineering and study of decoherence of ions in Paul traps," *Journal of Modern Optics*, vol. 47, no. 14-15, pp. 2573–2582, 2000. [Online]. Available: http://dx.doi.org/10.1080/09500340008232182

[367] E. Peik, J. Abel, T. Becker, J. von Zanthier, and H. Walther, "Sideband cooling of ions in radio-frequency traps," *Phys. Rev. A*, vol. 60, pp. 439–449, July 1999. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.60.439

[368] V. Letchumanan, G. Wilpers, M. Brownnutt, P. Gill, and A. G. Sinclair, "Zero-point cooling and heating-rate measurements of a single $^{88}\mathrm{Sr}^{+}$ ion," *Phys. Rev. A*, vol. 75, p. 063425, June 2007. [Online]. Available: http://dx.doi.org/10.1103/10.1103/PhysRevA.75.063425

[369] V. Letchumanan, M. A. Wilson, P. Gill, and A. G. Sinclair, "Lifetime measurement of the metastable $4d^2D_{52}$ state in $^{88}\mathrm{Sr}^{+}$ using a single trapped ion," *Phys. Rev. A*, vol. 72, p. 012509, July 2005. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.72.012509

[370] N. Yu and L. Maleki, "Lifetime measurements of the $4f^{14}5d$ metastable states in single ytterbium ions," *Phys. Rev. A*, vol. 61, p. 022507, Jan 2000.

[371] T. Noel, M. R. Dietrich, N. Kurz, G. Shu, J. Wright, and B. B. Blinov, "Adiabatic passage in the presence of noise," *Phys. Rev. A*, vol. 85, p. 023401, Feb 2012. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.85.023401

[372] S. Weidt, J. Randall, S. C. Webster, E. D. Standing, A. Rodriguez, A. E. Webb, B. Lekitsch, and W. K. Hensinger, "Ground-state cooling of a trapped ion using long-wavelength radiation," *Phys. Rev. Lett.*, vol. 115, p. 013002, Jun 2015. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.115.013002

[373] D. Hayes, D. N. Matsukevich, P. Maunz, D. Hucul, Q. Quraishi, S. Olmschenk, W. Campbell, J. Mizrahi, C. Senko, and C. Monroe, "Entanglement of Atomic Qubits Using an Optical Frequency Comb," *Phys. Rev. Lett.*, vol. 104, p. 140501, Apr 2010. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.104.140501

[374] R. G. DeVoe and C. Kurtsiefer, "Experimental study of anomalous heating and trap instabilities in a microscopic $^{137}\mathrm{Ba}$ ion trap," *Phys. Rev. A*, vol. 65, p. 063407, Jun 2002. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.65.063407

[375] M. Dietrich, "Barium Ions for Quantum Computation," Ph.D. dissertation, University of Washington, 2009.

[376] P. Anisimov and O. Kocharovskaya, “Decaying-dressed-state analysis of a coherently driven three-level Λ system,” *Journal of Modern Optics*, vol. 55, no. 19-20, pp. 3159–3171, 2008. [Online]. Available: http://dx.doi.org/10.1080/09500340802302378

[377] C. Cohen-Tannoudji, “Dark resonances from optical pumping to cold atoms and molecules,” *Physica Scripta*, vol. 90, no. 8, p. 088013, 2015. [Online]. Available: http://dx.doi.org/10.1088/0031-8949/90/8/088013

[378] J. Roßnagel, K. N. Tolazzi, F. Schmidt-Kaler, and K. Singer, “Fast thermometry for trapped ions using dark resonances,” *New Journal of Physics*, vol. 17, no. 4, p. 045004, 2015. [Online]. Available: http://dx.doi.org/10.1088/1367-2630/17/4/045004

[379] M. J. McDonnell, D. N. Stacey, and A. M. Steane, “Laser linewidth effects in quantum state discrimination by electromagnetically induced transparency,” *Phys. Rev. A*, vol. 70, p. 053802, Nov 2004. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.70.053802

[380] B. Lounis and C. Cohen-Tannoudji, “Coherent population trapping and Fano profiles,” *J. Phys. II France*, vol. 2, no. 4, pp. 579–592, 1992. [Online]. Available: http://dx.doi.org/10.1051/jp2:1992153

[381] Y. Stalgies, I. Siemers, B. Appasamy, and P. E. Toschek, “Light shift and Fano resonances in a single cold ion,” *J. Opt. Soc. Am. B*, vol. 15, no. 10, pp. 2505–2514, Oct 1998. [Online]. Available: http://dx.doi.org/10.1364/JOSAB.15.002505

[382] G. Morigi, J. Eschner, and C. H. Keitel, “Ground State Laser Cooling Using Electromagnetically Induced Transparency,” *Phys. Rev. Lett.*, vol. 85, pp. 4458–4461, Nov 2000. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.85.4458

[383] U. Fano, “Effects of Configuration Interaction on Intensities and Phase Shifts,” *Phys. Rev.*, vol. 124, pp. 1866–1878, Dec 1961. [Online]. Available: http://dx.doi.org/10.1103/PhysRev.124.1866

[384] R. Lechner, C. Maier, C. Hempel, P. Jurcevic, B. P. Lanyon, T. Monz, M. Brownnutt, R. Blatt, and C. F. Roos, “Electromagnetically-induced-transparency ground-state cooling of long ion strings,” *Phys. Rev. A*, vol. 93, p. 053401, May 2016. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.93.053401

[385] H. C. Nägerl, C. Roos, D. Leibfried, H. Rohde, G. Thalhammer, J. Eschner, F. Schmidt-Kaler, and R. Blatt, “Investigating a qubit candidate: Spectroscopy

on the $S_{1/2}$ to $D_{5/2}$ transition of a trapped calcium ion in a linear Paul trap," *Phys. Rev. A*, vol. 61, p. 023405, Jan 2000. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.61.023405

[386] M. Schacht, "Spin State Detection and Manipulation and Parity Violation in a Single Trapped Ion," Ph.D. dissertation, University of Washington, 2000.

[387] T. W. Koerber, M. Schacht, W. Nagourney, and E. N. Fortson, "Radio frequency spectroscopy with a trapped Ba$^+$ ion: recent progress and prospects for measuring parity violation," *Journal of Physics B: Atomic, Molecular and Optical Physics*, vol. 36, no. 3, p. 637, 2003. [Online]. Available: http://dx.doi.org/10.1088/0953-4075/36/3/320

[388] M. Hettrich, T. Ruster, H. Kaufmann, C. F. Roos, C. T. Schmiegelow, F. Schmidt-Kaler, and U. G. Poschinger, "Measurement of dipole matrix elements with a single trapped ion," *Phys. Rev. Lett.*, vol. 115, p. 143003, Oct 2015. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.115.143003

[389] Q. A. Turchette, Kielpinski, B. E. King, D. Leibfried, D. M. Meekhof, C. J. Myatt, M. A. Rowe, C. A. Sackett, C. S. Wood, W. M. Itano, C. Monroe, and D. J. Wineland, "Heating of trapped ions from the quantum ground state," *Phys. Rev. A*, vol. 61, p. 063418, May 2000. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.61.063418

[390] D. Shiqian, "Control of Trapped-ion Motion for Molecular Spectroscopy and Quantum Information Processing," Ph.D. dissertation, National University of Singapore, September 2015.

[391] N. C. Lewty, B. L. Chuah, R. Cazan, M. D. Barrett, and B. K. Sahoo, "Experimental determination of the nuclear magnetic octupole moment of ^{137}Ba$^+$ ion," *Phys. Rev. A*, vol. 88, p. 012518, Jul 2013. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.88.012518

[392] M. Fewell, B. Shore, and K. Bergmann, "Coherent population transfer among three states: Full algebraic solutions and the relevance of non adiabatic processes to transfer by delayed pulses," *Australian journal of physics*, vol. 50, no. 2, pp. 281–308, 1997. [Online]. Available: http://dx.doi.org/10.1071/P96071

[393] V. Romanenko and L. Yatsenko, "Adiabatic population transfer in the three-level Λ-system: two-photon lineshape," *Optics communications*, vol. 140, no. 4, pp. 231–236, 1997. [Online]. Available: http://dx.doi.org/10.1016/S0030-4018(97)00152-1

[394] K. Bergmann, H. Theuer, and B. W. Shore, "Coherent population transfer among quantum states of atoms and molecules," *Rev. Mod. Phys.*, vol. 70, pp. 1003–1025, Jul 1998. [Online]. Available: http://dx.doi.org/10.1103/RevModPhys.70.1003

[395] D. Møller, J. L. Sørensen, J. B. Thomsen, and M. Drewsen, "Efficient qubit detection using alkaline-earth-metal ions and a double stimulated Raman adiabatic process," *Phys. Rev. A*, vol. 76, p. 062321, Dec 2007. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.76.062321

[396] N. V. Vitanov and S. Stenholm, "Adiabatic population transfer via multiple intermediate states," *Phys. Rev. A*, vol. 60, pp. 3820–3832, Nov 1999. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.60.3820

[397] I. Labazan, N. Krstulović, and S. Milošević, "Laser vaporization of $LiAlH_4$ solid samples," *Chemical physics letters*, vol. 428, no. 1, pp. 13–17, 2006. [Online]. Available: http://dx.doi.org/10.1016/j.cplett.2006.06.111

[398] M. Drewsen, A. Mortensen, R. Martinussen, P. Staanum, and J. L. Sørensen, "Nondestructive Identification of Cold and Extremely Localized Single Molecular Ions," *Phys. Rev. Lett.*, vol. 93, p. 243201, Dec 2004. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.93.243201

[399] J. E. Goeders, C. R. Clark, G. Vittorini, K. Wright, C. R. Viteri, and K. R. Brown, "Identifying single molecular ions by resolved sideband measurements," *The Journal of Physical Chemistry A*, vol. 117, no. 39, pp. 9725–9731, 2013. [Online]. Available: http://dx.doi.org/10.1021/jp312368a

[400] T. Müller, "Production of Cold Molecular Radicals by Laser Ablation," Ph.D. dissertation, Imperial College, University of London, March 2005.

[401] G. D. Gillen, C. M. Seck, and S. Guha, "Analytical beam propagation model for clipped focused-Gaussian beams using vector diffraction theory," *Opt. Express*, vol. 18, no. 5, pp. 4023–4040, Mar 2010. [Online]. Available: http://dx.doi.org/10.1364/OE.18.004023

[402] C. M. Seck, M. G. Kokish, M. R. Dietrich, and B. C. Odom, "Raman sideband cooling of a $^{138}Ba^+$ ion using a Zeeman interval," *Phys. Rev. A*, vol. 93, p. 053415, May 2016. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.93.053415

[403] C. M. Seck, P. J. Martin, E. C. Cook, B. C. Odom, and D. A. Steck, "Noise reduction of a Libbrecht–Hall style current driver," *Review of Scientific Instruments*, vol. 87, no. 6, p. 064703, 2016. [Online]. Available: http://dx.doi.org/10.1063/1.4953330

Appendices

APPENDIX A

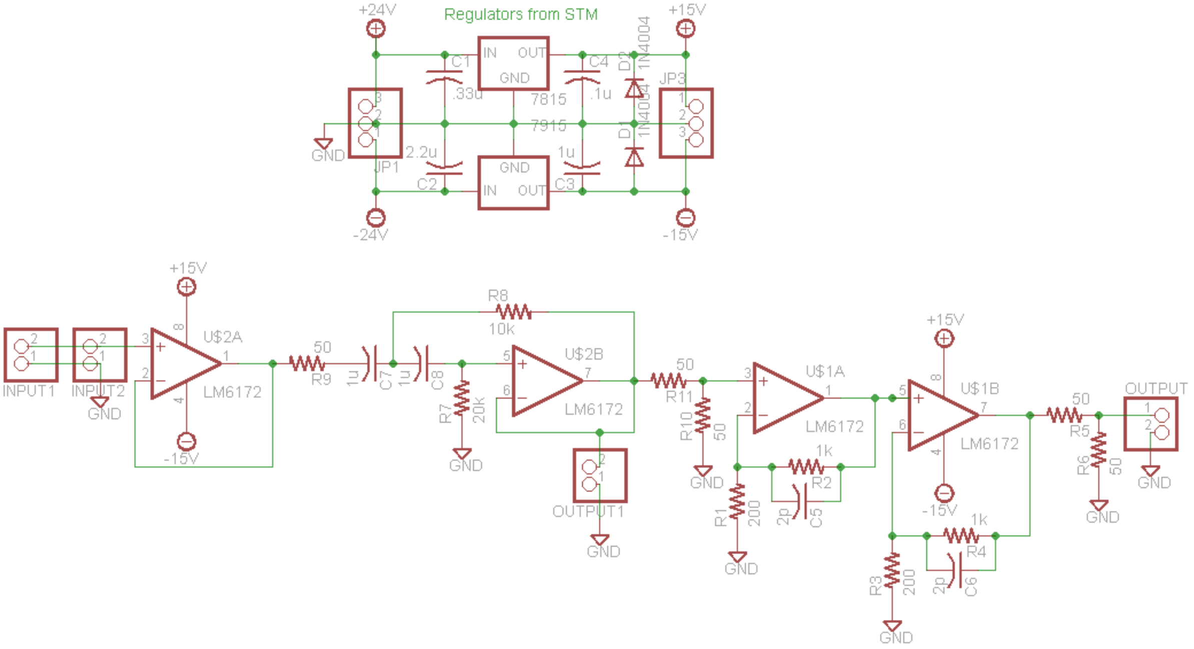
Circuit Diagrams

This appendix contains the most-recent versions of the circuits relevant to this work.

A.1. ECDL Project

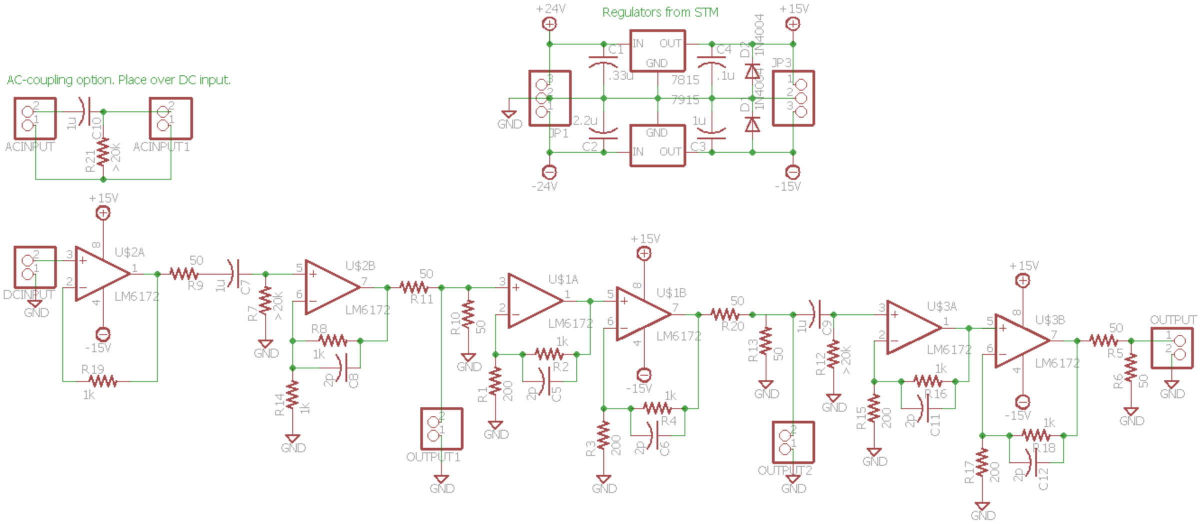
A.1.1. 20 dB Amplifier

The amplifier below provides 18.8 dB of gain for AC signals with a $f_{-3\mathrm{dB}} = 5$ MHz and is used to amplify current driver noise for the Agilent N9320A spectrum analyzer.



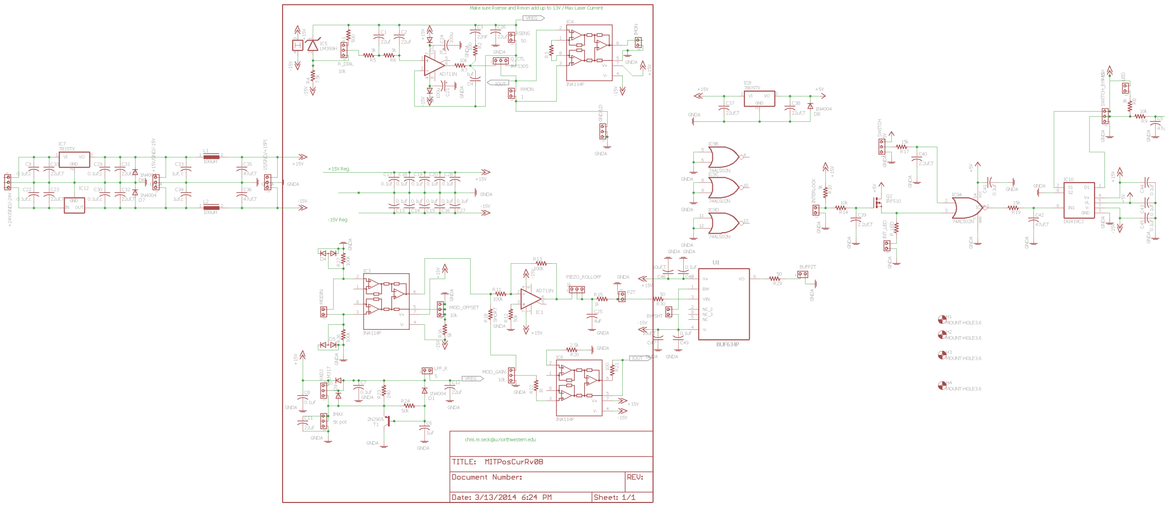
A.1.2. 40 dB Amplifier

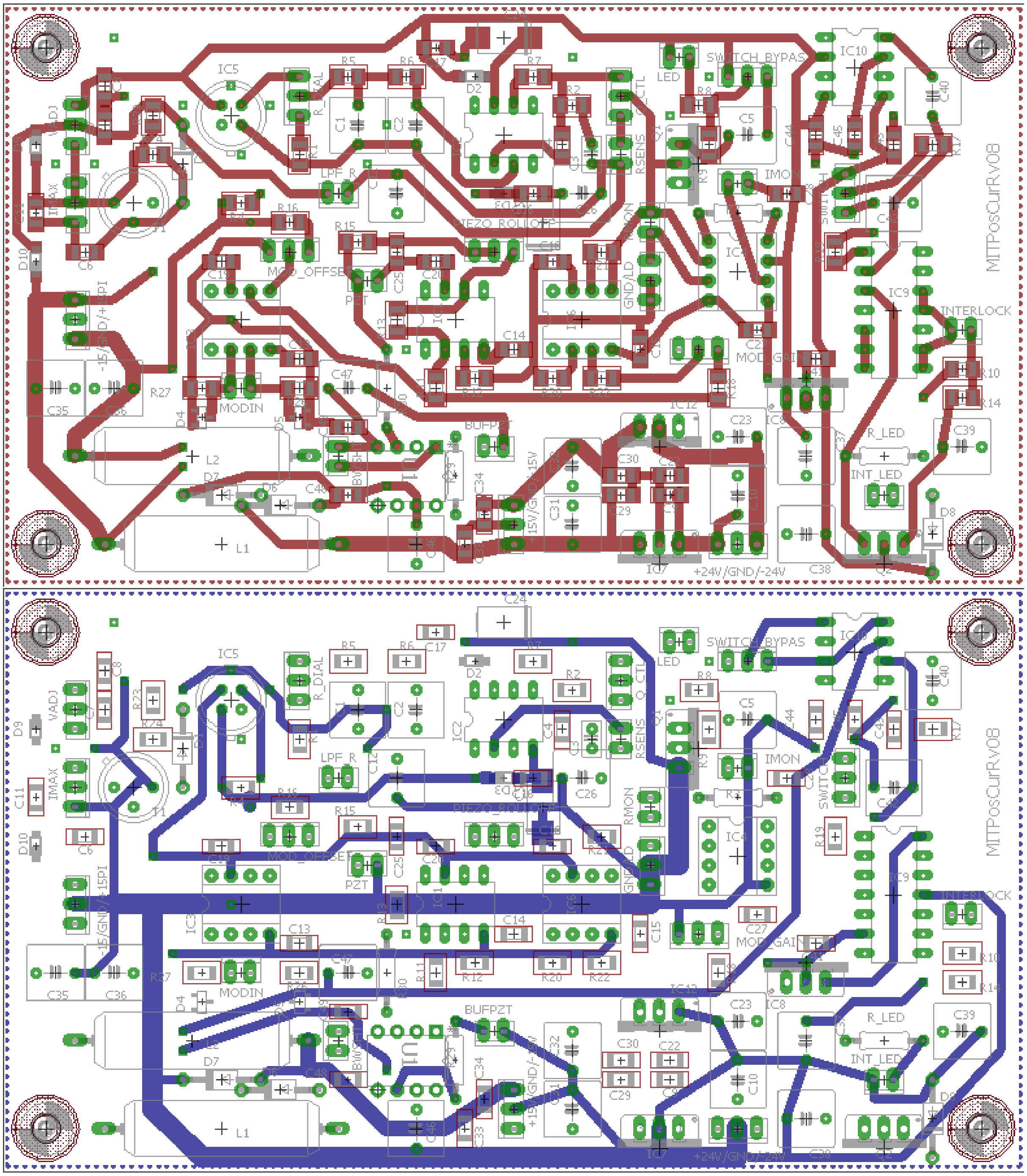
The amplifier below provides 34.5 dB of gain for AC signals with a $f_{-3\text{dB}} = 3.5$ MHz and is used to amplify current driver noise for the Agilent N9320A spectrum analyzer.



A.1.3. Diode Laser Current Driver

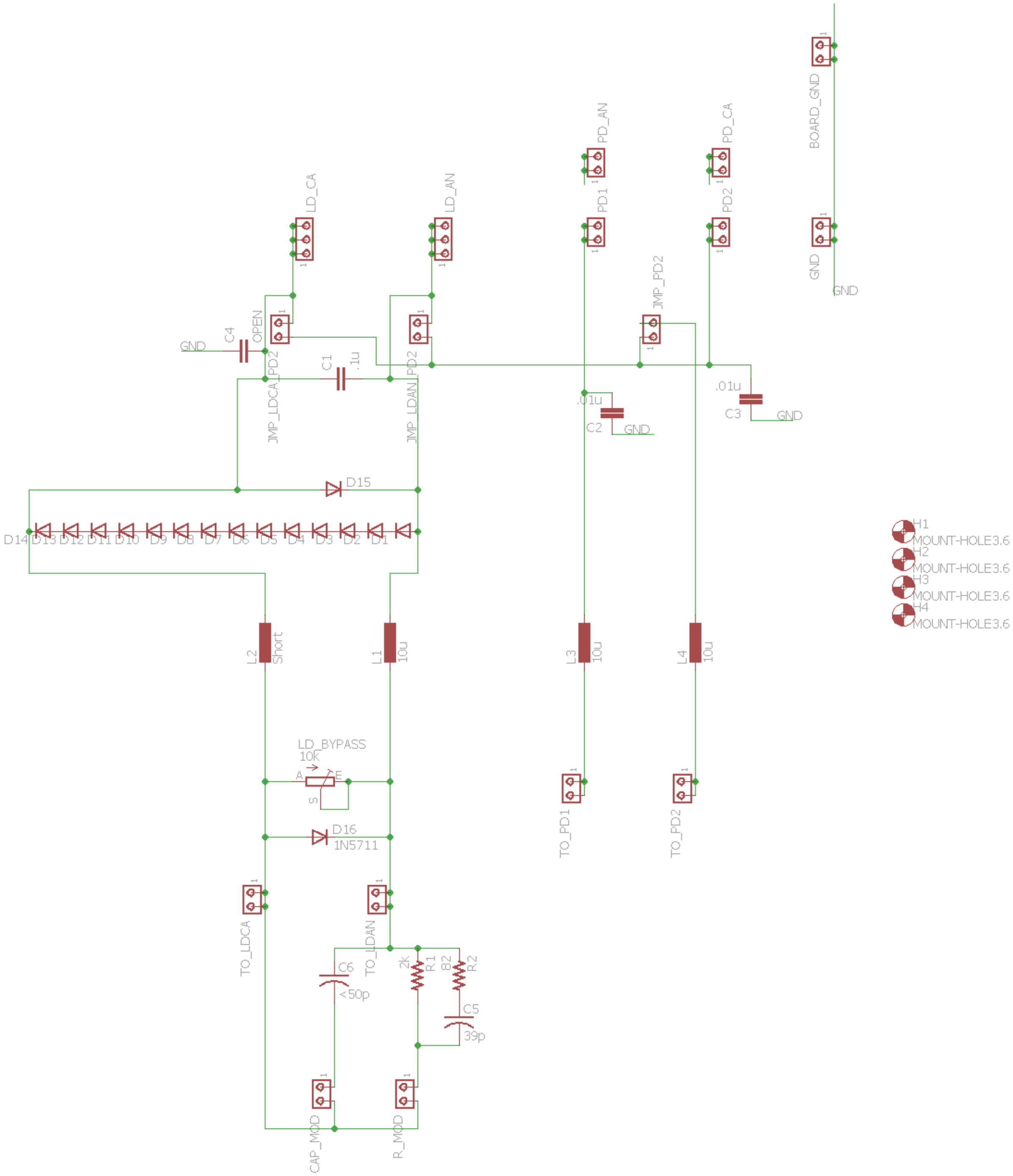
The in-house Libbrecht-Hall style current driver implementation. Notes/changes: C7 to 47 μF; C3 to 100 nF; omit C4; IC2 is an AD8671 on a Logical Systems PA-SOD3SM18-08 adapter; Q_CTL is an IRF9Z14 and is case mounted with 100 μH inductor on the output; remove R26; replace R27 with a 10 Ω and 10 nF in parallel; short R22 and R20; replace Mod_Offset with a 1 (2) kΩ potentiometer; R16 to 500 Ω (1 kΩ); replace Mod_Gain with a 1 (2) kΩ trimmer; replace R21 with 1 kΩ; RSENS, RMON, and LPF_R are TO-220 package and case mounted; the 7815 is case mounted; and the BWSHT pin is shorted. INAs typically are INA114 or INA217 with the latter providing higher current sinking for the current feed-forward. If a laser diode drive current or operating voltage is high, RSENS, LPF_R, and RMON values can be decreased. Additionally, the above changes *do not* reflect any of the recommended changes described in Ch. 4. The circuit is called MITPosCurRv08.

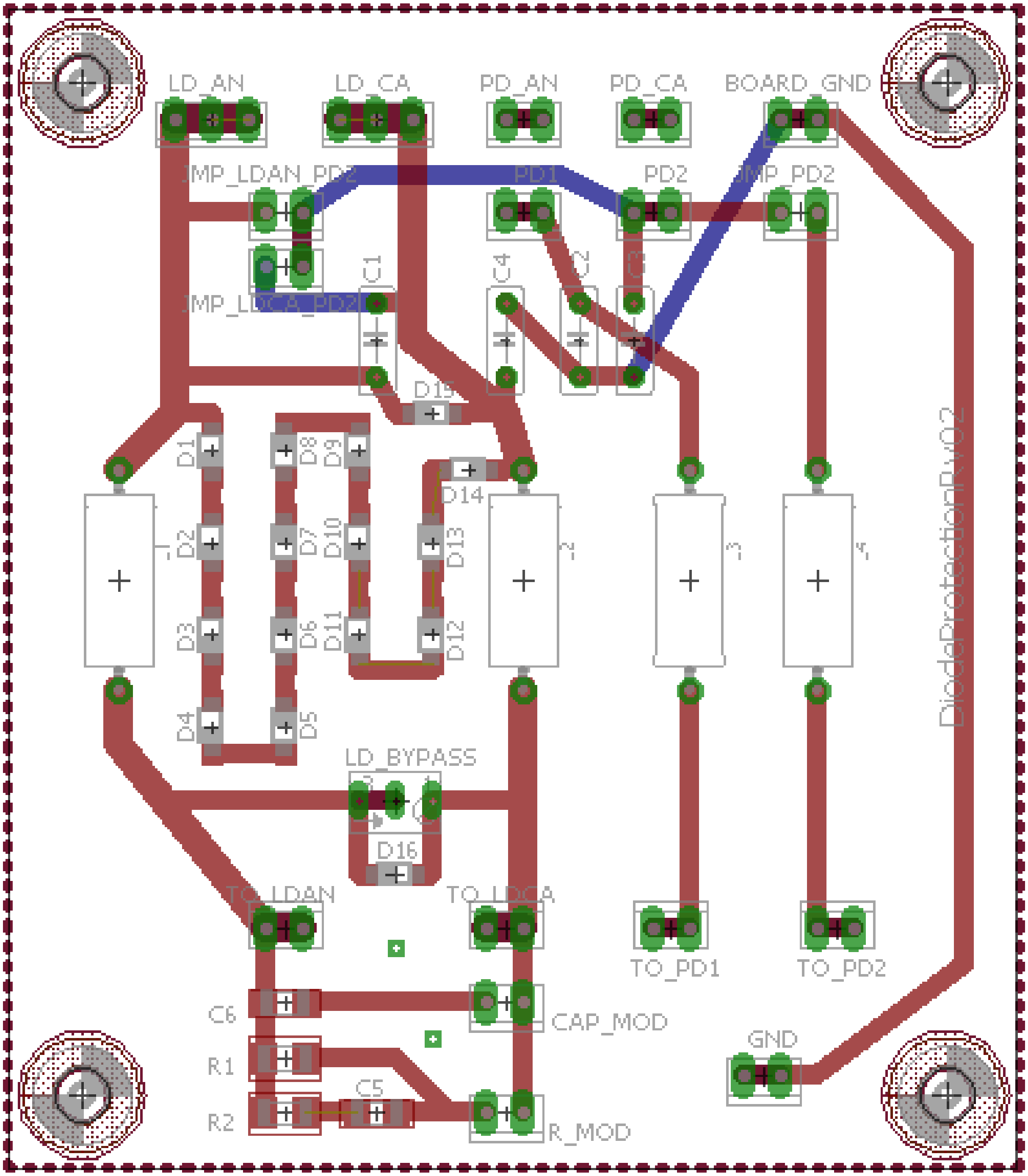



MITPosCurrRv08
IC5
R5
R6
C17
R7
C24
SWITCH_BYPAS
IC10
LED
R8
C5
C40
VADJ
IMAX
T1
C11
C6
D9
D10
R23
R24
R_DIAL
C1
C2
R1
D2
R2
IC2
C4
C3
Q_CTL
RSENS
Q1
R3
IMON
C44
C45
C43
R17
LPF_R
C12
D3
R16
R15
PIEZO_ROLLOFF
C18
C26
RMON
R4
IC4
SWITCH
C41
R19
C19
MOD_OFFSET
PZT
C25
C20
IC3
IC1
C13
C14
IC6
GND/LD
C15
C27
MOD_GAIN
IC9
INTERLOCK
R10
R14
-15/GND/+15PI
C35
C36
R27
MODIN
D4
D5
C47
R13
R11
R12
R20
R22
R18
C30
IC12
C23
IC8
R_LED
C39
INT_LED
L2
D7
D6
C48
BUFPZT
R30
R29
C34
+15V/GND/-15V
C32
C31
C22
C29
C10
C37
C38
D8
L1
C46
C33
IC7
+24V/GND/-24V
Q2

A.1.4. Diode Laser Protection Circuit

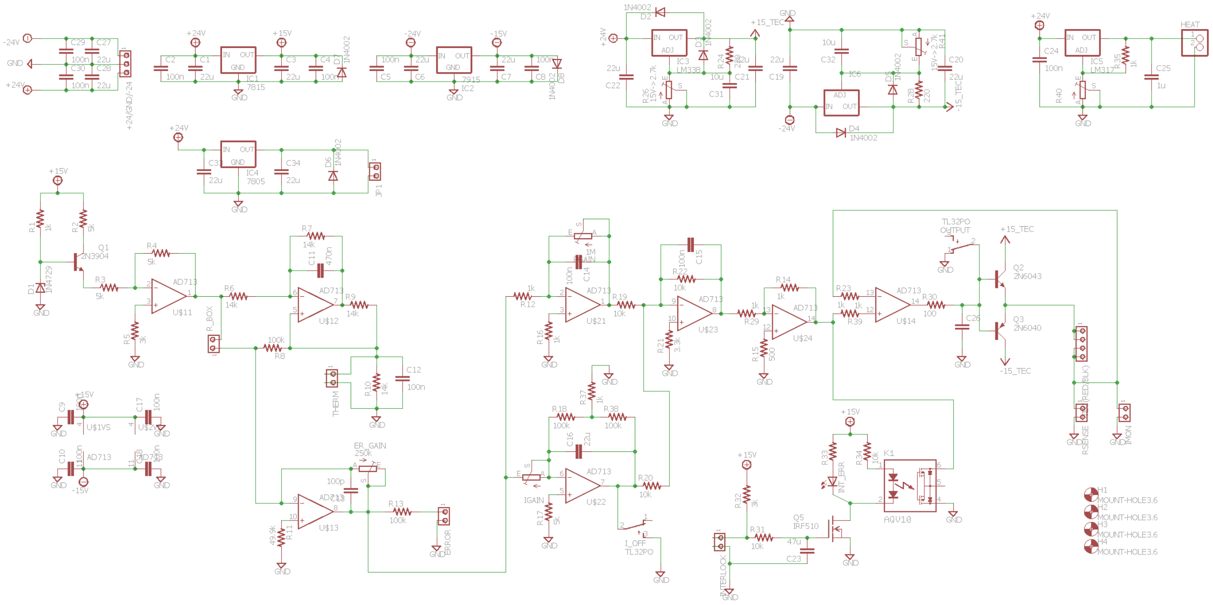
Diode protection circuit is placed inside of the ECDL enclosure. Wire leads go from the output of the board to the laser diode using a 3M 203-6585-50-0602J, 203-6585-50-0602J, 204-6585-50-0602J, or 204-6585-50-0602J socket. The potentiometer is shorted whenever equipment is being connected or disconnected from the ECDL system, cables disconnected, or drivers powered off. The design is modular to work for any 3- or 4-pin laser diode configuration. The circuit is called DiodeProtectionRv02.

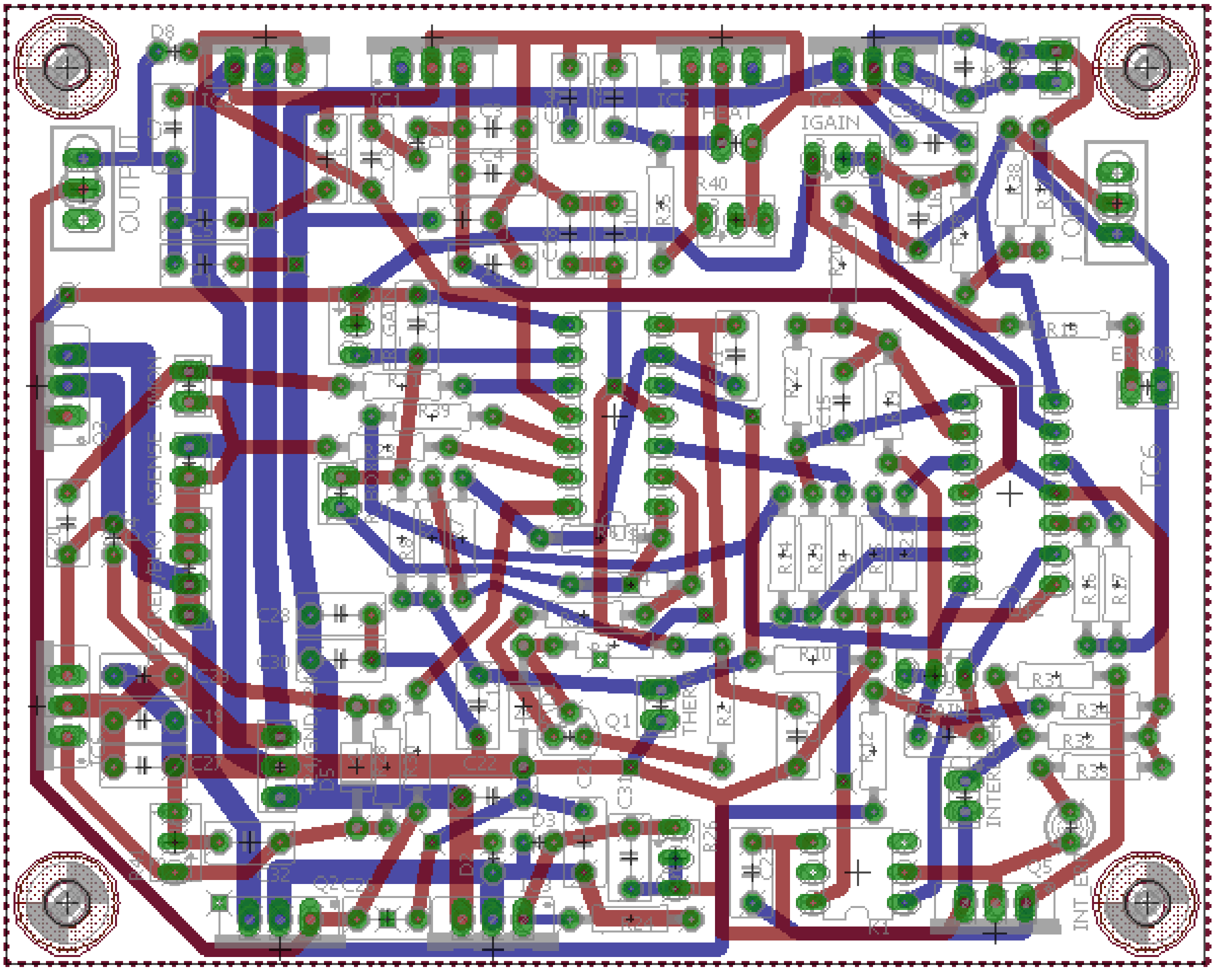

BOARD_GND
GND
GND
PD_AN
PD1
PD_CA
PD2
LD_CA
LD_AN
JMP_PD2
OPEN
JMP_LDCA_PD2
JMP_LDAN_PD2
GND
C4
C1
.1u
.01u
C2
GND
.01u
C3
GND
D15
D14
D13
D12
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
D1
H1
MOUNT-HOLE3.6
H2
MOUNT-HOLE3.6
H3
MOUNT-HOLE3.6
H4
MOUNT-HOLE3.6
L2
Short
L1
10u
L3
10u
L4
10u
LD_BYPASS
10k
A
E
S
D16
1N5711
TO_PD1
TO_PD2
TO_LDCA
TO_LDAN
C6
<50p
2k
R1
82
R2
C5
39p
CAP_MOD
R_MOD


LD_AN
LD_CA
PD_AN
PD_CA
BOARD_GND
JMP_LDAN_PD2
PD1
PD2
JMP_PD2
JMP_LDCA_PD2
C1
C4
C2
C3
D15
D14
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
J1
J2
J3
J4
DiodeProtectionRv02
LD_BYPASS
D16
TO_LDAN
TO_LDCA
TO_PD1
TO_PD2
C6
CAP_MOD
R1
R2
C5
R_MOD
GND

A.1.5. ECDL Temperature Controller

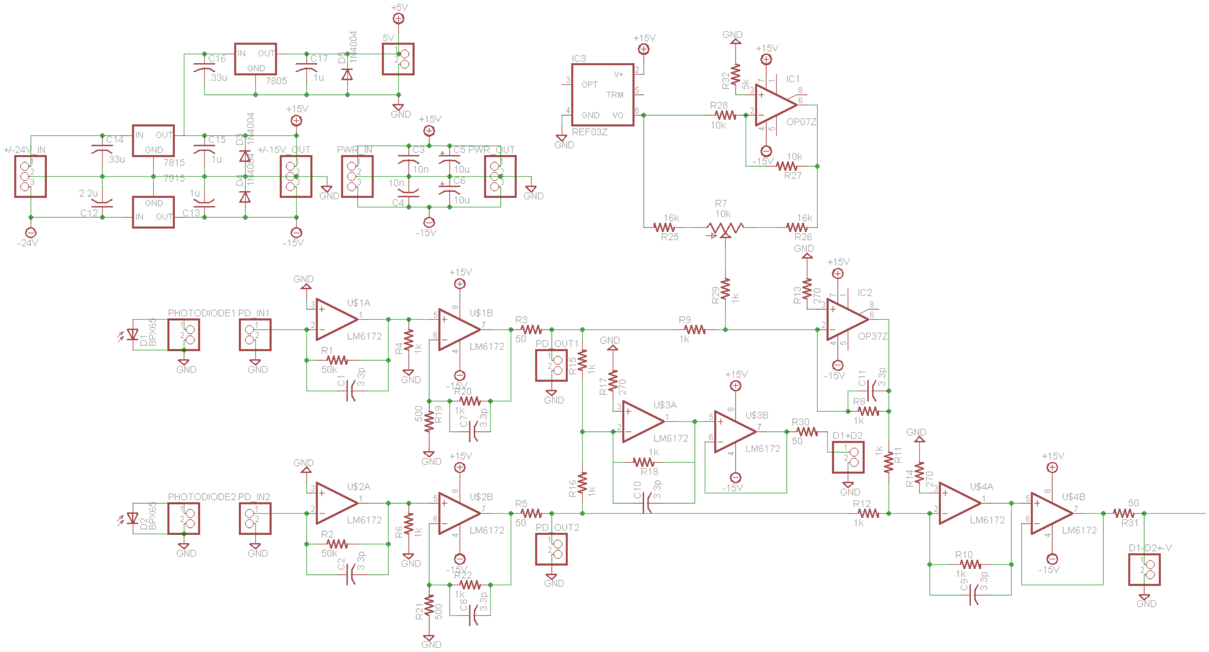
The thermistor is a 10 kΩ device. R_{box} is typically a 100 kΩ potentiometer in series with a 100 kΩ resistor. Typically $ER_{GAIN} = 25.04$ kΩ, $P_{GAIN} = 65.87$ kΩ, and $I_{GAIN} = 18.81$ kΩ. Q2, Q3, IC3, and IC6 are mounted to the case for heatsinking. R29, R14, R15, and U$24 form a invertor. This part of the circuit is not required if the circuit will be used on a diode laser system with TEC. In this case, do not install the resistors. In addition, short U$23 pin 8 and U$24 pin 14 with a jumper wire using the pads of R29 and R14 for connections. Also, physically cut pin 12, 13, and 14 on U$2, i.e., do not let those pins connect to the board. The circuit is called TC6.




D8
IC1
IC5
IC4
HEAT
IGAIN
R40
OUTPUT
R15
ERROR
TC6
RSENSE
R29
R39
R22
C15
R14
R23
R46
R47
C28
C30
R10
R31
R32
R33
Q1
C27
C22
C19
R12
INTER
R25
D3
D2
C31
K1
INT EN

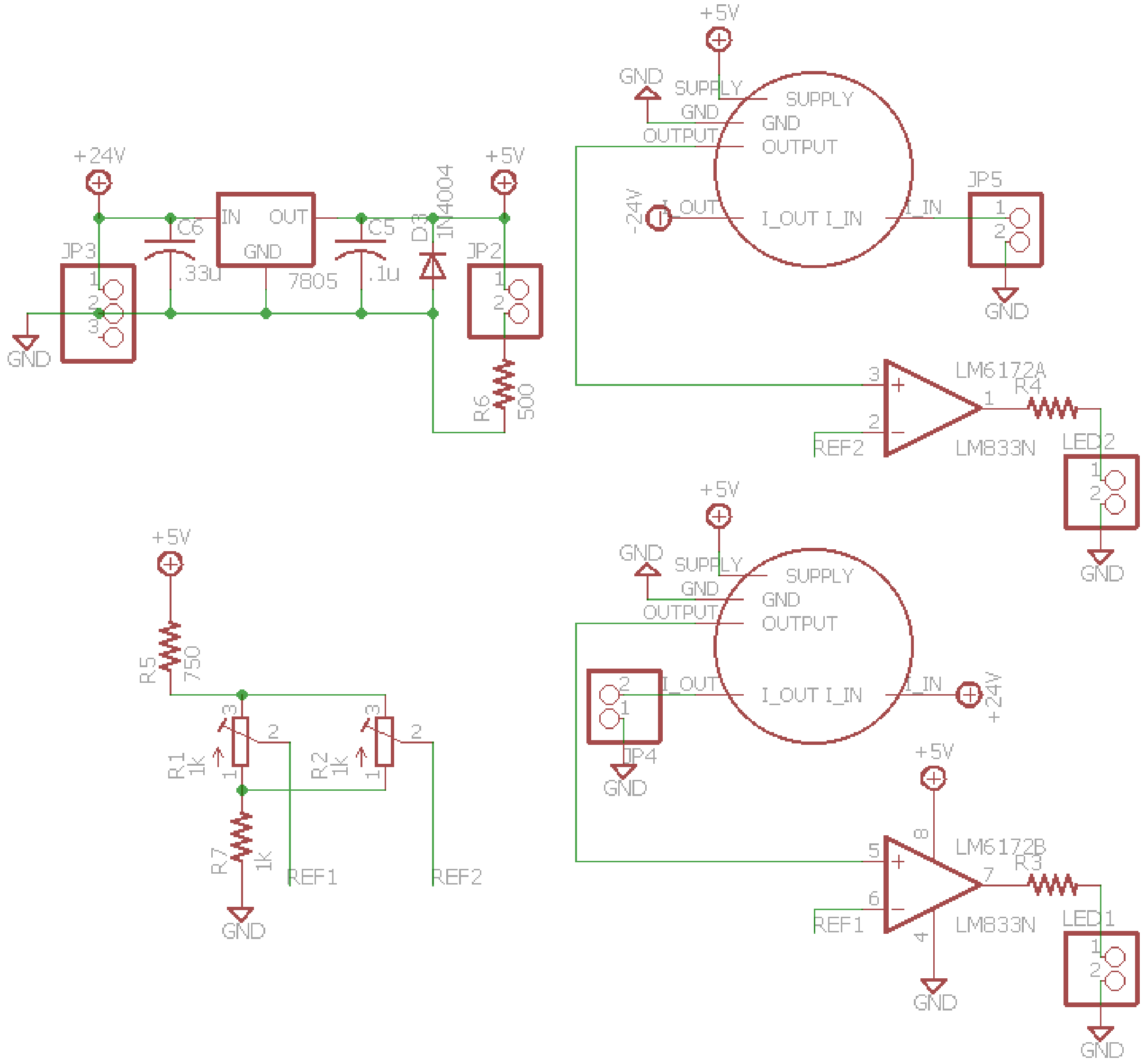
A.1.6. Polarization Spectroscopy

The circuit below generates the transmission and polarization spectroscopy signal for the Rb reference laser project. This is version 4.



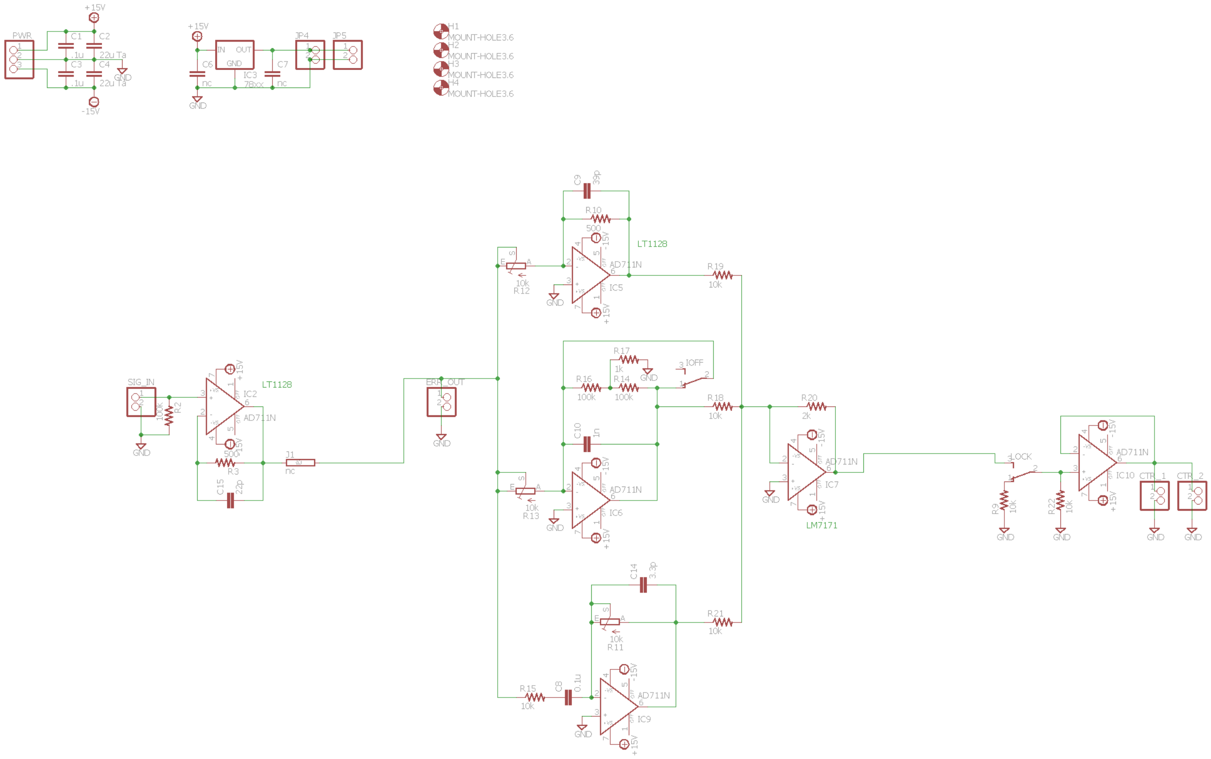
A.1.7. PowerOne Supply Current Overload Monitor

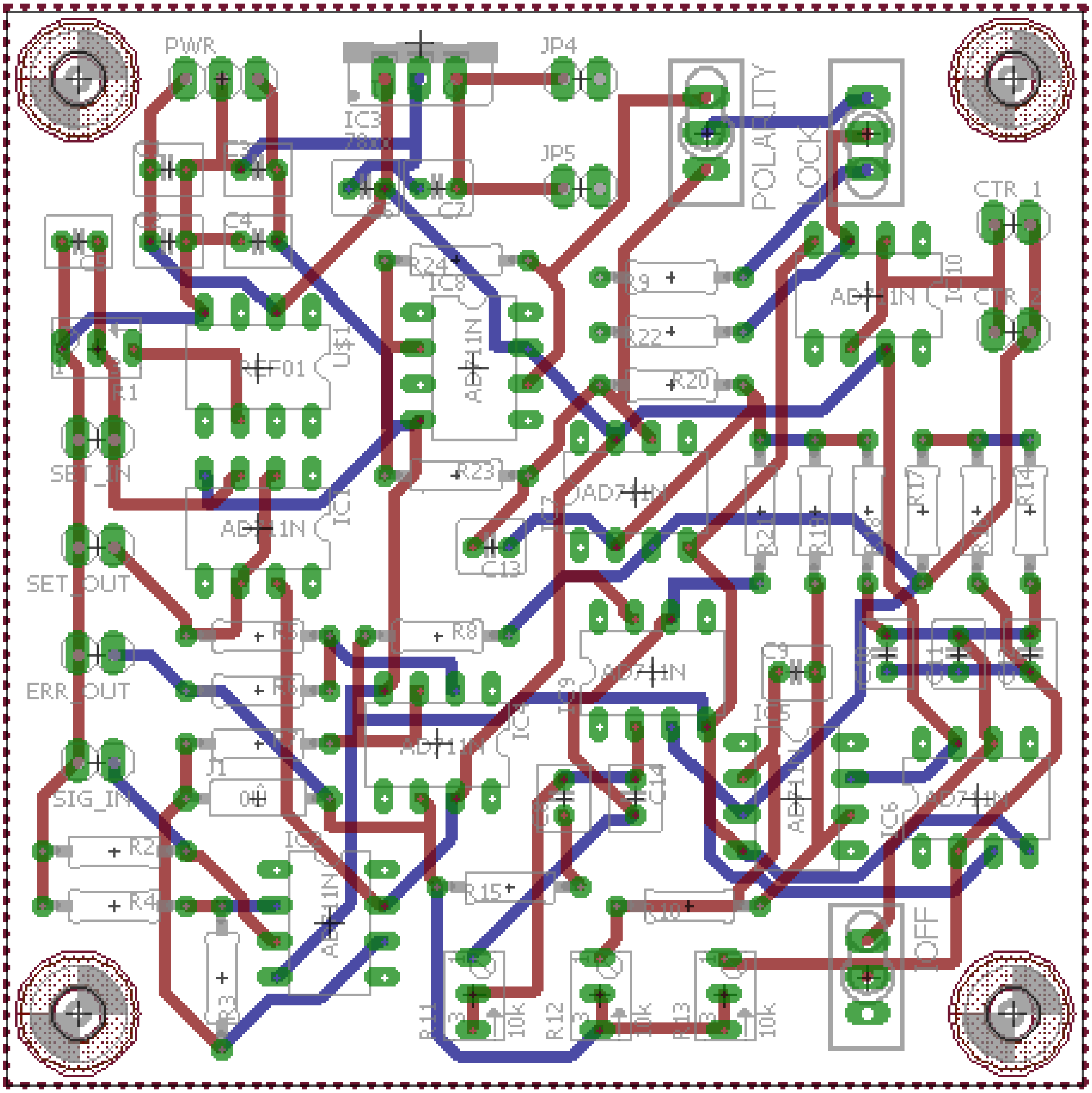
Current overload monitor that is used on the Rb Reference laser PowerOne supply.



A.1.8. Reference Laser PI Servo

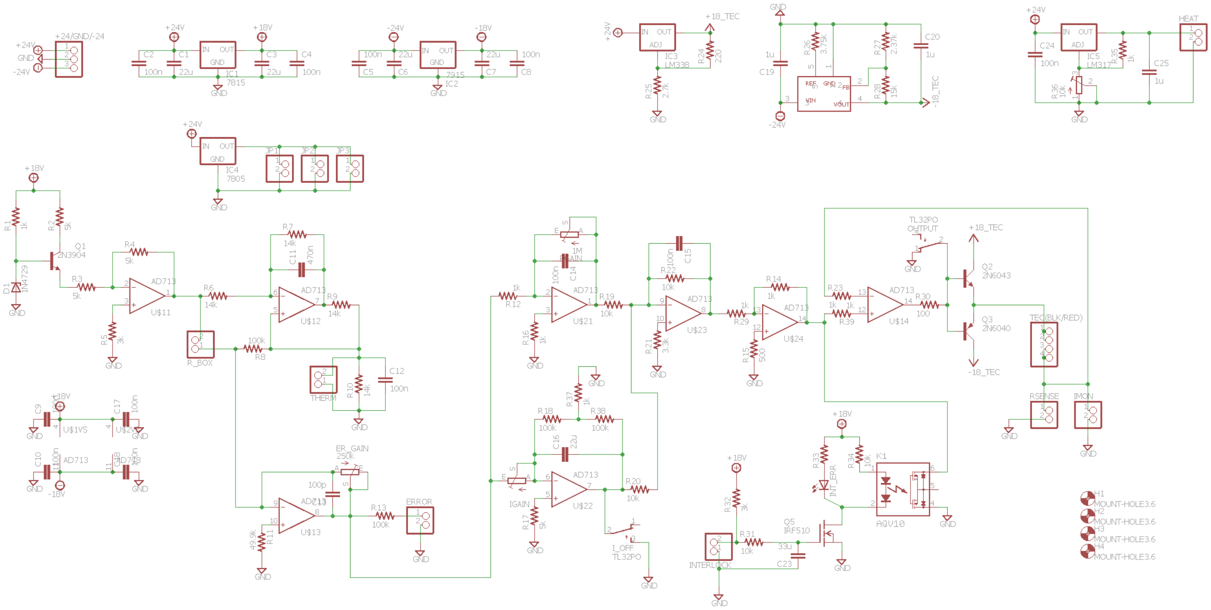
PI servo loop for locking the Rb reference laser. Component values and op-amps likely differ in the actual PI servo circuit. This is modified from the circuit called PID_v20.

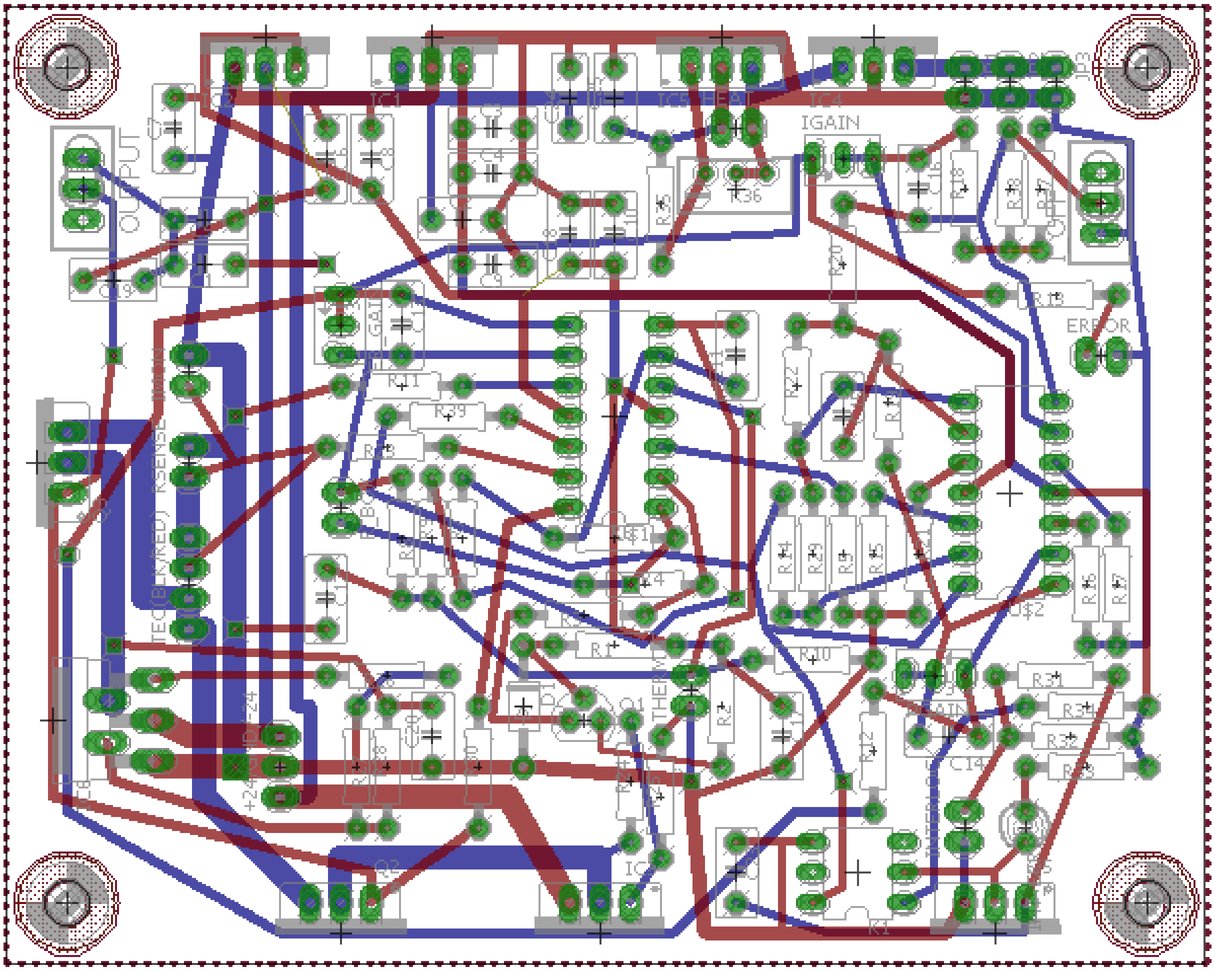



PWR
IC3
JP4
JP5
POLARITY
LOCK
CTR_1
CTR_2
C3
C4
C5
C7
R24
IC8
AD711N
U$1
REF01
R1
SET_IN
SET_OUT
ERR_OUT
SIG_IN
R2
R4
R3
J1
0Ω
R5
R6
R7
R8
R9
R22
R20
R23
C13
AD711N
AD711N
AD711N
AD711N
AD741N
AD741N
AD741N
IC1
IC2
IC4
IC5
IC6
IC7
IC9
IC10
R21
R19
R18
R17
R14
C14
R15
R10
R11
R12
R13
10k
10k
10k
OFF

A.1.9. SHG Module Temperature Controller

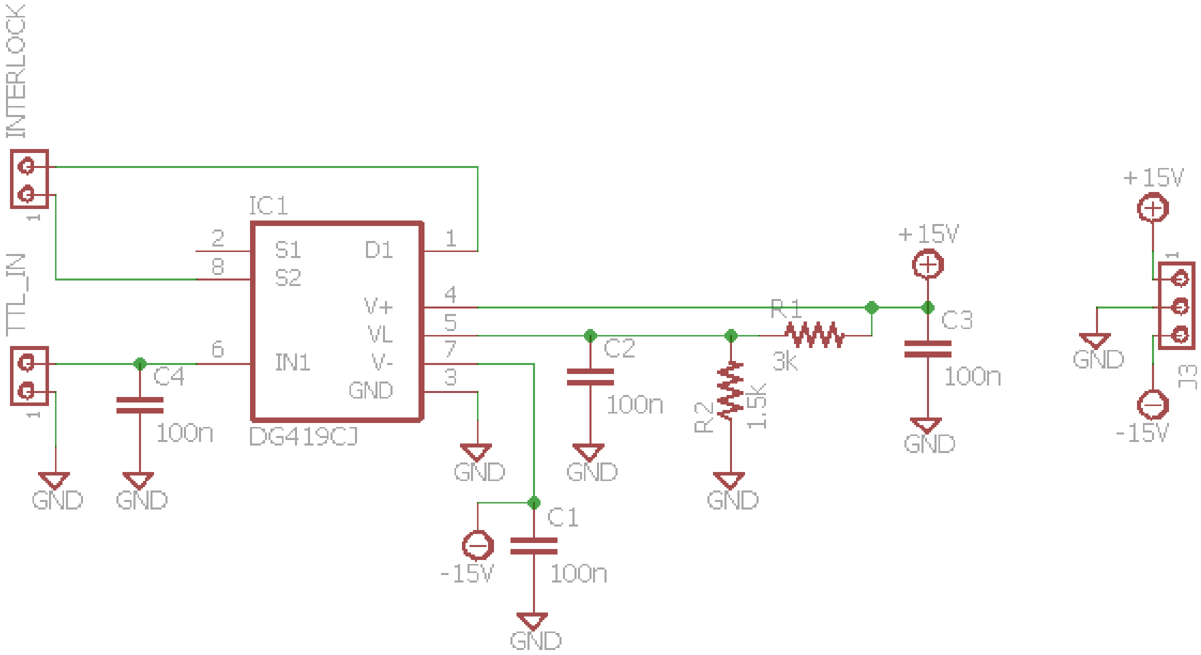
The SHG temperature controller uses TC4, an old version of the ECDL temperature controller for the current-limit feature of the LT1185CT regulator (unlabeled -18 V TEC regulator). The LM338, LM317, and Q2 are omitted. The LT1185CT regulator and Q3 are case mounted for heatsinking. This is modified from the circuit called TC4, and is called TC4SHG.




OUTPUT
IC1
IC5
HEAT
IC4
IGAIN
R36
R13
ERROR
R11
R39
RSENSE
TEC(BLK/RED)
U$1
U$2
R10
THERM
Q1
Q2
R31
R32
R33
R34
C14
K1

A.1.10. TTL Interlock Switch

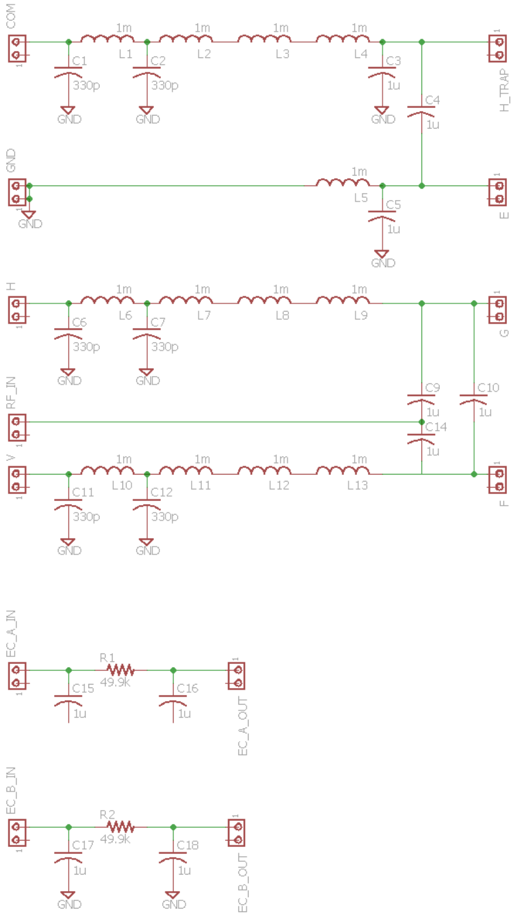
This circuit uses a DG419 switch in order to use the interlock input on the ECDL controller rear panel to switch the laser controller output on and off with a TTL (or other logic-level) input signal.



A.2. AlH$^+$ Rotational Cooling Project

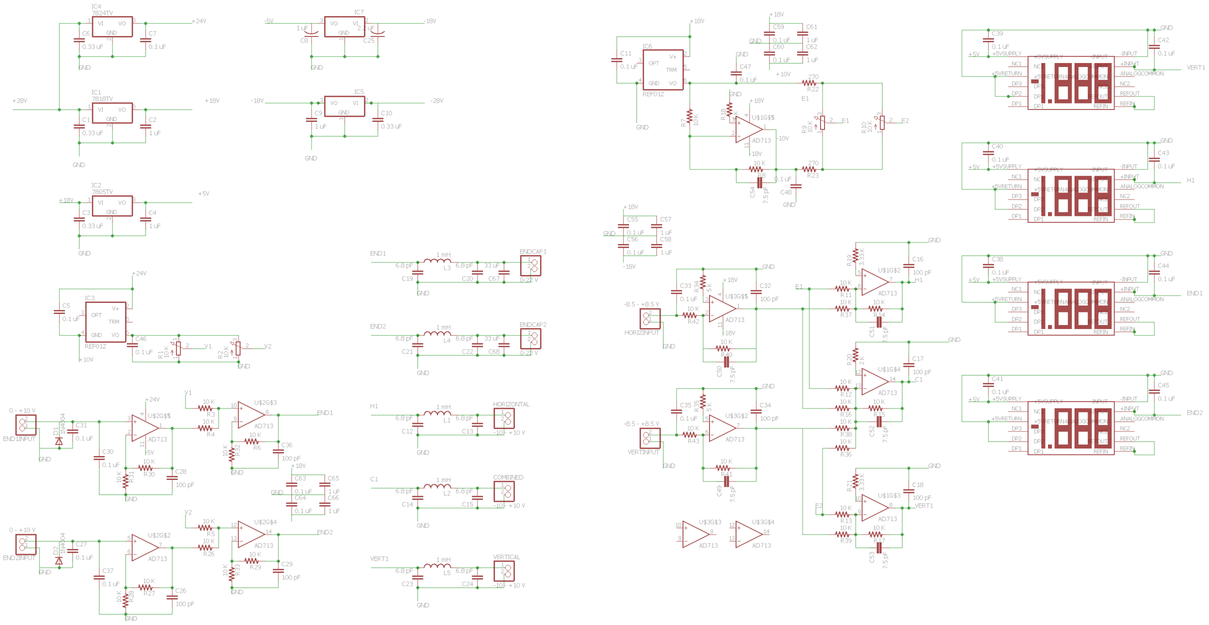
A.2.1. AlH$^+$ Filter Box

This circuit is used for RF and TOFMS switching voltage pickup filtering in addition to combining the DC compensation voltages with the applied RF trap drive voltage in the AlH$^+$ apparatus. All 1 μF capacitors are AVX SV09AC105KAA rated to 1 kV. The circuit is called Marie Box.



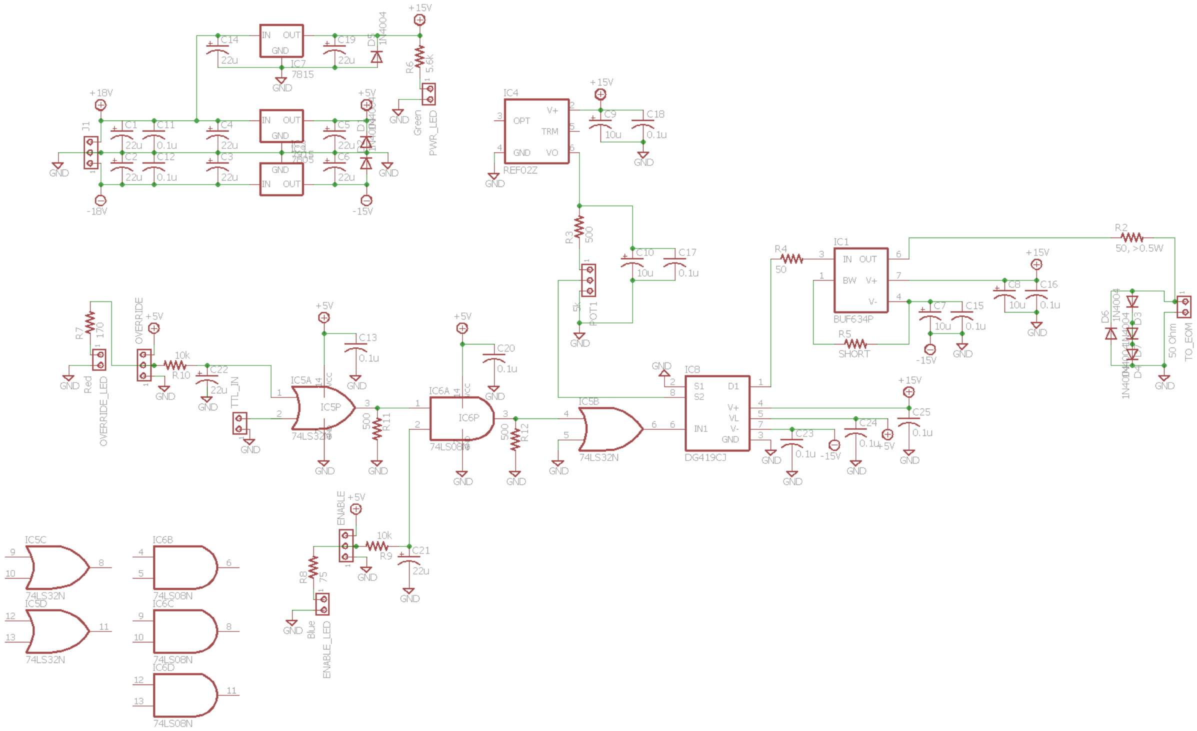
A.2.2. DC Trap Supply

This circuit provides compensation voltages for the AlH$^+$ ion trap, set with potentiometers and/or analog control voltages along the horizontal and vertical directions by summing the two voltages onto one of the grounded RF trap rods. Front panel displays show the set output voltages for each channel. This is later replaced directly with NI AO channels.



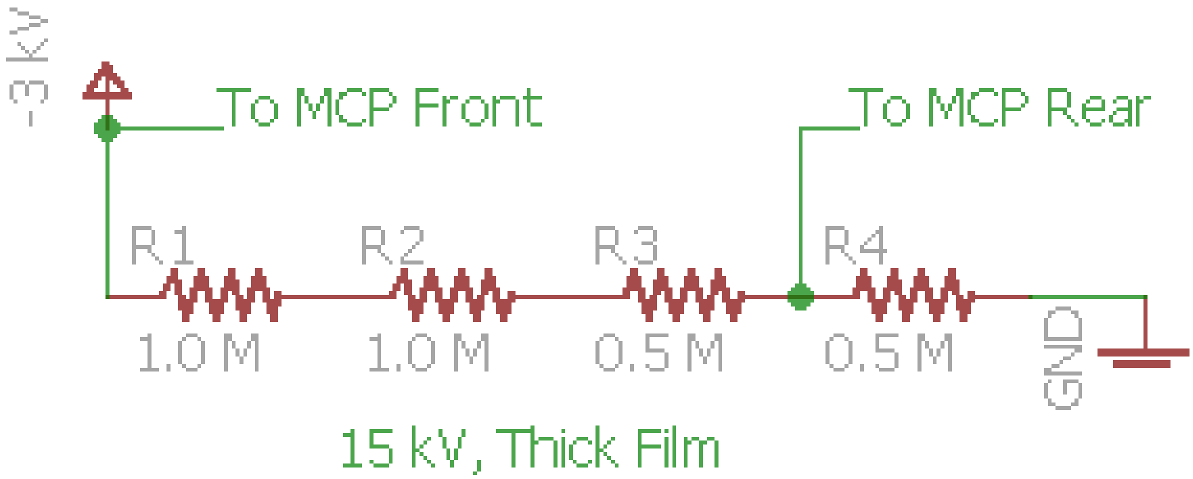
A.2.3. EOM Logic Level Shifter/Driver

This circuit shifts a TTL-logic signal into a square-pulse analog-amplitude signal for the ConOptics 302A EOM driver/amplifier. Note that the input and output loads on the BUF634 chip are carefully chosen to minimize the rise time and ringing in the circuit when switching. Additionally, the circuit includes an enable and manual override switch on the front panel for user convenience. The circuit is called EOM TTL to DC Switch.



A.2.4. MCP Voltage Divider

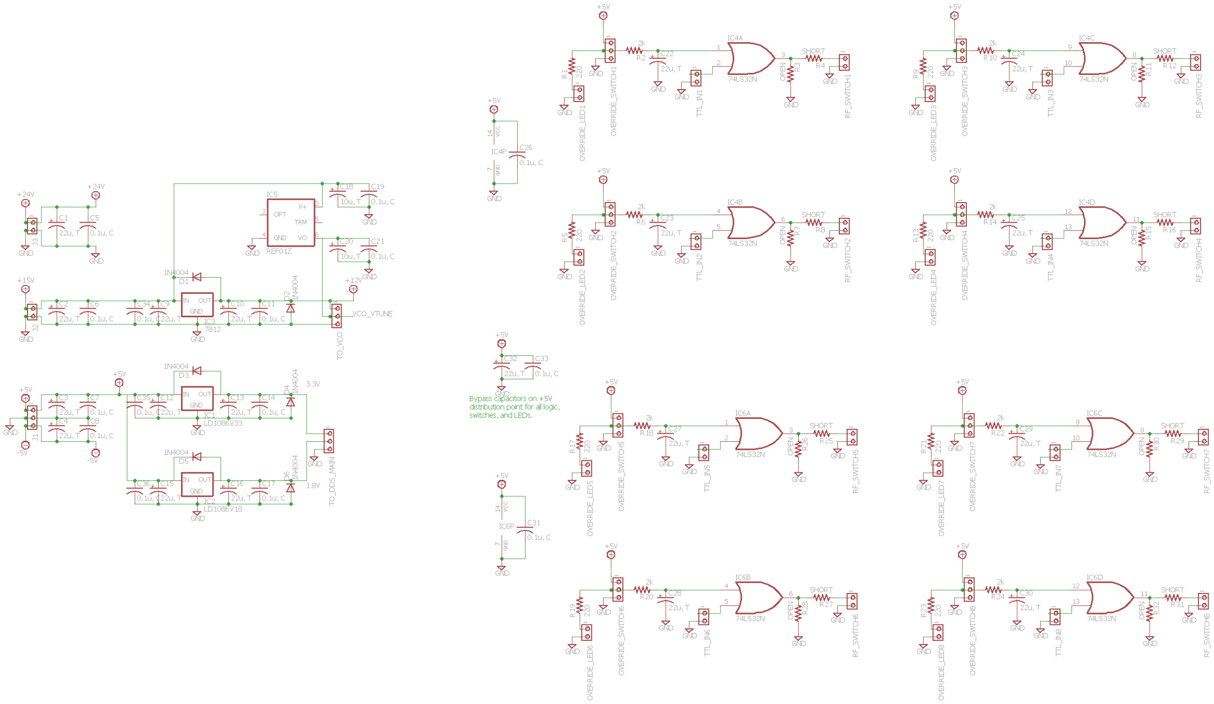
The MCP voltage divider circuit is used to supply a -0.5 kV anode bias to the MCP while maintaining 2.5 kV across the device. In this configuration, the MCP front surface is held at -3 kV and rear surface at -500 V with the anode connected through 50 Ω to ground. The 1.0 MΩ resistors are TE Connectivity HB31M0FZRE. The 0.5 MΩ resistors are Ohmite SM108035003FE. All resistors are rated to ≥ 15 kV and potted in Resin Technology Group KONA 870 FTLV-DP thermal epoxy. The circuit is called MCP Voltage Divider.



A.3. Motional Ground State Cooling Project

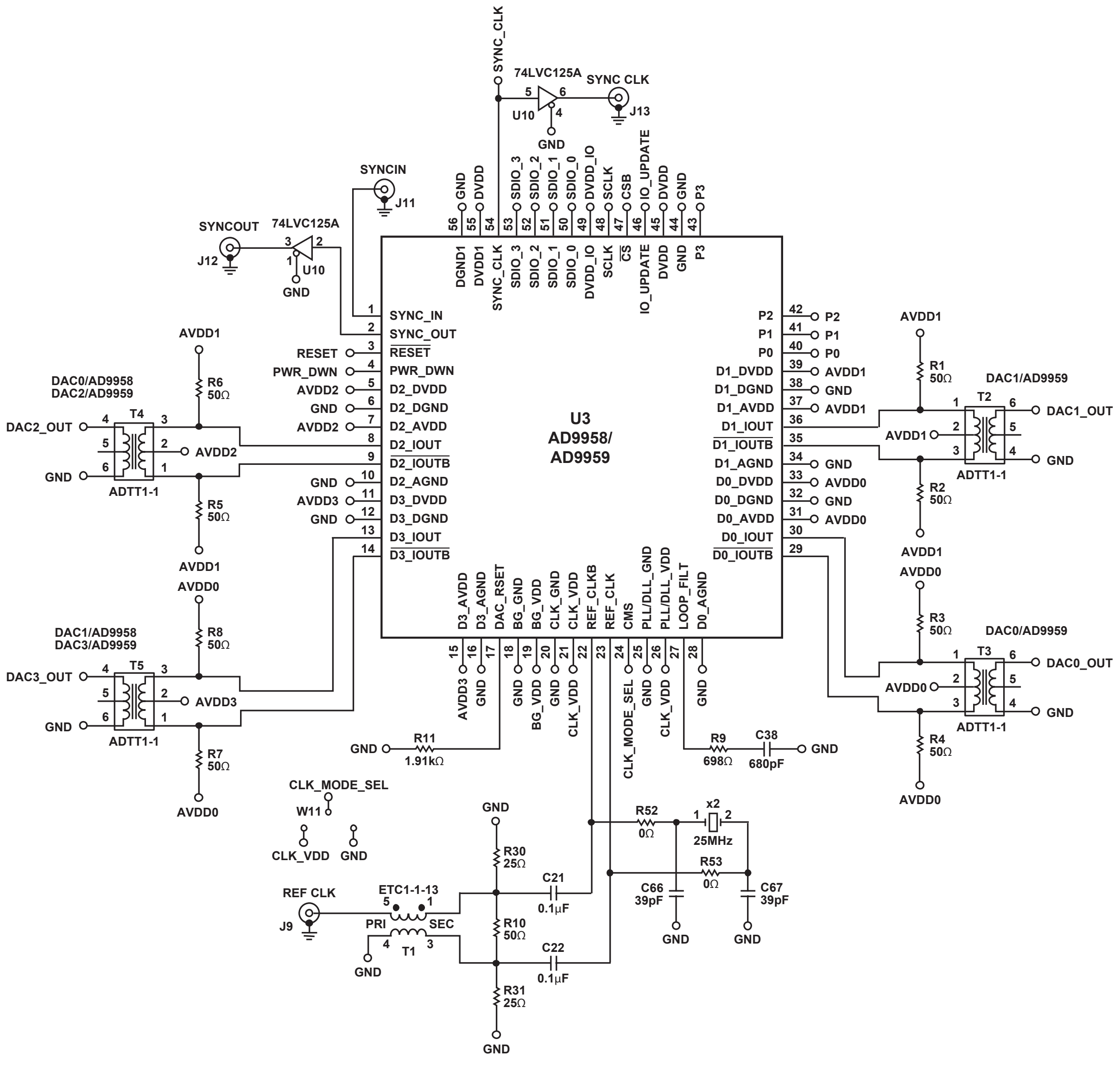
A.3.1. DDS AOM System Support Electronics

This circuit provides logic buffering and a manual override of each AOM channel, DDS supply voltages, and VCO supply and control voltages. Note that the +3.3 V regulator is case mounted for heatsinking. The circuit is called DDS AOM Switch Logic and Regulation v4.



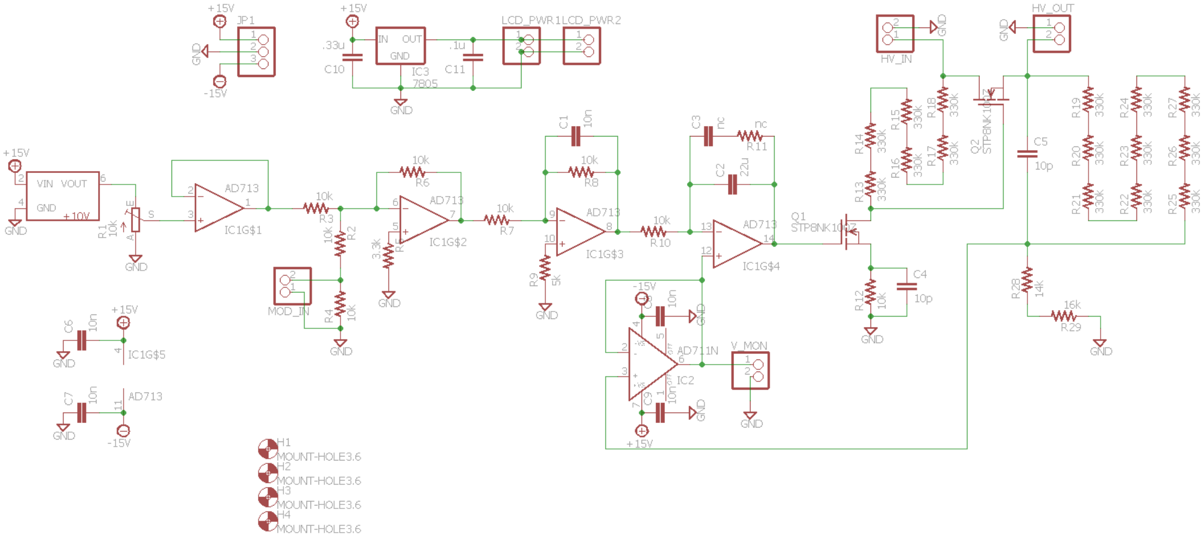
A.3.2. DDS Evaluation Board

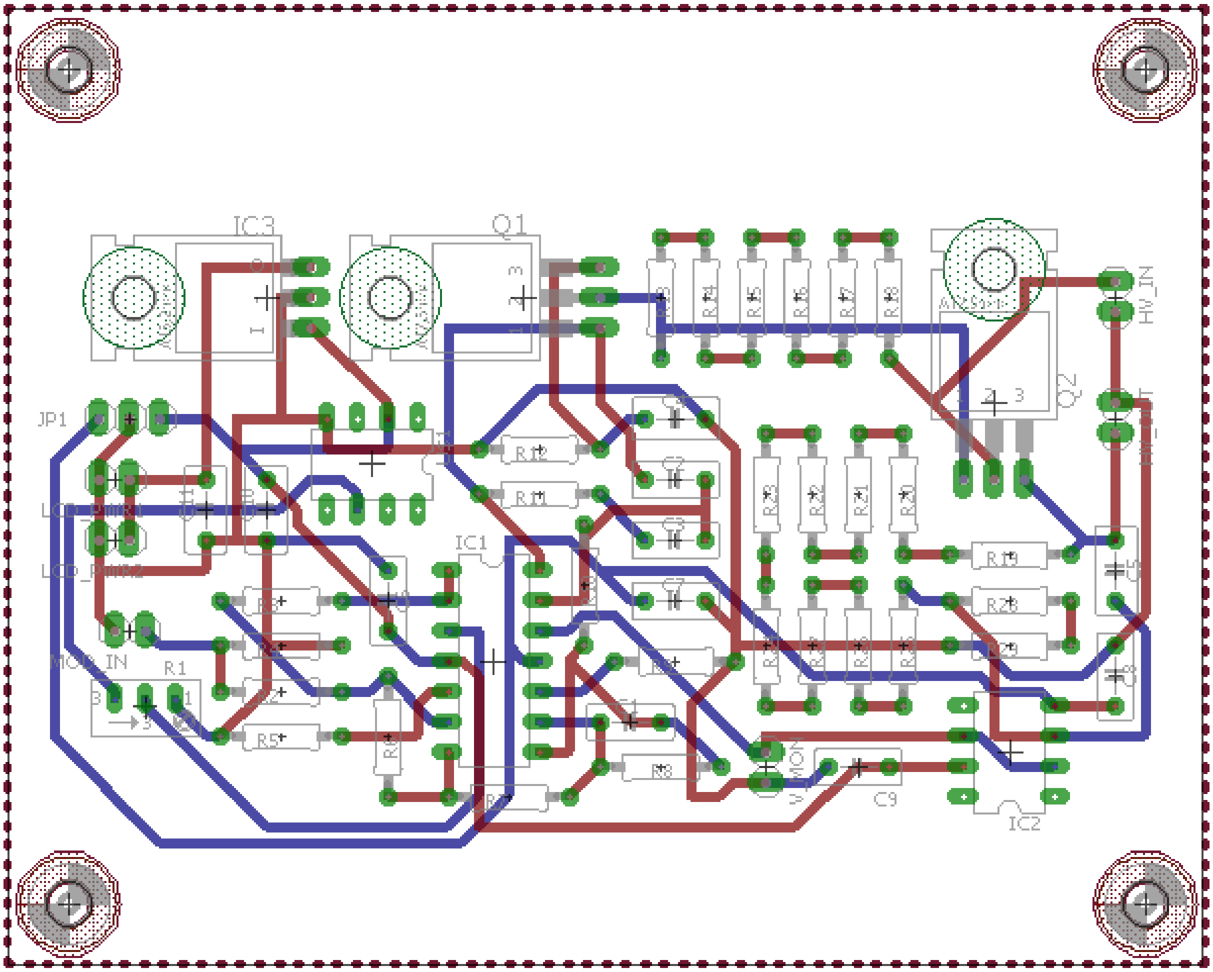
A simplified schematic of the AD9959 evaluation board. Original sourced from Analog Device's datasheet in case the schematic is revised in the future. Note that all modifications to run on a crystal oscillator clock are located on the lower portion of the schematic.



A.3.3. High Voltage Distributor

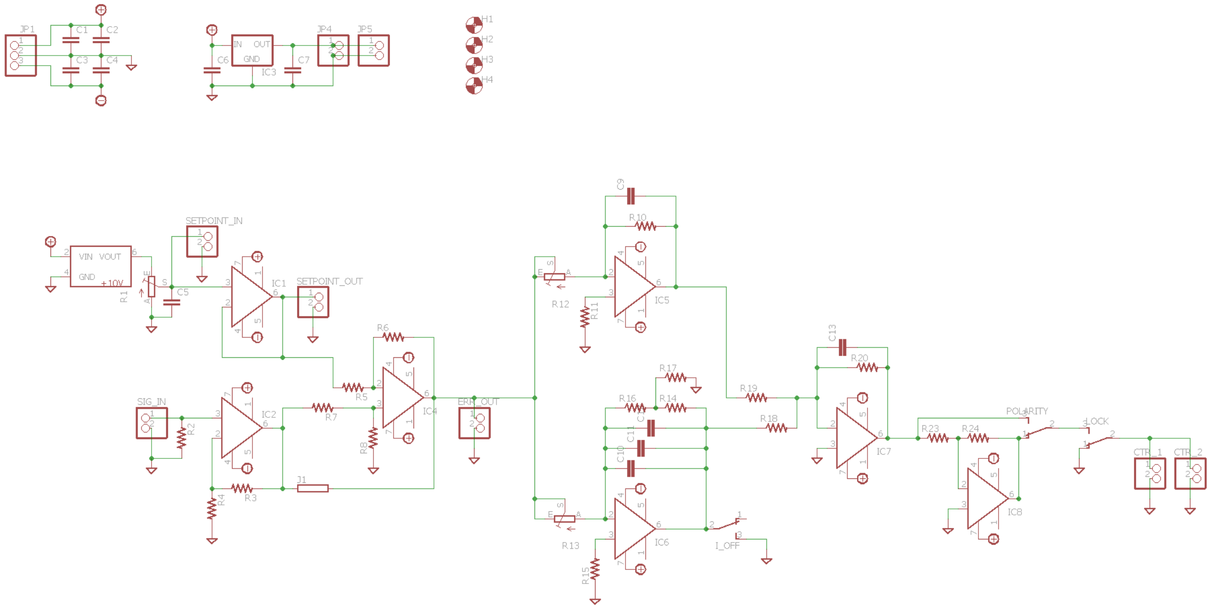
High voltage amplifier/distributor circuit that outputs an input voltage with 40x gain. The HV source is external and must be provided. The HV regulator circuit is used to supply endcap and compensation voltages on the single-ion experiment. For the experiment, the voltage reference and buffer subsections are not installed. Also note that the MOSFETs are replaced over failures with STP3N150 or STP8NK100Z MOSFETs for additional circuit longevity. Original circuit design by YWL and is called HV Regulator Circuit v10.

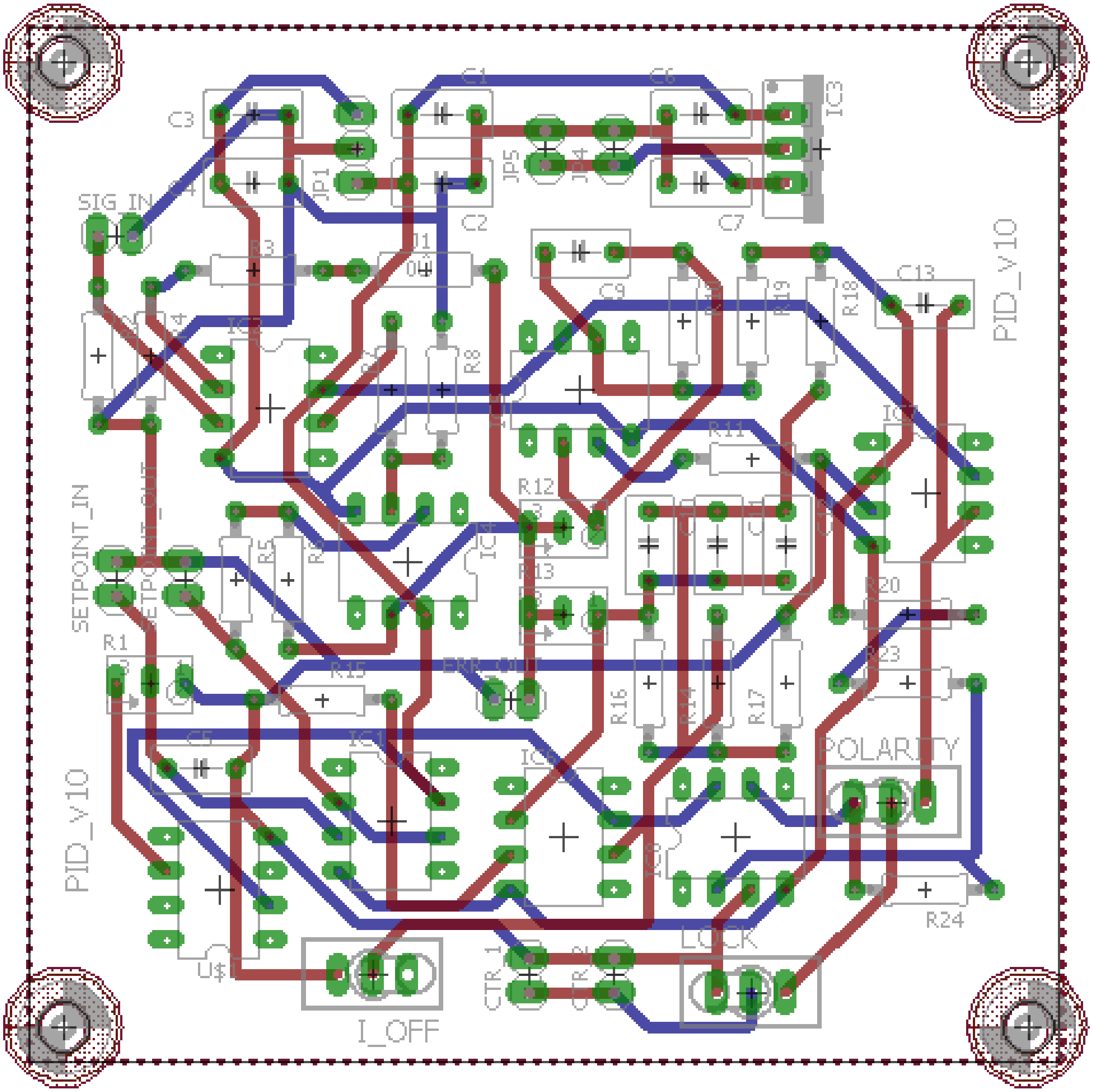



IC3
Q1
JP1
LCD_PWR1
LCD_PWR2
MOD_IN
R1
R3
R4
R2
R5
R6
IC1
R12
R11
R7
R8
C1
R9
R10
C7
R13
R14
R15
R16
R17
R18
Q2
HV_IN
HV_OUT
R23
R22
R21
R20
R19
R28
R29
R24
R25
R26
R27
C5
C6
V_MON
C9
IC2

A.3.4. Laser Power Control PI Servo

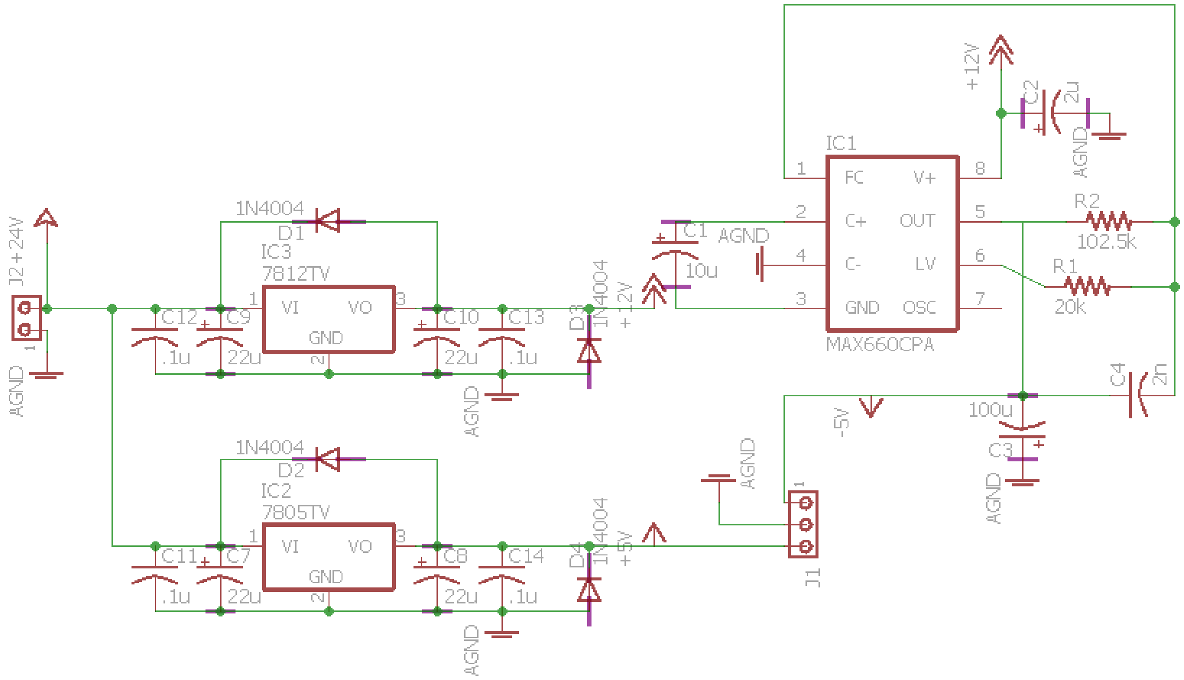
PI servo loop for laser power control on the single-ion apparatus. The set point is set via PC, so the voltage reference and associated potentiometer are omitted. The input signal is from the TIA outputs of App. A.3.6. Because the control output is sent directly to the 50 Ω load of an IntraAction AOM driver, a voltage-clamping Zener diode and BUF634 buffer chip is added to prevent current saturation of the last operational amplifier in the circuit. Component values specifying the various gains on the circuit may be different than in the diagram. Original circuit and board design by YWL. Modifications are designed by CMS. The circuit is called PID v10.




C3
C1
C6
IC3
JP5
JP4
JP1
C4
C2
C7
SIG_IN
R3
C9
R19
R18
C13
PID_V10
R8
R11
R12
IC4
R5
SETPOINT_IN
SETPOINT_OUT
R13
R20
R1
R15
ERR_OUT
R16
R14
R17
R23
C5
IC1
POLARITY
PID_V10
R24
LOCK
CTR_1
I_OFF

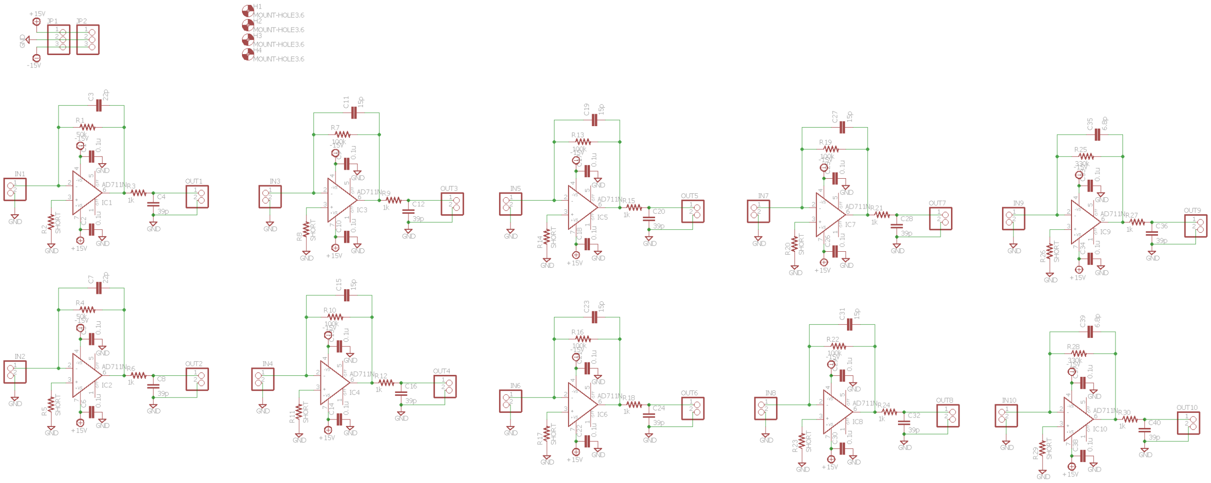
A.3.5. Single-pole RF Switch Driver

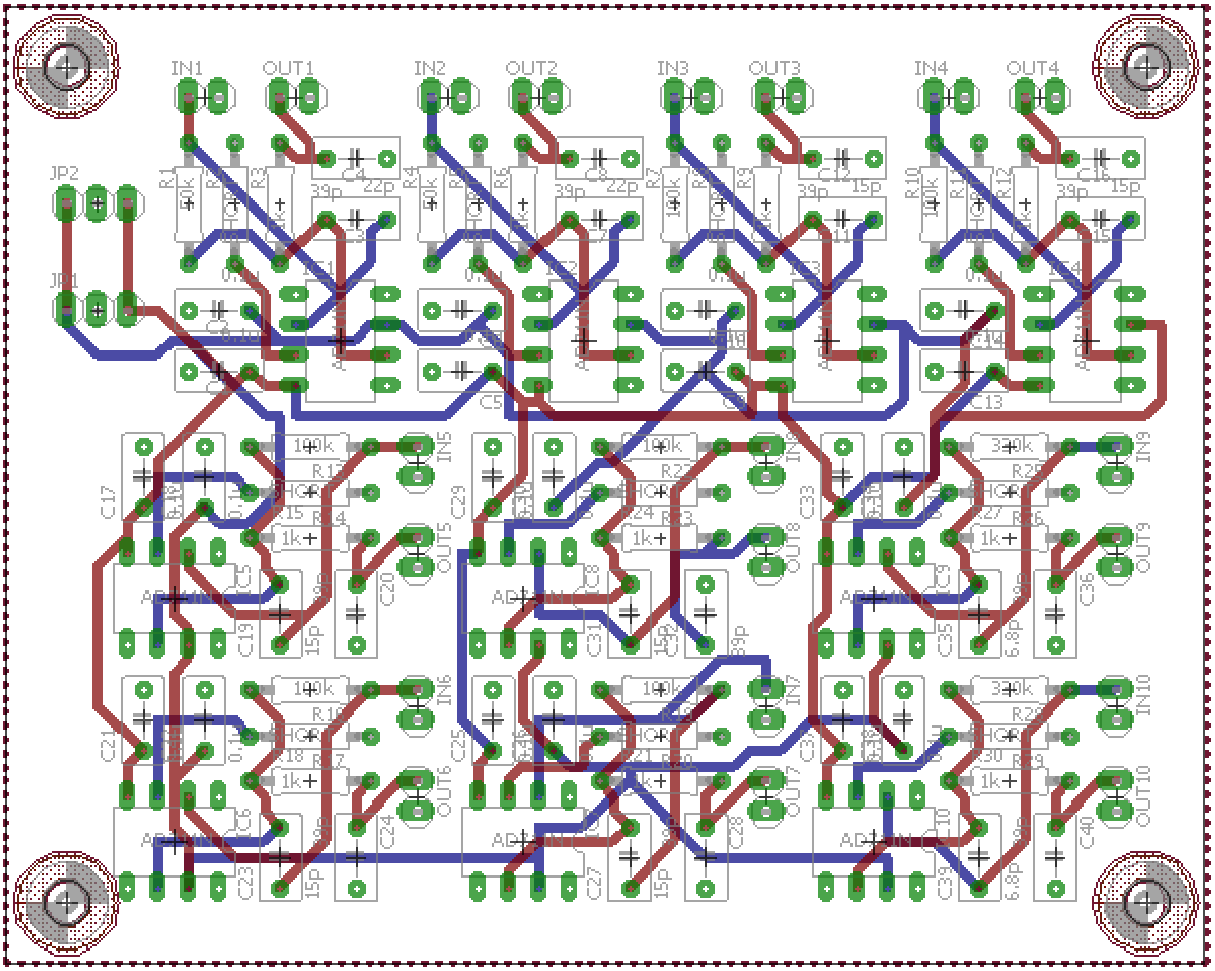
±5V supply to power Minicircuits ZYSWA-50-DR RF switches from a single-polarity supply such as that found in the IntraAction AOM drivers. The circuit is called ±5 V supply for IntraAction AOM Drivers.



A.3.6. Transimpedance Amplifiers

Transimpedance amplifiers with various gains on a single circuit board. The values for the feedback capacitors are calculated according to Sec. 3.3.1.1 estimating the total load capacitance to be < 0.450 nF. Original circuit and board design by YWL and component values modified by CMS. The circuit is called Photodiode Transimpedance Amplifier v10-1.

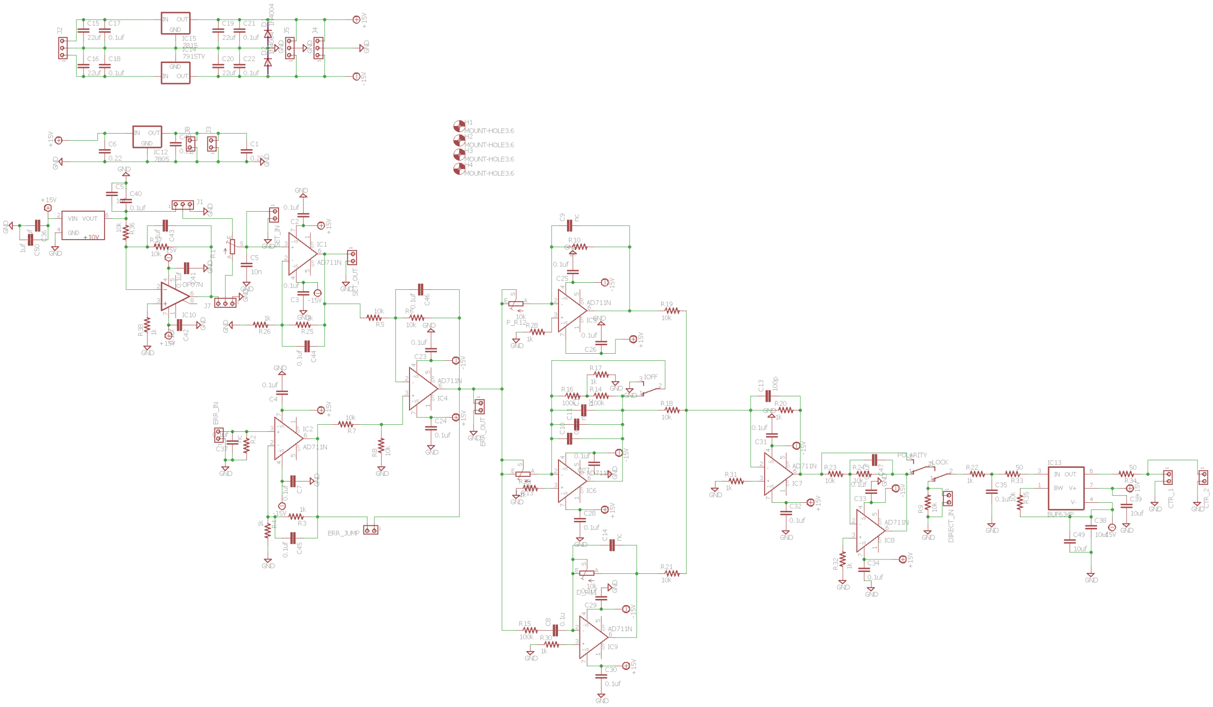


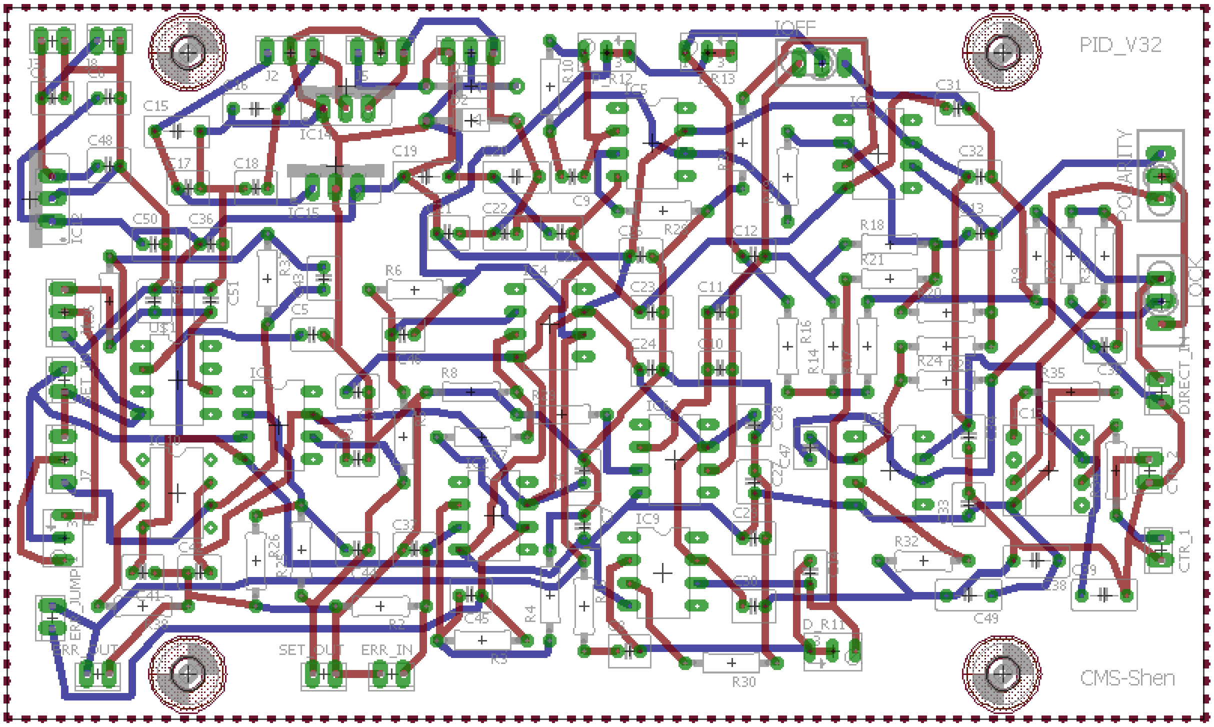

IN1
OUT1
IN2
OUT2
IN3
OUT3
IN4
OUT4
JP2
JP1
R1
R3
R4
R6
R7
R9
R10
R12
50k
100k
1k
C4
C8
C12
C16
39p
22p
15p
C5
C13
100k
330k
R15
R18
R24
R27
R30
1k
IN5
IN6
IN7
IN8
IN9
IN10
OUT5
OUT6
OUT7
OUT8
OUT9
OUT10
C17
C19
C20
C21
C23
C24
C25
C27
C28
C29
C31
C33
C35
C36
C39
C40
0.1u
15p
6.8p
AD

A.4. Miscellaneous

A.4.1. General Purpose Control PID Servo

General purpose PID servo loop circuit. Design by CMS. Layout by PRS. This circuit is called PID v32.




PID_V32
IOFF
POLARITY
LOCK
DIRECT_IN
CTR_1
ERR_JUMP
ERR_OUT
SET_OUT
ERR_IN
CMS-Shen
J2
IC14
IC15
IC9
C15
C16
C17
C18
C19
C22
C9
C23
C24
C11
C12
C10
C31
C32
C36
C48
C49
C50
C51
C45
C38
C39
R6
R8
R10
R12
R13
R14
R16
R18
R21
R24
R26
R30
R32
R35
R3
R4
D_R11

APPENDIX B

LEVEL Input Files

LEVEL [284] is used to calculate molecular rovibrational states and transitions between them. The code below are input files used to generate data used for BaF^+, ICl^+, IF^+, and AlH^+.

B.1. BaF^+ Ground State Input File

The input file below uses the ground state potential energy surface and permanent dipole moment as calculated by MZI to generate data used for vibrational overtone transition investigations for REMPD depletion rotational state preparation.

```
 56 138 9 19 1   1          %1 IAN1 IMN1 IAN2 IMN2 CHARGE NUMPOT
'BaF+ GS, Marcin new potentials, PQR branches, high v between-level transitions
   '
 0.0015 0.5 40.0  1.d-6          %4 RH RMIN RMAX EPS
 201  0  1 51000      %5 NTP LPPOT IOMEG VLIM
 4 2  4  5  0                    %6 NUSE IR2 ILR NCN CNN
 1.D0  1 0.d0      %7 RFACT EFACT VSHIFT & B-state turning points
1.4     100210.8167
1.428   92856.79324
1.456   84662.22302
1.484   75953.98532
1.512   67058.95946
1.54    58304.02472
1.568   50016.06041
1.596   42521.94584
1.624   36092.41549
1.652   30651.71768
1.68    25992.05647
1.708   21906.16392
```

1.736	18255.42284
1.764	15022.16737
1.792	12201.04344
1.82	9784.121096
1.848	7738.16439
1.876	6015.573999
1.904	4568.601057
1.932	3357.280352
1.96	2362.071106
1.988	1566.742124
2.016	955.0228689
2.044	509.9819847
2.072	214.1397885
2.1	50
2.128	1.226091797
2.156	56.12054908
2.184	204.1456101
2.212	434.8010683
2.24	738.827432
2.268	1108.460466
2.296	1536.024953
2.324	2014.104647
2.352	2536.881464
2.38	3099.146374
2.408	3695.697675
2.436	4321.867498
2.464	4973.928504
2.492	5648.249093
2.52	6341.266717
2.548	7050.097198
2.576	7772.241391
2.604	8505.204976
2.632	9246.697497
2.66	9994.963446
2.688	10748.334
2.716	11505.15772
2.744	12264.0752
2.772	13023.96935
2.8	13783.73041
2.828	14542.32025
2.856	15298.98735

2.884	16053.05179
2.912	16803.83705
2.94	17550.77744
2.968	18293.44094
2.996	19031.40345
3.024	19764.26313
3.052	20491.75543
3.08	21213.66813
3.108	21929.7895
3.136	22639.94328
3.164	23344.01566
3.192	24041.89918
3.22	24733.49301
3.248	25418.76122
3.276	26097.70472
3.304	26770.32485
3.332	27436.62221
3.36	28096.59542
3.388	28750.2428
3.416	29397.56226
3.444	30038.54483
3.472	30673.17585
3.5	31301.44049
3.528	31923.32027
3.556	32538.78215
3.584	33147.78945
3.612	33750.30476
3.64	34346.26737
3.668	34935.58847
3.696	35518.17757
3.724	36093.94384
3.752	36662.79441
3.78	37224.63567
3.808	37779.37373
3.836	38326.89503
3.864	38867.05125
3.892	39399.69057
3.92	39924.65889
3.948	40441.7801
3.976	40950.86556
4.004	41451.72641

4.032	41944.12715
4.06	42427.70989
4.088	42902.09693
4.116	43366.90105
4.144	43821.57563
4.172	44265.44174
4.2	44697.81644
4.228	45117.95667
4.256	45524.87888
4.284	45917.53936
4.312	46294.8944
4.34	46655.89924
4.368	46999.50789
4.396	47324.67428
4.424	47630.43836
4.452	47916.37085
4.48	48182.24478
4.508	48427.83772
4.536	48653.28927
4.564	48859.37686
4.592	49046.94284
4.62	49216.85635
4.648	49370.24955
4.676	49508.4039
4.704	49632.60252
4.732	49744.09679
4.76	49844.05489
4.788	49933.63149
4.816	50013.97033
4.844	50086.03106
4.872	50150.62062
4.9	50208.54131
4.928	50260.55576
4.956	50307.26788
4.984	50349.24189
5.012	50387.03909
5.04	50421.12411
5.068	50451.84504
5.096	50479.54308
5.124	50504.54938
5.152	50527.13333

5.18	50547.54072
5.208	50566.01694
5.236	50582.77178
5.264	50597.95238
5.292	50611.69948
5.32	50624.1527
5.348	50635.44027
5.376	50645.68402
5.404	50655.00564
5.432	50663.50966
5.46	50671.25561
5.488	50678.29567
5.516	50684.68198
5.544	50690.46511
5.572	50695.69438
5.6	50700.41908
5.628	50704.68991
5.656	50708.56341
5.684	50712.09755
5.712	50715.34933
5.74	50718.34392
5.768	50721.0681
5.796	50723.50636
5.824	50725.65025
5.852	50727.5349
5.88	50729.212
5.908	50730.73312
5.936	50732.13494
5.964	50733.42782
5.992	50734.61948
6.02	50735.71661
6.048	50736.71603
6.076	50737.60896
6.104	50738.38653
6.132	50739.04321
6.16	50739.5822
6.188	50740.0081
6.216	50740.32605
6.244	50740.55009
6.272	50740.70165
6.3	50740.80241

```
6.328  50740.8696
6.356  50740.90282
6.384  50740.89727
6.412  50740.84829
6.44   50740.75653
6.468  50740.62903
6.496  50740.4732
6.524  50740.29551
6.552  50740.09674
6.58   50739.87547
6.608  50739.63029
6.636  50739.36265
6.664  50739.07902
6.692  50738.78634
6.72   50738.49091
6.748  50738.19238
6.776  50737.88662
6.804  50737.56948
6.832  50737.23887
6.86   50736.8981
6.888  50736.55134
6.916  50736.2028
6.944  50735.85664
6.972  50735.51706
7      50735.18825
-18  1  2  -4  2 1  -1 0    %17 NLEV1 AUTO1 LCDC LXPCT NJM JDJR IWR LPRWF
  0 0                       %18 IV(1) IJ(1)
 1  20   1.D0               %19 MORDR IRFN RREF
55 0                        %21 NRFN RFLIM
4 0  4  5 0                 %22  NUSEF IRF ILRF NCNF CNNF
1.D0 1                      %23 RFACTF, MFACTF
1.5    -1.20184157821967
1.6    -2.29199857163190
1.7    -3.32451699808873
1.8    -4.30114261823431
1.9    -5.21855587410028
2      -6.05795941604652
2.1    -6.83908142206498
2.2    -7.57022841181733
2.3    -8.25973392196332
2.4    -8.91368152982799
```

2.5	-9.53653791021105
2.6	-10.13175153003130
2.7	-10.70188356512550
2.8	-11.24851109145620
2.9	-11.77193274470330
3	-12.27156952746940
3.1	-12.74543890549390
3.2	-13.19035241551790
3.3	-13.60099535907450
3.4	-13.97063943617580
3.5	-14.28861562665200
3.6	-14.54047586007080
3.7	-14.70548193444760
3.8	-14.75405249888170
3.9	-14.64430701272820
4	-14.32002259464840
4.1	-13.70736003517550
4.2	-12.73648235126670
4.3	-11.38641075495100
4.4	-9.75686757858566
4.5	-8.08624143487455
4.6	-6.63236524541499
4.7	-5.52256831767720
4.8	-4.74382856024562
4.9	-4.22224195335692
5	-3.88213352954333
5.1	-3.66388955817169
5.2	-3.52980561994226
5.3	-3.45150049814973
5.4	-3.41190537188402
5.5	-3.39885379813753
5.6	-3.40480287503558
5.7	-3.42439983123907
5.8	-3.45375027957383
5.9	-3.49039501240291
6	-3.53231520169981
6.1	-3.57798602639177
6.2	-3.62693922573299
6.3	-3.67810654812736
6.4	-3.73101741999105
6.5	-3.78520485136839

```
6.6    -3.84038070330609
6.7    -3.89626324163312
6.8    -3.95283546317759
6.9    -4.00985220710829
14 1 -1 1 1  %25 NLEV2 AUTO2 J2DL J2DU J2DD
4 5 6 7 8 9 10 11 12 13 14 15 16 17  %26 IV2
```

B.2. IF$^+$ Input File

The input file below uses the ground and excited state potential energy surfaces and transition dipole moment as calculated by MZI.

```
 53 127  9 19  1   2       %1 IAN1 IMN1 IAN2 IMN2 CHARGE NUMPOT
'Modelling G St to Exc State of IF+ using Marcins New Molpro'
 0.0015 0.5 40.0  1.d-6         %4 RH RMIN RMAX EPS
 276  0  1 19683      %5 NTP LPPOT IOMEG VLIM
 4 2  4  5 0                    %6 NUSE IR2 ILR NCN CNN
 1.D0  1 0.d0      %7 RFACT EFACT VSHIFT & B-state turning points
1.30   96727.25906
1.32   92824.08384
1.34   87279.78138
1.36   80504.6335
1.38   72908.92201
1.40   64902.9287
1.42   56896.9354
1.44   49301.2239
1.46   42513.20793
1.48   36634.33511
1.50   31470.08696
1.52   26846.00342
1.54   22719.33043
1.56   19079.38052
1.58   15895.68864
1.60   13118.01214
1.62   10698.95007
1.64   8605.907646
1.66   6810.0525
1.68   5283.95105
1.70   4001.568523
1.72   2937.963139
1.74   2072.32182
1.76   1384.943958
1.78   857.9756851
1.80   475.4098738
1.82   221.8785847
1.84   84.24945903
1.86   50
```

1.88	107.7799458
1.90	247.4112696
1.92	459.1098698
1.94	734.4634778
1.96	1065.436749
1.98	1444.77554
2.00	1866.006909
2.02	2322.901632
2.04	2810.069488
2.06	3322.351348
2.08	3855.078831
2.10	4404.074308
2.12	4965.318811
2.14	5535.342663
2.16	6110.827437
2.18	6688.774975
2.20	7266.507394
2.22	7841.462139
2.24	8411.482283
2.26	8974.520579
2.28	9528.720008
2.30	10072.41379
2.32	10604.0886
2.34	11122.81193
2.36	11627.78383
2.38	12117.91357
2.40	12591.81965
2.42	13048.38315
2.44	13487.58606
2.46	13909.67243
2.48	14314.58397
2.50	14701.96005
2.52	15071.52823
2.54	15423.42134
2.56	15757.87484
2.58	16075.15331
2.60	16375.55049
2.62	16659.38636
2.64	16927.08091
2.66	17179.08008
2.68	17415.85201

2.70	17637.88706
2.72	17845.66856
2.74	18039.64784
2.76	18220.26629
2.78	18387.92081
2.80	18542.96378
2.82	18685.70658
2.84	18816.30426
2.86	18934.86613
2.88	19041.34897
2.90	19135.55692
2.92	19217.23888
2.94	19285.94909
2.96	19341.21331
2.98	19383.02064
3.00	19411.82356
3.02	19428.43317
3.04	19435.01436
3.06	19434.05814
3.08	19427.77082
3.10	19418.07407
3.12	19406.67533
3.14	19394.47475
3.16	19382.16075
3.18	19370.19421
3.20	19358.80839
3.22	19348.20591
3.24	19338.50626
3.26	19329.80635
3.28	19322.16205
3.30	19315.58816
3.32	19310.078
3.34	19305.54598
3.36	19301.88797
3.38	19299.02571
3.40	19296.90688
3.42	19295.48163
3.44	19294.70558
3.46	19294.53576
3.48	19294.92953
3.50	19295.84464

3.52	19297.24006
3.54	19299.07962
3.56	19301.32832
3.58	19303.94971
3.60	19306.90593
3.62	19310.16124
3.64	19313.68857
3.66	19317.46301
3.68	19321.45912
3.70	19325.65086
3.72	19330.01363
3.74	19334.52838
3.76	19339.17741
3.78	19343.94075
3.80	19348.79617
3.82	19353.72605
3.84	19358.73144
3.86	19363.81728
3.88	19368.96922
3.90	19374.15373
3.92	19379.34221
3.94	19384.52918
3.96	19389.7148
3.98	19394.89469
4.00	19400.05997
4.02	19405.20234
4.04	19410.31654
4.06	19415.39806
4.08	19420.4426
4.10	19425.44607
4.12	19430.40417
4.14	19435.31186
4.16	19440.164
4.18	19444.95741
4.20	19449.69092
4.22	19454.3631
4.24	19458.97126
4.26	19463.51247
4.28	19467.98503
4.30	19472.38851
4.32	19476.72235

4.34	19480.98535
4.36	19485.17618
4.38	19489.29412
4.40	19493.33911
4.42	19497.3111
4.44	19501.20993
4.46	19505.03546
4.48	19508.78797
4.50	19512.46812
4.52	19516.07662
4.54	19519.61422
4.56	19523.08163
4.58	19526.47953
4.60	19529.80844
4.62	19533.06906
4.64	19536.26252
4.66	19539.39009
4.68	19542.45296
4.70	19545.45226
4.72	19548.38908
4.74	19551.26445
4.76	19554.0794
4.78	19556.83521
4.80	19559.53346
4.82	19562.17566
4.84	19564.76296
4.86	19567.29643
4.88	19569.77703
4.90	19572.20566
4.92	19574.58353
4.94	19576.91331
4.96	19579.19793
4.98	19581.43789
5.00	19583.63126
5.02	19585.77693
5.04	19587.87738
5.06	19589.93595
5.08	19591.95444
5.10	19593.93319
5.12	19595.87264
5.14	19597.77409

5.16	19599.63903
5.18	19601.46801
5.20	19603.26066
5.22	19605.01766
5.24	19606.74392
5.26	19608.44513
5.28	19610.12053
5.30	19611.76293
5.32	19613.36698
5.34	19614.93597
5.36	19616.47523
5.38	19617.98838
5.40	19619.4773
5.42	19620.94307
5.44	19622.3839
5.46	19623.79735
5.48	19625.18336
5.50	19626.54423
5.52	19627.88204
5.54	19629.19763
5.56	19630.49151
5.58	19631.7638
5.60	19633.01421
5.62	19634.24318
5.64	19635.454
5.66	19636.65053
5.68	19637.83234
5.70	19638.99477
5.72	19640.13423
5.74	19641.2522
5.76	19642.35138
5.78	19643.43346
5.80	19644.49907
5.82	19645.54856
5.84	19646.58125
5.86	19647.59639
5.88	19648.59672
5.90	19649.58858
5.92	19650.57572
5.94	19651.55108
5.96	19652.50522

5.98	19653.43651
6.00	19654.35108
6.02	19655.25418
6.04	19656.14606
6.06	19657.02575
6.08	19657.89276
6.10	19658.74707
6.12	19659.58883
6.14	19660.41888
6.16	19661.23821
6.18	19662.04698
6.20	19662.84457
6.22	19663.63055
6.24	19664.40549
6.26	19665.17018
6.28	19665.92501
6.30	19666.66993
6.32	19667.40497
6.34	19668.13027
6.36	19668.84606
6.38	19669.55245
6.40	19670.24949
6.42	19670.93729
6.44	19671.61626
6.46	19672.28686
6.48	19672.94919
6.50	19673.60299
6.52	19674.2487
6.54	19674.8896
6.56	19675.53003
6.58	19676.18138
6.60	19676.86212
6.62	19677.57529
6.64	19678.26085
6.66	19678.84735
6.68	19679.36281
6.70	19679.9347
6.72	19680.643
6.74	19681.3603
6.76	19681.91327
6.78	19682.26425

```
6.80   19682.51128
276  0  1  19683     %5b NTP LPPOT IOMEG VLIM
 4 2  4  5 0               %6b NUSE IR2 ILR NCN CNN
 1.D0 1 0.d0    %7b RFACT EFACT VSHIFT & B-state turning points
1.30   101429.9879
1.32   97542.55175
1.34   92018.76692
1.36   85267.72058
1.38   77698.4999
1.40   69720.19205
1.42   61741.8842
1.44   54172.66352
1.46   47408.79697
1.48   41551.68671
1.50   36407.87007
1.52   31803.76868
1.54   27696.62214
1.56   24075.52193
1.58   20909.94914
1.60   18149.7745
1.62   15747.69663
1.64   13671.13621
1.66   11891.25524
1.68   10380.61372
1.70   9113.169673
1.72   8063.967701
1.74   7212.155592
1.76   6537.987021
1.78   6023.557812
1.80   5652.805935
1.82   5410.306511
1.84   5282.862894
1.86   5257.886018
1.88   5323.948831
1.90   5470.786296
1.92   5688.525919
1.94   5968.663283
1.96   6303.069554
1.98   6684.387826
2.00   7106.033124
2.02   7561.659927
```

2.04	8045.746285
2.06	8552.996424
2.08	9078.581026
2.10	9618.137223
2.12	10167.44938
2.14	10722.80965
2.16	11280.64887
2.18	11837.66795
2.20	12390.8378
2.22	12937.21476
2.24	13474.14975
2.26	13999.07021
2.28	14509.47023
2.30	15002.91049
2.32	15477.03811
2.34	15929.83446
2.36	16359.34771
2.38	16763.24058
2.40	17138.79033
2.42	17483.66498
2.44	17797.16247
2.46	18079.02171
2.48	18329.75136
2.50	18550.62987
2.52	18743.27845
2.54	18910.55562
2.56	19055.61623
2.58	19181.31587
2.60	19290.21085
2.62	19384.70025
2.64	19466.6061
2.66	19537.59351
2.68	19599.03609
2.70	19652.01592
2.72	19697.52356
2.74	19736.23407
2.76	19768.73829
2.78	19795.5043
2.80	19816.8774
2.82	19832.80546
2.84	19841.66808

2.86	19841.49059
2.88	19831.16834
2.90	19810.46668
2.92	19779.95424
2.94	19743.26155
2.96	19704.70618
2.98	19666.80213
3.00	19630.25986
3.02	19595.62994
3.04	19563.13714
3.06	19532.92031
3.08	19505.01565
3.10	19479.35668
3.12	19455.86779
3.14	19434.45474
3.16	19415.01836
3.18	19397.45253
3.20	19381.64425
3.22	19367.49026
3.24	19354.92753
3.26	19343.90245
3.28	19334.34676
3.30	19326.17759
3.32	19319.30276
3.34	19313.59547
3.36	19308.92165
3.38	19305.17945
3.40	19302.29926
3.42	19300.21486
3.44	19298.86826
3.46	19298.20357
3.48	19298.16694
3.50	19298.70653
3.52	19299.77262
3.54	19301.32359
3.56	19303.31978
3.58	19305.72005
3.60	19308.48173
3.62	19311.56468
3.64	19314.93915
3.66	19318.57797

3.68	19322.45391
3.70	19326.53972
3.72	19330.80938
3.74	19335.24197
3.76	19339.81775
3.78	19344.51538
3.80	19349.31192
3.82	19354.18913
3.84	19359.14774
3.86	19364.19244
3.88	19369.30833
3.90	19374.46099
3.92	19379.62101
3.94	19384.78243
3.96	19389.94497
3.98	19395.10408
4.00	19400.25091
4.02	19405.3771
4.04	19410.47688
4.06	19415.5452
4.08	19420.57734
4.10	19425.56897
4.12	19430.51573
4.14	19435.41305
4.16	19440.25638
4.18	19445.04258
4.20	19449.76993
4.22	19454.43654
4.24	19459.03955
4.26	19463.57593
4.28	19468.04398
4.30	19472.44337
4.32	19476.77359
4.34	19481.03317
4.36	19485.22048
4.38	19489.33482
4.40	19493.37642
4.42	19497.34548
4.44	19501.2418
4.46	19505.06511
4.48	19508.8157

4.50	19512.49445
4.52	19516.10212
4.54	19519.63893
4.56	19523.10501
4.58	19526.50103
4.60	19529.82819
4.62	19533.08767
4.64	19536.28041
4.66	19539.40728
4.68	19542.4693
4.70	19545.46763
4.72	19548.40341
4.74	19551.27792
4.76	19554.0924
4.78	19556.84818
4.80	19559.54663
4.82	19562.18902
4.84	19564.77629
4.86	19567.3093
4.88	19569.78903
4.90	19572.21663
4.92	19574.5937
4.94	19576.92362
4.96	19579.21004
4.98	19581.45298
5.00	19583.64882
5.02	19585.79518
5.04	19587.89521
5.06	19589.95336
5.08	19591.97187
5.10	19593.95074
5.12	19595.89019
5.14	19597.79162
5.16	19599.65667
5.18	19601.48579
5.20	19603.27822
5.22	19605.03439
5.24	19606.75965
5.26	19608.4603
5.28	19610.1357
5.30	19611.77829

5.32	19613.38242
5.34	19614.95137
5.36	19616.49057
5.38	19618.00414
5.40	19619.49486
5.42	19620.96424
5.44	19622.40889
5.46	19623.82438
5.48	19625.21037
5.50	19626.57057
5.52	19627.90822
5.54	19629.22396
5.56	19630.51775
5.58	19631.78981
5.60	19633.04055
5.62	19634.2706
5.64	19635.48145
5.66	19636.67474
5.68	19637.85065
5.70	19639.00794
5.72	19640.14553
5.74	19641.26337
5.76	19642.36174
5.78	19643.44224
5.80	19644.50785
5.82	19645.56032
5.84	19646.59618
5.86	19647.611
5.88	19648.60875
5.90	19649.60175
5.92	19650.59768
5.94	19651.58424
5.96	19652.54471
5.98	19653.47595
6.00	19654.38839
6.02	19655.29083
6.04	19656.18315
6.06	19657.06309
6.08	19657.92991
6.10	19658.78438
6.12	19659.62716

```
6.14   19660.45825
6.16   19661.27746
6.18   19662.0851
6.20   19662.88188
6.22   19663.6684
6.24   19664.44452
6.26   19665.20994
6.28   19665.96476
6.30   19666.70944
6.32   19667.44439
6.34   19668.16974
6.36   19668.88557
6.38   19669.59195
6.40   19670.28899
6.42   19670.97684
6.44   19671.65581
6.46   19672.32627
6.48   19672.98846
6.50   19673.6425
6.52   19674.28904
6.54   19674.93095
6.56   19675.57191
6.58   19676.22325
6.60   19676.90382
6.62   19677.61697
6.64   19678.30257
6.66   19678.88897
6.68   19679.40429
6.70   19679.9764
6.72   19680.68552
6.74   19681.40382
6.76   19681.95738
6.78   19682.30839
6.80   19682.55517
-6  1  2  -4  50 1  -1 0    %17 NLEV1 AUTO1 LCDC LXPCT NJM JDJR IWR LPRWF
  0 0                       %18 IV(1) IJ(1)
 1  20   1.D0               %19 MORDR IRFN RREF
109 0                       %21 NRFN RFLIM
4 0 4  5 0                  %22  NUSEF IRF ILRF NCNF CNNF
1.D0 1                      %23 RFACTF, MFACTF
1.4    0.00106851730132
```

1.45	0.00108476848029
1.5	0.00108057591416
1.55	0.00104208779950
1.6	0.00096747642349
1.65	0.00086351742833
1.7	0.00074126058182
1.75	0.00061270529675
1.8	0.00048780767595
1.85	0.00037305690118
1.9	0.00027227661464
1.95	0.00018749072440
2	0.00011948624405
2.05	0.00006828211851
2.1	0.00003292931631
2.15	0.00001152517440
2.2	0.00000152693005
2.25	0.00000063601330
2.3	0.00000803509002
2.35	0.00002568982954
2.4	0.00005925694594
2.45	0.00012305945605
2.5	0.00020477590399
2.55	0.00029208522579
2.6	0.00036439821467
2.65	0.00041315095525
2.7	0.00043677295837
2.75	0.00043295621071
2.8	0.00039390414349
2.85	0.00029791811824
2.9	0.00012998400670
2.95	0.00003016082490
3	0.00000963888783
3.05	0.00000486337651
3.1	0.00000280759635
3.15	0.00000166364877
3.2	0.00000100560624
3.25	0.00000062510830
3.3	0.00000040553101
3.35	0.00000025395350
3.4	0.00000015090109
3.45	0.00000009353160

```
3.5     0.00000006051124
3.55    0.00000004029622
3.6     0.00000002718782
3.65    0.00000001838048
3.7     0.00000001246180
3.75    0.00000000844976
3.8     0.00000000572956
3.85    0.00000000387893
3.9     0.00000000263483
3.95    0.00000000178322
4       0.00000000121279
4.05    0.00000000082467
4.1     0.00000000055966
4.15    0.00000000038520
4.2     0.00000000026580
4.25    0.00000000018147
4.3     0.00000000012802
4.35    0.00000000009203
4.4     0.00000000006517
4.45    0.00000000004748
4.5     0.00000000003497
4.55    0.00000000002730
4.6     0.00000000001927
4.65    0.00000000001568
4.7     0.00000000001139
4.75    0.00000000000928
4.8     0.00000000000755
4.85    0.00000000000587
4.9     0.00000000000559
4.95    0.00000000000447
5       0.00000000000430
5.05    0.00000000000346
5.1     0.00000000000328
5.15    0.00000000000257
5.2     0.00000000000239
5.25    0.00000000000203
5.3     0.00000000000315
5.35    0.00000000000313
5.4     0.00000000000310
5.45    0.00000000000262
5.5     0.00000000000252
```

```
5.55   0.00000000000259
5.6    0.00000000000257
5.65   0.00000000000226
5.7    0.00000000000127
5.75   0.00000000000020
5.8    0.00000000000046
5.85   0.00000000000056
5.9    0.00000000000048
5.95   0.00000000000046
6      0.00000000000043
6.05   0.00000000000046
6.1    0.00000000000048
6.15   0.00000000000056
6.2    0.00000000000056
6.25   0.00000000000056
6.3    0.00000000000048
6.35   0.00000000000056
6.4    0.00000000000066
6.45   0.00000000000066
6.5    0.00000000000066
6.55   0.00000000000071
6.6    0.00000000000081
6.65   0.00000000000048
6.7    0.00000000000048
6.75   0.00000000000046
6.8    0.00000000000056
4 1 -1 1 1   %25 NLEV2 AUTO2 J2DL J2DU J2DD
0 1 2 3     %26 IV2
```

B.3. ICl^+ Input File

The input file below uses the ground and excited state potential energy surfaces and transition dipole moment as calculated by MZI.

```
 53 127 9 19 1   2          %1 IAN1 IMN1 IAN2 IMN2 CHARGE NUMPOT
'ICl+ ES to GS, Marcin potentials, PQR branches, ICl+ Dipole Moments'
 0.0015 0.5 40.0  1.d-6         %4 RH RMIN RMAX EPS
 191  0  1 14800      %5 NTP LPPOT IOMEG VLIM
 4 2  4  5 0                    %6 NUSE IR2 ILR NCN CNN
 1.D0  1 0.d0      %7 RFACT EFACT VSHIFT & B-state turning points
1.90   20933.82371
1.92   19026.19101
1.94   16960.97842
1.96   14843.2392
1.98   12778.02661
2.00   10870.3939
2.02   9200.123106
2.04   7745.911237
2.06   6462.057743
2.08   5322.956283
2.10   4323.094739
2.12   3456.656743
2.14   2713.114296
2.16   2080.840757
2.18   1550.032682
2.20   1112.709815
2.22   761.1140242
2.24   488.058572
2.26   286.527215
2.28   150.1394803
2.30   73.15066602
2.32   50
2.34   75.75185992
2.36   145.6440834
2.38   255.3094761
2.40   400.7758115
2.42   578.2009907
2.44   784.1947361
2.46   1015.491512
```

2.48	1269.096882
2.50	1542.287511
2.52	1832.431012
2.54	2137.211646
2.56	2454.400901
2.58	2781.955781
2.60	3118.018796
2.62	3460.800211
2.64	3808.74903
2.66	4160.37733
2.68	4514.275114
2.70	4869.110311
2.72	5223.664534
2.74	5577.160576
2.76	5928.90217
2.78	6277.517902
2.80	6620.961208
2.82	6958.180598
2.84	7292.222289
2.86	7626.69438
2.88	7954.566444
2.90	8258.169531
2.92	8533.284411
2.94	8831.340918
2.96	9213.526306
2.98	9645.466291
3.00	9997.225057
3.02	10179.31538
3.04	10280.66377
3.06	10430.63488
3.08	10662.79061
3.10	10914.89019
3.12	11134.53245
3.14	11325.336
3.16	11504.45792
3.18	11678.32629
3.20	11842.64018
3.22	11993.75907
3.24	12132.54995
3.26	12260.9767
3.28	12380.3133

3.30	12491.14383
3.32	12594.12122
3.34	12690.29402
3.36	12780.79706
3.38	12866.47661
3.40	12947.89029
3.42	13025.55561
3.44	13099.87981
3.46	13171.23817
3.48	13239.90623
3.50	13306.05972
3.52	13369.83669
3.54	13431.24188
3.56	13490.24044
3.58	13546.65389
3.60	13600.16011
3.62	13650.26598
3.64	13695.81933
3.66	13735.54545
3.68	13769.14026
3.70	13797.27028
3.72	13820.76307
3.74	13840.9214
3.76	13859.14255
3.78	13876.26454
3.80	13892.56616
3.82	13908.28257
3.84	13923.57169
3.86	13938.57016
3.88	13953.36971
3.90	13968.01715
3.92	13982.55234
3.94	13996.99515
3.96	14011.36019
3.98	14025.65536
4.00	14039.88192
4.02	14054.04009
4.04	14068.12705
4.06	14082.13914
4.08	14096.07026
4.10	14109.91189

4.12	14123.65573
4.14	14137.29487
4.16	14150.82271
4.18	14164.23238
4.20	14177.51666
4.22	14190.66857
4.24	14203.68201
4.26	14216.55116
4.28	14229.2707
4.30	14241.8359
4.32	14254.2423
4.34	14266.48659
4.36	14278.56571
4.38	14290.47692
4.40	14302.21776
4.42	14313.78608
4.44	14325.18094
4.46	14336.40167
4.48	14347.44797
4.50	14358.31986
4.52	14369.01732
4.54	14379.54008
4.56	14389.88786
4.58	14400.06244
4.60	14410.06759
4.62	14419.90499
4.64	14429.56762
4.66	14439.04738
4.68	14448.36207
4.70	14457.55532
4.72	14466.6569
4.74	14475.63669
4.76	14484.4509
4.78	14493.08536
4.80	14501.55559
4.82	14509.87527
4.84	14518.04565
4.86	14526.06501
4.88	14533.93491
4.90	14541.66019
4.92	14549.24523

```
4.94   14556.69194
4.96   14564.00164
4.98   14571.17666
5.00   14578.22027
5.02   14585.13563
5.04   14591.92516
5.06   14598.59113
5.08   14605.13564
5.10   14611.56067
5.12   14617.86827
5.14   14624.06087
5.16   14630.14098
5.18   14636.11101
5.20   14641.97327
5.22   14647.72995
5.24   14653.38292
5.26   14658.93399
5.28   14664.38522
5.30   14669.739
5.32   14674.99754
5.34   14680.16232
5.36   14685.23466
5.38   14690.21647
5.40   14695.11027
5.42   14699.91836
5.44   14704.64218
5.46   14709.28296
5.48   14713.84225
5.50   14718.32191
5.52   14722.72374
5.54   14727.04919
5.56   14731.29965
5.58   14735.47675
5.60   14739.58241
5.62   14743.61814
5.64   14747.58371
5.66   14751.47847
5.68   14755.30202
5.70   14759.0542
191  0  1  14800     %5b NTP LPPOT IOMEG VLIM
 4 2  4  5 0                 %6b NUSE IR2 ILR NCN CNN
 1.D0 1 0.d0    %7b RFACT EFACT VSHIFT & B-state turning points
1.90   24999.35862
1.92   23113.88466
1.94   21072.54974
1.96   18979.26116
1.98   16937.92623
2.00   15052.45228
2.02   13401.82544
2.04   11965.34719
2.06   10698.24649
2.08   9575.27138
2.10   8590.688961
2.12   7738.520467
2.14   7008.408968
2.16   6388.975945
2.18   5870.520733
2.20   5445.020522
2.22   5104.676879
2.24   4842.297079
2.26   4650.866634
2.28   4523.987702
2.30   4455.879085
2.32   4440.939649
2.34   4474.181259
2.36   4550.785604
2.38   4666.31555
2.40   4816.715132
2.42   4998.05107
2.44   5206.814512
2.46   5439.613346
2.48   5693.29997
2.50   5964.971287
2.52   6251.797723
2.54   6551.201248
2.56   6860.670826
2.58   7177.789789
2.60   7500.235839
2.62   7825.700411
2.64   8151.913451
2.66   8476.608041
2.68   8797.367896
```

2.70	9111.627362
2.72	9416.801361
2.74	9710.253097
2.76	9989.299765
2.78	10250.49777
2.80	10489.64273
2.82	10704.16908
2.84	10898.19888
2.86	11077.09543
2.88	11236.31722
2.90	11361.41791
2.92	11452.04523
2.94	11565.94566
2.96	11770.95075
2.98	12032.77989
3.00	12215.04035
3.02	12224.09152
3.04	12155.05993
3.06	12145.84431
3.08	12232.69252
3.10	12350.20161
3.12	12443.33858
3.14	12516.22882
3.16	12587.28733
3.18	12663.42877
3.20	12740.06756
3.22	12813.2175
3.24	12883.29006
3.26	12951.75016
3.28	13019.00488
3.30	13084.40351
3.32	13147.55028
3.34	13209.25334
3.36	13270.52607
3.38	13330.18088
3.40	13384.82916
3.42	13432.1802
3.44	13474.71757
3.46	13516.05375
3.48	13558.31336
3.50	13602.13319

3.52	13647.32103
3.54	13690.62753
3.56	13728.1307
3.58	13758.01657
3.60	13780.57923
3.62	13796.65683
3.64	13808.89723
3.66	13820.32908
3.68	13832.33346
3.70	13844.64388
3.72	13857.0012
3.74	13869.46227
3.76	13882.15812
3.78	13895.10927
3.80	13908.22568
3.82	13921.42999
3.84	13934.71478
3.86	13948.08961
3.88	13961.55311
3.90	13975.09301
3.92	13988.6968
3.94	14002.35297
3.96	14016.05021
3.98	14029.77597
4.00	14043.51642
4.02	14057.25752
4.04	14070.98442
4.06	14084.6822
4.08	14098.33731
4.10	14111.93764
4.12	14125.47118
4.14	14138.9261
4.16	14152.29073
4.18	14165.55502
4.20	14178.7106
4.22	14191.74921
4.24	14204.66268
4.26	14217.44292
4.28	14230.08312
4.30	14242.57772
4.32	14254.9214

4.34	14267.10957
4.36	14279.13785
4.38	14291.00299
4.40	14302.7028
4.42	14314.23512
4.44	14325.59776
4.46	14336.78854
4.48	14347.80682
4.50	14358.65346
4.52	14369.329
4.54	14379.83233
4.56	14390.16207
4.58	14400.31959
4.60	14410.30901
4.62	14420.13227
4.64	14429.78211
4.66	14439.2501
4.68	14448.55395
4.70	14457.73748
4.72	14466.83057
4.74	14475.80274
4.76	14484.60971
4.78	14493.23711
4.80	14501.70044
4.82	14510.0135
4.84	14518.17783
4.86	14526.19208
4.88	14534.05765
4.90	14541.77871
4.92	14549.35916
4.94	14556.80142
4.96	14564.10752
4.98	14571.27987
5.00	14578.32123
5.02	14585.23427
5.04	14592.02111
5.06	14598.68375
5.08	14605.22456
5.10	14611.64626
5.12	14617.9515
5.14	14624.1426

```
5.16   14630.22176
5.18   14636.19103
5.20   14642.05228
5.22   14647.8074
5.24   14653.45849
5.26   14659.00771
5.28   14664.45721
5.30   14669.80923
5.32   14675.06588
5.34   14680.22889
5.36   14685.29991
5.38   14690.28085
5.40   14695.17392
5.42   14699.98117
5.44   14704.70408
5.46   14709.34399
5.48   14713.90244
5.50   14718.38116
5.52   14722.78192
5.54   14727.1065
5.56   14731.35665
5.58   14735.53389
5.60   14739.63948
5.62   14743.67443
5.64   14747.63897
5.66   14751.53311
5.68   14755.35663
5.70   14759.10907
-6  1  2  -4  50 1  -1 0    %17 NLEV1 AUTO1 LCDC LXPCT NJM JDJR IWR LPRWF
  0 0                       %18 IV(1) IJ(1)
 1  20   1.D0               %19 MORDR IRFN RREF
100 0                       %21 NRFN RFLIM
4 0  4  5 0                 %22  NUSEF IRF ILRF NCNF CNNF
1.D0 1                      %23 RFACTF, MFACTF
1.95   0.00001823257082
2      0.00000975709576
2.05   0.00000584546212
2.1    0.00000325554646
2.15   0.00000155700556
2.2    0.00000046975354
2.25   0.00000001133084
```

2.3	0.00000024288226
2.35	0.00000123880712
2.4	0.00000307851641
2.45	0.00000583119359
2.5	0.00000956416788
2.55	0.00001438835492
2.6	0.00002049591874
2.65	0.00002842919526
2.7	0.00003952620475
2.75	0.00005570207645
2.8	0.00007786040606
2.85	0.00010024255482
2.9	0.00011284229404
2.95	0.00011492992999
3	0.00011069289745
3.05	0.00010308796872
3.1	0.00009246274787
3.15	0.00007924265749
3.2	0.00006404376788
3.25	0.00004876313174
3.3	0.00003579646447
3.35	0.00002622743623
3.4	0.00001837902199
3.45	0.00000009611743
3.5	0.00000003544212
3.55	0.00000011553142
3.6	0.00000120457727
3.65	0.00000044303650
3.7	0.00000022212990
3.75	0.00000012430472
3.8	0.00000007677143
3.85	0.00000005054678
3.9	0.00000003440121
3.95	0.00000002409027
4	0.00000001676619
4.05	0.00000001176730
4.1	0.00000000826736
4.15	0.00000000590505
4.2	0.00000000424103
4.25	0.00000000313864
4.3	0.00000000224035

4.35	0.00000000164549
4.4	0.00000000122029
4.45	0.00000000091113
4.5	0.00000000069547
4.55	0.00000000053334
4.6	0.00000000042359
4.65	0.00000000033617
4.7	0.00000000027159
4.75	0.00000000022349
4.8	0.00000000018497
4.85	0.00000000015682
4.9	0.00000000013446
4.95	0.00000000011699
5	0.00000000010354
5.05	0.00000000009265
5.1	0.00000000008344
5.15	0.00000000007362
5.2	0.00000000006837
5.25	0.00000000006439
5.3	0.00000000006060
5.35	0.00000000005764
5.4	0.00000000005505
5.45	0.00000000005299
5.5	0.00000000004994
5.55	0.00000000004841
5.6	0.00000000004676
5.65	0.00000000004489
5.7	0.00000000004305
5.75	0.00000000004175
5.8	0.00000000004101
5.85	0.00000000003894
5.9	0.00000000003814
5.95	0.00000000003717
6	0.00000000003666
6.05	0.00000000003551
6.1	0.00000000003438
6.15	0.00000000009102
6.2	0.00000000003204
6.25	0.00000000003187
6.3	0.00000000003061
6.35	0.00000000003080

```
6.4    0.00000000002962
6.45   0.00000000002916
6.5    0.00000000002875
6.55   0.00000000002817
6.6    0.00000000002754
6.65   0.00000000002724
6.7    0.00000000002676
6.75   0.00000000002635
6.8    0.00000000002554
6.85   0.00000000002660
6.9    0.00000000002598
4 1 -1 1 1   %25 NLEV2 AUTO2 J2DL J2DU J2DD
0 1 2 3     %26 IV2
```

B.4. AlH$^+$ Vibrational Decay Lifetime Input File

The input file below uses the updated potential energy surface and permanent dipole moment of the $\mathrm{X}^2\Sigma^+$ state to calculate the $\left|\mathrm{X}^2\Sigma^+, v=1, N=1\right\rangle \rightarrow \left|\mathrm{X}^2\Sigma^+, v=0\right\rangle$ transition lifetime.

```
 13 27  1 1  1   1         %1 IAN1 IMN1 IAN2 IMN2 CHARGE NUMPOT
’AlH+ GS Vibrational Transitions w GA Tech V(r) and M(r)_com (center of mass?)’
 0.0015 0.5 40.0  1.d-6         %4 RH RMIN RMAX EPS
 46  0  0 6085      %5 NTP LPPOT IOMEG VLIM
 4 2  4  5 0                    %6 NUSE IR2 ILR NCN CNN
 1.0D+00 1.0D+00 0      %7 RFACT EFACT VSHIFT & B-state turning points
0.8    119937.90630
0.88   82577.45146
0.96   55688.34464
1.04   36572.57402
1.12   23176.39918
1.2    13968.20739
1.28   7819.56211
1.36   3897.63207
1.44   1584.39930
1.52   417.89008
1.6    50.00000
1.75   636.68497
1.9    1922.41602
2.05   3202.06820
2.2    4165.89119
2.35   4780.64247
2.5    5138.32072
2.65   5342.81383
2.8    5467.21132
2.95   5553.01885
3.1    5620.70647
3.25   5679.11510
3.4    5731.44459
3.55   5778.53236
3.7    5820.46237
3.85   5857.26136
4.0    5889.13072
```

```
4.15   5916.45718
4.3    5939.73742
4.45   5959.49710
4.6    5976.24017
4.75   5990.42587
4.9    6002.45972
5.05   6012.69121
5.2    6021.41536
5.35   6028.87794
5.5    6035.28272
6.0    6050.95344
6.5    6060.85028
7.0    6067.36618
7.5    6071.81794
8.0    6074.95678
8.5    6077.22195
9.0    6078.88878
9.5    6080.13684
10     6081.08618
-10 1  2  -4 3 1  -1 0    %17 NLEV1 AUTO1 LCDC LXPCT NJM JDJR IWR LPRWF
  0 0                     %18 IV(1) IJ(1)
 1  20  1.D0              %19 MORDR IRFN RREF
46 0                      %21 NRFN RFLIM
4 2 4  5 0                %22  NUSEF IRF ILRF NCNF CNNF
1.0D+00 1.0D+00                  %23 RFACTF, MFACTF
0.80   -0.248106588
0.88   -0.278866697
0.96   -0.317996546
1.04   -0.358799143
1.12   -0.390103084
1.20   -0.409506544
1.28   -0.416840345
1.36   -0.411279285
1.44   -0.3913279
1.52   -0.355168895
1.60   -0.301168993
1.75   -0.150711355
1.90   0.047493693
2.05   0.243300019
2.20   0.375677582
2.35   0.417267511
```

```
2.50   0.383037157
2.65   0.303627522
2.80   0.204501055
2.95   0.10122778
3.10   0.001798285
3.25   -0.090332509
3.40   -0.174136264
3.55   -0.249791537
3.70   -0.318023997
3.85   -0.379756475
4.00   -0.435922275
4.15   -0.487394817
4.30   -0.534953147
4.45   -0.579282138
4.60   -0.620971195
4.75   -0.660511714
4.90   -0.698301783
5.05   -0.734656885
5.20   -0.769824482
5.35   -0.803998817
5.50   -0.837333383
6.00   -0.943796321
6.50   -1.04518085
7.00   -1.143172885
7.50   -1.238773417
8.00   -1.332616103
8.50   -1.425130802
9.00   -1.516621254
9.50   -1.60730943
10.00  -1.697360415
1 1 -1 1 1   %25 NLEV2 AUTO2 J2DL J2DU J2DD
0    %26 IV2
```

APPENDIX C

LabVIEW Code

This section contains most LabVIEW code used for both experiments. Original code credit given where notable.

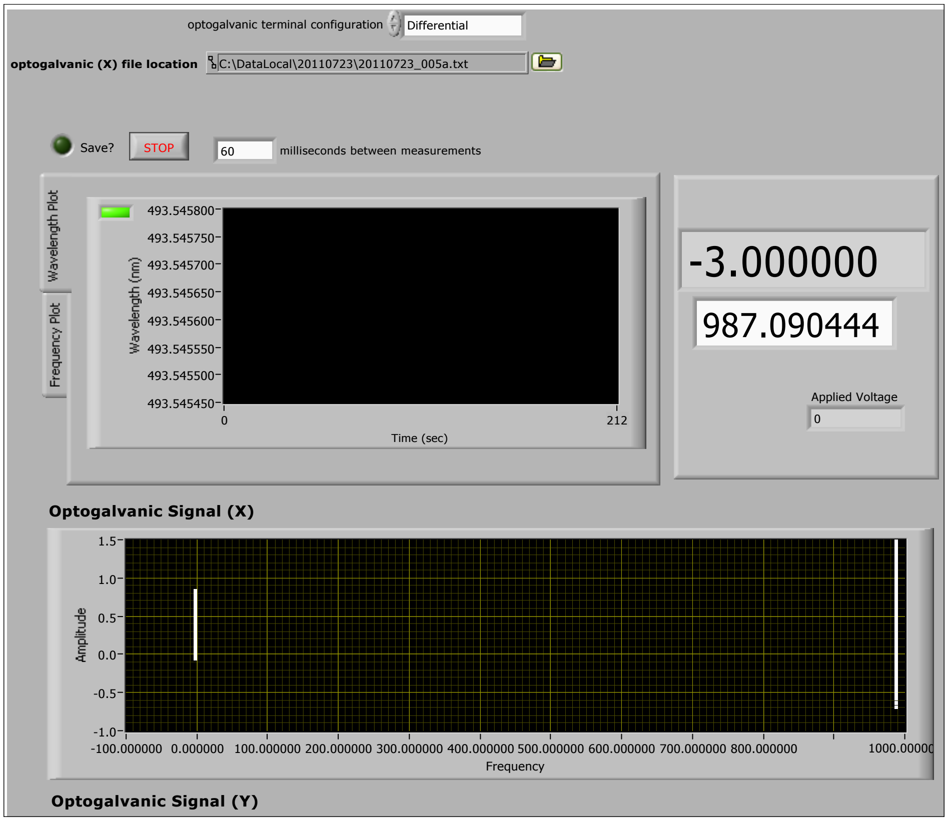
C.1. ECDL Project

C.1.1. Wavelength Meter Lock Computer

C.1.1.1. Hollow-cathode Cell Logger. The code below records a continuous stream of wavelength meter readings and input voltage to a NI AI channel.

Page 1

record_optigalvanic_v2.vi
C:\LabviewCode\record_optigalvanic_v2.vi
Last modified on 7/25/2012 at 6:40 PM
Printed on 7/4/2016 at 11:46 AM



optogalvanic terminal configuration: Differential

optogalvanic (X) file location: C:\DataLocal\20110723\20110723_005a.txt

Save?

60 milliseconds between measurements

-3.000000

987.090444

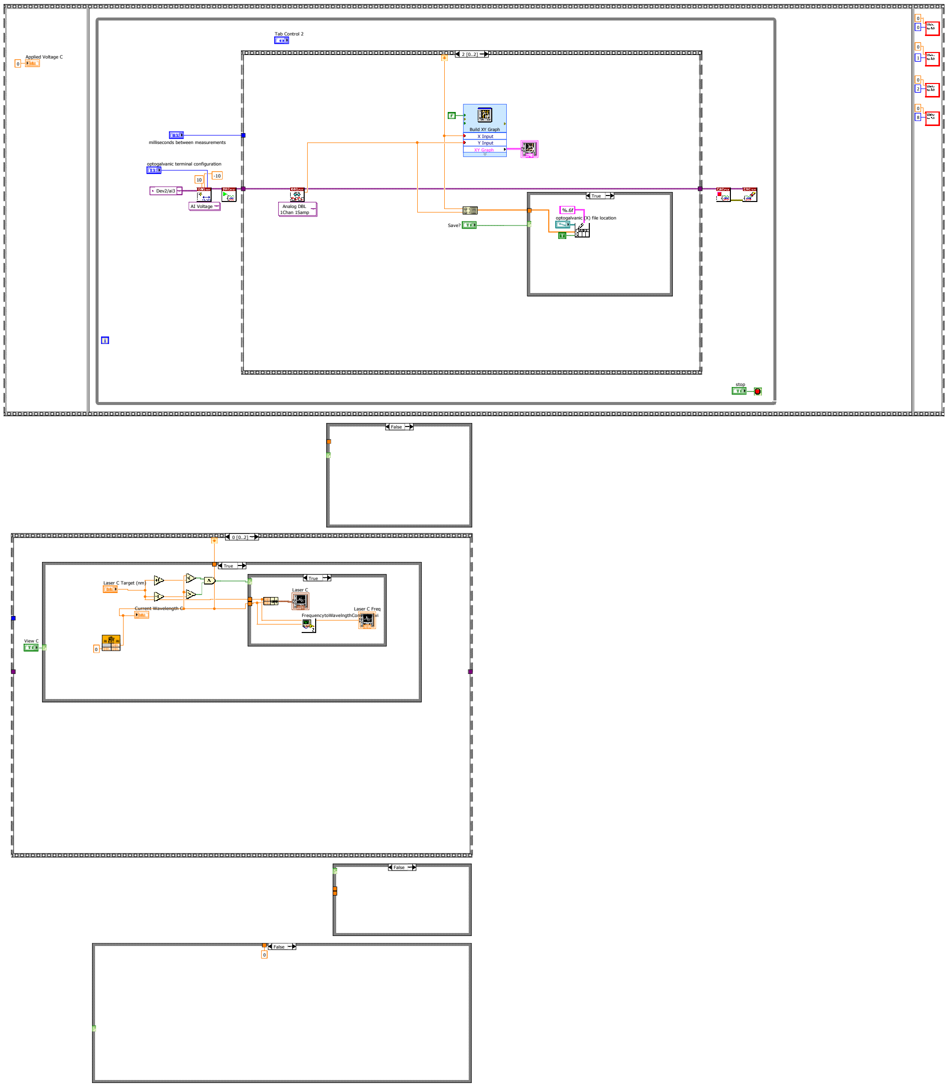
Applied Voltage: 0

Optogalvanic Signal (X)

Optogalvanic Signal (Y)

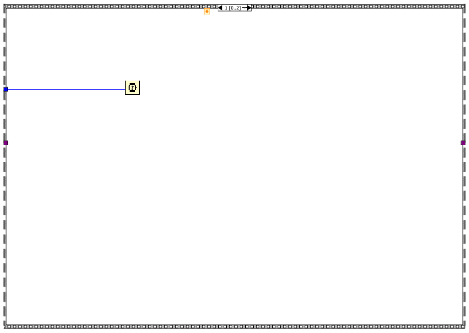
Page 1

record_optigalvanic_v2.vi
C:\LabviewCode\record_optigalvanic_v2.vi
Last modified on 7/25/2012 at 6:40 PM
Printed on 7/4/2016 at 11:46 AM



Page 2

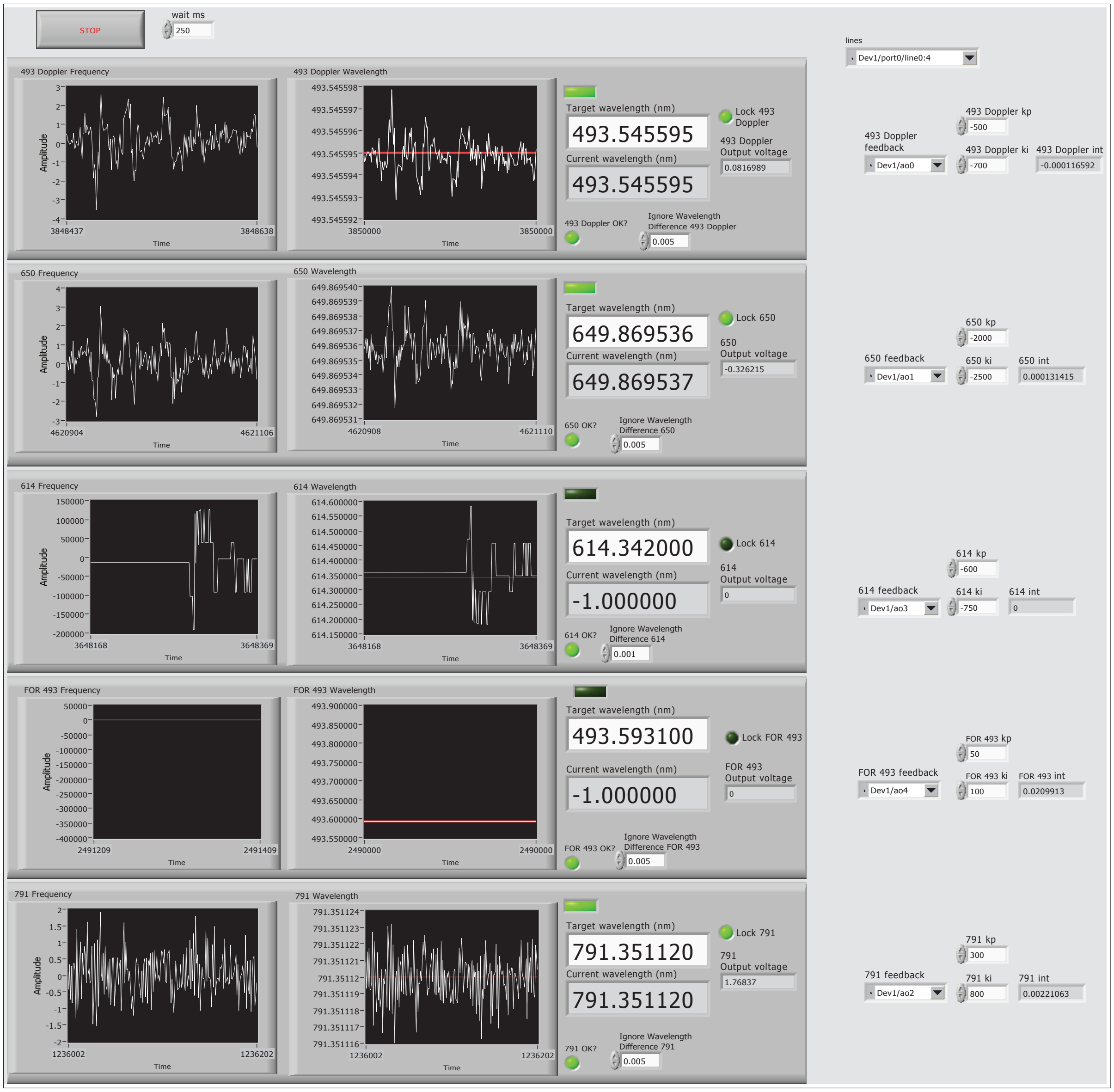
record_optigalvanic_v2.vi
C:\LabviewCode\record_optigalvanic_v2.vi
Last modified on 7/25/2012 at 6:40 PM
Printed on 7/4/2016 at 11:46 AM



C.1.1.2. Wavelength Meter Laser Locking. The code below provides laser frequency locking by sequentially switching between lasers using 80 MHz AOMs before combining all beamlines using polarization and dichroics. The controlling PC uses a PCI-6722 for interfacing. The Wavelength meter is a High Finesse WS-U 10. The unincluded sub VIs generate an error signal by subtraction and convert wavelength difference to a frequency difference.

Page 1

medusa_v29.vi
C:\Documents and Settings\David\Desktop\medusa_v29.vi
Last modified on 3/1/2016 at 5:24 PM
Printed on 6/6/2016 at 9:23 AM



wait ms 250

lines Dev1/port0/line0:4

Target wavelength (nm) 493.545595
Current wavelength (nm) 493.545595
● Lock 493 Doppler
493 Doppler Output voltage 0.0816989
493 Doppler OK? ●
Ignore Wavelength Difference 493 Doppler 0.005

493 Doppler feedback Dev1/ao0
493 Doppler kp -500
493 Doppler ki -700
493 Doppler int -0.000116592

Target wavelength (nm) 649.869536
Current wavelength (nm) 649.869537
● Lock 650
650 Output voltage -0.326215
650 OK? ●
Ignore Wavelength Difference 650 0.005

650 feedback Dev1/ao1
650 kp -2000
650 ki -2500
650 int 0.000131415

Target wavelength (nm) 614.342000
Current wavelength (nm) -1.000000
● Lock 614
614 Output voltage 0
614 OK? ●
Ignore Wavelength Difference 614 0.001

614 feedback Dev1/ao3
614 kp -600
614 ki -750
614 int 0

Target wavelength (nm) 493.593100
Current wavelength (nm) -1.000000
● Lock FOR 493
FOR 493 Output voltage 0
FOR 493 OK? ●
Ignore Wavelength Difference FOR 493 0.005

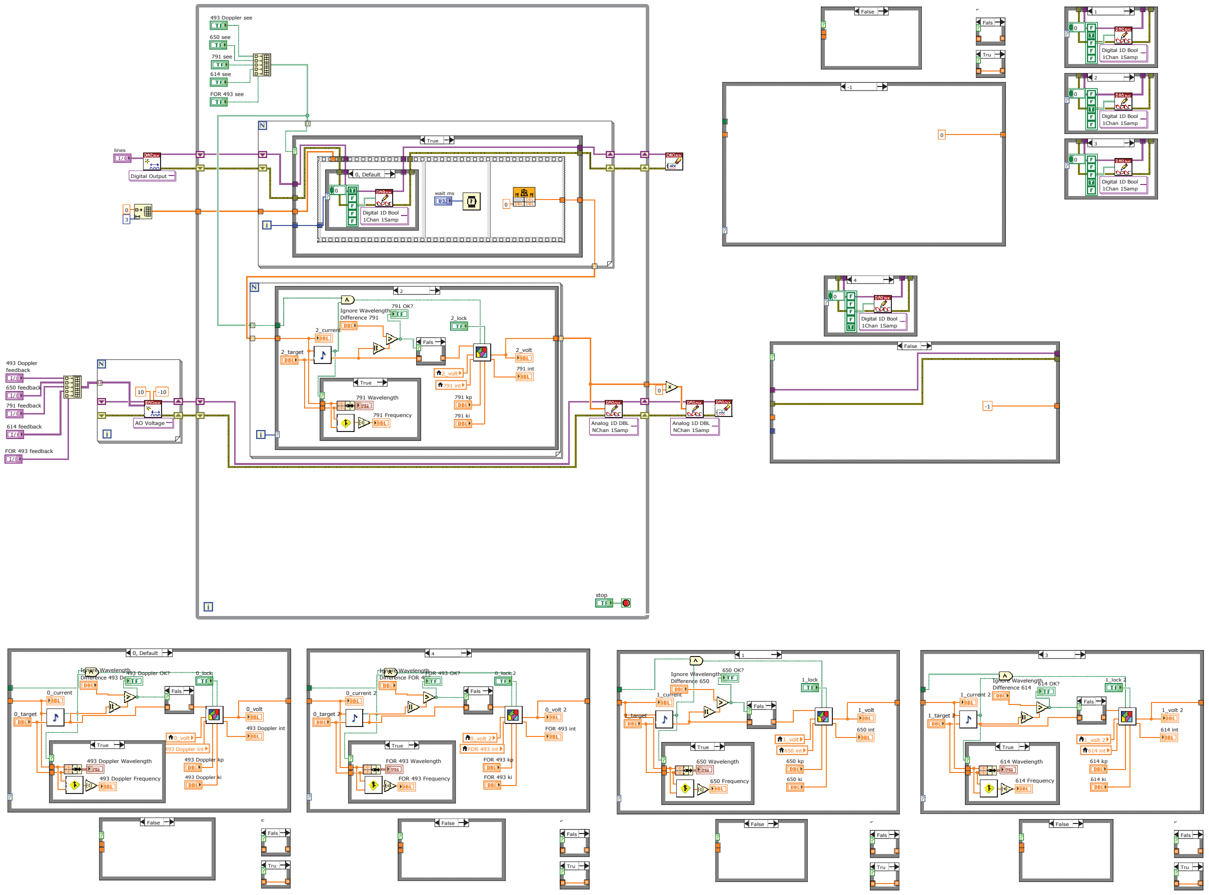
FOR 493 feedback Dev1/ao4
FOR 493 kp 50
FOR 493 ki 100
FOR 493 int 0.0209913

Target wavelength (nm) 791.351120
Current wavelength (nm) 791.351120
● Lock 791
791 Output voltage 1.76837
791 OK? ●
Ignore Wavelength Difference 791 0.005

791 feedback Dev1/ao2
791 kp 300
791 ki 800
791 int 0.00221063

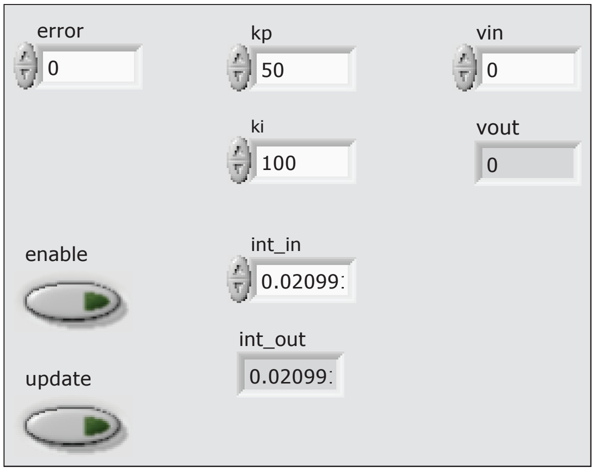
Page 1

medusa_v29.vi
C:\Documents and Settings\David\Desktop\medusa_v29.vi
Last modified on 3/1/2016 at 5:24 PM
Printed on 6/6/2016 at 9:24 AM



Page 1

medusa_PID.vi
C:\Documents and Settings\David\Desktop\medusa_PID.vi
Last modified on 1/7/2016 at 9:42 AM
Printed on 6/6/2016 at 9:26 AM



error 0

kp 50

vin 0

ki 100

vout 0

enable

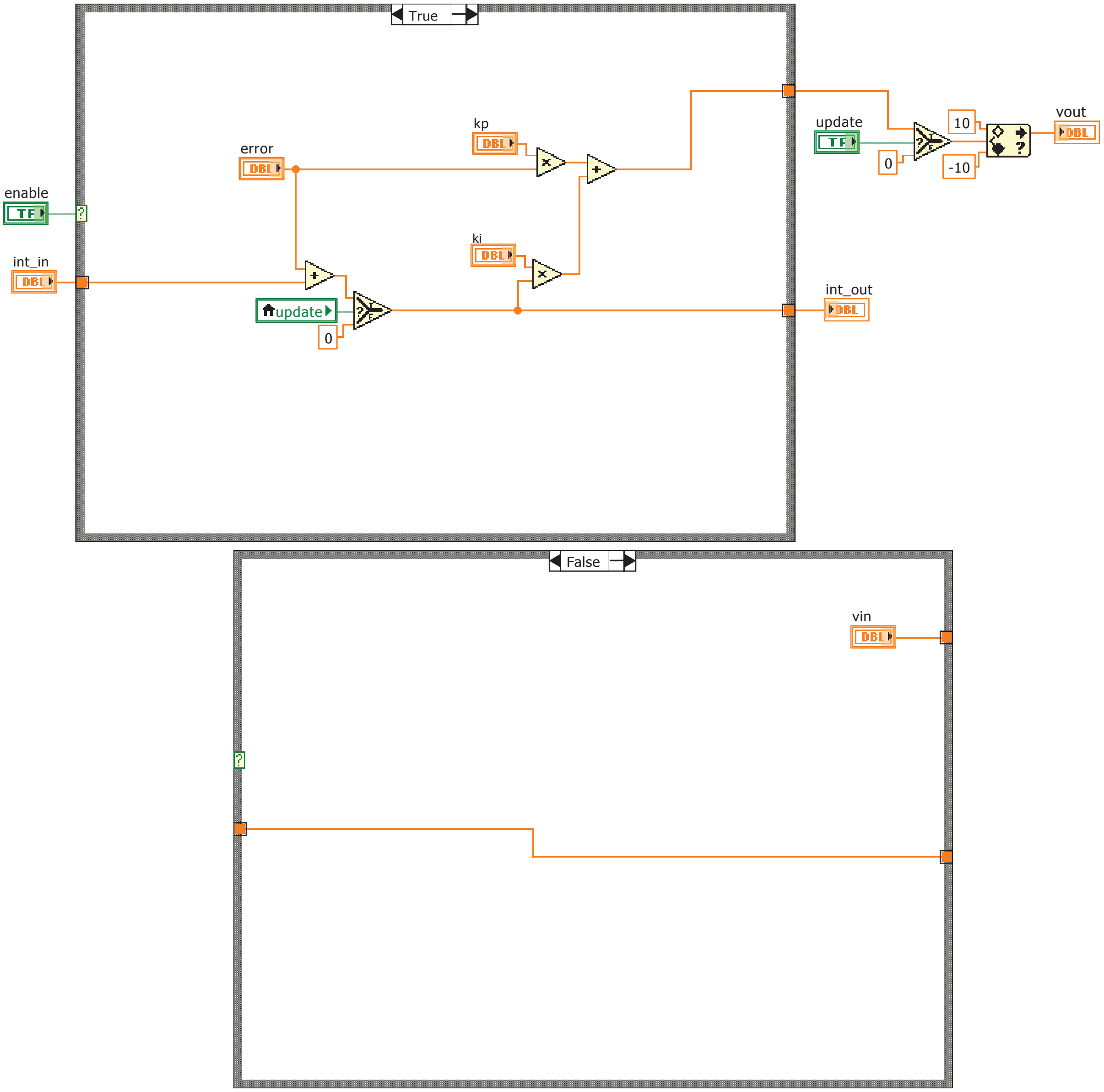
int_in 0.02099

int_out 0.02099

update

Page 1

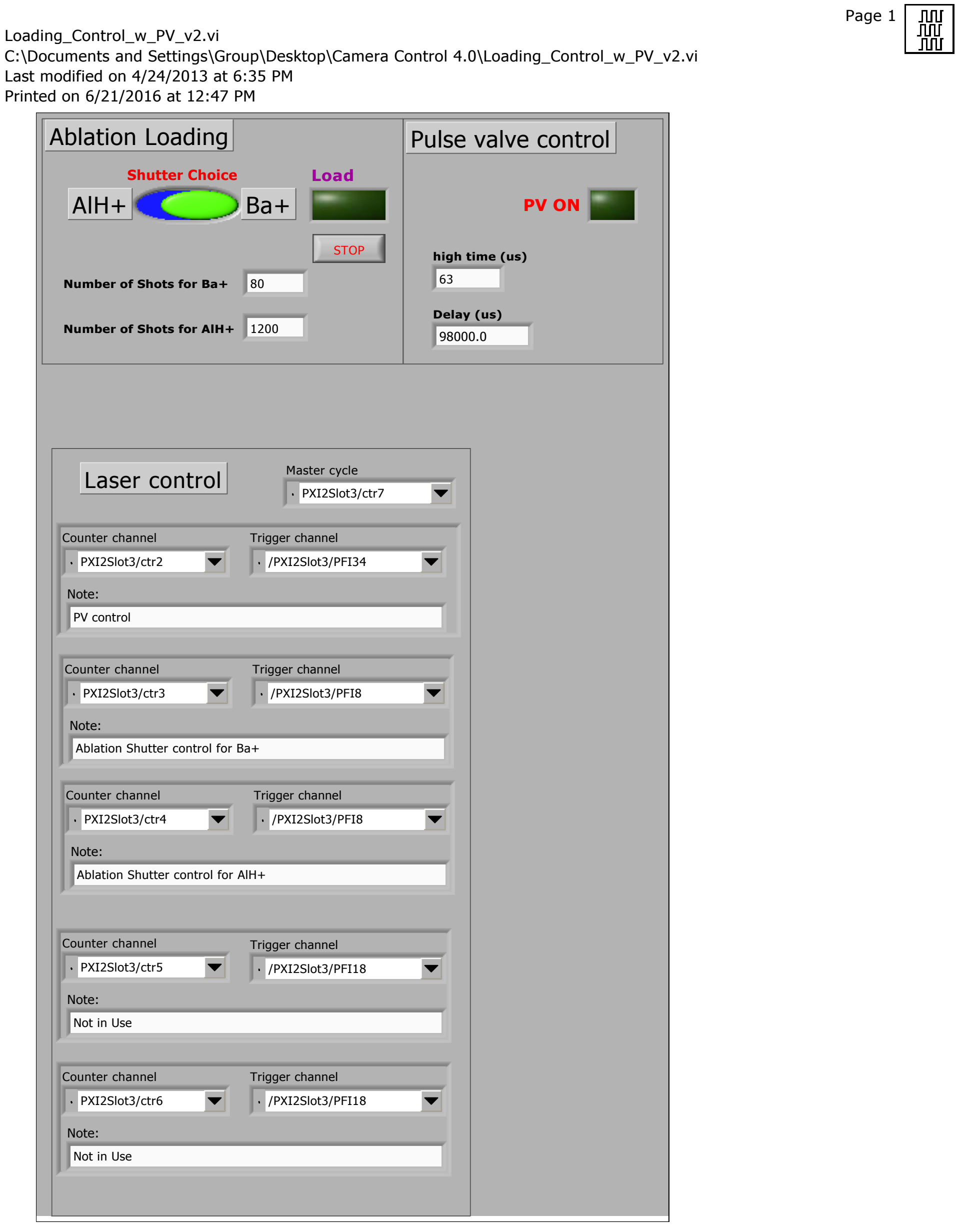
medusa_PID.vi
C:\Documents and Settings\David\Desktop\medusa_PID.vi
Last modified on 1/7/2016 at 9:42 AM
Printed on 6/6/2016 at 9:26 AM



C.2. AlH$^+$ Rotational Cooling Project

C.2.1. Ablation Load Control

This code opens the correct shutter for ablation of either Ba or Al. Pulse valve control and synchronization is also included.



Page 1

Loading_Control_w_PV_v2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\Loading_Control_w_PV_v2.vi
Last modified on 4/24/2013 at 6:35 PM
Printed on 6/21/2016 at 12:47 PM

Ablation Loading

Shutter Choice: AlH+ / Ba+

Load

STOP

Number of Shots for Ba+ 80

Number of Shots for AlH+ 1200

Pulse valve control

PV ON

high time (us) 63

Delay (us) 98000.0

Laser control

Master cycle: PXI2Slot3/ctr7

Counter channel: PXI2Slot3/ctr2
Trigger channel: /PXI2Slot3/PFI34
Note: PV control

Counter channel: PXI2Slot3/ctr3
Trigger channel: /PXI2Slot3/PFI8
Note: Ablation Shutter control for Ba+

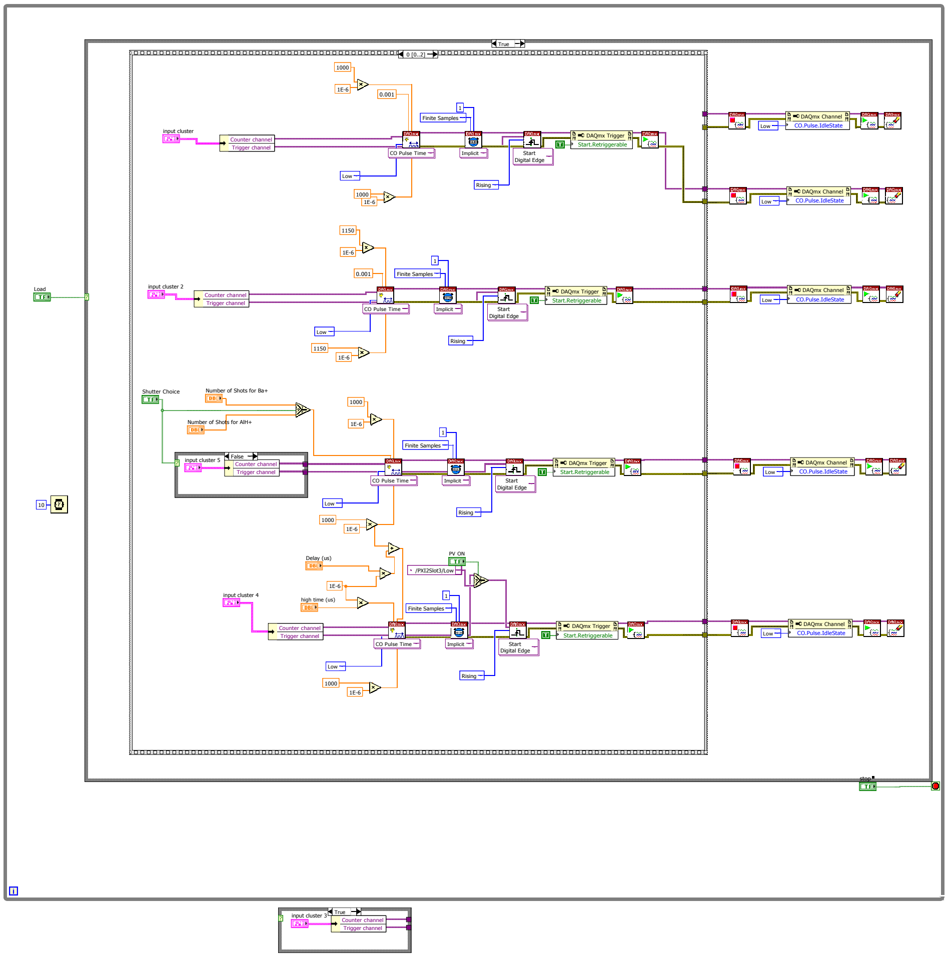
Counter channel: PXI2Slot3/ctr4
Trigger channel: /PXI2Slot3/PFI8
Note: Ablation Shutter control for AlH+

Counter channel: PXI2Slot3/ctr5
Trigger channel: /PXI2Slot3/PFI18
Note: Not in Use

Counter channel: PXI2Slot3/ctr6
Trigger channel: /PXI2Slot3/PFI18
Note: Not in Use

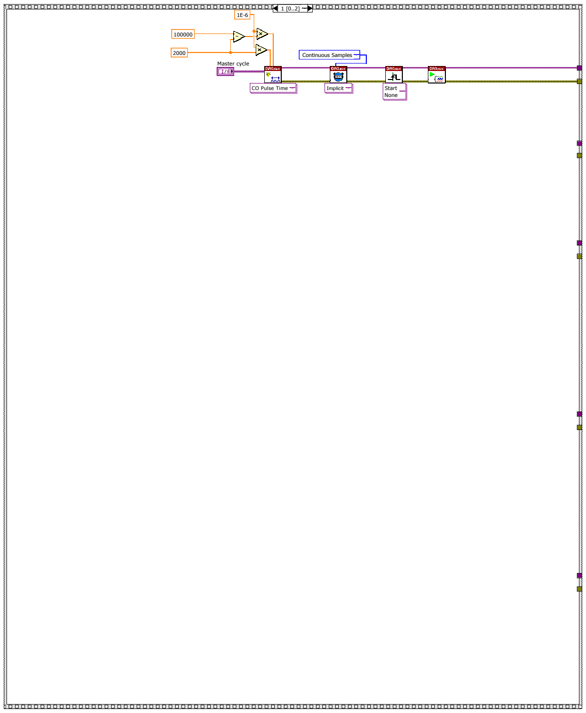
Page 1

Loading_Control_w_PV_v2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\Loading_Control_w_PV_v2.vi
Last modified on 4/24/2013 at 6:35 PM
Printed on 6/21/2016 at 12:47 PM



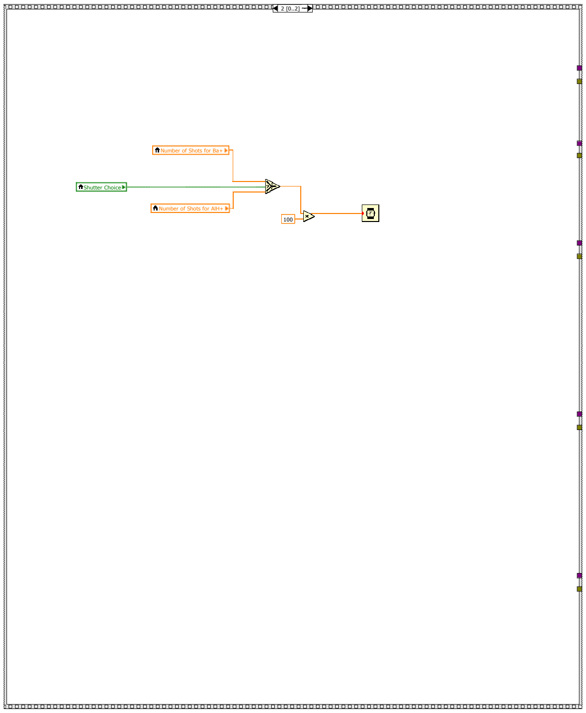
Page 2

Loading_Control_w_PV_v2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\Loading_Control_w_PV_v2.vi
Last modified on 4/24/2013 at 6:35 PM
Printed on 6/21/2016 at 12:47 PM



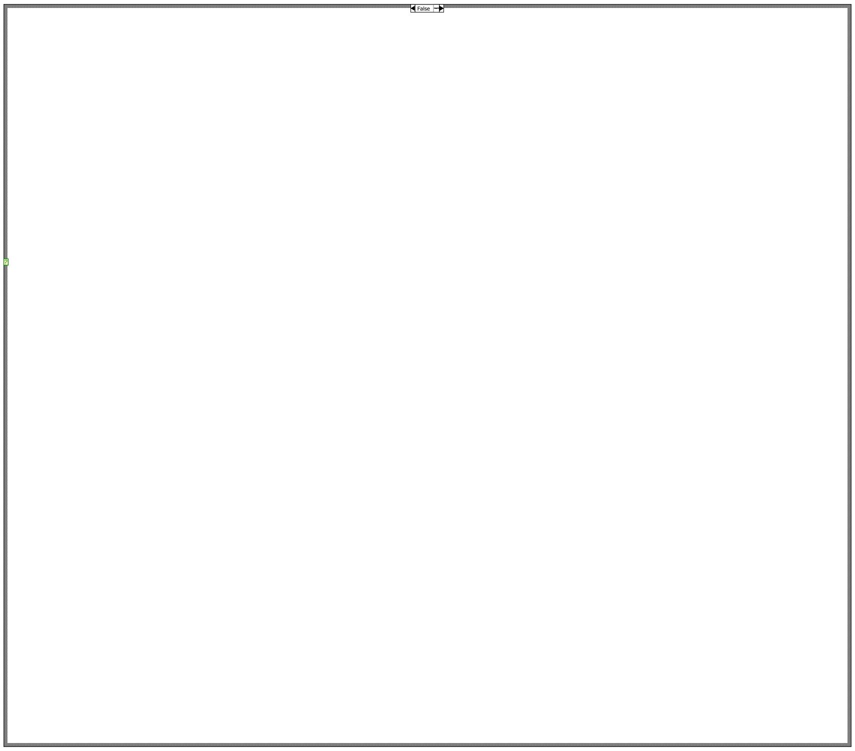
Page 3

Loading_Control_w_PV_v2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\Loading_Control_w_PV_v2.vi
Last modified on 4/24/2013 at 6:35 PM
Printed on 6/21/2016 at 12:47 PM



Page 4

Loading_Control_w_PV_v2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\Loading_Control_w_PV_v2.vi
Last modified on 4/24/2013 at 6:35 PM
Printed on 6/21/2016 at 12:47 PM

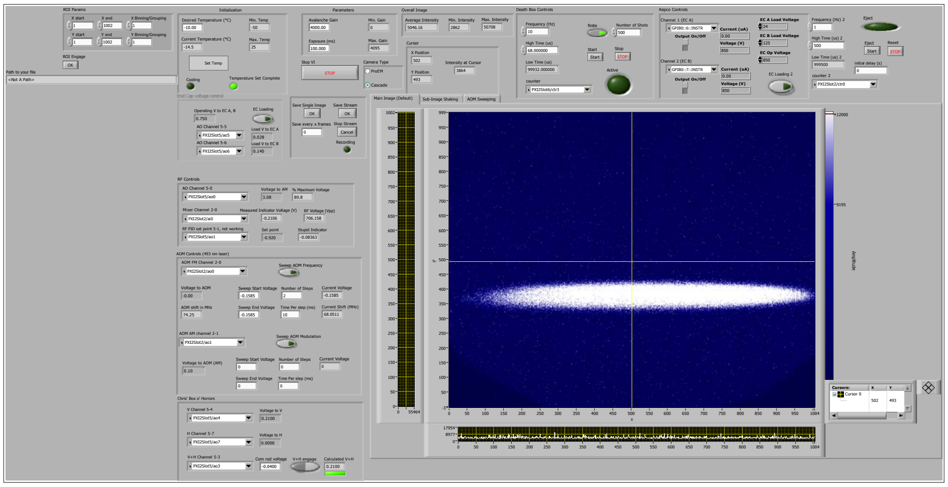


C.2.2. Apparatus Master Control

Apparatus master control including camera, pulse valve, trap voltages, and TOF initialization.

Page 1

camera control version 10.2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\camera control version 10.2.vi
Last modified on 9/4/2015 at 10:13 AM
Printed on 6/21/2016 at 12:43 PM



ROI Params
X start 1
X end 1002
X Binning/Grouping 1
Y start 1
Y end 1002
Y Binning/Grouping 1

Path to your file
<Not A Path>

Initialization
Desired Temperature (°C) -10.00
Min. Temp -50
Current Temperature (°C) -14.5
Max. Temp 25

Parameters
Avalanche Gain 4000.00
Min. Gain 0
Exposure (ms) 100.000
Max. Gain 4095

Camera Type
○ ProEM
● Cascade

Overall Image
Average Intensity 5046.16
Min. Intensity 2862
Max. Intensity 50708

Cursor
X Position 502
Y Position 493
Intensity at Cursor 3864

Death Box Controls
Frequency (Hz) 10
Number of Shots 500
High Time (us) 68.000000
Low Time (us) 99932.000000
counter PXI2Slot6/ctr3

Kepco Controls
Channel 1 (EC A) GPIB0::6::INSTR
Current (uA) 0.00
Voltage (V) 850
Channel 2 (EC B) GPIB0::7::INSTR
Current (uA) 0.00
Voltage (V) 850
EC A Load Voltage 24
EC B Load Voltage 125
EC Op Voltage 850
Frequency (Hz) 2 1
High Time (us) 2 500
Low Time (us) 2 999500
initial delay (s) 0
counter 2 PXI2Slot2/ctr0

End Cap voltage control
Operating V to EC A, B 0.750
AO Channel 5-5 PXI2Slot5/ao5
Load V to EC A 0.028
AO Channel 5-6 PXI2Slot5/ao6
Load V to EC B 0.140

Save every x frames 0

RF Controls
AO Channel 5-0 PXI2Slot5/ao0
Voltage to AM 3.08
% Maximum Voltage 80.8
Mixer Channel 2-0 PXI2Slot2/ai0
Measured Indicator Voltage (V) -0.2106
RF Voltage (Vpp) 706.158
RF PID set point 5-1, not working PXI2Slot5/ao1
Set point -0.920
Stupid Indicator -0.08363

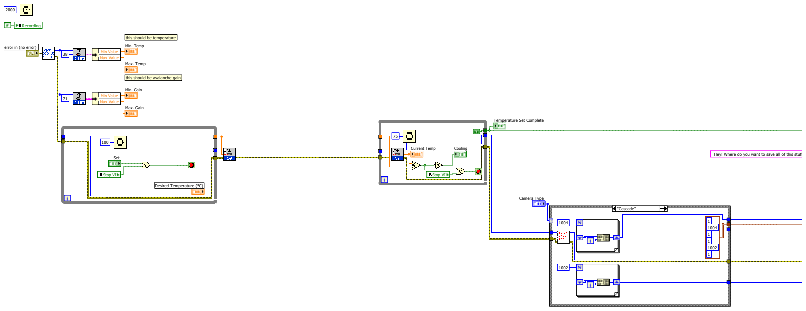
AOM Controls (493 nm laser)
AOM FM Channel 2-0 PXI2Slot2/ao0
Voltage to AOM -0.00
Sweep Start Voltage -0.1585
Number of Steps 2
Current Voltage -0.1585
AOM shift in MHz 74.25
Sweep End Voltage -0.1585
Time Per step (ms) 10
Current Shift (MHz) 68.0511
AOM AM channel 2-1 PXI2Slot2/ao1
Voltage to AOM (AM) 0.10
Sweep Start Voltage 0
Number of Steps 0
Current Voltage 0
Sweep End Voltage 0
Time Per step (ms) 0

Chris' Box o' Horrors
V Channel 5-4 PXI2Slot5/ao4
Voltage to V 0.2100
H Channel 5-7 PXI2Slot5/ao7
Voltage to H 0.0000
V+H Channel 5-3 PXI2Slot5/ao3
Com rod voltage -0.0400
Calculated V+H 0.2100

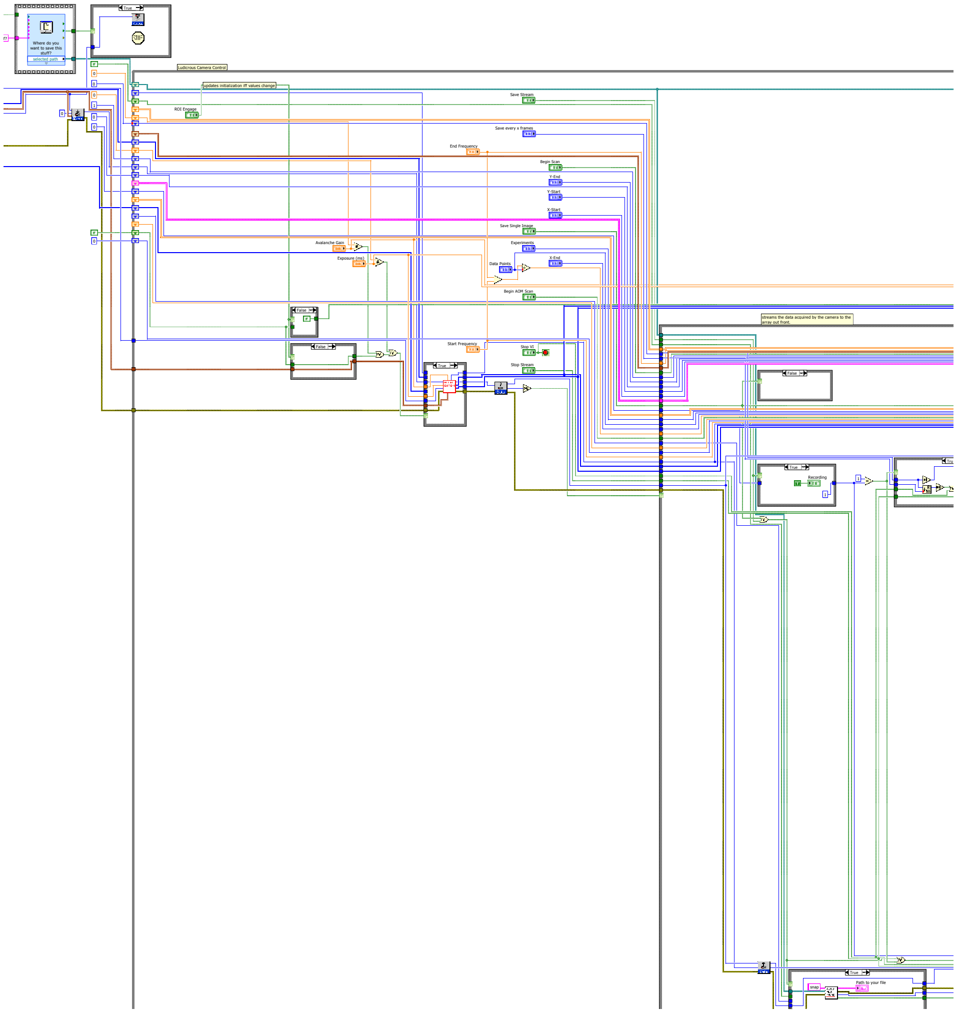
Cursors:	X	Y
Cursor 0	502	493

Page 1

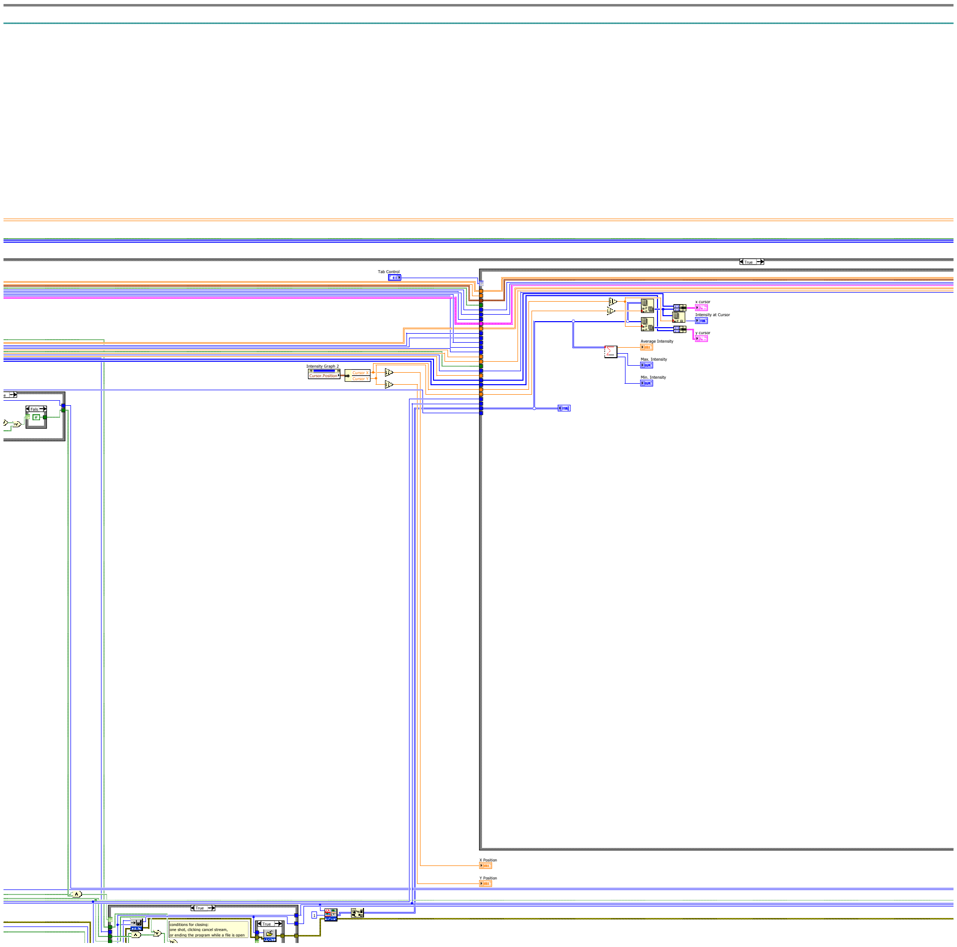
camera control version 10.2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\camera control version 10.2.vi
Last modified on 9/4/2015 at 10:13 AM
Printed on 6/21/2016 at 12:42 PM



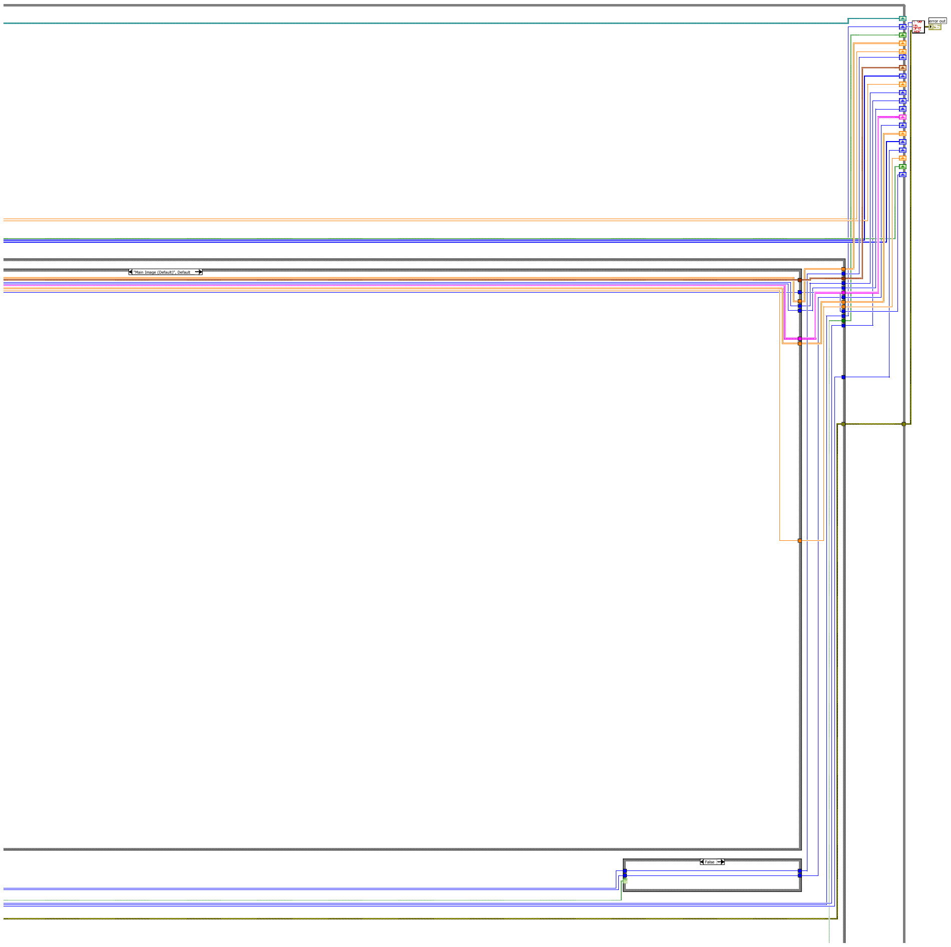
camera control version 10.2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\camera control version 10.2.vi
Last modified on 9/4/2015 at 10:13 AM
Printed on 6/21/2016 at 12:42 PM



camera control version 10.2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\camera control version 10.2.vi
Last modified on 9/4/2015 at 10:13 AM
Printed on 6/21/2016 at 12:42 PM

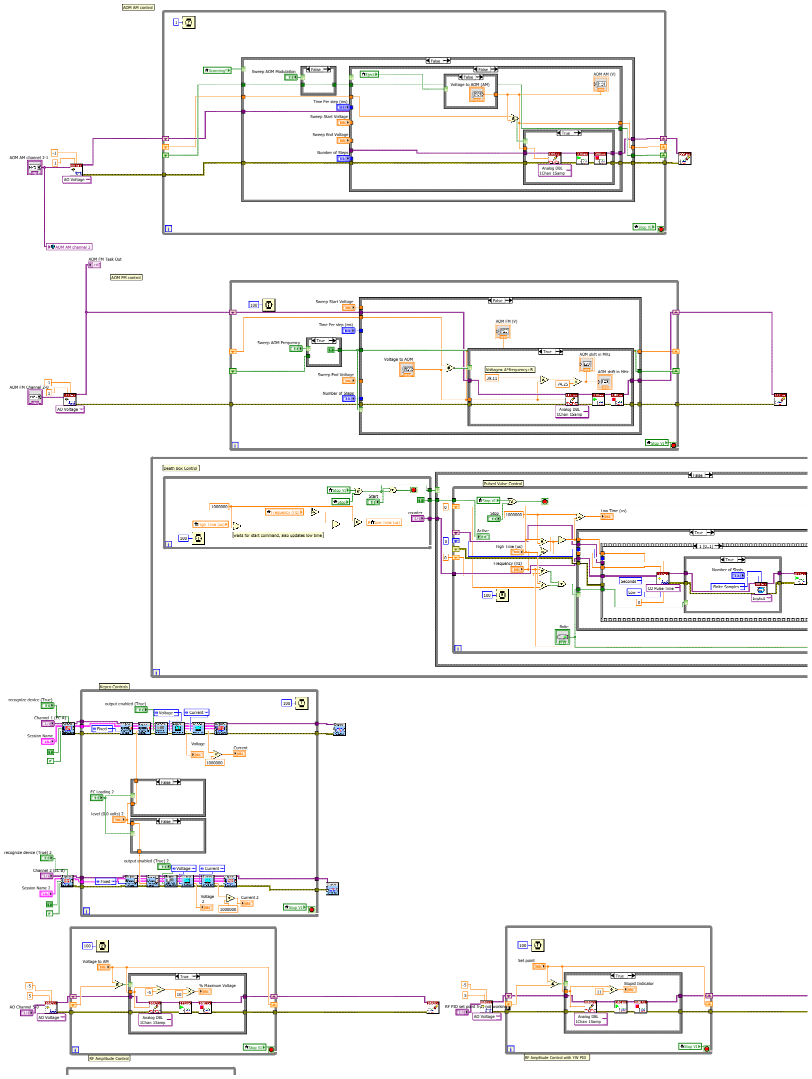


camera control version 10.2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\camera control version 10.2.vi
Last modified on 9/4/2015 at 10:13 AM
Printed on 6/21/2016 at 12:42 PM



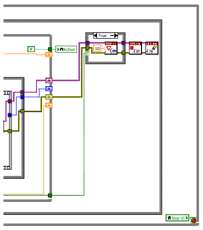
Page 5

camera control version 10.2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\camera control version 10.2.vi
Last modified on 9/4/2015 at 10:13 AM
Printed on 6/21/2016 at 12:42 PM



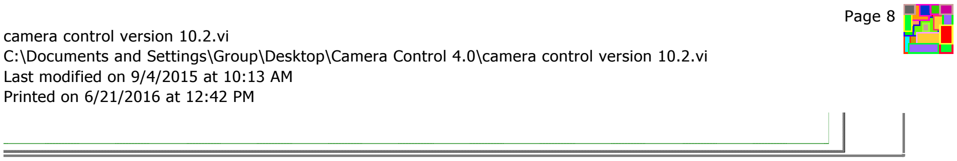
Page 6

camera control version 10.2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\camera control version 10.2.vi
Last modified on 9/4/2015 at 10:13 AM
Printed on 6/21/2016 at 12:42 PM



Page 7

camera control version 10.2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\camera control version 10.2.vi
Last modified on 9/4/2015 at 10:13 AM
Printed on 6/21/2016 at 12:42 PM

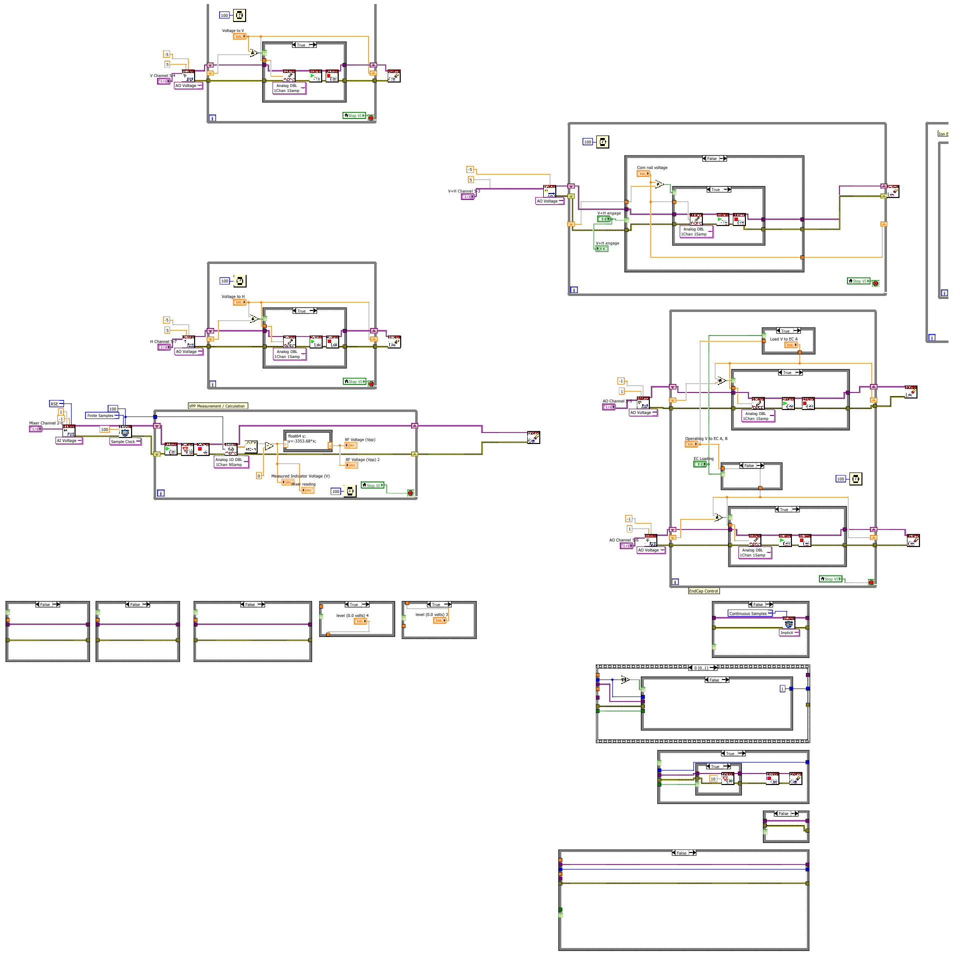


Page 8

camera control version 10.2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\camera control version 10.2.vi
Last modified on 9/4/2015 at 10:13 AM
Printed on 6/21/2016 at 12:42 PM

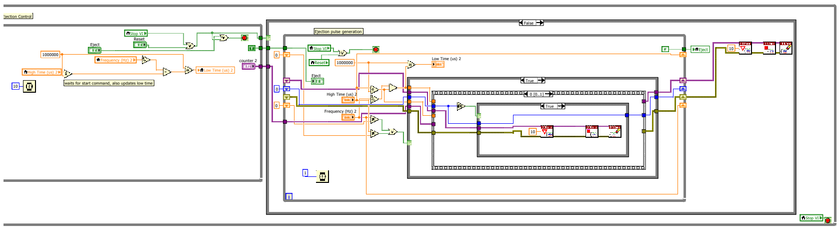
Page 9

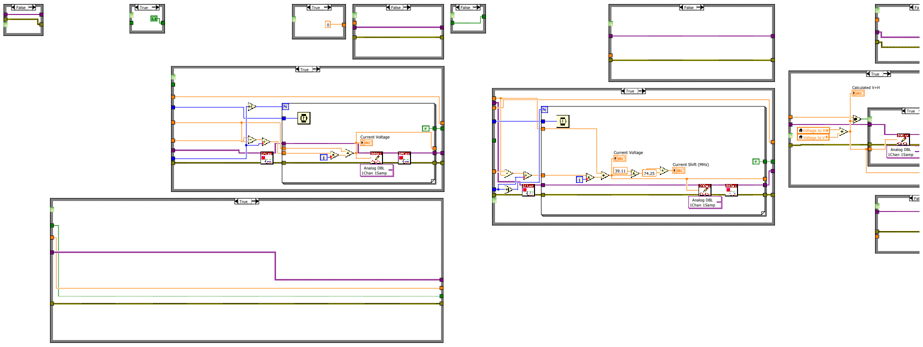
camera control version 10.2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\camera control version 10.2.vi
Last modified on 9/4/2015 at 10:13 AM
Printed on 6/21/2016 at 12:42 PM



Page 10

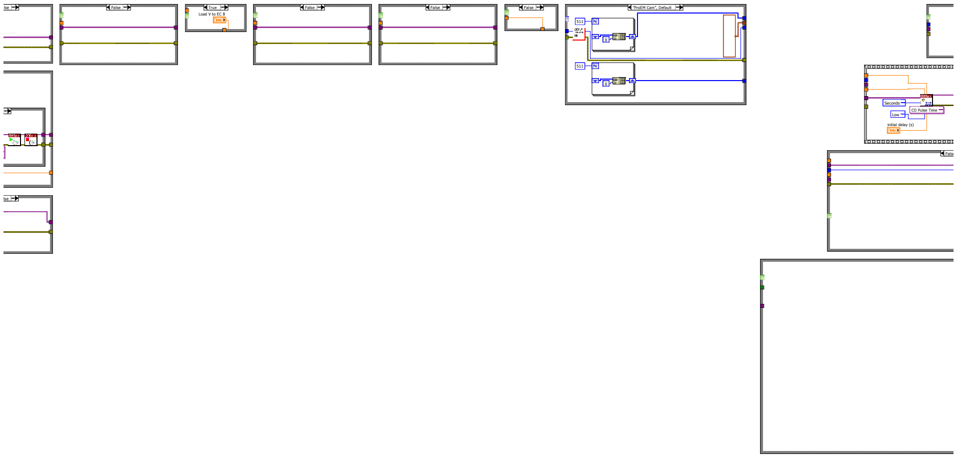
camera control version 10.2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\camera control version 10.2.vi
Last modified on 9/4/2015 at 10:13 AM
Printed on 6/21/2016 at 12:42 PM



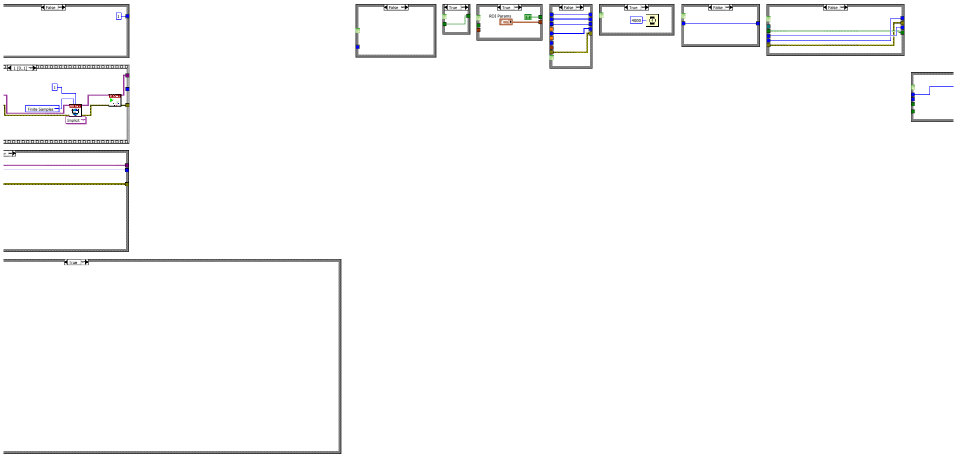


Page 11

camera control version 10.2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\camera control version 10.2.vi
Last modified on 9/4/2015 at 10:13 AM
Printed on 6/21/2016 at 12:42 PM

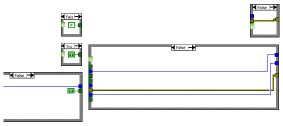


camera control version 10.2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\camera control version 10.2.vi
Last modified on 9/4/2015 at 10:13 AM
Printed on 6/21/2016 at 12:42 PM

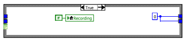


Page 13

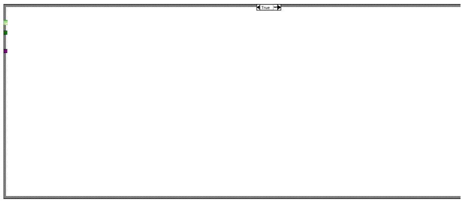
camera control version 10.2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\camera control version 10.2.vi
Last modified on 9/4/2015 at 10:13 AM
Printed on 6/21/2016 at 12:42 PM



camera control version 10.2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\camera control version 10.2.vi
Last modified on 9/4/2015 at 10:13 AM
Printed on 6/21/2016 at 12:42 PM



camera control version 10.2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\camera control version 10.2.vi
Last modified on 9/4/2015 at 10:13 AM
Printed on 6/21/2016 at 12:42 PM

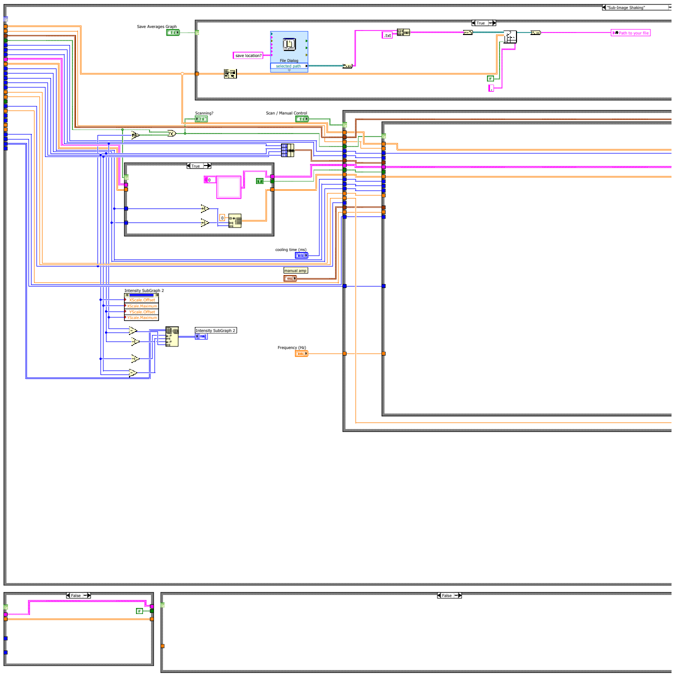


Page 16

camera control version 10.2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\camera control version 10.2.vi
Last modified on 9/4/2015 at 10:13 AM
Printed on 6/21/2016 at 12:42 PM

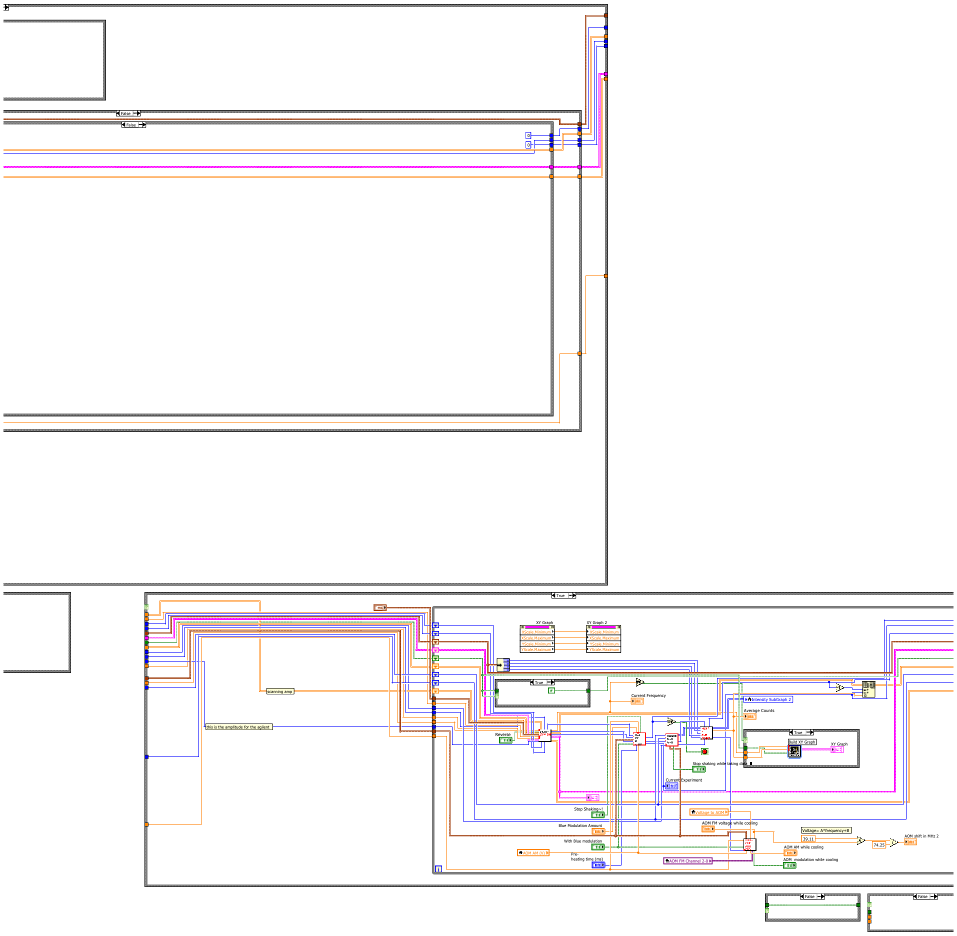
Page 17

camera control version 10.2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\camera control version 10.2.vi
Last modified on 9/4/2015 at 10:13 AM
Printed on 6/21/2016 at 12:42 PM

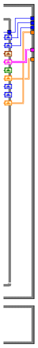


Page 18

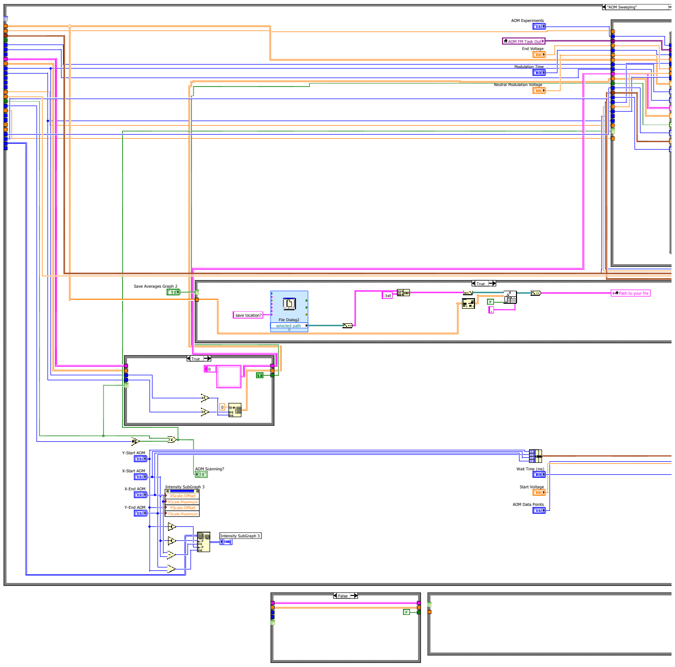
camera control version 10.2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\camera control version 10.2.vi
Last modified on 9/4/2015 at 10:13 AM
Printed on 6/21/2016 at 12:42 PM



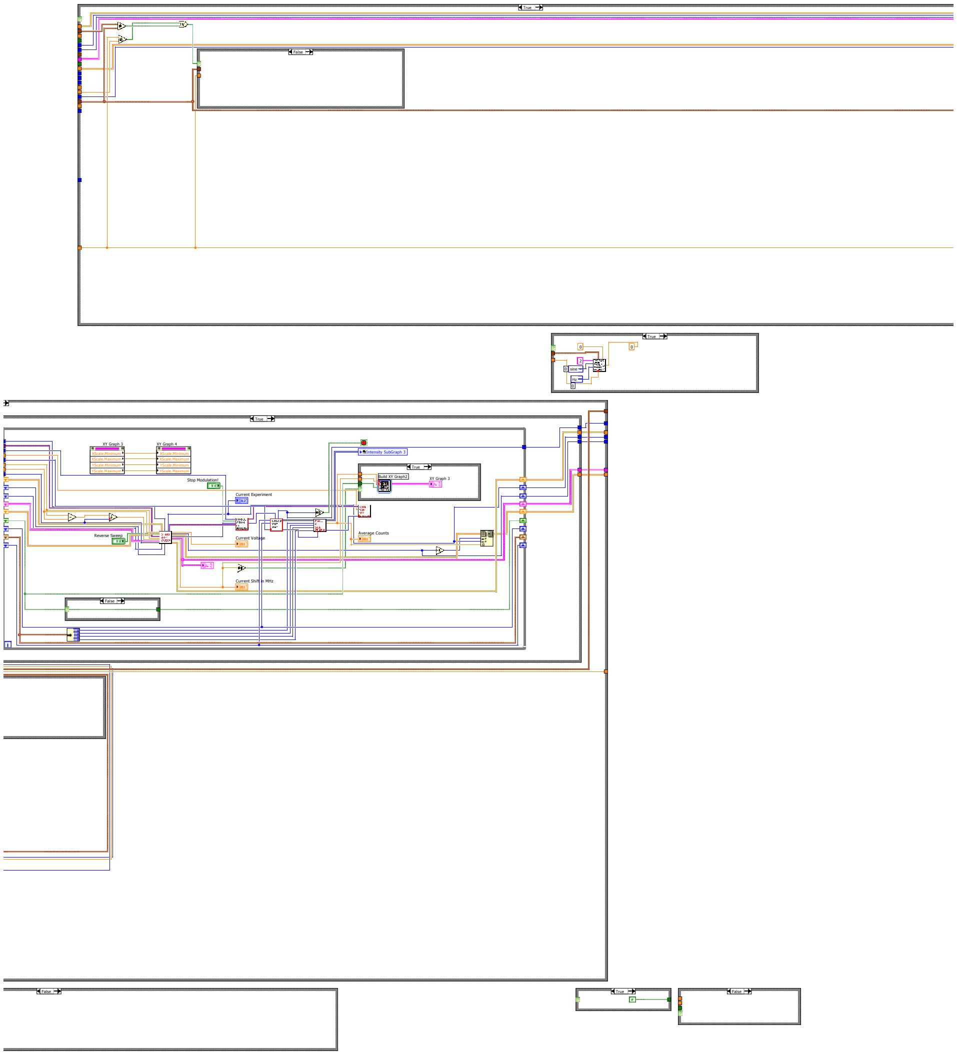
camera control version 10.2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\camera control version 10.2.vi
Last modified on 9/4/2015 at 10:13 AM
Printed on 6/21/2016 at 12:42 PM



camera control version 10.2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\camera control version 10.2.vi
Last modified on 9/4/2015 at 10:13 AM
Printed on 6/21/2016 at 12:42 PM



camera control version 10.2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\camera control version 10.2.vi
Last modified on 9/4/2015 at 10:13 AM
Printed on 6/21/2016 at 12:42 PM



camera control version 10.2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\camera control version 10.2.vi
Last modified on 9/4/2015 at 10:13 AM
Printed on 6/21/2016 at 12:42 PM

Page 24

camera control version 10.2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\camera control version 10.2.vi
Last modified on 9/4/2015 at 10:13 AM
Printed on 6/21/2016 at 12:42 PM

Page 25

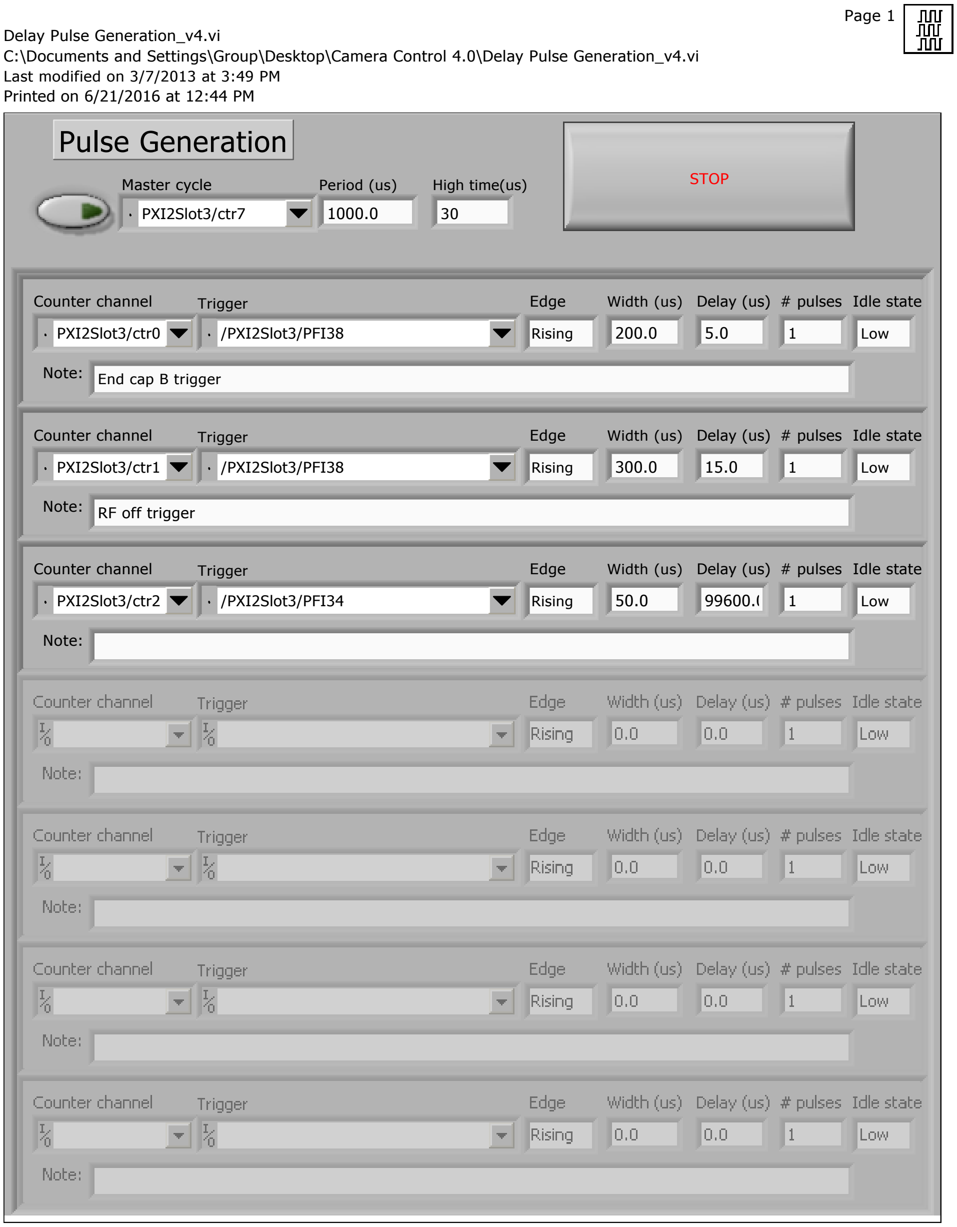
camera control version 10.2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\camera control version 10.2.vi
Last modified on 9/4/2015 at 10:13 AM
Printed on 6/21/2016 at 12:42 PM

Page 26

camera control version 10.2.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\camera control version 10.2.vi
Last modified on 9/4/2015 at 10:13 AM
Printed on 6/21/2016 at 12:43 PM

C.2.3. Pulse Delay Generator

This code generates the TTL timing pulses for synchronizing the Nd:YAG lasers and pulse valve with a variable delay between each.



Page 1

Delay Pulse Generation_v4.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\Delay Pulse Generation_v4.vi
Last modified on 3/7/2013 at 3:49 PM
Printed on 6/21/2016 at 12:44 PM

Pulse Generation

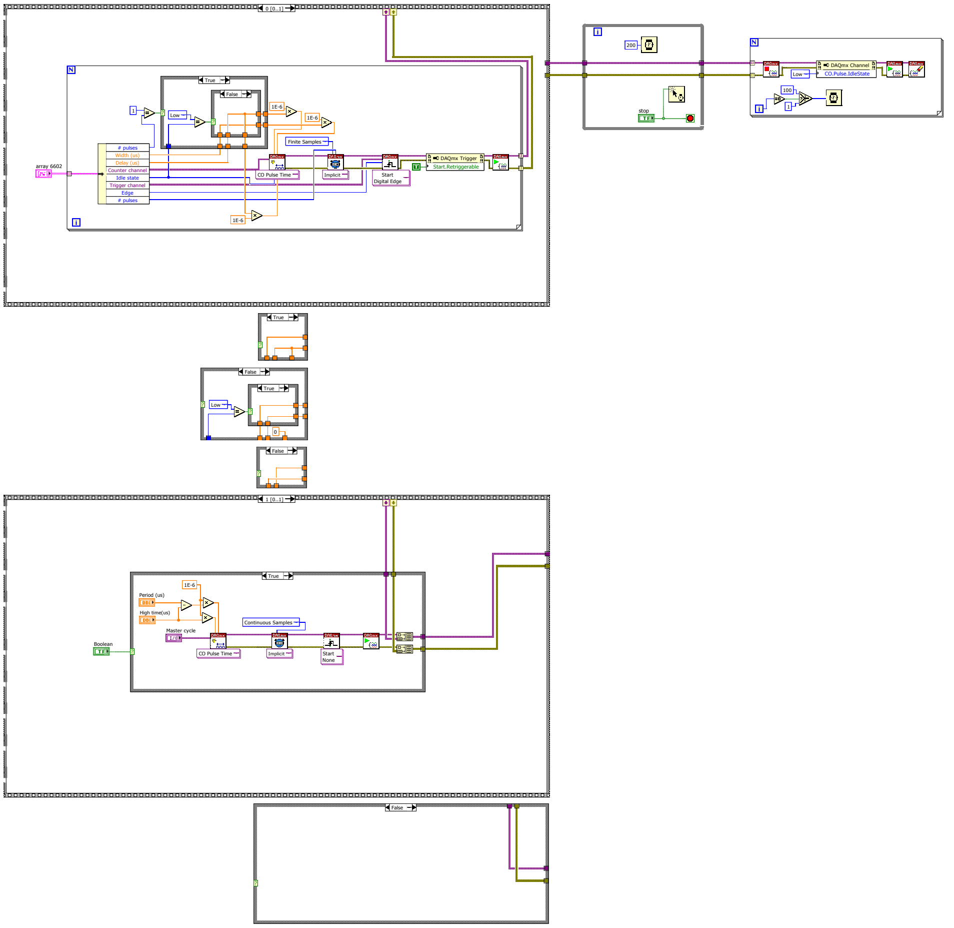
Master cycle	Period (us)	High time(us)
PXI2Slot3/ctr7	1000.0	30

STOP

Counter channel	Trigger	Edge	Width (us)	Delay (us)	# pulses	Idle state	Note
PXI2Slot3/ctr0	/PXI2Slot3/PFI38	Rising	200.0	5.0	1	Low	End cap B trigger
PXI2Slot3/ctr1	/PXI2Slot3/PFI38	Rising	300.0	15.0	1	Low	RF off trigger
PXI2Slot3/ctr2	/PXI2Slot3/PFI34	Rising	50.0	99600.(	1	Low	
		Rising	0.0	0.0	1	Low	
		Rising	0.0	0.0	1	Low	
		Rising	0.0	0.0	1	Low	
		Rising	0.0	0.0	1	Low	

Page 1

Delay Pulse Generation_v4.vi
C:\Documents and Settings\Group\Desktop\Camera Control 4.0\Delay Pulse Generation_v4.vi
Last modified on 3/7/2013 at 3:49 PM
Printed on 6/21/2016 at 12:44 PM



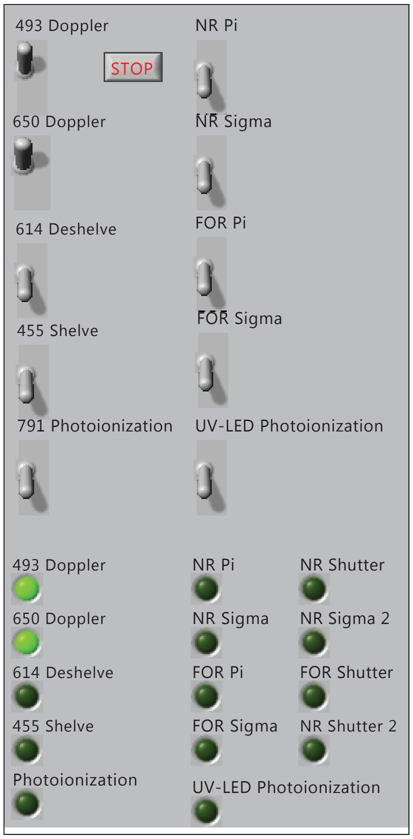
C.3. Motional Ground State Cooling Project

C.3.1. AOM Controls

The code below provides control over switching AOMs via assigning the associated 6602 counter channel a task with a new default state and run the task. In the block diagram, all top-level False boxes are empty, and all inner state boxes contain inverted constants for the opposite Boolean case. Created by MRD and modified by CMS.

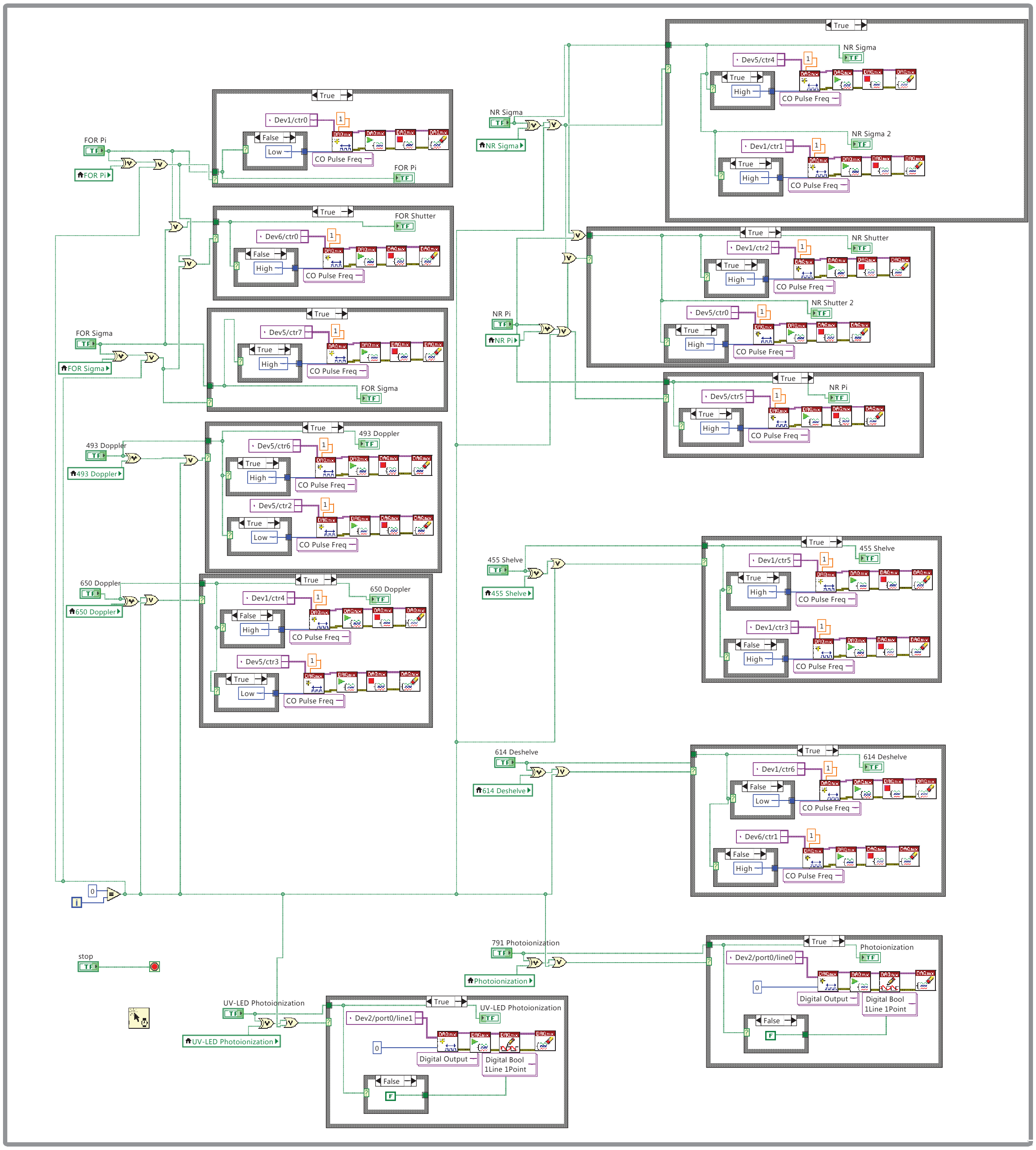
Page 1

AOM Controls.vi
C:\Users\lynx\Documents\LabView Program\AOM Controls.vi
Last modified on 5/6/2016 at 4:13 PM
Printed on 5/6/2016 at 4:14 PM



Page 1

AOM Controls.vi
C:\Users\lynx\Documents\LabView Program\AOM Controls.vi
Last modified on 5/6/2016 at 4:13 PM
Printed on 5/6/2016 at 4:13 PM

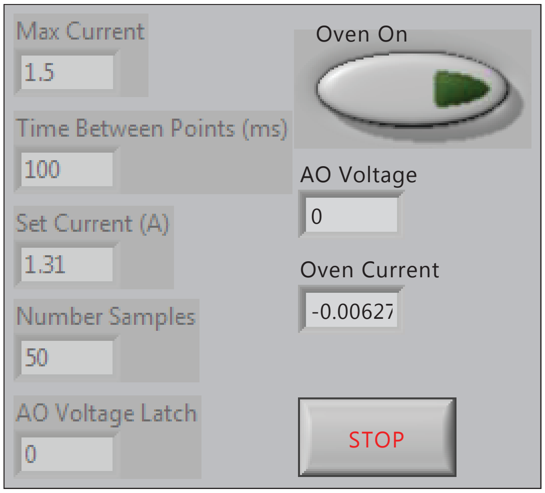


C.3.2. Barium Oven Control

The code below provides control over the barium oven current supply to ramp the current up and down slowly.

Page 1

Ba Oven Control.vi
C:\Users\lynx\Documents\LabView Program\Single Program\Ba Oven Control.vi
Last modified on 11/30/2015 at 12:19 PM
Printed on 5/6/2016 at 4:16 PM



Max Current
1.5

Time Between Points (ms)
100

Set Current (A)
1.31

Number Samples
50

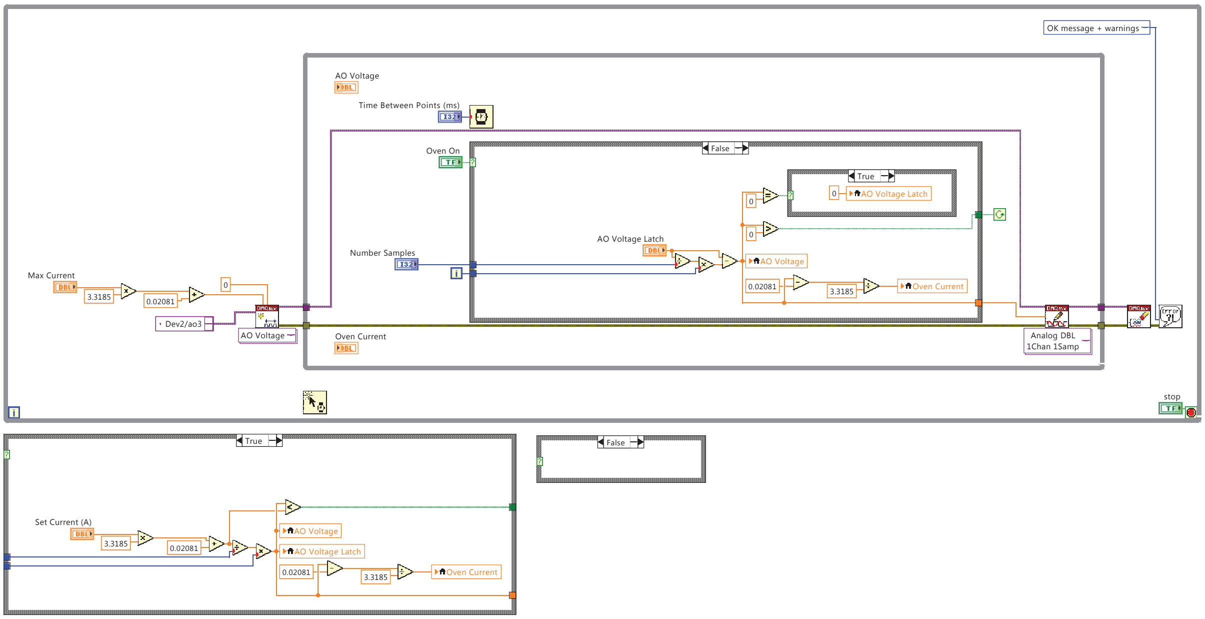
AO Voltage Latch
0

AO Voltage
0

Oven Current
-0.00627

Page 1

Ba Oven Control.vi
C:\Users\lynx\Documents\LabView Program\Single Program\Ba Oven Control.vi
Last modified on 11/30/2015 at 12:19 PM
Printed on 5/6/2016 at 4:16 PM

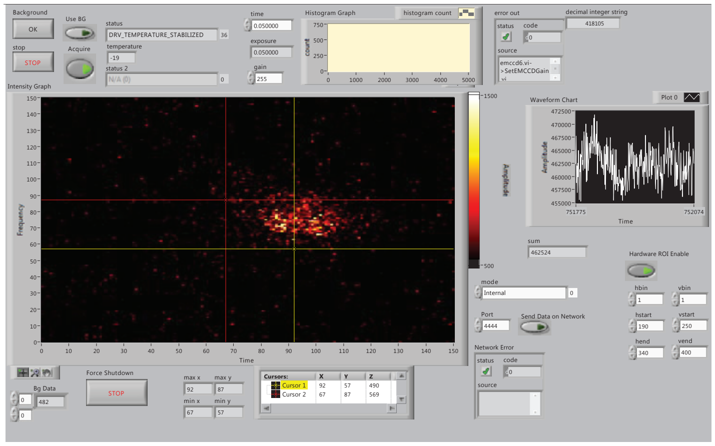


C.3.3. Camera Program

The code below provides EMCCD camera control including background subtraction, camera counts vs time, and data server functions. Created by YWL. Modified by MRD.

Page 1

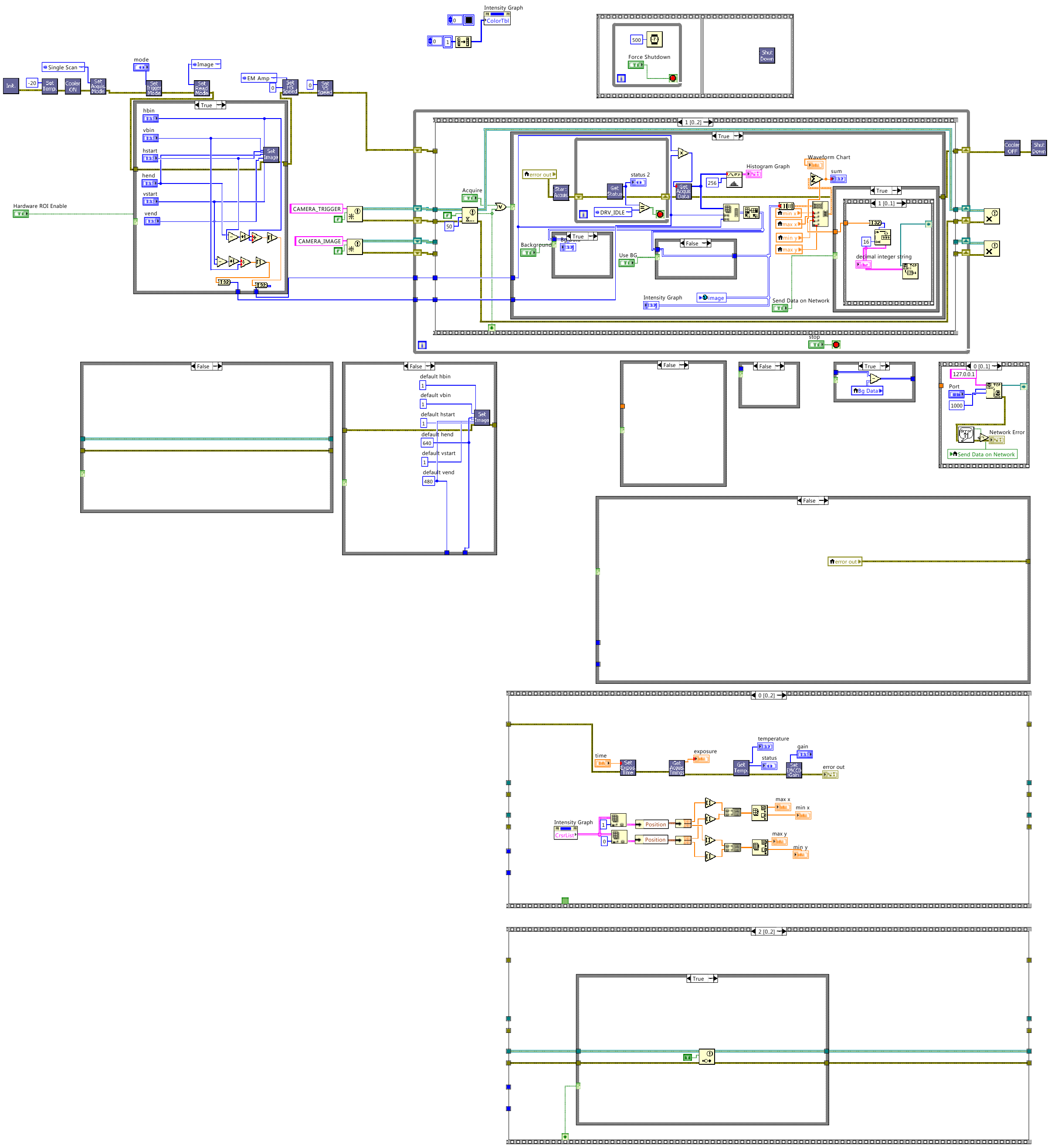
emccd6.vi
C:\Users\lynx\Documents\LabView Program\Single Program\emccd6.vi
Last modified on 1/28/2016 at 5:10 PM
Printed on 5/6/2016 at 4:16 PM



status: DRV_TEMPERATURE_STABILIZED 36
temperature: -19
status 2: N/A (0) 0
time: 0.050000
exposure: 0.050000
gain: 255
error out: status ✓ code 0
source: emccd6.vi->SetEMCCDGain vi
decimal integer string: 418105
sum: 462524
mode: Internal 0
Port: 4444
Network Error: status ✓ code 0
source:
hbin: 1
vbin: 1
hstart: 190
vstart: 250
hend: 340
vend: 400
Bg Data: 0 0 482
max x: 92
max y: 87
min x: 67
min y: 57

Cursors:	X	Y	Z
Cursor 1	92	57	490
Cursor 2	67	87	569

emccd6.vi
C:\Users\lynx\Documents\LabView Program\Single Program\emccd6.vi
Last modified on 1/28/2016 at 5:10 PM
Printed on 5/6/2016 at 4:19 PM

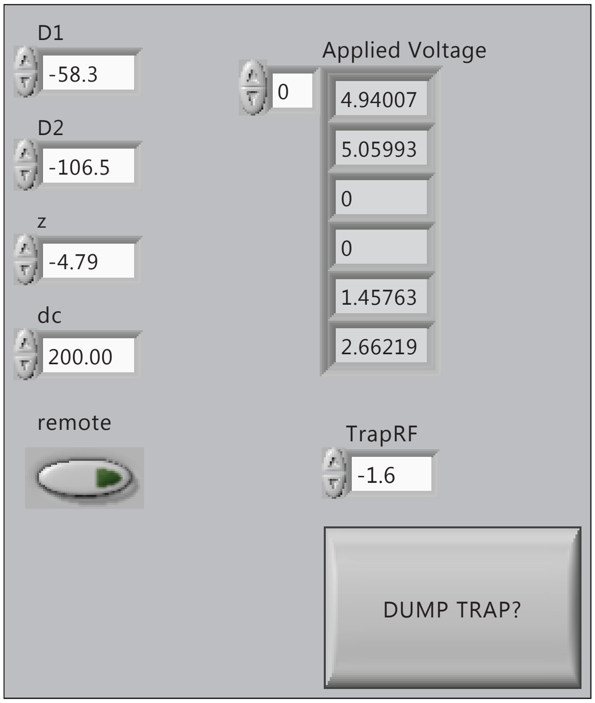


C.3.4. DC Voltage Control

The code below provides the DC analog signals to the high voltage amplifier circuit (App. A.3.3) and RF AM voltage. Created by YWL.

Page 1

DC Voltage Control.vi
C:\Users\lynx\Documents\LabView Program\Orange\DC Voltage Control.vi
Last modified on 1/23/2016 at 9:13 AM
Printed on 5/10/2016 at 10:48 AM



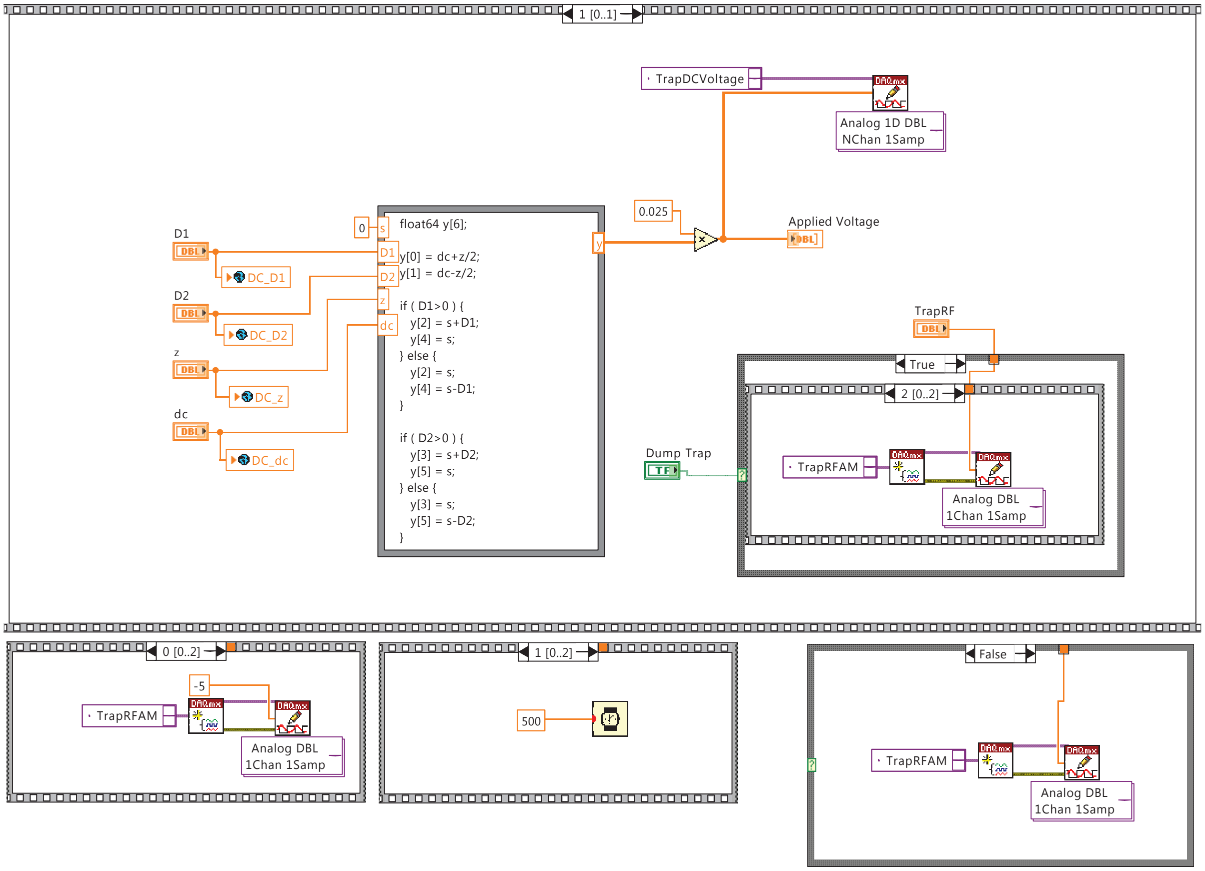
D1: -58.3
D2: -106.5
z: -4.79
dc: 200.00

Applied Voltage: 0
4.94007
5.05993
0
0
1.45763
2.66219

TrapRF: -1.6

Page 1

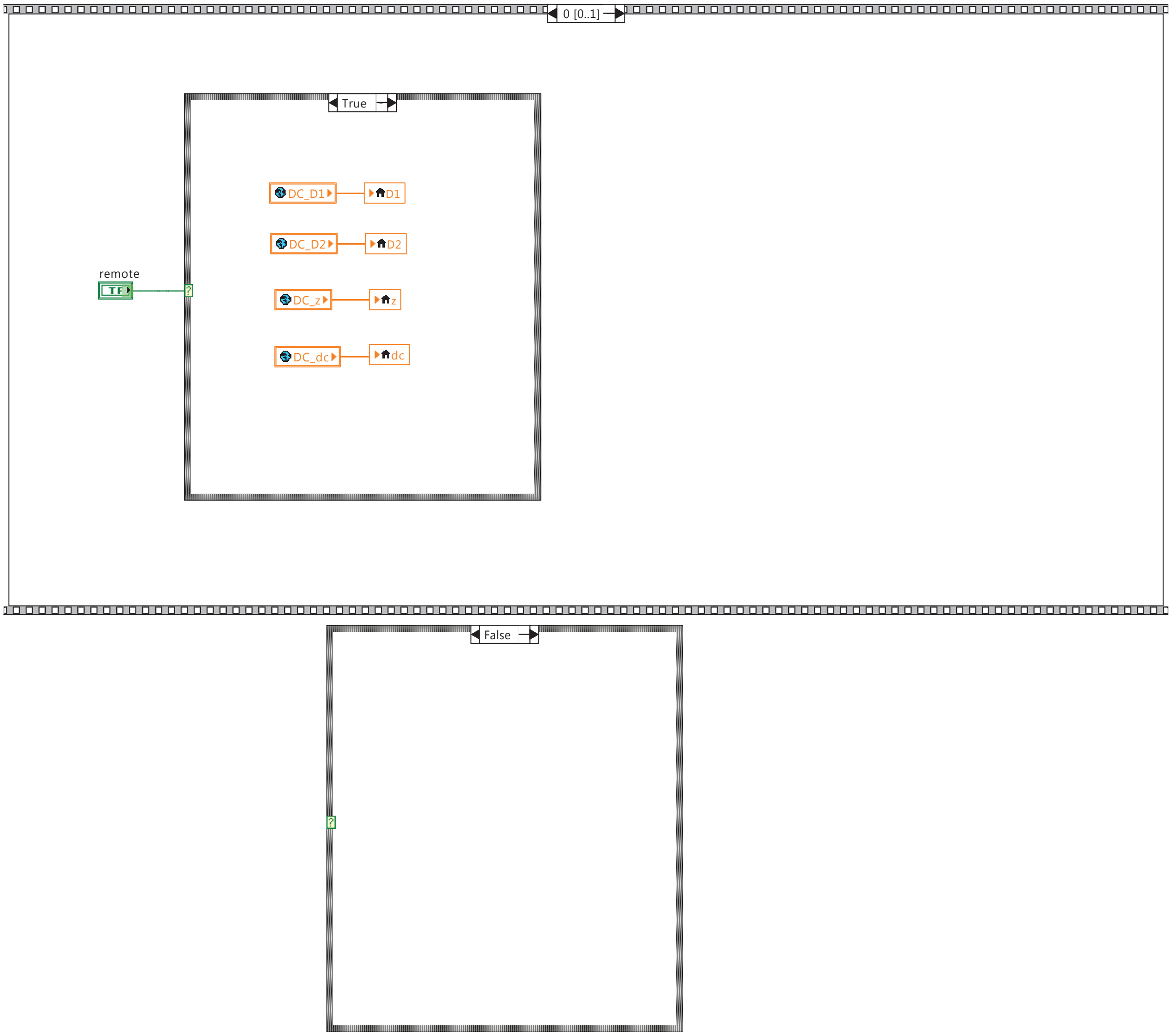
DC Voltage Control.vi
C:\Users\lynx\Documents\LabView Program\Orange\DC Voltage Control.vi
Last modified on 1/23/2016 at 9:13 AM
Printed on 5/10/2016 at 10:48 AM



```
float64 y[6];

y[0] = dc+z/2;
y[1] = dc-z/2;

if ( D1>0 ) {
   y[2] = s+D1;
   y[4] = s;
} else {
   y[2] = s;
   y[4] = s-D1;
}

if ( D2>0 ) {
   y[3] = s+D2;
   y[5] = s;
} else {
   y[3] = s;
   y[5] = s-D2;
}
```

Page 2

DC Voltage Control.vi
C:\Users\lynx\Documents\LabView Program\Orange\DC Voltage Control.vi
Last modified on 1/23/2016 at 9:13 AM
Printed on 5/10/2016 at 10:48 AM

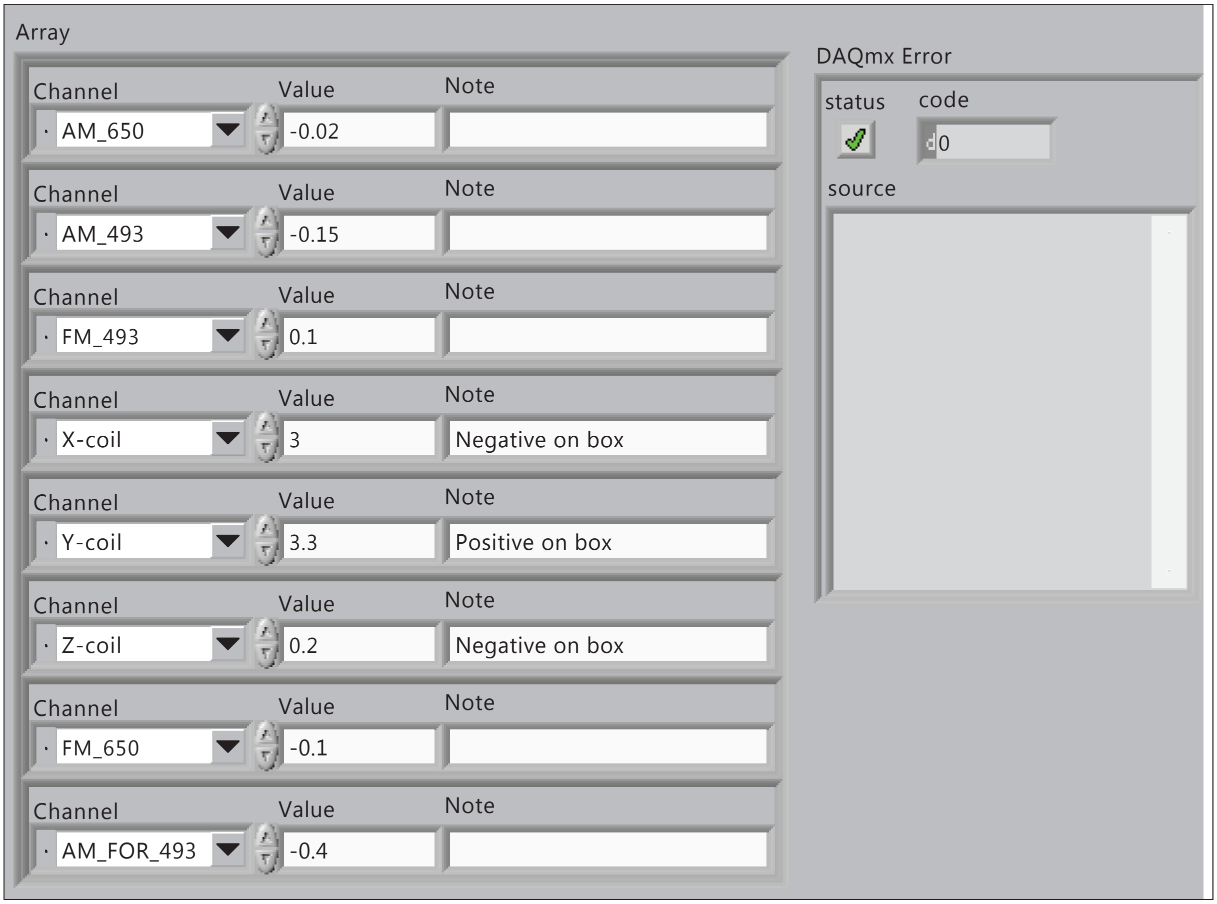


C.3.5. Laser and Magnetic Field Control

The code below provides frequency and intensity control of laser sources and magnetic field set points. Created by YWL.

Page 1

LaserControl.vi
C:\Users\lynx\Documents\LabView Program\Orange\LaserControl.vi
Last modified on 1/19/2016 at 10:34 AM
Printed on 5/6/2016 at 4:15 PM



Array

Channel	Value	Note
AM_650	-0.02	
AM_493	-0.15	
FM_493	0.1	
X-coil	3	Negative on box
Y-coil	3.3	Positive on box
Z-coil	0.2	Negative on box
FM_650	-0.1	
AM_FOR_493	-0.4	

DAQmx Error

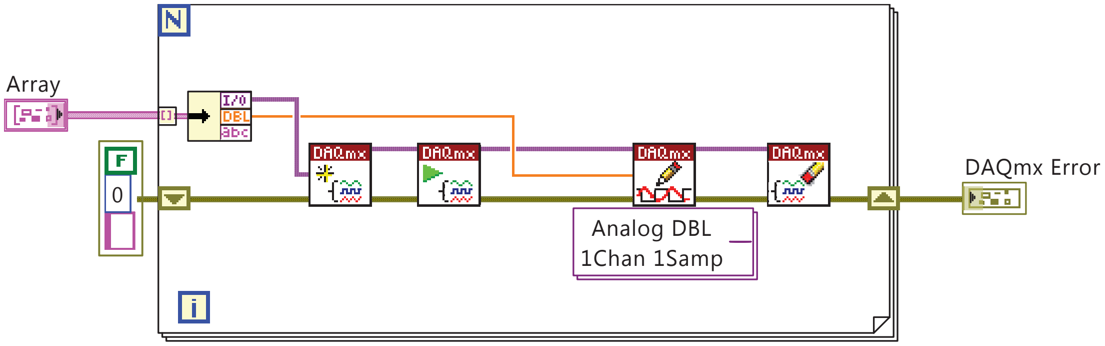
status ✓

code 0

source

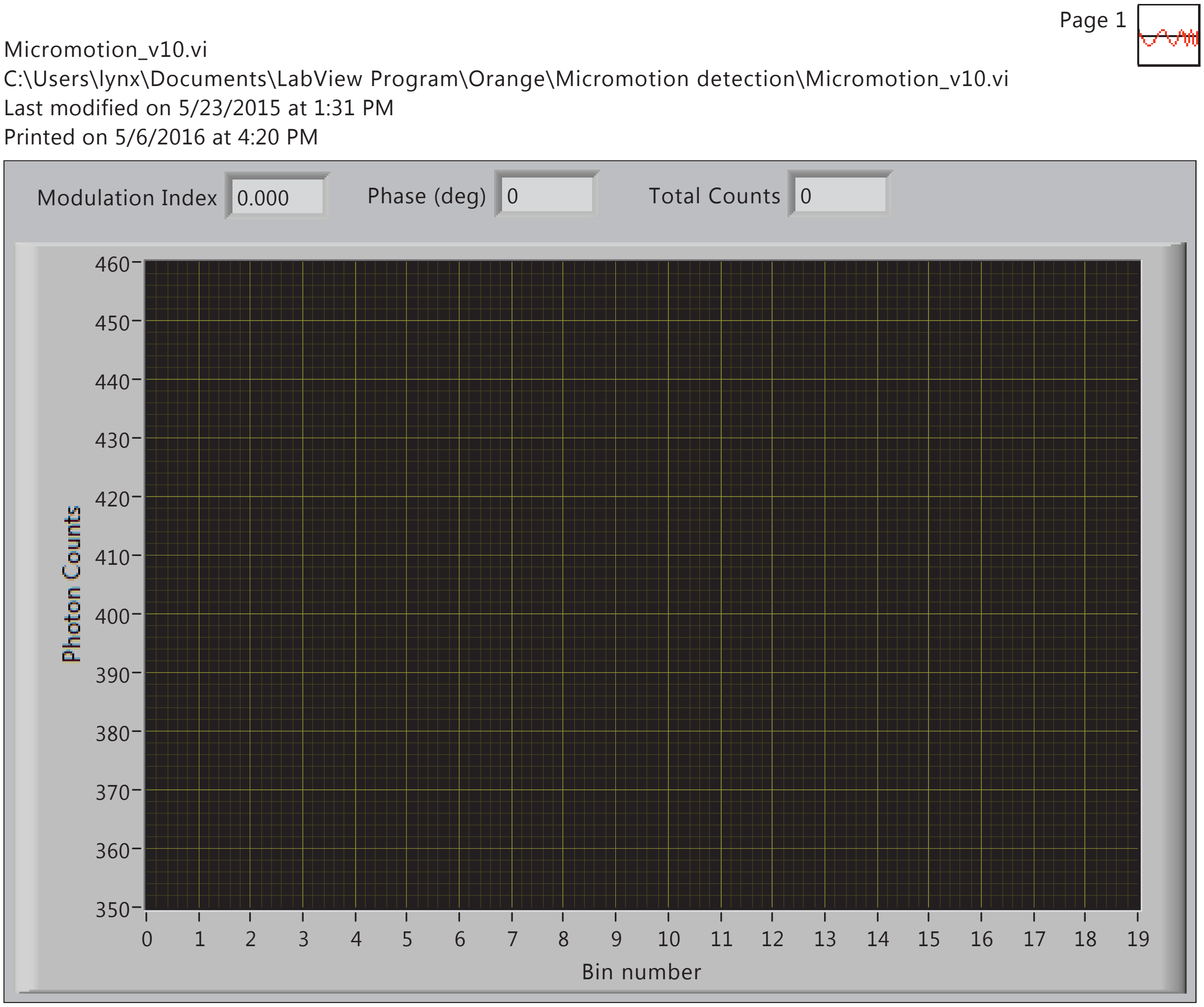
Page 1

LaserControl.vi
C:\Users\lynx\Documents\LabView Program\Orange\LaserControl.vi
Last modified on 1/19/2016 at 10:34 AM
Printed on 5/6/2016 at 4:15 PM



C.3.6. Micromotion Compensation

The code below provides excess micromotion compensation for the single-ion work using a Stanford Research Systems SR400 photon counter. Created by YWL.



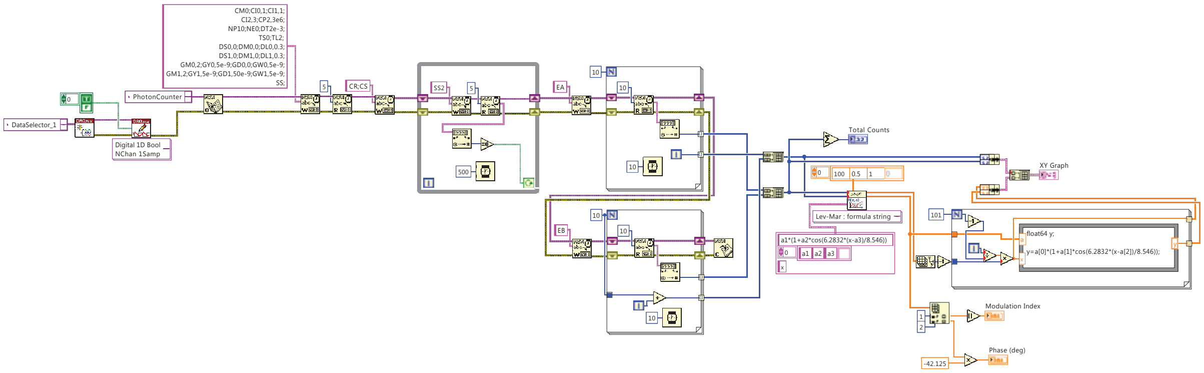
Page 1

Micromotion_v10.vi
C:\Users\lynx\Documents\LabView Program\Orange\Micromotion detection\Micromotion_v10.vi
Last modified on 5/23/2015 at 1:31 PM
Printed on 5/6/2016 at 4:20 PM

Modulation Index 0.000 Phase (deg) 0 Total Counts 0

Page 1

Micromotion_v10.vi
C:\Users\lynx\Documents\LabView Program\Orange\Micromotion detection\Micromotion_v10.vi
Last modified on 5/23/2015 at 1:31 PM
Printed on 5/6/2016 at 4:21 PM



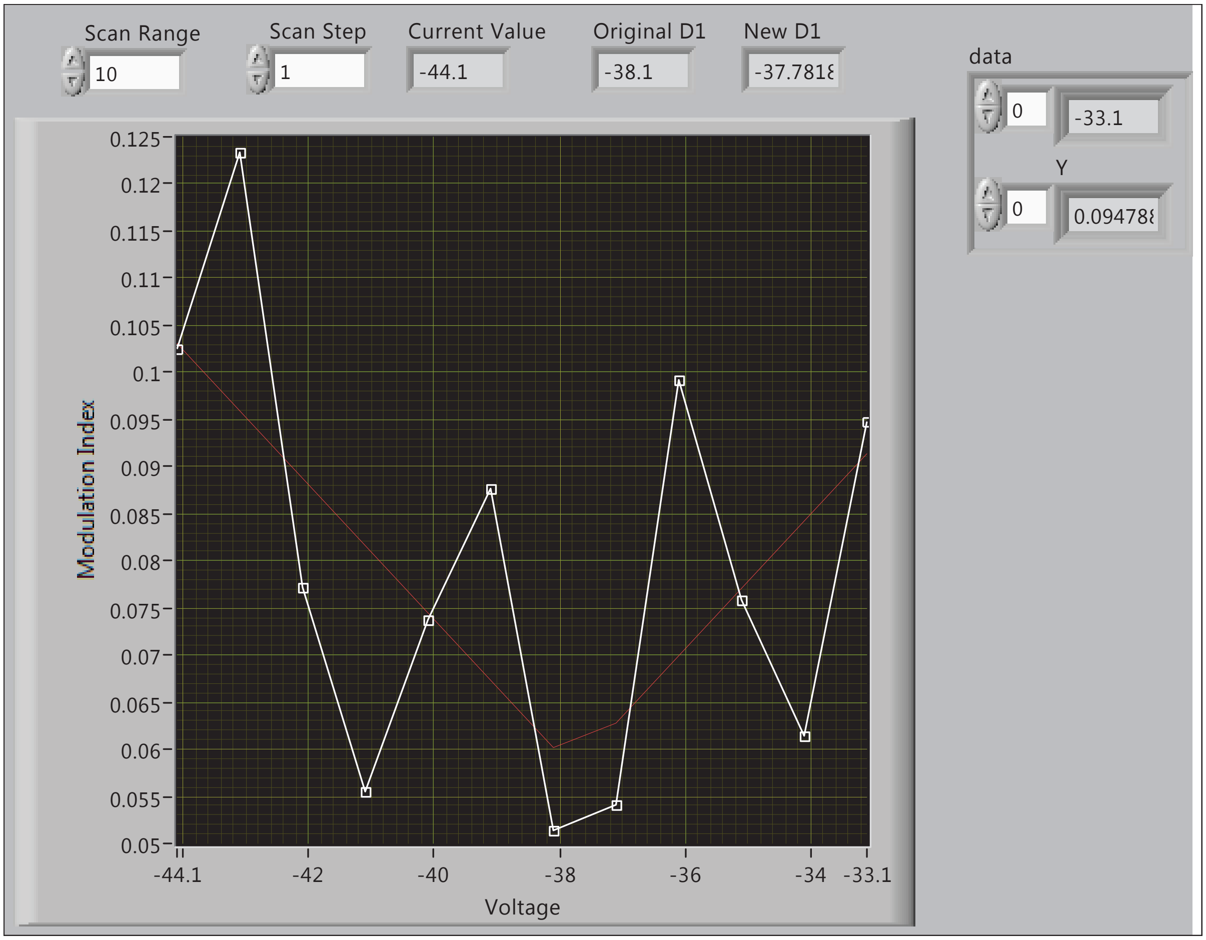
```
CM0;CI0,1;CI1,1;
CI2,3;CP2,3e6;
NP10;NE0;DT2e-3;
TS0;TL2;
DS0,0;DM0,0;DL0,0.3;
DS1,0;DM1,0;DL1,0.3;
GM0,2;GY0,5e-9;GD0,0;GW0,5e-9;
GM1,2;GY1,5e-9;GD1,50e-9;GW1,5e-9;
SS;
```

```
a1*(1+a2*cos(6.2832*(x-a3)/8.546))
```

```
float64 y;

y=a[0]*(1+a[1]*cos(6.2832*(x-a[2])/8.546));
```

Page 1

Micromotion Scan_D1_v10.vi
C:\Users\lynx\Documents\LabView Program\Orange\Micromotion detection\Micromotion Scan_D1_v10.vi
Last modified on 7/17/2015 at 1:16 PM
Printed on 5/6/2016 at 4:21 PM



Scan Range 10

Scan Step 1

Current Value -44.1

Original D1 -38.1

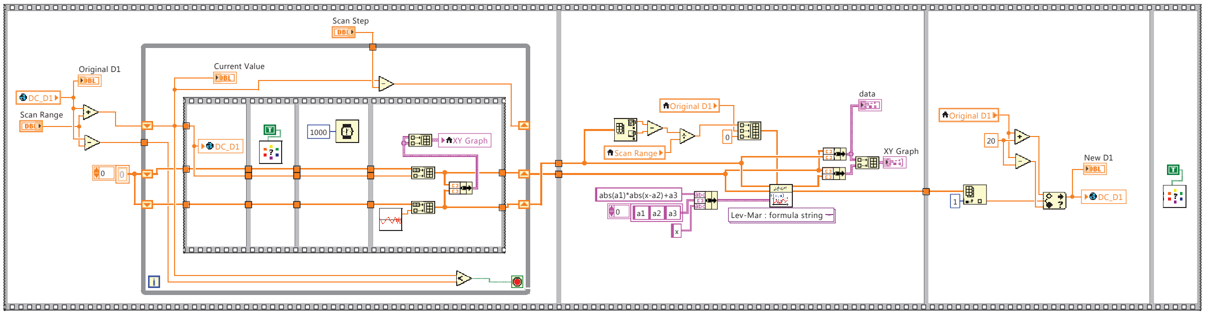
New D1 -37.7818

data 0 -33.1

Y 0 0.094788

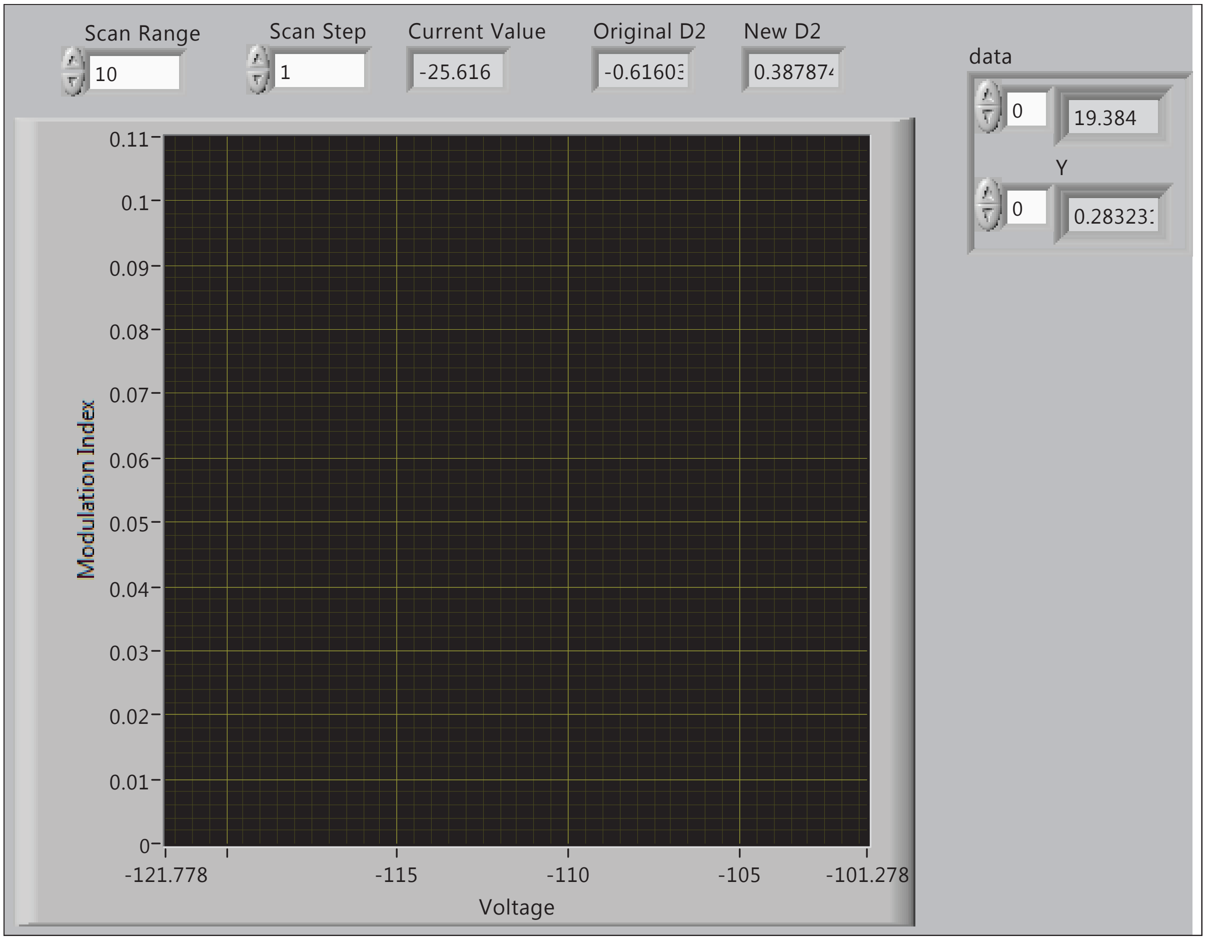
Page 1

Micromotion Scan_D1_v10.vi
C:\Users\lynx\Documents\LabView Program\Orange\Micromotion detection\Micromotion Scan_D1_v10.vi
Last modified on 7/17/2015 at 1:16 PM
Printed on 5/6/2016 at 4:22 PM



Page 1

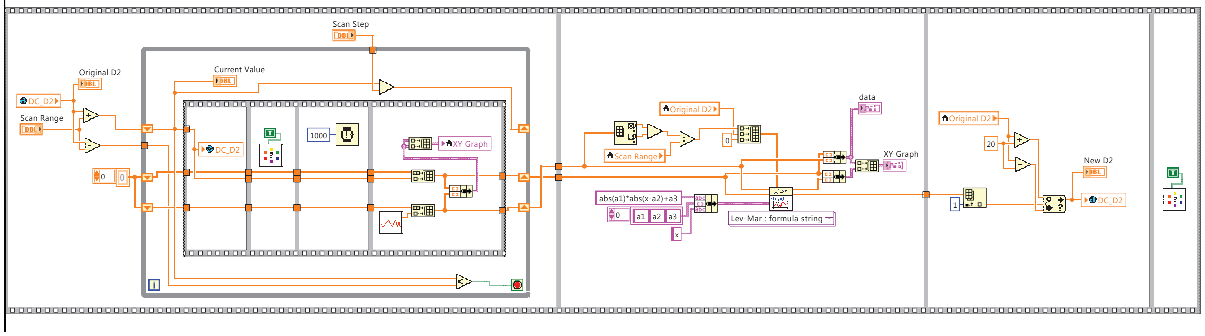
Micromotion Scan_D2_v10.vi
C:\Users\lynx\Documents\LabView Program\Orange\Micromotion detection\Micromotion Scan_D2_v10.vi
Last modified on 7/21/2015 at 11:06 AM
Printed on 5/6/2016 at 4:22 PM



Scan Range: 10
Scan Step: 1
Current Value: -25.616
Original D2: -0.61603
New D2: 0.387874
data: 0 | 19.384
Y: 0 | 0.283231

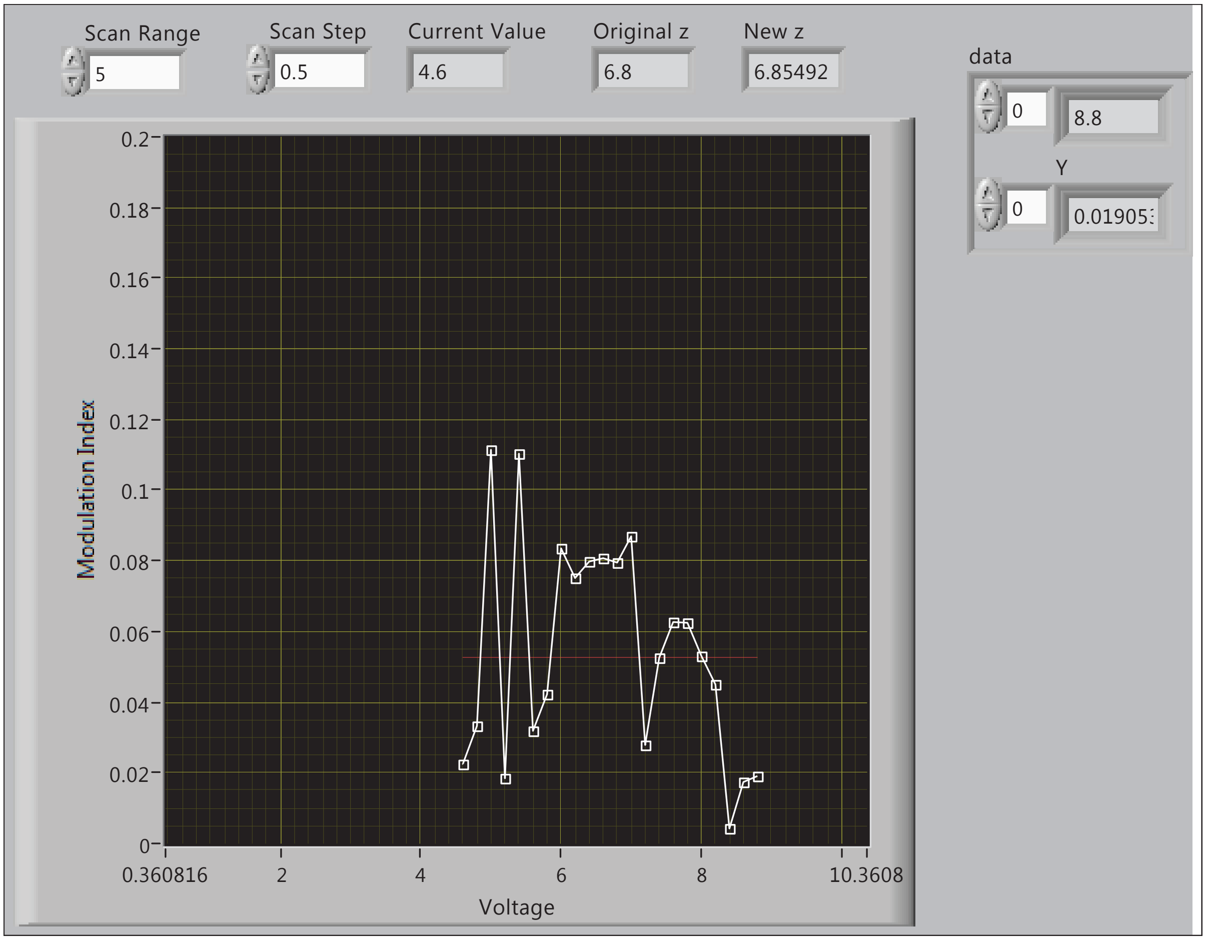
Page 1

Micromotion Scan_D2_v10.vi
C:\Users\lynx\Documents\LabView Program\Orange\Micromotion detection\Micromotion Scan_D2_v10.vi
Last modified on 7/21/2015 at 11:06 AM
Printed on 5/6/2016 at 4:22 PM



Page 1

Micromotion Scan_z_v10.vi
C:\Users\lynx\Documents\LabView Program\Orange\Micromotion detection\Micromotion Scan_z_v10.vi
Last modified on 1/22/2016 at 4:18 PM
Printed on 5/6/2016 at 4:22 PM



Scan Range: 5

Scan Step: 0.5

Current Value: 4.6

Original z: 6.8

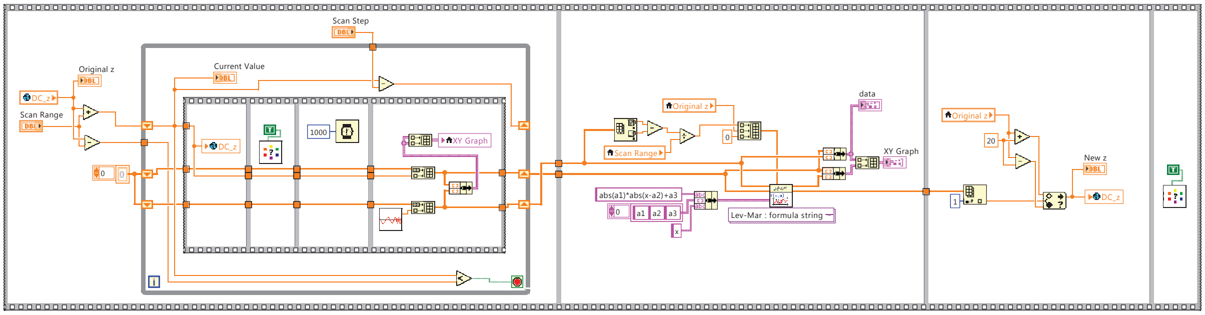
New z: 6.85492

data: 0 8.8

Y: 0 0.01905:

Page 1

Micromotion Scan_z_v10.vi
C:\Users\lynx\Documents\LabView Program\Orange\Micromotion detection\Micromotion Scan_z_v10.vi
Last modified on 1/22/2016 at 4:18 PM
Printed on 5/6/2016 at 4:23 PM

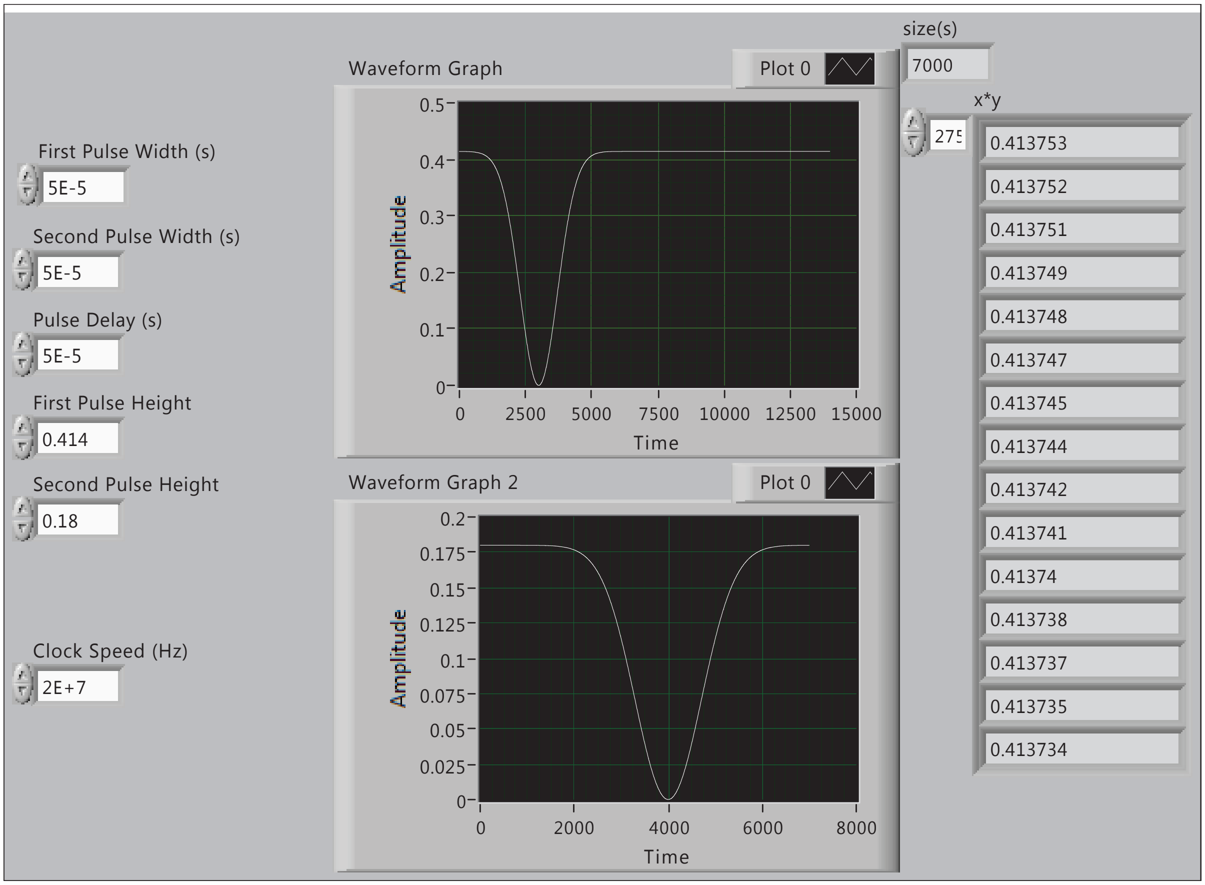


C.3.7. STIRAP Pulse Programming

The code below programs the SRS DS345 function generators to output the arbitrary waveform for the STIRAP pulses. Created by MRD.

Page 1

STIRAP Sequence.vi
C:\Users\lynx\Documents\LabView Program\SRS Smooth Pulse\STIRAP Sequence.vi
Last modified on 10/12/2015 at 4:18 PM
Printed on 5/6/2016 at 4:23 PM



First Pulse Width (s): 5E-5

Second Pulse Width (s): 5E-5

Pulse Delay (s): 5E-5

First Pulse Height: 0.414

Second Pulse Height: 0.18

Clock Speed (Hz): 2E+7

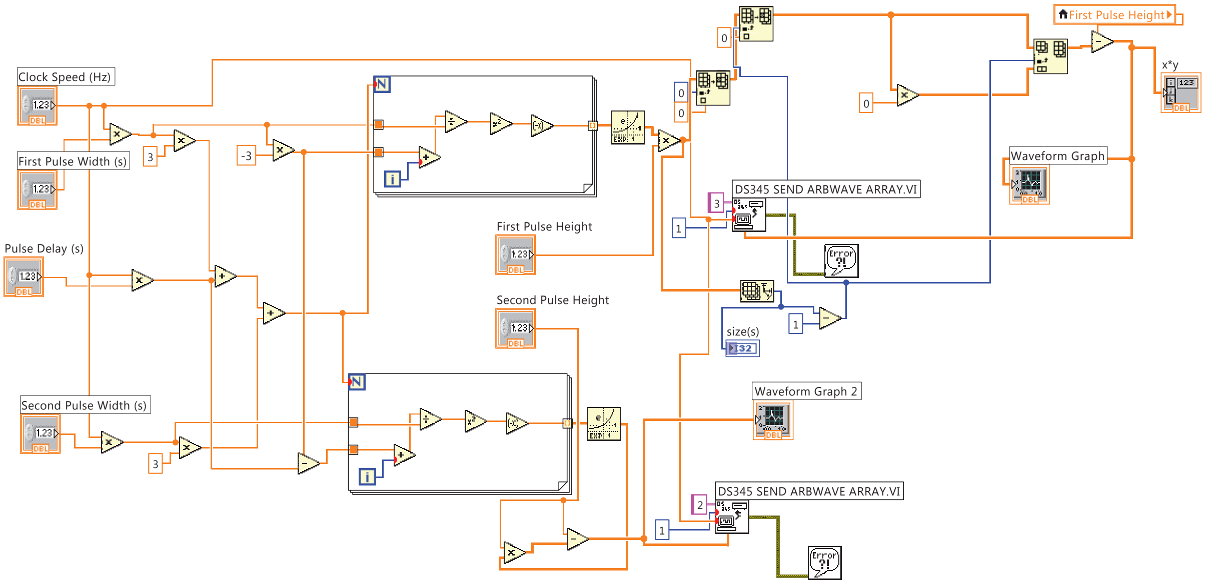
size(s): 7000

x*y: 275

0.413753
0.413752
0.413751
0.413749
0.413748
0.413747
0.413745
0.413744
0.413742
0.413741
0.41374
0.413738
0.413737
0.413735
0.413734

Page 1

STIRAP Sequence.vi
C:\Users\lynx\Documents\LabView Program\SRS Smooth Pulse\STIRAP Sequence.vi
Last modified on 10/12/2015 at 4:18 PM
Printed on 5/6/2016 at 4:23 PM



APPENDIX D

Octave (MatLab) Code

The GNU Octave environment is used for various modeling and data analysis applications including for the AlH^+ rotational cooling experiment, Monte Carlo simulations, and SHG cavity crystal length modeling.

D.1. ECDL Project

D.1.1. Condenser for Agilent N9320A Spectrum Analyzer CSV Files

The code below reads an input set of CSV-format data, converts the dBm to dBm/$\sqrt{\text{Hz}}$, and condenses the data into a single file. The code here is for six input files, but can be reduced or expanded as needed.

```
#Resets
clear all
close all

#Filenames to import and condense.
title1 = "TRACE399";
title2 = "TRACE400";
title3 = "TRACE401";
title4 = "TRACE402";
title5 = "TRACE403";
title6 = "TRACE404";
file1 = strcat(title1,".csv");
file2 = strcat(title2,".csv");
file3 = strcat(title3,".csv");
file4 = strcat(title4,".csv");
file5 = strcat(title5,".csv");
```

```
file6 = strcat(title6,".csv");

#Import .csv file data directly from Spectrum Analyzer and
#calculate spectral power density with RBW read from file
rawdata1 = csvread(file1,0,0);
RBW1 = rawdata1(6,2);
rawdata1 = csvread(file1,12,1);
data1(:,1) = rawdata1(:,1);
data1(:,2) = rawdata1(:,3)-10*log10(RBW1);

rawdata2 = csvread(file2,0,0);
RBW2 = rawdata2(6,2);
rawdata2 = csvread(file2,12,1);
data2(:,1) = rawdata2(:,1);
data2(:,2) = rawdata2(:,3)-10*log10(RBW2);

rawdata3 = csvread(file3,0,0);
RBW3 = rawdata3(6,2);
rawdata3 = csvread(file3,12,1);
data3(:,1) = rawdata3(:,1);
data3(:,2) = rawdata3(:,3)-10*log10(RBW3);

rawdata4 = csvread(file4,0,0);
RBW4 = rawdata4(6,2);
rawdata4 = csvread(file4,12,1);
data4(:,1) = rawdata4(:,1);
data4(:,2) = rawdata4(:,3)-10*log10(RBW4);

rawdata5 = csvread(file5,0,0);
RBW5 = rawdata5(6,2);
rawdata5 = csvread(file5,12,1);
data5(:,1) = rawdata5(:,1);
data5(:,2) = rawdata5(:,3)-10*log10(RBW5);

rawdata6 = csvread(file6,0,0);
RBW6 = rawdata6(6,2);
rawdata6 = csvread(file6,12,1);
data6(:,1) = rawdata6(:,1);
data6(:,2) = rawdata6(:,3)-10*log10(RBW6);

data = [data1;data2;data3;data4;data5;data6];
```

```
titleout = strcat(title1,title2,title3,title4,title5,title6,".csv");
csvwrite(titleout, data)
#save("-ascii","titleout","data")

display("Condensing Complete")

#Plot the Spectral Voltage Density
figure(1)
semilogx(data(:,1),data(:,2))
title(titleout);
#axis([10 400000 -0.001 0.01]);
xlabel ("Frequency (Hz)");
ylabel ("Amplitude (dBm/Hz)");
```

D.1.2. Polarization Spectroscopy Noise Integrator

The code below takes in an input polarization spectroscopy power spectral density trace, calculates the total power between 4 kHz and 400 kHz, and estimates the RMS frequency and current noise.

```
%Resets
clear all
close all

%Current to Frequency Ratio is MHz/uA.
dFi = 0.525;

%Polarization Spectroscopy Error Signal mV/MHz
dFv = 9.2;

%Filename to import. This should have been passed through a condenser first.
file1 = "TRACE340TRACE341TRACE342.csv";
title1 = file1;

%Import .csv file data.
%Calculate 3rd column to be power/Hz assuming 50 Ohm system.
%Calculate 4rd column to be mVrms/root(Hz).
%Calculate 5rd column to be MHz/root(Hz).
%Calculate 6th column to be uArms/root(Hz).
data = csvread(file1,0,0);
for i = 1: length(data);
       data(i,3) = 10^((data(i,2)-30)/10);
       data(i,4) = sqrt(data(i,3)*50)*1000;
       data(i,5) = data(i,4)/dFv;
       data(i,6) = data(i,5)/dFi;
endfor

%Plots!
figure(1)
semilogx(data(:,1),data(:,2))
title(title1);
%axis([10000 100000 -0.001 0.01]);
xlabel ("Frequency (Hz)");
```

```
ylabel ("Amplitude (dBm/Hz)");

figure(2)
loglog(data(:,1),data(:,3))
title(title1);
%axis([10000 100000 -0.001 0.01]);
xlabel ("Frequency (Hz)");
ylabel ("Amplitude (mW/Hz)");

figure(3)
loglog(data(:,1),data(:,4))
title(title1);
%axis([10000 100000 -0.001 0.01]);
xlabel ("Frequency (Hz)");
ylabel ("Amplitude (mVrms/root(Hz))");

figure(4)
loglog(data(:,1),data(:,5))
title(title1);
%axis([10000 100000 -0.001 0.01]);
xlabel ("Frequency (Hz)");
ylabel ("Amplitude (MHz/root(Hz))");

figure(5)
loglog(data(:,1),data(:,4))
title(title1);
%axis([10000 100000 -0.001 0.01]);
xlabel ("Frequency (Hz)");
ylabel ("Amplitude (uArms/root(Hz))");

%Save data matrix for later plotting.
%dlmwrite(strcat(title1,".txt"), data, "\t");

%Numerically Integrate the power/Hz dataset using trapezoidal rule.
P = trapz(data(:,1),data(:,3));
mVRMS = sqrt(P*50)*1000;
fRMS = mVRMS/dFv;
uARMS = fRMS/dFi;

%Display calculated noise
```

```
display(["Total noise power over 50 Ohms in input range is ",num2str(P,3)," W
    ."])
display(["Total voltage noise in input range is ",num2str(mVRMS,3)," mVRMS."])
display(["Total frequency noise in input range is ",num2str(fRMS,3)," MHz."])
display(["Total current noise in input range is ",num2str(uARMS,3)," uArms."])
```

D.1.3. SHG Cavity Modeling

The code below models the SHG output power of an optical SHG cavity for critical and non-critical phase matching with various crystals.

D.1.3.1. Critical Phase Matching with BBO. The code below models critical phase matching with BBO.

```
#clears
clear all
close all
home

#Uses Boyd-Kleinman theory to calculate the relative power output of a second
   harmonic generation crystal for non-critical phase matching

#Physical Constants
e_0 = 8.85419e-12; #vacuum permittivity (Ferads/meter)
c = 2.9979e8; #speed of light (meters/second)

#Define Cavity Parameters
r = .99; #reflectivity of input mirror
t = .0075; #total loss due to transmission or stray reflections in cavity
ceff = .8; #coupling efficiency coefficient
Pinput = .2; #input power

#Define Crystal Length using for loop on parameters listed blow
numpoints = 250;
maxlength = 50; #in mm
stepsize = (maxlength/numpoints)/1000;
for i = 1:(numpoints+1);
       L = 0 + (i-1/2)*stepsize

       #Input Beam Parameters
       lambda = 559e-9 ; #wavelength (meters)
       f = L/2; #laser focal distance (entrance of the crystal to laser focus);
           simplifying assumption sets laser focus at center of crystal (L/2)
```

```
#Crystal Parameters: The phase mismatch constant can be calculated in
   different ways. To first order, you can use the difference in
   refractive indices, since refractive index varies with wavelength.
   Alternatively, if your phase matching eliminates that difference to
   first order, you may be able to approximate a higher-order effect
   with the fluctuations of the refractive index due to optical
   inhomogeneity in your crystal. Set ri_2=ri_1 to use the fluctuation
   method.
deff = 1.81; #effective nonlinear coefficient of your crystal (
   picometers per volt)
alpha_1 = .03 ; #absorption coefficient for the first harmonic (
   fractional absorption per meter)
alpha_2 = .4; #absorption coefficient for the second harmonic (
   fractional absorption per meter)
ri_1 = (1.671597); #refractive index of the crystal for the input beam
ri_2 = ri_1; #refractive index of the crystal for the second harmonic
   beam -- set equal to ri_1 if using refractive index fluctuations
   rather than phase mismatch
delta_n_ri = 1e-6*L*100+(7.3e-6)*.1; #fluctuation in refractive index -
   set to zero if using different refractive indices between harmonics.
    Note that this should be a unitless quantity, you may have to
   multiply your data by temperature or length.
rho = 4.8; #walk-off angle in degrees. For NCPM, this should be zero.
xi = 1.39; #optimum focusing parameter; optimized at 1.39 for this case.

#Numerical Integration accuracy
intpts = 1000;

############################################################
####                        User Input Section Ends Here
                  ####
############################################################


#Calculated Parameters
k = 2*pi/lambda; #wave number
delta_k = 2*k*(ri_1-ri_2); #wavenumber fluctuations of laser in crystal
kappa = L*(alpha_1-1/2*alpha_2)/(2*xi); #absorption factor
mu = (L-2*f)/L; #displacement factor (displacement of laser focus from
   center of crystal)
alpha=alpha_1-1/2*alpha_2; #relative apsorption factor
```

```
rho = 4.8*pi/180;
Beta = 1/2*rho*sqrt(L*k/xi);


#phase mismatch parameter (chooses phase mismatch method)
if ri_2 == ri_1
        sigma = 1/2*L/xi*k*delta_n_ri;
else
        sigma = L/xi*delta_k;
end

#Numerically Solving the Boyd-Kleinman Equation:
#coefficients
coeff_0=(16*pi^2*deff^2)/(lambda^3*e_0*ri_1*ri_2*c)*10^-24;
coeff_1=L*exp(-(alpha_1+1/2*alpha_2)*L);

#Set up a numerical integration. Remember to only take the real part!
uplim = xi*(1+mu);
lowlim = -xi*(1-mu);
#f = @(x)@(y) exp(mu*L*alpha)/(4*xi)*exp(-kappa*(x+y)-Beta*Beta*(x-y)*(x
    -y))*((1+x*y)*cos(sigma*(x-y))+(x-y)*sin(sigma*(x-y)))/((1+x*x)*(1+y
    *y));
#Since Octave does not perform numerical integration in more than 1D
    well, must use brute force.
ystep = (uplim-lowlim)/intpts;
h = 0;
for j = 1:intpts
        y = lowlim + (j-1/2)*ystep;
        f =  @(x) exp(mu*L*alpha)/(4*xi)*exp(-kappa*(x+y)-Beta*Beta*(x-y)
            *(x-y))*((1+x*y)*cos(sigma*(x-y))+(x-y)*sin(sigma*(x-y)))
            /((1+x*x)*(1+y*y));
        xintegral = quad(f,lowlim,uplim);
        if xintegral == NaN
                tosum = 0;
        else
                tosum = xintegral*ystep;
        end
        data(j,1) = y;
        data(j,2) = tosum;
        h = h + tosum;
end
```

```
       #calculate the simulated conversion coefficient
       eta = coeff_0*coeff_1*h;

       #Define parameters to store in a matrix to plot
       hplot(i,1) = L*1000; #L in mm
       hplot(i,2) = h;
       etaplot(i,1) = L*1000; #L in mm
       etaplot(i,2) = eta;

       #Including the Cavity Effects: must call a function to obtain a
          numerical solution for the inter-cavity power
       f = @(P) -r*(1-alpha_1*L)*(1-t)*eta*(P^2)-2*sqrt(r*(1-alpha_1*L)*(1-t))*
          P*sqrt(1-eta*P)+(r*(1-alpha_1*L)*(1-t)+1)*P-(1-r)*ceff*Pinput;
       [Pcav, info] = fsolve(f,1);

       #Define parameters to store in a matrix to plot
       Pcavplot(i,1) = L*1000; #L in mm
       Pcavplot(i,2) = Pcav;

       #Define the SHG power using the inter-cavity fundamental power for
          plotting
       PSHG(i,1) = L*1000; #L in mm
       PSHG(i,2) = eta*Pcav*Pcav;
endfor

#make the plots
figure(1)
plot(hplot(:,1),hplot(:,2),’-.’)
xlabel("Crystal Length, L (mm)")
ylabel("h")
title("Function of Optimizable Parameters vs. Crystal Length")

figure(2)
plot(etaplot(:,1),etaplot(:,2),’-.’)
xlabel("Crystal Length, L (mm)")
ylabel("eta (1/W)")
title("Conversion Coefficient vs. Crystal Length")

figure(3)
plot(Pcavplot(:,1),Pcavplot(:,2),’-.’)
```

```
xlabel("Crystal Length, L (mm)")
ylabel("Pcav (W)")
title("Inter-cavity Fundamental Power vs. Crystal Length")

figure(4)
plot(PSHG(:,1),PSHG(:,2),'-.')
xlabel("Crystal Length, L (mm)")
ylabel("P SHG (W)")
title("SHG Power vs. Crystal Length")
```

D.1.3.2. Critical Phase Matching with BiBO. The code below models critical phase matching with BiBO for 385 nm SHG.

```
#clears
clear all
close all
home

#Uses Boyd-Kleinman theory to calculate the relative power output of a second
   harmonic generation crystal
#for non-critical phase matching

#Physical Constants
e_0 = 8.85419e-12; #vacuum permittivity (Ferads/meter)
c = 2.9979e8; #speed of light (meters/second)

#Define Cavity Parameters
r = .978; #reflectivity of input mirror
t = .01; #total loss due to transmission or stray reflections in cavity
ceff = .8; #coupling efficiency coefficient
Pinput = 1.0; #input power in Watts

#Input Beam Parameters
lambda = 770e-9 ; #wavelength (meters)

#Crystal Parameters: The phase mismatch constant can be calculated in different
    ways.
#To first order, you can use the difference in refractive indices, since
   refractive index varies with wavelength.
#Alternatively, if your phase matching eliminates that difference to first
   order, you may be able to approximate
#a higher-order effect with the fluctuations of the refractive index due to
   optical inhomogeneity in your crystal.
#Set ri_2=ri_1 to use the fluctuation method.
deff = 2.2; #effective nonlinear coefficient of your crystal (picometers per
   volt)
alpha_1 = .03 ; #absorption coefficient for the first harmonic (fractional
   absorption per meter)
alpha_2 = .4; #absorption coefficient for the second harmonic (fractional
   absorption per meter)
```

```
ri_1 = (1.852170 + 1.796213)/2; #refractive index of the crystal for the input
   beam
ri_2 = ri_1; #refractive index of the crystal for the second harmonic beam --
   set equal to ri_1 if using refractive
#index fluctuations rather than phase mismatch
#ri_2 = (1.930884 + 1.869970)/2;
#delta_n_ri = 0; #See Comment Below if using different method
rho = 5.1; #walk-off angle in degrees. For NCPM, this should be zero.
xi = 1.39; #optimum focusing parameter; optimized at 1.39 for this case.

#Numerical Integration accuracy
intpts = 500;

#Change below if these are used:
#delta_n_ri = 1e-6*L*100+(7.3e-6-2.1*lambda)*.1; #fluctuation in refractive
   index - set to zero if using different
#refractive indices between harmonics. Note that this should be a unitless
   quantity, you may have to multiply your data by temperature or length.

numpoints = 50;
maxlength = 25; #in mm

############################################################
####                 User Input Section Ends Here              ####
############################################################


#Calculated Parameters
k = 2*pi/lambda; #wave number
delta_k = 2*k*(ri_1-ri_2); #wavenumber fluctuations of laser in crystal
alpha=alpha_1-1/2*alpha_2; #relative apsorption factor
rho = 4.8*pi/180;

#Define Crystal Length using for loop on parameters listed blow
stepsize = (maxlength/numpoints)/1000;
for i = 1:(numpoints);
       L = 0 + (i-1/2)*stepsize

       #Calculated Parameters
       f = L/2; #laser focal distance (entrance of the crystal to laser focus);
           simplifying assumption sets laser focus at center of crystal (L/2)
```

```
kappa = L*(alpha_1-1/2*alpha_2)/(2*xi); #absorption factor
mu = (L-2*f)/L; #displacement factor (displacement of laser focus from
  center of crystal)
Beta = 1/2*rho*sqrt(L*k/xi);
delta_n_ri = 1e-6*L*100+(7.3e-6-2.1*lambda)*.1; #fluctuation in
  refractive index - set to zero if using different

#phase mismatch parameter (chooses phase mismatch method)
if ri_2 == ri_1
    sigma = 1/2*L/xi*k*delta_n_ri;
else
    sigma = L/xi*delta_k;
end

#Numerically Solving the Boyd-Kleinman Equation:
#coefficients
coeff_0=(16*pi^2*deff^2)/(lambda^3*e_0*ri_1*ri_2*c)*10^-24;
coeff_1=L*exp(-(alpha_1+1/2*alpha_2)*L);

#Set up a numerical integration. Remember to only take the real part!
uplim = xi*(1+mu);
lowlim = -xi*(1-mu);
#f = @(x)@(y) exp(mu*L*alpha)/(4*xi)*exp(-kappa*(x+y)-Beta*Beta*(x-y)*(x
  -y))*((1+x*y)*cos(sigma*(x-y))+(x-y)*sin(sigma*(x-y)))/((1+x*x)*(1+y
  *y));
#Since Octave does not perform numerical integration in more than 1D
  well, must use brute force.
ystep = (uplim-lowlim)/intpts;
h = 0;
for j = 1:intpts
    y = lowlim + (j-1/2)*ystep;
    f =  @(x) exp(mu*L*alpha)/(4*xi)*exp(-kappa*(x+y)-Beta*Beta*(x-y)
      *(x-y))*((1+x*y)*cos(sigma*(x-y))+(x-y)*sin(sigma*(x-y)))
      /((1+x*x)*(1+y*y));
    xintegral = quad(f,lowlim,uplim);
    if xintegral == NaN
        tosum = 0;
    else
        tosum = xintegral*ystep;
    end
    data(j,1) = y;
```

```
              data(j,2) = tosum;
              h = h + tosum;
       end

       #calculate the simulated conversion coefficient
       eta = coeff_0*coeff_1*h;

       #Define parameters to store in a matrix to plot
       hplot(i,1) = L*1000; #L in mm
       hplot(i,2) = h;
       etaplot(i,1) = L*1000; #L in mm
       etaplot(i,2) = eta;

       #Including the Cavity Effects: must call a function to obtain a
          numerical solution for the inter-cavity power
       f = @(P) -r*(1-alpha_1*L)*(1-t)*eta*(P^2)-2*sqrt(r*(1-alpha_1*L)*(1-t))*
          P*sqrt(1-eta*P)+(r*(1-alpha_1*L)*(1-t)+1)*P-(1-r)*ceff*Pinput;
       [Pcav, info] = fsolve(f,1);

       #Define parameters to store in a matrix to plot
       Pcavplot(i,1) = L*1000; #L in mm
       Pcavplot(i,2) = Pcav;

       #Define the SHG power using the inter-cavity fundamental power for
          plotting
       PSHG(i,1) = L*1000; #L in mm
       PSHG(i,2) = eta*Pcav*Pcav;
endfor

#make the plots
figure(1)
plot(hplot(:,1),hplot(:,2),'-.')
xlabel("Crystal Length, L (mm)")
ylabel("h")
title("Function of Optimizable Parameters vs. Crystal Length")

figure(2)
plot(etaplot(:,1),etaplot(:,2),'-.')
xlabel("Crystal Length, L (mm)")
ylabel("eta (1/W)")
title("Conversion Coefficient vs. Crystal Length")
```

```
figure(3)
plot(Pcavplot(:,1),Pcavplot(:,2),'-.')
xlabel("Crystal Length, L (mm)")
ylabel("Pcav (W)")
title("Inter-cavity Fundamental Power vs. Crystal Length")

figure(4)
plot(PSHG(:,1),PSHG(:,2),'-.')
xlabel("Crystal Length, L (mm)")
ylabel("P SHG (W)")
title("SHG Power vs. Crystal Length")
```

D.1.3.3. Non-critical Phase Matching with KNbO$_3$. The code below models non-critical phase matching with KNbO$_3$ for 493 nm SHG.

```
#clears
clear all
close all
home

#Uses Boyd-Kleinman theory to calculate the relative power output of a second
   harmonic generation crystal for non-critical phase matching

#Physical Constants
e_0 = 8.85419e-12; #vacuum permittivity (Ferads/meter)
c = 2.9979e8; #speed of light (meters/second)

#Define Cavity Parameters
r = .978; #reflectivity of input mirror
t = .01; #total loss due to transmission or stray reflections in cavity
ceff = .8; #coupling efficiency coefficient
Pinput = .120; #input power in Watts

#Define Crystal Length using for loop on parameters listed blow
numpoints = 250;
maxlength = 200; #in mm
stepsize = (maxlength/numpoints)/1000;
for i = 1:(numpoints+1);
        L = 0 + (i-1)*stepsize

        #Input Beam Parameters
        lambda = 987e-9 ; #wavelength (meters)
        f = L/2; #laser focal distance (entrance of the crystal to laser focus);
            simplifying assumption sets laser focus at center of crystal (L/2)
```

```
#Crystal Parameters: The phase mismatch constant can be calculated in
   different ways. To first order, you can use the difference in
   refractive indices, since refractive index varies with wavelength.
   Alternatively, if your phase matching eliminates that difference to
   first order, you may be able to approximate a higher-order effect
   with the fluctuations of the refractive index due to optical
   inhomogeneity in your crystal. Set ri_2=ri_1 to use the fluctuation
   method.
deff = 9.22 ; #effective nonlinear coefficient of your crystal (
   picometers per volt)
alpha_1 = .03 ; #absorption coefficient for the first harmonic (
   fractional absorption per meter)
alpha_2 = .3 ; #absorption coefficient for the second harmonic (
   fractional absorption per meter)
ri_1 = (2.264059 + 2.225508)/2; #refractive index of the crystal for the
    input beam
ri_2 = ri_1; #refractive index of the crystal for the second harmonic
   beam -- set equal to ri_1 if using refractive index fluctuations
   rather than phase mismatch
delta_n_ri = 1e-6*L*100+(7.3e-6-2.1*lambda)*.1; #fluctuation in
   refractive index - set to zero if using different refractive indices
    between harmonics. Note that this should be a unitless quantity,
   you may have to multiply your data by temperature or length.
rho = 0; #walk-off angle in degrees. For NCPM, this should be zero.
xi = 2.84; #optimum focusing parameter; optimized at 2.84 for NCPM.

############################################################
####                         User Input Section Ends Here
                  ####
############################################################


#Calculated Parameters
k = 2*pi/lambda; #wave number
delta_k = 2*k*(ri_1-ri_2); #wavenumber fluctuations of laser in crystal
xi = 2.84; #optimum focusing parameter; optimized at 2.84 for NCPM
kappa = L*(alpha_1-1/2*alpha_2)/(2*xi); #absorption factor
mu = (L-2*f)/L; #displacement factor (displacement of laser focus from
   center of crystal)
alpha=alpha_1-1/2*alpha_2; #relative apsorption factor
Beta = 1/2*rho*sqrt(L*k/xi);
```

```
#phase mismatch parameter (chooses phase mismatch method)
if ri_2 == ri_1
       sigma = 1/2*L/xi*k*delta_n_ri;
else
       sigma = L/xi*delta_k;
end

#Numerically Solving the Boyd-Kleinman Equation:
#coefficients
coeff_0=(16*pi^2*deff^2)/(lambda^3*e_0*ri_1*ri_2*c)*10^-24;
coeff_1=L*exp(-(alpha_1+1/2*alpha_2)*L);
coeff_2=exp(mu*alpha*L)/(4*xi);

#integrand expressions to integrate after decoupling and splitting into
   real and imaginary components
realintegrandx = @(x) exp(-kappa*x)*(cos(sigma*x)+x*sin(sigma*x))/(x
   ^2+1);
realintegrandy = @(y) exp(-kappa*y)*(cos(sigma*y)+y*sin(sigma*y))/(y
   ^2+1);
imgintegrandx = @(x) exp(-kappa*x)*(sin(sigma*x)-x*cos(sigma*x))/(x^2+1)
   ;
imgintegrandy = @(y) exp(-kappa*y)*(-sin(sigma*y)+y*cos(sigma*y))/(y
   ^2+1);

#perform numerical integrations
realintegralx = quad(realintegrandx,-xi*(1-mu),xi*(1+mu));
realintegraly = quad(realintegrandy,-xi*(1-mu),xi*(1+mu));
imgintegralx = quad(imgintegrandx,-xi*(1-mu),xi*(1+mu));
imgintegraly = quad(imgintegrandy,-xi*(1-mu),xi*(1+mu));

#calculate the function of optimizable parameters
h = coeff_2*(realintegralx*realintegraly-imgintegralx*imgintegraly);

#calculate the simulated conversion coefficient
eta = coeff_0*coeff_1*h;

#Define parameters to store in a matrix to plot
hplot(i,1) = L*1000; #L in mm
hplot(i,2) = h;
etaplot(i,1) = L*1000; #L in mm
```

```
        etaplot(i,2) = eta;

        #Including the Cavity Effects: must call a function to obtain a
           numerical solution for the inter-cavity power
        f = @(P) -r*(1-alpha_1*L)*(1-t)*eta*(P^2)-2*sqrt(r*(1-alpha_1*L)*(1-t))*
           P*sqrt(1-eta*P)+(r*(1-alpha_1*L)*(1-t)+1)*P-(1-r)*ceff*Pinput;
        [Pcav, info] = fsolve(f,1);

        #Define parameters to store in a matrix to plot
        Pcavplot(i,1) = L*1000; #L in mm
        Pcavplot(i,2) = Pcav;

        #Define the SHG power using the inter-cavity fundamental power for
           plotting
        PSHG(i,1) = L*1000; #L in mm
        PSHG(i,2) = eta*Pcav*Pcav;
end

#make the plots
figure(1)
plot(hplot(:,1),hplot(:,2),'-.')
xlabel("Crystal Length, L (mm)")
ylabel("h")
title("Function of Optimizable Parameters vs. Crystal Length")

figure(2)
plot(etaplot(:,1),etaplot(:,2),'-.')
xlabel("Crystal Length, L (mm)")
ylabel("eta (1/W)")
title("Conversion Coefficient vs. Crystal Length")

figure(3)
plot(Pcavplot(:,1),Pcavplot(:,2),'-.')
xlabel("Crystal Length, L (mm)")
ylabel("Pcav (W)")
title("Inter-cavity Fundamental Power vs. Crystal Length")

figure(4)
plot(PSHG(:,1),PSHG(:,2),'-.')
xlabel("Crystal Length, L (mm)")
ylabel("P SHG (W)")
```

```
title("SHG Power vs. Crystal Length")
```

D.1.3.4. Non-critical Phase Matching with LBO. The code below models non-critical phase matching with LBO for 614 nm SHG.

```
#clears
clear all
close all
home

#Uses Boyd-Kleinman theory to calculate the relative power output of a second
   harmonic generation crystal for non-critical phase matching

#Physical Constants
e_0 = 8.85419e-12; #vacuum permittivity (Ferads/meter)
c = 2.9979e8; #speed of light (meters/second)

#Define Cavity Parameters
r = .978; #reflectivity of input mirror
t = .01; #total loss due to transmission or stray reflections in cavity
ceff = .8; #coupling efficiency coefficient
Pinput = 0.200; #input power

#Define Crystal Length using for loop on parameters listed blow
numpoints = 90;
maxlength = 30; #in mm
stepsize = (maxlength/numpoints)/1000;
for i = 1:(numpoints+1);
       L = 0 + (i-1)*stepsize

       #Input Beam Parameters
       lambda = 1228e-9 ; #wavelength (meters)
       f = L/2; #laser focal distance (entrance of the crystal to laser focus);
           simplifying assumption sets laser focus at center of crystal (L/2)
```

```
#Crystal Parameters: The phase mismatch constant can be calculated in
   different ways. To first order, you can use the difference in
   refractive indices, since refractive index varies with wavelength.
   Alternatively, if your phase matching eliminates that difference to
   first order, you may be able to approximate a higher-order effect
   with the fluctuations of the refractive index due to optical
   inhomogeneity in your crystal. Set ri_2=ri_1 to use the fluctuation
   method.
deff = .834 ; #effective nonlinear coefficient of your crystal (
   picometers per volt)
alpha_1 = .1 ; #absorption coefficient for the first harmonic (
   fractional absorption per meter)
alpha_2 = .3 ; #absorption coefficient for the second harmonic (
   fractional absorption per meter)
ri_1 = 1.602289; #refractive index of the crystal for the input beam
ri_2 = ri_1; #refractive index of the crystal for the second harmonic
   beam -- set equal to ri_1 if using refractive index fluctuations
   rather than phase mismatch
delta_n_ri = 1e-6*L*100+(7.3e-6-2.1*lambda)*.1; #fluctuation in
   refractive index - set to zero if using different refractive indices
    between harmonics. Note that this should be a unitless quantity,
   you may have to multiply your data by temperature or length.
rho = 0; #walk-off angle in degrees. For NCPM, this should be zero.
xi = 2.84; #optimum focusing parameter; optimized at 2.84 for NCPM.

############################################################
####                        User Input Section Ends Here
                 ####
############################################################


#Calculated Parameters
k = 2*pi/lambda; #wave number
delta_k = 2*k*(ri_1-ri_2); #wavenumber fluctuations of laser in crystal
xi = 2.84; #optimum focusing parameter; optimized at 2.84 for NCPM
kappa = L*(alpha_1-1/2*alpha_2)/(2*xi); #absorption factor
mu = (L-2*f)/L; #displacement factor (displacement of laser focus from
   center of crystal)
alpha=alpha_1-1/2*alpha_2; #relative apsorption factor
Beta = 1/2*rho*sqrt(L*k/xi);
```

```
#phase mismatch parameter (chooses phase mismatch method)
if ri_2 == ri_1
       sigma = 1/2*L/xi*k*delta_n_ri;
else
       sigma = L/xi*delta_k;
end

#Numerically Solving the Boyd-Kleinman Equation:
#coefficients
coeff_0=(16*pi^2*deff^2)/(lambda^3*e_0*ri_1*ri_2*c)*10^-24;
coeff_1=L*exp(-(alpha_1+1/2*alpha_2)*L);
coeff_2=exp(mu*alpha*L)/(4*xi);

#integrand expressions to integrate after decoupling and splitting into
   real and imaginary components
realintegrandx = @(x) exp(-kappa*x)*(cos(sigma*x)+x*sin(sigma*x))/(x
   ^2+1);
realintegrandy = @(y) exp(-kappa*y)*(cos(sigma*y)+y*sin(sigma*y))/(y
   ^2+1);
imgintegrandx = @(x) exp(-kappa*x)*(sin(sigma*x)-x*cos(sigma*x))/(x^2+1)
   ;
imgintegrandy = @(y) exp(-kappa*y)*(-sin(sigma*y)+y*cos(sigma*y))/(y
   ^2+1);

#perform numerical integrations
realintegralx = quad(realintegrandx,-xi*(1-mu),xi*(1+mu));
realintegraly = quad(realintegrandy,-xi*(1-mu),xi*(1+mu));
imgintegralx = quad(imgintegrandx,-xi*(1-mu),xi*(1+mu));
imgintegraly = quad(imgintegrandy,-xi*(1-mu),xi*(1+mu));

#calculate the function of optimizable parameters
h = coeff_2*(realintegralx*realintegraly-imgintegralx*imgintegraly);

#calculate the simulated conversion coefficient
eta = coeff_0*coeff_1*h;

#Define parameters to store in a matrix to plot
hplot(i,1) = L*1000; #L in mm
hplot(i,2) = h;
etaplot(i,1) = L*1000; #L in mm
etaplot(i,2) = eta;
```

```
        #Including the Cavity Effects: must call a function to obtain a
           numerical solution for the inter-cavity power
        f = @(P) -r*(1-alpha_1*L)*(1-t)*eta*(P^2)-2*sqrt(r*(1-alpha_1*L)*(1-t))*
           P*sqrt(1-eta*P)+(r*(1-alpha_1*L)*(1-t)+1)*P-(1-r)*ceff*Pinput;
        [Pcav, info] = fsolve(f,1);

        #Define parameters to store in a matrix to plot
        Pcavplot(i,1) = L*1000; #L in mm
        Pcavplot(i,2) = Pcav;

        #Define the SHG power using the inter-cavity fundamental power for
           plotting
        PSHG(i,1) = L*1000; #L in mm
        PSHG(i,2) = eta*Pcav*Pcav;
end

#make the plots
figure(1)
plot(hplot(:,1),hplot(:,2),’-.’)
xlabel("Crystal Length, L (mm)")
ylabel("h")
title("Function of Optimizable Parameters vs. Crystal Length")

figure(2)
plot(etaplot(:,1),etaplot(:,2),’-.’)
xlabel("Crystal Length, L (mm)")
ylabel("eta (1/W)")
title("Conversion Coefficient vs. Crystal Length")

figure(3)
plot(Pcavplot(:,1),Pcavplot(:,2),’-.’)
xlabel("Crystal Length, L (mm)")
ylabel("Pcav (W)")
title("Inter-cavity Fundamental Power vs. Crystal Length")

figure(4)
plot(PSHG(:,1),PSHG(:,2),’-.’)
xlabel("Crystal Length, L (mm)")
ylabel("P SHG (W)")
title("SHG Power vs. Crystal Length")
```

D.2. AlH^+ Rotational Cooling Project

D.2.1. Monte Carlo Statistical Analysis Code

This section contains the toy Monte Carlo model used for the AlH^+ statistical analysis.

D.2.1.1. Broadband Rotational Optical Cooling Loss Fraction. Broadband rotational optical cooling loss fraction code.

```
%This script calculates the loss fraction Limit.
%Resets and format for long number strings
clear all
close all
format long g

%Loss fraction and uncertainty.
%If fx = 0, the experimental loss fraction is calculated and used. BEWARE!
   Counting with dfx near 0 can then be off due to histogram granularity.
fx = 0.;
dfx = 0.053;
numpts = 10000;

%Insert Data Below with pX = theory Al+ population fraction and nX = number of
   trials
p0 = 0.483;
n0Al = 64;
n0AlH = 68;
p1 = 0.517;
n1Al = 145;
n1AlH = 174;

%Processing...
n0 = n0Al + n0AlH;
n1 = n1Al + n1AlH;
p0exp = n0Al/n0;
p1exp = n1Al/n1;
fexp = p0exp/p1exp;
fxexp = 1 - p0exp - p1exp;

%Calculate actual fx if input fx = 0.
```

```
if fx == 0
       fx1 = fxexp;
       else
       fx1 = fx;
endif
fx2 = fx1 + dfx;

%Make Random Ntuple Set
for i = 1:numpts
       nt(i,1) = binornd(n0,p0);
       nt(i,2) = binornd(n1,p1);
       nt(i,3) = binornd(n0+n1,fx2);
endfor
nt(:,4) = nt(:,1) ./ n0;
nt(:,5) = nt(:,2) ./ n1;
nt(:,6) = nt(:,3) ./ (n0+n1);
nt(:,7) = 1 - nt(:,4) - nt(:,5) + nt(:,6);

%Loss fraction histogram, fit, numerically integrate.
%Make the loss fraction histogram and fit by turning exponential fit into a
   polynomial fit.
for i = 1:25
       bins1(i) = (i-1)*(0.5+0.25)/25-0.25;
endfor
[n1, x1b] = hist(nt(:,7),bins1);
%Exclude zeros and low values of the histogram for fitting since log(0) = -
   infinity kills the fit.
%dy1 is a root-(N+1) uncertainty for a weighted fit via wpolyfit.
j1 = 1;
for i = 1:length(n1)
       if n1(i) > 0 && n1(i) > max(n1)/100
              y1(j1) = log(n1(i));
              x1(j1) = x1b(i);
              dy1(j1) = (sqrt(n1(i)+1))/n1(i);
              j1 = j1 + 1;
       else
       endif
endfor
coef = wpolyfit(x1,y1,dy1,2);
a1 = coef(3);
b1 = coef(2);
```

```
c1 = coef(1);
sigma1 = sqrt(-1/(2*c1))
offset1 = -b1/(2*c1)
amplitude1 = exp(a1-b1^2/(4*c1))
%Generate fit curve for plotting.
for i = 1:1001
       fit1(i,1) = (i-1)*(0.5+0.25)/1000-0.25;
       fit1(i,2) = amplitude1*exp(-(fit1(i,1)-offset1)^2/(2*sigma1^2));
endfor

%Plot histogram and curve fit.
figure(1)
hold on
bar(x1b,n1)
plot(fit1(:,1),fit1(:,2),'b','linewidth',2)
plot([fxexp,fxexp],[0,max(n1)],'r','linewidth',2)
legend("Simulated Histogram","Gaussian fit","Experimental Result")
xlabel("Loss Fraction, fx")
ylabel("Counts")
title({["Random Number Simulation with ",num2str(numpts)," Attempts"],"Loss
   Fraction Upper Limit"});
hold off

%Calculate the percentage of the ntuple that has less ions than observed.
j2 = 1;
j3 = 2;
j4 = 1;
nte(1) = 0;
for i = 1:length(nt)
       if nt(i,7) < fxexp
              ntl(j2) = nt(i,7);
              j2 = j2 + 1;
       elseif nt(i,7) == fxexp
              nte(j3) = nt(i,7);
              j3 = j3 + 1;
       elseif nt(i,7) > fxexp
              ntu(j4) = nt(i,7);
              j4 = j4 + 1;
       endif
endfor
perc = (length(ntl)+(length(nte)-1)/2)/numpts*100;
```

```
%Display results.
display(" ")
display(["Loss Fraction = (",num2str(fx1)," + ",num2str(dfx),") = ",num2str(fx2
   )," with ",num2str(numpts)," points yields ",num2str(perc,3),"% of the
   Ntuple consistent with experiment."])
```

D.2.1.2. Broadband Rotational Optical Cooling Temperature Negative Parity Limit.

Broadband rotational optical cooling temperature negative parity limit code.

```
%This script calculates the N = 1 Parity Pumping Temperature Limit using the
   population fraction ratio in N = 3 / N = 1.
%Resets and format for long number strings
clear all
close all
format long g

%Temperature Guess in K, number of rotational states to consider, and number of
    points in Ntuple.
%If T = 0, the experimental temperature is calculated and used. BEWARE!
   Counting with dT near 0 can then be off due to histogram granularity.
T = 0.;
dT = 0.;
numstates = 20;
numpts = 100000;

%Thermal Population Numbers
ntAl(1) = 20;
ntAlH(1) = 330;
ntAl(2) = 28;
ntAlH(2) = 378;
ntAl(3) = 99;
ntAlH(3) = 487;
ntAl(4) = 37;
ntAlH(4) = 184;
ntAl(5) = 35;
ntAlH(5) = 198;
ntAl(6) = 33;
ntAlH(6) = 206;
ntAl(7) = 39;
ntAlH(7) = 320;
ntAl(8) = 27;
ntAlH(8) = 321;
%PROC Simulation Probabilities
pp(1) = 0.483;
pp(2) = 0.517;
pp(3) = 0.;
pp(4) = 0.;
pp(5) = 0.;
pp(6) = 0.;
pp(7) = 0.;
pp(8) = 0.;
%PROC Cooling Numbers
npAl(1) = 64;
npAlH(1) = 68;
npAl(2) = 145;
npAlH(2) = 174;
npAl(3) = 0;
npAlH(3) = 82;
%npAl(4) = 6;
npAl(4) = 2;
%npAl(4) = 1;
%npAl(4) = 0;
npAlH(4) = 291;
npAl(5) = 0;
npAlH(5) = 138;
npAl(6) = 0;
npAlH(6) = 166;
npAl(7) = 0;
npAlH(7) = 274;
npAl(7) = 0;
npAlH(7) = 100;
%Parity Pumping Simulation Probabilities
ppp(1) = 1.;
ppp(2) = 0.;
ppp(3) = 0.;
ppp(4) = 0.;
%Parity Pumping Numbers
nppAl(1) = 76;
nppAlH(1) = 4;
nppAl(2) = 1;
nppAlH(2) = 49;
nppAl(3) = 0;
nppAlH(3) = 11;
nppAl(4) = 0;
nppAlH(4) = 16;

%Calculating2
```

```
ntexp = ntAl + ntAlH;
npexp = npAl + npAlH;
nppexp = nppAl + nppAlH;
ptexp = ntAl ./ ntexp;
ppexp = npAl ./ npexp;
pppexp = nppAl ./ nppexp;

%Molecule Properties and constants.
hc = 1.23984193E-4; %eV*cm
B = 6.5632; %cm^-1
k = 8.61733E-5; %ev/K

%Set up Partition Function input parameters only looking at + parity.
numstates = numstates + 1;
for i = 2:2:numstates
       N(i) = i-1;
       g(i) = 2*(2*N(i)+1);
       E(i) = hc*B*N(i)*(N(i)+1);
endfor

%Calculate actual temperature if input temperature is set to 0.
R = ppexp(4)/ppexp(2);
if T == 0
       T1 = ((E(4)-E(2))/log(g(4)/(R*g(2))))/k;
       else
       T1 = T;
endif
T2 = T1 + dT;

%Map out thermal distribution probabilities starting by calculating Z.
Z = 0;
for i = 2:2:numstates
       Z = Z + g(i)*exp(-E(i)/(k*T2));
endfor
for i = 2:2:numstates
       pt(i) = pp(2)/Z*g(i)*exp(-E(i)/(k*T2));
endfor

%Make Random Ntuple Set of N = 1 and N = 3.
for i = 1:numpts
       nt(i,1) = binornd(npexp(2),pt(2))/npexp(2);
       nt(i,2) = binornd(npexp(4),pt(4))/npexp(4);
endfor
nt(:,3) = nt(:,2) ./ nt(:,1);

%Make the histogram and fit by turning exponential fit into a polynomial fit.
for i = 1:20
       bins1(i) = (i-1)*(0.20-0.)/19;
endfor
[n1, x1b] = hist(nt(:,3),bins1);
%Exclude zeros and low values of the histogram for fitting since log(0) = -
   infinity kills the fit.
%dy1 is a root-(N+1) uncertainty for a weighted fit via wpolyfit.
j1 = 1;
for i = 1:length(n1)
       if n1(i) > 0 && n1(i) > max(n1)/100
              y1(j1) = log(n1(i));
              x1(j1) = x1b(i);
              dy1(j1) = (sqrt(n1(i)+1))/n1(i);
              j1 = j1 + 1;
       else
       endif
endfor
coef = wpolyfit(x1,y1,dy1,2);
a1 = coef(3);
b1 = coef(2);
c1 = coef(1);
sigma1 = sqrt(-1/(2*c1))
offset1 = -b1/(2*c1)
amplitude1 = exp(a1-b1^2/(4*c1))
%Generate fit curve for plotting.
for i = 1:1001
       fit1(i,1) = (i-1)*(0.20-0.)/1000;
       fit1(i,2) = amplitude1*exp(-(fit1(i,1)-offset1)^2/(2*sigma1^2));
endfor

%Plot histogram and curve fit.
figure(1)
hold on
bar(x1b,n1)
plot(fit1(:,1),fit1(:,2),'b','linewidth',2)
plot([ppexp(4)/ppexp(2),ppexp(4)/ppexp(2)],[0,max(n1)],'r','linewidth',2)
```

```
legend("Simulated Histogram","Gaussian fit","Experimental Result")
xlabel("Population Fraction Ratio of N=3/N=1")
ylabel("Counts")
title({["Random Number Simulation with ",num2str(numpts)," Attempts"],"Positive
     Parity Temperature Limit"});
hold off

%Calculate the percentage of the ntuple that has less ions than observed.
j2 = 2;
j3 = 2;
j4 = 2;
ntl(1) = 0;
nte(1) = 0;
ntu(1) = 0;
for i = 1:length(nt)
        if nt(i,3) < ppexp(4)/ppexp(2)
                ntl(j2) = nt(i,3);
                j2 = j2 + 1;
        elseif nt(i,3) == ppexp(4)/ppexp(2)
                nte(j3) = nt(i,3);
                j3 = j3 + 1;
        elseif nt(i,3) > ppexp(4)/ppexp(2)
                ntu(j4) = nt(i,3);
                j4 = j4 + 1;
        endif
endfor
perc = (length(ntl)-1+(length(nte)-1)/2)/numpts*100;

%Display results.
display(["At T = (",num2str(T1)," + ",num2str(dT),") K = ",num2str(T2)," K with
     ",num2str(numpts)," points yields ",num2str(perc,3),"% of the Ntuple
    consistent with experiment."])
```

D.2.1.3. Broadband Rotational Optical Cooling Temperature Positive Parity Limit. Broadband rotational optical cooling temperature positive parity limit code.

```
%This script calculates the N = 0, 2 (Positive Parity) Temperature Limit.
%Resets and format for long number strings
clear all
close all
format long g

%Temperature Guess in K, number of rotational states to consider, and number of
    points in Ntuple.
T = 11.8;
numstates = 20;
numpts = 100000;

%Thermal Population Numbers
ntAl(1) = 20;
ntAlH(1) = 330;
ntAl(2) = 28;
ntAlH(2) = 378;
ntAl(3) = 99;
ntAlH(3) = 487;
ntAl(4) = 37;
ntAlH(4) = 184;
ntAl(5) = 35;
ntAlH(5) = 198;
ntAl(6) = 33;
ntAlH(6) = 206;
ntAl(7) = 39;
ntAlH(7) = 320;
ntAl(8) = 27;
ntAlH(8) = 321;
%PROC Simulation Probabilities
pp(1) = 0.483;
pp(2) = 0.517;
pp(3) = 0.;
pp(4) = 0.;
pp(5) = 0.;
pp(6) = 0.;
pp(7) = 0.;
pp(8) = 0.;
%PROC Cooling Numbers
npAl(1) = 64;
npAlH(1) = 68;
npAl(2) = 145;
npAlH(2) = 174;
npAl(3) = 0;
npAlH(3) = 82;
npAl(4) = 6;
%npAl(4) = 2;
%npAl(4) = 1;
%npAl(4) = 0;
npAlH(4) = 291;
npAl(5) = 0;
npAlH(5) = 138;
npAl(6) = 0;
npAlH(6) = 166;
npAl(7) = 0;
npAlH(7) = 274;
npAl(7) = 0;
npAlH(7) = 100;
%Parity Pumping Simulation Probabilities
ppp(1) = 1.;
ppp(2) = 0.;
ppp(3) = 0.;
ppp(4) = 0.;
%Parity Pumping Numbers
nppAl(1) = 76;
nppAlH(1) = 4;
nppAl(2) = 1;
nppAlH(2) = 49;
nppAl(3) = 0;
nppAlH(3) = 11;
nppAl(4) = 0;
nppAlH(4) = 16;

%Molecule Properties and constants.
hc = 1.23984193E-4; %eV*cm
B = 6.5632; %cm^-1
k = 8.61733E-5; %ev/K
kT = k*T;
```

```
%Calculating2
ntexp = ntAl + ntAlH;
npexp = npAl + npAlH;
nppexp = nppAl + nppAlH;
ptexp = ntAl ./ ntexp;
ppexp = npAl ./ npexp;
pppexp = nppAl ./ nppexp;

%Set up Partition Function with initialization of Z only looking at + parity.
Z = 0;
numstates = numstates + 1;
for i = 1:2:numstates
       N(i) = i-1;
       g(i) = 2*(2*N(i)+1);
       E(i) = hc*B*N(i)*(N(i)+1);
       Z = Z + g(i)*exp(-E(i)/kT);
endfor

%Map out thermal distribution probabilities.
for i = 1:2:numstates
       pt(i) = pp(1)/Z*g(i)*exp(-E(i)/kT);
endfor

%Make Random Ntuple Set of N = 0 and N = 2.
for i = 1:numpts
       nt(i,1) = binornd(npexp(1),pt(1));
       nt(i,2) = binornd(npexp(3),pt(3));
endfor

%Make the histogram and fit by turning exponential fit into a polynomial fit.
for i = 1:20
       bins1(i) = i-1;
endfor
[n1, x1b] = hist(nt(:,2),bins1);
%Exclude zeros and low values of the histogram for fitting since log(0) = -
   infinity kills the fit.
%dy1 is a root-(N+1) uncertainty for a weighted fit via wpolyfit.
j1 = 1;
for i = 1:length(n1)
       if n1(i) > 0 && n1(i) > max(n1)/100
```

```
              y1(j1) = log(n1(i));
              x1(j1) = x1b(i);
              dy1(j1) = (sqrt(n1(i)+1))/n1(i);
              j1 = j1 + 1;
       else
       endif
endfor
coef = wpolyfit(x1,y1,dy1,2);
a1 = coef(3);
b1 = coef(2);
c1 = coef(1);
sigma1 = sqrt(-1/(2*c1))
offset1 = -b1/(2*c1)
amplitude1 = exp(a1-b1^2/(4*c1))
%Generate fit curve for plotting.
for i = 1:1001
       fit1(i,1) = (i-1)*(20)/1000;
       fit1(i,2) = amplitude1*exp(-(fit1(i,1)-offset1)^2/(2*sigma1^2));
endfor

%Plot histogram and curve fit.
figure(1)
hold on
bar(x1b,n1)
plot(fit1(:,1),fit1(:,2),'b','linewidth',2)
plot([npAl(3),npAl(3)],[0,max(n1)],'r','linewidth',2)
legend("Simulated Histogram","Gaussian fit","Experimental Result")
xlabel("Ions in N = 2")
ylabel("Counts")
title({["Random Number Simulation with ",num2str(numpts)," Attempts"],"Positive
    Parity Temperature Limit"});
hold off

%Calculate the percentage of the ntuple that has less ions than observed.
j2 = 2;
j3 = 2;
j4 = 2;
ntl(1) = 0;
nte(1) = 0;
ntu(1) = 0;
for i = 1:length(nt)
```

```
        if nt(i,2) < npAl(3)
                ntl(j2) = nt(i,2);
                j2 = j2 + 1;
        elseif nt(i,2) == npAl(3)
                nte(j3) = nt(i,2);
                j3 = j3 + 1;
        elseif nt(i,2) > npAl(3)
                ntu(j4) = nt(i,2);
                j4 = j4 + 1;
        endif
endfor
perc = (length(ntl)-1+(length(nte)-1)/2)/numpts*100;

%Display results.
display(["At T = ",num2str(T)," K with ",num2str(numpts)," points yields ",
    num2str(perc,3),"% of the Ntuple consistent with experiment."])
```

D.2.1.4. Model Consistency. Model consistency code.

```
%Resets and format for long number strings
clear all
close all
format long g

%Insert Data Below with pX = theory Al+ population fraction and nX = number of
   trials
p0 = 0.483;
n0Al = 64;
n0AlH = 68;
p1 = 0.517;
n1Al = 145;
n1AlH = 174;

%Confidence parameters:
numpts = 100000;

%Processing...
n0 = n0Al + n0AlH;
n1 = n1Al + n1AlH;
p0exp = n0Al/n0;
p1exp = n1Al/n1;
fexp = p0exp/p1exp;
fxexp = 1 - p0exp - p1exp;

%Make Random Ntuple Set
for i = 1:numpts
       nt(i,1) = binornd(n0,p0);
       nt(i,2) = binornd(n1,p1);
       nt(i,3) = nt(i,1)/n0;
       nt(i,4) = nt(i,2)/n1;
       nt(i,5) = nt(i,3)/nt(i,4);
       nt(i,6) = 1 - nt(i,3) - nt(i,4);
       %Line below uses experimental probabilities for loss fraction.
       %nt(i,6) = 1 - binornd(n0,p0exp)/n0 - binornd(n1,p1exp)/n1;
endfor

%Make the histogram and fit by turning exponential fit into a polynomial fit.
for i = 1:25
```

```
        bins1(i) = (i-1)*(1.5-0.5)/25+0.5;
endfor
[n1, x1b] = hist(nt(:,5),bins1);
%Exclude zeros and low values of the histogram for fitting since log(0) = -
   infinity kills the fit.
%dy1 is a root-(N+1) uncertainty for a weighted fit via wpolyfit.
j1 = 1;
for i = 1:length(n1)
        if n1(i) > 0 && n1(i) > max(n1)/100
                y1(j1) = log(n1(i));
                x1(j1) = x1b(i);
                dy1(j1) = (sqrt(n1(i)+1))/n1(i);
                j1 = j1 + 1;
        else
        endif
endfor
coef = wpolyfit(x1,y1,dy1,2);
a1 = coef(3);
b1 = coef(2);
c1 = coef(1);
sigma1 = sqrt(-1/(2*c1))
offset1 = -b1/(2*c1)
amplitude1 = exp(a1-b1^2/(4*c1))
%Generate fit curve for plotting.
for i = 1:1001
        fit1(i,1) = (i-1)*(1.5-0.5)/1000+0.5;
        fit1(i,2) = amplitude1*exp(-(fit1(i,1)-offset1)^2/(2*sigma1^2));
endfor

%Plot histogram and curve fit.
figure(1)
hold on
bar(x1b,n1)
plot(fit1(:,1),fit1(:,2),'b','linewidth',2)
plot([fexp,fexp],[0,max(n1)],'r','linewidth',2)
legend("Simulated Histogram","Gaussian fit","Experimental Result")
xlabel("f0/f1")
ylabel("Counts")
title({["Random Number Simulation with ",num2str(numpts)," Attempts"],"
   Population Fraction Ratio"});
hold off
```

```
%Calculate number of sigma discrepancy.
nsigma1 = abs((offset1-fexp)/sigma1);
display(" ")
display(["Experimental fraction ratio is ",num2str(nsigma1,3)," sigma from the
   simulation result given the experimental statistics."])

%Loss fraction histogram, fit, numerically integrate.
%Make the loss fraction histogram and fit by turning exponential fit into a
   polynomial fit.
for i = 1:25
       bins2(i) = (i-1)*(0.25+0.25)/25-0.25;
endfor
[n2, x2b] = hist(nt(:,6),bins2);
%Exclude zeros and low values of the histogram for fitting since log(0) = -
   infinity kills the fit.
%dy1 is a root-(N+1) uncertainty for a weighted fit via wpolyfit.
j2 = 1;
for i = 1:length(n2)
       if n2(i) > 0 && n2(i) > max(n2)/100
              y2(j2) = log(n2(i));
              x2(j2) = x2b(i);
              dy2(j2) = (sqrt(n2(i)+1))/n2(i);
              j2 = j2 + 1;
       else
       endif
endfor
coef = wpolyfit(x2,y2,dy2,2);
a2 = coef(3);
b2 = coef(2);
c2 = coef(1);
sigma2 = sqrt(-1/(2*c2))
offset2 = -b2/(2*c2)
amplitude2 = exp(a2-b2^2/(4*c2))
%Generate fit curve for plotting.
for i = 1:1001
       fit2(i,1) = (i-1)*(0.25+0.25)/1000-0.25;
       fit2(i,2) = amplitude2*exp(-(fit2(i,1)-offset2)^2/(2*sigma2^2));
endfor

%Plot histogram and curve fit.
```

```
figure(2)
hold on
bar(x2b,n2)
plot(fit2(:,1),fit2(:,2),'b','linewidth',2)
plot([fxexp,fxexp],[0,max(n2)],'r','linewidth',2)
legend("Simulated Histogram","Gaussian fit","Experimental Result")
xlabel("Loss Fraction, fx")
ylabel("Counts")
title({["Random Number Simulation with ",num2str(numpts)," Attempts"],"Loss
   Fraction"});
hold off

%Calculate number of sigma discrepancy for loss fraction.
nsigma2 = abs((offset2-fxexp)/sigma2);
display(" ")
display(["Experimental loss fraction is ",num2str(nsigma2,3)," sigma from the
   simulation result given the experimental statistics."])

%Calculate the percentage of the ratio ntuple that is less than observed.
j2 = 1;
j3 = 2;
j4 = 1;
nte(1) = 0;
for i = 1:length(nt)
       if nt(i,5) < fexp
              ntl(j2) = nt(i,5);
              j2 = j2 + 1;
       elseif nt(i,5) == fexp
              nte(j3) = nt(i,5);
              j3 = j3 + 1;
       elseif nt(i,5) > fexp
              ntu(j4) = nt(i,5);
              j4 = j4 + 1;
       endif
endfor
perc = (length(ntl)+(length(nte)-1)/2)/numpts*100;

%Display results.
display(" ")
display(["PROC population fraction ratio = (",num2str(fexp),") with ",num2str(
   numpts)," points yields ",num2str(perc,3),"% of the Ntuple consistent with
   experiment."])

%Output histogram data to import for plotting.
for i = 1:length(x1b)
       op(i,1) = x1b(i);
       op(i,2) = n1(i);
endfor
dlmwrite("AlH PROC Fraction Ration Simulation Histogram.txt", op, ",");
```

D.2.1.5. Predissociation Fraction. Predissociation fraction code.

```
%This script calculates the predissociation fraction Limit.
%Resets and format for long number strings
clear all
close all
format long g

%Predissociation fraction and uncertainty.
%If pd = 0, the experimental predissociation fraction is calculated and used.
   BEWARE! Counting with dpd near 0 can then be off due to histogram
   granularity.
pd = 0.0015;
dpd = 0.;
numpts = 100000;

%Insert Al+/AlH+ population from N >= 2
%nAl = 6;
%nAl = 2;
%nAl = 1;
nAl = 0;
nAlH = 1051;

%Processing...
nexp = nAl + nAlH;

%Calculate actual pd fraction if input pd = 0.
if pd == 0
       pd1 = nAl/nexp;
       else
       pd1 = pd;
endif
pd2 = pd1 + dpd;

%Make Random Ntuple Set
for i = 1:numpts
       nt(i,1) = binornd(nexp,pd2);
endfor

%Make the histogram and fit by turning exponential fit into a polynomial fit.
for i = 1:25
```

```
       bins1(i) = i-1;
endfor
[n1, x1b] = hist(nt(:,1),bins1);
%Exclude zeros and low values of the histogram for fitting since log(0) = -
   infinity kills the fit.
%dy1 is a root-(N+1) uncertainty for a weighted fit via wpolyfit.
j1 = 1;
for i = 1:length(n1)
       if n1(i) > 0 && n1(i) > max(n1)/100
              y1(j1) = log(n1(i));
              x1(j1) = x1b(i);
              dy1(j1) = (sqrt(n1(i)+1))/n1(i);
              j1 = j1 + 1;
       else
       endif
endfor
coef = wpolyfit(x1,y1,dy1,2);
a1 = coef(3);
b1 = coef(2);
c1 = coef(1);
sigma1 = sqrt(-1/(2*c1))
offset1 = -b1/(2*c1)
amplitude1 = exp(a1-b1^2/(4*c1))
%Generate fit curve for plotting.
for i = 1:1001
       fit1(i,1) = (i-1)*(25)/1000;
       fit1(i,2) = amplitude1*exp(-(fit1(i,1)-offset1)^2/(2*sigma1^2));
endfor

%Plot histogram and curve fit.
figure(1)
hold on
bar(x1b,n1)
plot(fit1(:,1),fit1(:,2),'b','linewidth',2)
plot([nAl,nAl],[0,max(n1)],'r','linewidth',2)
legend("Simulated Histogram","Gaussian fit","Experimental Result")
xlabel("Predissociated Ions")
ylabel("Counts")
title({["Random Number Simulation with ",num2str(numpts)," Attempts"],"
   Predissociation Fraction Limit"});
hold off
```

```
%Calculate the percentage of the ntuple that has less ions than observed.
j2 = 2;
j3 = 2;
j4 = 2;
ntl(1) = 0;
nte(1) = 0;
ntu(1) = 0;
for i = 1:length(nt)
       if nt(i) < nAl
              ntl(j2) = nt(i);
              j2 = j2 + 1;
       elseif nt(i) == nAl
              nte(j3) = nt(i);
              j3 = j3 + 1;
       elseif nt(i) > nAl
              ntu(j4) = nt(i);
              j4 = j4 + 1;
       endif
endfor
perc = (length(ntl)-1+(length(nte)-1)/2)/numpts*100;

%Display results.
display(["Predissociation fraction = (",num2str(pd1)," + ",num2str(dpd),") = ",
   num2str(pd2)," with ",num2str(numpts)," points yields ",num2str(perc,3),"%
   of the Ntuple consistent with experiment."])
```

D.2.1.6. REMPD Efficiency. REMPD efficiency code.

```
%This script calculates the N = 1 Parity Pumping Temperature Limit using the
   population fraction ratio in N = 1 / N = 0.
%Resets and format for long number strings
clear all
close all
format long g

%REMPD and dREMPD efficiency guesses, Temperature in K, number of rotational
   states to consider, and number of points in Ntuple.
%If R = 0, the experimental REMPD efficiency is calculated and used. BEWARE!
   Counting with dR near 0 can then be off due to histogram granularity.
R = 0.;
dR = 0.;
T = 293;
numstates = 20;
numpts = 1000;

%Thermal Population Numbers
ntAl(1) = 20;
ntAlH(1) = 330;
ntAl(2) = 28;
ntAlH(2) = 378;
ntAl(3) = 99;
ntAlH(3) = 487;
ntAl(4) = 37;
ntAlH(4) = 184;
ntAl(5) = 35;
ntAlH(5) = 198;
ntAl(6) = 33;
ntAlH(6) = 206;
ntAl(7) = 39;
ntAlH(7) = 320;
ntAl(8) = 27;
ntAlH(8) = 321;
%PROC Simulation Probabilities
pp(1) = 0.483;
pp(2) = 0.517;
pp(3) = 0.;
pp(4) = 0.;
pp(5) = 0.;
pp(6) = 0.;
pp(7) = 0.;
pp(8) = 0.;
%PROC Cooling Numbers
npAl(1) = 64;
npAlH(1) = 68;
npAl(2) = 145;
npAlH(2) = 174;
npAl(3) = 0;
npAlH(3) = 82;
npAl(4) = 6;
%npAl(4) = 2;
%npAl(4) = 1;
%npAl(4) = 0;
npAlH(4) = 291;
npAl(5) = 0;
npAlH(5) = 138;
npAl(6) = 0;
npAlH(6) = 166;
npAl(7) = 0;
npAlH(7) = 274;
npAl(7) = 0;
npAlH(7) = 100;
%Parity Pumping Simulation Probabilities
ppp(1) = 1.;
ppp(2) = 0.;
ppp(3) = 0.;
ppp(4) = 0.;
%Parity Pumping Numbers
nppAl(1) = 76;
nppAlH(1) = 4;
nppAl(2) = 1;
nppAlH(2) = 49;
nppAl(3) = 0;
nppAlH(3) = 11;
nppAl(4) = 0;
nppAlH(4) = 16;

%Calculating2
ntexp = ntAl + ntAlH;
```

```
npexp = npAl + npAlH;
nppexp = nppAl + nppAlH;
ptexp = ntAl ./ ntexp;
ppexp = npAl ./ npexp;
pppexp = nppAl ./ nppexp;

%Molecule Properties and constants.
hc = 1.23984193E-4; %eV*cm
B = 6.5632; %cm^-1
k = 8.61733E-5; %ev/K

%Set up Partition Function input parameters looking at all states.
numstates = numstates + 1;
for i = 1:numstates
       N(i) = i-1;
       g(i) = 2*(2*N(i)+1);
       E(i) = hc*B*N(i)*(N(i)+1);
endfor

%Map out thermal distribution probabilities starting by calculating Z.
Z = 0;
for i = 1:numstates
       Z = Z + g(i)*exp(-E(i)/(k*T));
endfor
for i = 1:numstates
       pt(i) = 1/Z*g(i)*exp(-E(i)/(k*T));
endfor




%Calculate actual REMPD efficiency if input efficiency is set to 0.
pts = pt(1) + pt(2) + pt(3) + pt(4) + pt(5) + pt(6) + pt(7) + pt(8);
tsexp = sum(ptexp);
if R == 0
       R1 = 1 - (pts - tsexp);
       else
       R1 = R;
endif
R2 = R1 + dR;
```

```
R3 = 1 - R;
R4 = 1 - R1;

%Make Random Ntuple Set of N = 1 to 7.
for i = 1:numpts
       nt(i,1) = binornd(ntexp(1),pt(1) - R4)/ntexp(1);
       nt(i,2) = binornd(ntexp(2),pt(2) - R4)/ntexp(2);
       nt(i,3) = binornd(ntexp(3),pt(3) - R4)/ntexp(3);
       nt(i,4) = binornd(ntexp(4),pt(4) - R4)/ntexp(4);
       nt(i,5) = binornd(ntexp(5),pt(5) - R4)/ntexp(5);
       nt(i,6) = binornd(ntexp(6),pt(6) - R4)/ntexp(6);
       nt(i,7) = binornd(ntexp(7),pt(7) - R4)/ntexp(7);
       nt(i,8) = binornd(ntexp(8),pt(8) - R4)/ntexp(8);
endfor
nt(:,9) = nt(:,1) + nt(:,2) + nt(:,3) + nt(:,4) + nt(:,5) + nt(:,6) + nt(:,7) +
    nt(:,8);
nt(:,10) = 1 - (pts - nt(:,9));

%Make the histogram and fit by turning exponential fit into a polynomial fit.
for i = 1:20
       bins1(i) = (i-1)*(1.1-0.85)/19 + 0.85;
endfor
[n1, x1b] = hist(nt(:,10),bins1);
%Exclude zeros and low values of the histogram for fitting since log(0) = -
   infinity kills the fit.
%dy1 is a root-(N+1) uncertainty for a weighted fit via wpolyfit.
j1 = 1;
for i = 1:length(n1)
       if n1(i) > 0 && n1(i) > max(n1)/100
              y1(j1) = log(n1(i));
              x1(j1) = x1b(i);
              dy1(j1) = (sqrt(n1(i)+1))/n1(i);
              j1 = j1 + 1;
       else
       endif
endfor
coef = wpolyfit(x1,y1,dy1,2);
a1 = coef(3);
b1 = coef(2);
c1 = coef(1);
sigma1 = sqrt(-1/(2*c1))
```

```
offset1 = -b1/(2*c1)
amplitude1 = exp(a1-b1^2/(4*c1))
%Generate fit curve for plotting.
for i = 1:1001
       fit1(i,1) = (i-1)*(1.1-0.85)/1000 + 0.85;
       fit1(i,2) = amplitude1*exp(-(fit1(i,1)-offset1)^2/(2*sigma1^2));
endfor

%Plot histogram and curve fit.
figure(1)
hold on
bar(x1b,n1)
plot(fit1(:,1),fit1(:,2),'b','linewidth',2)
plot([tsexp,tsexp],[0,max(n1)],'r','linewidth',2)
legend("Simulated Histogram","Gaussian fit","Experimental Result")
xlabel("Population Fraction Sum of N=0 to N=7")
ylabel("Counts")
title({["Random Number Simulation with ",num2str(numpts)," Attempts"],"REMPD
   Efficiency Limit"});
hold off

%Calculate the percentage of the ntuple that has less ions than observed.
j2 = 1;
for i = 1:length(nt)
       if nt(i,10) <= tsexp
              ntc(j2) = nt(i,10);
              j2 = j2 + 1;
       else
       endif
endfor
perc = length(ntc)/numpts*100;

%Display results.
display(["At R = (",num2str(R)," + ",num2str(dR),") = ",num2str(R2)," with ",
   num2str(numpts)," points yields ",num2str(perc,3),"% of the Ntuple
   consistent with experiment."])
```

D.2.1.7. Vibrationally-assisted Broadband Rotational Optical Cooling Loss Fraction.

Vibrationally-assisted broadband rotational optical cooling loss fraction code.

```
%This script calculates the loss fraction Limit.
%Resets and format for long number strings
clear all
close all
format long g

%Loss fraction and uncertainty.
%If fx = 0, the experimental loss fraction is calculated and used. BEWARE!
   Counting with dfx near 0 can then be off due to histogram granularity.
f1 = 0.;
df1 = -0.0134;
numpts = 100000;

%Insert Data Below with pX = theory Al+ population fraction and nX = number of
   trials
p0 = 1;
n0Al = 104;
n0AlH = 5;
p1 = 0;
n1Al = 1;
n1AlH = 49;

%Processing...
n0 = n0Al + n0AlH;
n1 = n1Al + n1AlH;
p0exp = n0Al/n0;
p1exp = n1Al/n1;
fexp = p0exp/p1exp;
fxexp = 1 - p0exp;

%Calculate actual fx if input fx = 0.
if f1 == 0
       f11 = p1exp;
       else
       f11 = f1;
endif
f12 = f11 + df1;
```

```
%Make Random Ntuple Set
for i = 1:numpts
       nt(i,1) = binornd(n1,f12);
endfor
nt(:,2) = nt(:,1) ./ n1;

%Loss fraction histogram, fit, numerically integrate.
%Make the loss fraction histogram and fit by turning exponential fit into a
   polynomial fit.
for i = 1:16
       bins1(i) = (i-1)*(0.15+0.00)/10-0.00;
endfor
[n1, x1b] = hist(nt(:,2),bins1);
%Exclude zeros and low values of the histogram for fitting since log(0) = -
   infinity kills the fit.
%dy1 is a root-(N+1) uncertainty for a weighted fit via wpolyfit.
j1 = 1;
for i = 1:length(n1)
       if n1(i) > 0 && n1(i) > max(n1)/100
              y1(j1) = log(n1(i));
              x1(j1) = x1b(i);
              dy1(j1) = (sqrt(n1(i)+1))/n1(i);
              j1 = j1 + 1;
       else
       endif
endfor
coef = wpolyfit(x1,y1,dy1,2);
a1 = coef(3);
b1 = coef(2);
c1 = coef(1);
sigma1 = sqrt(-1/(2*c1))
offset1 = -b1/(2*c1)
amplitude1 = exp(a1-b1^2/(4*c1))
%Generate fit curve for plotting.
for i = 1:1001
       fit1(i,1) = (i-1)*(0.15+0.0)/1000-0.0;
       fit1(i,2) = amplitude1*exp(-(fit1(i,1)-offset1)^2/(2*sigma1^2));
endfor

%Plot histogram and curve fit.
```

```
figure(1)
hold on
bar(x1b,n1)
plot(fit1(:,1),fit1(:,2),'b','linewidth',2)
plot([p1exp,p1exp],[0,max(n1)],'r','linewidth',2)
legend("Simulated Histogram","Gaussian fit","Experimental Result")
xlabel("N = 1 Population Fraction")
ylabel("Counts")
title({["Random Number Simulation with ",num2str(numpts)," Attempts"],"N = 1
   Population Fraction Statistical Uncertainties"});
hold off
%Calculate the percentage of the ntuple that has less ions than observed.
j2 = 1;
j3 = 2;
j4 = 1;
nte(1) = 0;
for i = 1:length(nt)
       if nt(i,2) < p1exp
              ntl(j2) = nt(i,2);
              j2 = j2 + 1;
       elseif nt(i,2) == p1exp
              nte(j3) = nt(i,2);
              j3 = j3 + 1;
       elseif nt(i,2) > p1exp
              ntu(j4) = nt(i,2);
              j4 = j4 + 1;
       endif
endfor
perc = (length(ntl)+(length(nte)-1)/2)/numpts*100;
%Display results.
display(" ")
display(["Loss Fraction = (",num2str(f11)," + ",num2str(df1),") = ",num2str(f12
   )," with ",num2str(numpts)," points yields ",num2str(perc,3),"% of the
   Ntuple consistent with experiment."])
%Output histogram data to import for plotting.
%for i = 1:length(x1b)
%      op(i,1) = x1b(i);
%      op(i,2) = n1(i);
%endfor
%dlmwrite("AlH PP Loss Fraction Simulation Histogram.txt", op, ",");
```

D.2.1.8. Vibrationally-assisted Broadband Rotational Optical Cooling Temperature.

Vibrationally-assisted broadband rotational optical cooling temperature code.

```
%This script calculates the N = 1 Parity Pumping Temperature Limit using the
   population fraction ratio in N = 1 / N = 0.
%Resets and format for long number strings
clear all
close all
format long g

%Temperature Guess in K, number of rotational states to consider, and number of
    points in Ntuple.
%If T = 0, the experimental temperature is calculated and used. BEWARE!
   Counting with dT near 0 can then be off due to histogram granularity.
T = 0.;
dT = -0.27;
numstates = 20;
numpts = 1000;

%Thermal Population Numbers
ntAl(1) = 20;
ntAlH(1) = 330;
ntAl(2) = 28;
ntAlH(2) = 378;
ntAl(3) = 99;
ntAlH(3) = 487;
ntAl(4) = 37;
ntAlH(4) = 184;
ntAl(5) = 35;
ntAlH(5) = 198;
ntAl(6) = 33;
ntAlH(6) = 206;
ntAl(7) = 39;
ntAlH(7) = 320;
ntAl(8) = 27;
ntAlH(8) = 321;
%PROC Simulation Probabilities
pp(1) = 0.483;
pp(2) = 0.517;
pp(3) = 0.;
pp(4) = 0.;
pp(5) = 0.;
pp(6) = 0.;
pp(7) = 0.;
pp(8) = 0.;
%PROC Cooling Numbers
npAl(1) = 64;
npAlH(1) = 68;
npAl(2) = 145;
npAlH(2) = 174;
npAl(3) = 0;
npAlH(3) = 82;
npAl(4) = 6;
%npAl(4) = 2;
%npAl(4) = 1;
%npAl(4) = 0;
npAlH(4) = 291;
npAl(5) = 0;
npAlH(5) = 138;
npAl(6) = 0;
npAlH(6) = 166;
npAl(7) = 0;
npAlH(7) = 274;
npAl(7) = 0;
npAlH(7) = 100;
%Parity Pumping Simulation Probabilities
ppp(1) = 1.;
ppp(2) = 0.;
ppp(3) = 0.;
ppp(4) = 0.;
%Parity Pumping Numbers
nppAl(1) = 104;
nppAlH(1) = 5;
nppAl(2) = 1;
nppAlH(2) = 49;
nppAl(3) = 0;
nppAlH(3) = 11;
nppAl(4) = 0;
nppAlH(4) = 16;

%Calculating2
```

```
ntexp = ntAl + ntAlH;
npexp = npAl + npAlH;
nppexp = nppAl + nppAlH;
ptexp = ntAl ./ ntexp;
ppexp = npAl ./ npexp;
pppexp = nppAl ./ nppexp;

%Molecule Properties and constants.
hc = 1.23984193E-4; %eV*cm
B = 6.5632; %cm^-1
k = 8.61733E-5; %ev/K

%Set up Partition Function input parameters looking at all states.
numstates = numstates + 1;
for i = 1:numstates
       N(i) = i-1;
       g(i) = 2*(2*N(i)+1);
       E(i) = hc*B*N(i)*(N(i)+1);
endfor

%Calculate actual temperature if input temperature is set to 0.
R = pppexp(2)/pppexp(1);
if T == 0
       T1 = ((E(2)-E(1))/log(g(2)/(R*g(1))))/k;
       else
       T1 = T;
endif
T2 = T1 + dT;

%Map out thermal distribution probabilities starting by calculating Z.
Z = 0;
for i = 1:numstates
       Z = Z + g(i)*exp(-E(i)/(k*T2));
endfor
for i = 1:numstates
       pt(i) = 1/Z*g(i)*exp(-E(i)/(k*T2));
endfor

%Make Random Ntuple Set of N = 0 and N = 1.
for i = 1:numpts
       nt(i,1) = binornd(nppexp(1),pt(1))/nppexp(1);
       nt(i,2) = binornd(nppexp(2),pt(2))/nppexp(2);
endfor
nt(:,3) = nt(:,2) ./ nt(:,1);

%Make the histogram and fit by turning exponential fit into a polynomial fit.
for i = 1:20
       bins1(i) = (i-1)*(0.30-0.)/19;
endfor
[n1, x1b] = hist(nt(:,3),bins1);
%Exclude zeros and low values of the histogram for fitting since log(0) = -
   infinity kills the fit.
%dy1 is a root-(N+1) uncertainty for a weighted fit via wpolyfit.
j1 = 1;
for i = 1:length(n1)
       if n1(i) > 0 && n1(i) > max(n1)/100
              y1(j1) = log(n1(i));
              x1(j1) = x1b(i);
              dy1(j1) = (sqrt(n1(i)+1))/n1(i);
              j1 = j1 + 1;
       else
       endif
endfor
coef = wpolyfit(x1,y1,dy1,2);
a1 = coef(3);
b1 = coef(2);
c1 = coef(1);
sigma1 = sqrt(-1/(2*c1))
offset1 = -b1/(2*c1)
amplitude1 = exp(a1-b1^2/(4*c1))
%Generate fit curve for plotting.
for i = 1:1001
       fit1(i,1) = (i-1)*(0.30-0.)/1000;
       fit1(i,2) = amplitude1*exp(-(fit1(i,1)-offset1)^2/(2*sigma1^2));
endfor

%Plot histogram and curve fit.
figure(1)
hold on
bar(x1b,n1)
plot(fit1(:,1),fit1(:,2),'b','linewidth',2)
plot([pppexp(2)/pppexp(1),pppexp(2)/pppexp(1)],[0,max(n1)],'r','linewidth',2)
```

```
legend("Simulated Histogram","Gaussian fit","Experimental Result")
xlabel("Population Fraction Ratio of N=1/N=0")
ylabel("Counts")
title({["Random Number Simulation with ",num2str(numpts)," Attempts"],"Parity
   Pumping Temperature Limit"});
hold off

%Calculate the percentage of the ntuple that has less ions than observed.
j2 = 2;
j3 = 2;
j4 = 2;
ntl(1) = 0;
nte(1) = 0;
ntu(1) = 0;
for i = 1:length(nt)
        if nt(i,3) < pppexp(2)/pppexp(1)
                ntl(j2) = nt(i,3);
                j2 = j2 + 1;
        elseif nt(i,3) == pppexp(2)/pppexp(1)
                nte(j3) = nt(i,3);
                j3 = j3 + 1;
        elseif nt(i,3) > pppexp(2)/pppexp(1)
                ntu(j4) = nt(i,3);
                j4 = j4 + 1;
        endif
endfor
perc = (length(ntl)-1+(length(nte)-1)/2)/numpts*100;

%Display results.
display(["At T = (",num2str(T1)," + ",num2str(dT),") K = ",num2str(T2)," K with
    ",num2str(numpts)," points yields ",num2str(perc,3),"% of the Ntuple
   consistent with experiment."])
```

D.2.2. Rate Equation Simulations

The code below performs the rate equation simulation and is broken into three parts: (1) eq_distro.m, which sets the initial populations, blackbody temperature, total time, and time step for the simulation; (2) ratesolver.m, which uses the next section of code to generate the rate equations; and (3) rate_eqns.m, which sets up the system of coupled rate equations and includes the laser drives applied in the system.

Code below is used to simulate the fluorescence from AlH^+ using 1 repump laser at 100x saturation intensity.

D.2.2.1. eq_distro.m. eq_distro.m code.

```
%this calculates the equilibrium distribution of the internal states

%*****to run, SEND this file to octave***** (but read below for dependancies
    and user input)

%to run this program requires the following dependancies:

%              rate_eqns.m
%              ratesolver.m
%
%ratesolver.m requires user to input:

%gt : ground state on or off
%vxm : maximum ground state vibrational level
%Kxm : maximum ground state rotational level
%Jxm : maximum ground state total angular momentum quantum number

%et : excited state on or off
%vam : maximum excited state vibrational level
%Kam : maximum ground state rotational level
%Jam : maximum ground state total angular momentum quantum number


% T : blackbody temperature (in K)
% total_time : total time (in s) to simulate to (starting at t=0 seconds)
%t_step : maximum number of time steps (interval is set by total_time/t_step)
%
%matrix (the input file, see below) should have the format:
%
%      state'        v'     K'     J'     parity '      state"        v"
   K"     J"     parity"       E (cm-1)      A(s-1)
%
%where ' corresponds to the upper state and " corresponds to the lower state
%the Einstein A coefficient is assumed to have units of (s-1) and the energy is
    in (cm-1)
%if transitions are being drive by something other than blackbody radiation,
   you must include this by hand (see rate_eqns.m)


clear all
close all

%ground state defns

gt=1; % 1 for on 0 for off
vxm=3;
Kxm=4;
Jxm=4.5;

%excited state defns

et=1; % 1 for on 0 for off
vam=0;
Kam=1;
Jam=0.5;


T=300;
total_time=0.3;
t_step=3000;

%set the maximum number of states that need to be considered (assuming the
   doublet-Sigma <---> doublet-Pi transition)

sz=gt*(vxm+1)*(Kxm+Jxm+1/2)+et*(vam+1)*(2*(Kam+Jam-1)+1);
```

```
%define the initial population

p0=zeros(sz,1);


p0(1)=0.0;
p0(2)=0.322497372837403;
p0(3)=0.674333157024162;
p0(4)=0.0;
p0(5)=0.0;
p0(6)=0.0;
p0(7)=0.0;
p0(8)=0.0;
p0(9)=0.0;
p0(10)=0.0;
p0(11)=0.0;
p0(12)=0.0;
p0(13)=0.0;
p0(14)=0.0;
p0(15)=0.0;
p0(16)=0.0;
p0(17)=0.0;
p0(18)=0.0;
p0(19)=0.0;
p0(20)=0.0;
p0(21)=0.0;
%p0=p0*1.5661;
t=linspace(0,total_time,t_step);


matrix=dlmread('AlH_internal.txt','\t');                 %ENTER FILE NAME TO
   BE READ HERE FOR NON-ZERO A COEFFICIENTS
matrix_1=rate_eqns(T,gt,vxm,Kxm,Jxm,et,vam,Kam,Jam,matrix);
dlmwrite("matrix.txt", matrix_1);
printf("calculating. WAIT!")
prob_dist=lsode("ratesolver",p0,t);
printf("\n done")
dlmwrite("output_simulation.txt", prob_dist, "\t");
figure(1)
plot(t,prob_dist)
```

D.2.2.2. ratesolver.m. ratesolver.m code.

```
function BBReq = ratesolver(p,t)

%this function is called from eq_distro.m and uses rate_eqns.m
%to generate the rate equations.

matrix_1=dlmread("matrix.txt", ",");
BBReq=matrix_1*p;
endfunction
```

D.2.2.3. rate_eqns.m. rate_eqns.m code.

```
function soln=rate_eqns(T,gt,vxm,Kxm,Jxm,et,vam,Kam,Jam,IM)

    %rate_eqns(T,gt,vxm,Kxm,Jxm,et,vam,Kam,Jam,IM)
    %set up a system of coupled rate equations corresponding to excitations
       from the ^{2}\Sigma^{+} state to the ^{2}\Pi excited state (including
       transitions within the electronic ground state)
    %
    %the total number of equations is:
    %
    %          (vxm+1)*[Kxm+Jxm+1/2] for the ground state
    %          (vam+1)*(2(Kam+Jam-1)+1) for the excited state
    %
    %where vxm = maximum ground vibrational state
    %
    %      Jxm = maximum total angular momentum for the ground state
    %          vam = maximum excited vibrational state
    %          Kam = maximum rotational+lambda for the excited state
    %     Jam = maximum total angular momentum for the excited state
    %
    %
    %the equations are set up so that the ground state is listed first (
       starting from v=0) then the excited states are listed
    %
    %
    %
    %setup laser center frequency in cm-1

    %Here Elaser0 and Elaser1 are for the Parity Pumping.
    %Elaser0=27672.34738;
    %Elaser1=27690.83776;

    %Elaser2=27651.56357;Elaser3=27643.26467;Elaser4=27634.50352;Elaser5
       =27625.46747;Elaser6=27616.35945;Elaser7=27607.39008;Elaser8
       =27598.77501;Elaser9=27590.73538;Elaser10=27583.4997;Elaser11
       =27577.30611;
```

```
%Elaser12=27633.07319;Elaser13=27612.35062;Elaser14=27591.04139;Elaser15
   =27569.3138;Elaser16=27547.36616;Elaser17=27525.4124;Elaser18
   =27503.6741;Elaser19=27482.37788;Elaser20=27461.75586;Elaser21
   =27442.04751;Elaser22=27423.50196;

%This is the laser that drives k''=1 to k'=1 state
Elaser23=27707.91592;

%Elaser24=27651.56357;Elaser25=27643.26467;Elaser26=27634.50352;Elaser27
   =27625.46747;Elaser28=27616.35945;Elaser29=27607.39008;Elaser30
   =27598.77501;Elaser31=27590.73538;Elaser32=27583.4997;Elaser33
   =27577.30611;

%Elaser34=27661.94719;Elaser35=27647.83879;Elaser36=27634.6119;Elaser37
   =27622.27541;Elaser38=27610.86393;Elaser39=27600.43585;Elaser40
   =27591.07108;

%Here are the lasers that might accidentally hit for Parity Pumping
%Elaser51=27682.47762;Elaser52=27686.72681;Elaser53=27690.65719;Elaser54
   =27694.46075;Elaser55=27698.33712;Elaser56=27702.49099;Elaser57
   =27707.13251;

%This is the laser for v''=1 to v'=0 Q(1)repump
Elaser61=26202.67904;

%This is the laser for v''=2 to v'=0 Q(1)repump
%Elaser62=24845.44253;

%This is the laser for v''=3 to v'=0 Q(1)repump
%Elaser63=23636.20603;


   %Laser intensity goes here for B-coefficients
P = 100;


%set up the necessary constants
%
% speed of light
c=299792458;
%
```

```
% Plancks constant (hbar)
hbar=1.05457148e-034;
%
% Boltzmann's constant (k)
k=1.3806503e-023;
%
% set up the A and B matricies initially population all entries with zeros

max_index=gt*(vxm+1)*(Kxm+Jxm+1/2)+et*(vam+1)*(2*(Kam+Jam-1)+1);

Am=zeros(max_index,max_index);
Bm=zeros(max_index,max_index);
Bmrho=zeros(max_index,max_index);
Bmlaser=zeros(max_index,max_index);



%define the A, B, and Bmrho coefficients matrix

j=size(IM); %get the total number of rows in the input matrix

%for loop for each row in the input matrix

for i = 1:j(1)
    statep=IM(i,1);       %first start by obtain quantum numbers for upper
       and lower states for proper indexing of the matrix
    vp=IM(i,2);
    Kp=IM(i,3);
    Jp=IM(i,4);
    Pp=IM(i,5);

    statepp=IM(i,6);
    vpp=IM(i,7);
    Kpp=IM(i,8);
    Jpp=IM(i,9);
    Ppp=IM(i,10);


    if(statep==1)
       if(statepp==0)
          g=(2*Jp+1)/(2*Jpp+1);
    endif;
else
    g=1;
endif;


%determine row index from quantum numbers for lower state

if(statepp==0)  %if the lower state is the ground state
    row=vpp*(Kxm+Jxm+1/2)+(Kpp+Jpp+1/2);
elseif(statepp==1) %if the lower state is in the excited state
    row=gt*(vxm+1)*(Kxm+Jxm+1/2)+2*(Kpp+Jpp-1)+0.5*abs((-1)^(Kpp-1)-Ppp)
        +vpp*(2*(Kam+Jam-1)+1);
endif;

%determine column index from quantum numbers for upper state

if(statep==0)
    col=vp*(Kxm+Jxm+1/2)+(Kp+Jp+1/2);
elseif(statep==1)
    col=gt*(vxm+1)*(Kxm+Jxm+1/2)+2*(Kp+Jp-1)+0.5*abs((-1)^(Kp-1)-Pp)+vp
        *(2*(Kam+Jam-1)+1);
endif;


Am(col,col)+=-IM(i,12); %A coefficient correspond to loss from the upper
     state via spontaneous emission
Am(row,col)+=IM(i,12); %A coefficient correspond to gain from the upper
    state to lower state via spontaneous emission
if(Am(row,col)>0)            %avoids division by zeros, and if there's
    no A coefficient then there's no B coefficient, anyway!



    Bm(row,col)+=IM(i,12)*pi^2*c^3/(hbar*(IM(i,11)*100*c*2*pi)^3); %gain
         from upper state to lower state via stimulated emission
    Bmrho(row,col)+=IM(i,12)/(exp(hbar*(IM(i,11)*100*c*2*pi)/(k*T))-1);
        %gain from upper state to lower state via stimulated emission

    %Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser0)/(3.3e-5))^2)
        );
```

```
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser1)/(3.3e-5))^2)
    );
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser2)/(3.3e-5))^2)
    );
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser3)/(3.3e-5))^2)
    );
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser4)/(3.3e-5))^2)
    );
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser5)/(3.3e-5))^2)
    );
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser6)/(3.3e-5))^2)
    );
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser7)/(3.3e-5))^2)
    );
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser8)/(3.3e-5))^2)
    );
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser9)/(3.3e-5))^2)
    );
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser10)/(3.3e-5))
    ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser11)/(3.3e-5))
    ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser12)/(3.3e-5))
    ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser13)/(3.3e-5))
    ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser14)/(3.3e-5))
    ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser15)/(3.3e-5))
    ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser16)/(3.3e-5))
    ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser17)/(3.3e-5))
    ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser18)/(3.3e-5))
    ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser19)/(3.3e-5))
    ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser20)/(3.3e-5))
    ^2));
```

```
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser21)/(3.3e-5))
    ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser22)/(3.3e-5))
    ^2));
Bmlaser(row,col)+=P*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser23)/(3.3e-5))
    ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser24)/(3.3e-5))
    ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser25)/(3.3e-5))
    ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser26)/(3.3e-5))
    ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser27)/(3.3e-5))
    ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser28)/(3.3e-5))
    ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser29)/(3.3e-5))
    ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser30)/(3.3e-5))
    ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser31)/(3.3e-5))
    ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser32)/(3.3e-5))
    ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser33)/(3.3e-5))
    ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser34)/(3.3e-5))
    ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser35)/(3.3e-5))
    ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser36)/(3.3e-5))
    ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser37)/(3.3e-5))
    ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser38)/(3.3e-5))
    ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser39)/(3.3e-5))
    ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser40)/(3.3e-5))
    ^2));
```

```
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser51)/(3.3e-5))
   ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser52)/(3.3e-5))
   ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser53)/(3.3e-5))
   ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser54)/(3.3e-5))
   ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser55)/(3.3e-5))
   ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser56)/(3.3e-5))
   ^2));
%Bmlaser(row,col)+=IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser57)/(3.3e-5))
   ^2));
Bmlaser(row,col)+=P*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser61)/(3.3e-5))
   ^2));
          %Bmlaser(row,col)+=P*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser62
             )/(3.3e-5))^2));
          %Bmlaser(row,col)+=P*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser63
             )/(3.3e-5))^2));




Bm(col,col)+=-Bm(row,col); %loss from upper state to lower state
   from stimulated emission
Bmrho(col,col)+=-Bmrho(row,col); %loss from upper state to lower
   state from stimulated emission

%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser0)/(3.3e-5))
   ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser1)/(3.3e-5))
   ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser2)/(3.3e-5))
   ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser3)/(3.3e-5))
   ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser4)/(3.3e-5))
   ^2));
```

```
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser5)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser6)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser7)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser8)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser9)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser10)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser11)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser12)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser13)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser14)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser15)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser16)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser17)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser18)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser19)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser20)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser21)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser22)/(3.3e-5))
    ^2));
Bmlaser(col,col)+=-P*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser23)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser24)/(3.3e-5))
    ^2));
```

```
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser25)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser26)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser27)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser28)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser29)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser30)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser31)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser32)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser33)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser34)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser35)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser36)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser37)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser38)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser39)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser40)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser51)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser52)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser53)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser54)/(3.3e-5))
    ^2));
```

```
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser55)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser56)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser57)/(3.3e-5))
    ^2));
Bmlaser(col,col)+=-P*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser61)/(3.3e-5))
    ^2));
%Bmlaser(col,col)+=-P*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser62)/(3.3e-5)
    )^2));
            %Bmlaser(col,col)+=-P*IM(i,12)*(1/(1+(2*(IM(i,11)-
                Elaser63)/(3.3e-5))^2));



Bm(col,row)+=IM(i,12)*pi^2*c^3/(hbar*(IM(i,11)*100*c*2*pi)^3)*(2*Jp
    +1)/(2*Jpp+1); %gain from upper state to lower state via %
    stimulated absorption **note that you need to properly account
    for deg. factor
Bmrho(col,row)+=g*IM(i,12)/(exp(hbar*(IM(i,11)*100*c*2*pi)/(k*T))-1)
    *(2*Jp+1)/(2*Jpp+1); %gain from upper state to lower state via %
    stimulated absorption **note that you need to properly account
    for deg. factor

%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser0)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser1)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser2)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser3)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser4)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser5)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser6)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser7)/(3.3e-5))
    ^2));
```

```
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser8)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser9)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser10)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser11)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser12)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser13)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser14)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser15)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser16)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser17)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser18)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser19)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser20)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser21)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser22)/(3.3e-5))
    ^2));
Bmlaser(col,row)+=P*g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser23)/(3.3e-5)
    )^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser24)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser25)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser26)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser27)/(3.3e-5))
    ^2));
```

```
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser28)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser29)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser30)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser31)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser32)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser33)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser34)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser35)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser36)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser37)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser38)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser39)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser40)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser51)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser52)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser53)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser54)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser55)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser56)/(3.3e-5))
    ^2));
%Bmlaser(col,row)+=g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser57)/(3.3e-5))
    ^2));
```

```
Bmlaser(col,row)+=P*g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser61)/(3.3e-5)
    )^2));
%Bmlaser(col,row)+=P*g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser62)/(3.3e
    -5))^2));
            %Bmlaser(col,row)+=P*g*IM(i,12)*(1/(1+(2*(IM(i,11)-
                Elaser63)/(3.3e-5))^2));



Bm(row,row)+=-Bm(col,row); %loss from lower state to upper state
    from stimulated absorption
Bmrho(row,row)+=-Bmrho(col,row); %loss from lower state to upper
    state from stimulated absorption

%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser0)/(3.3e-5))
    ^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser1)/(3.3e-5))
    ^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser2)/(3.3e-5))
    ^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser3)/(3.3e-5))
    ^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser4)/(3.3e-5))
    ^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser5)/(3.3e-5))
    ^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser6)/(3.3e-5))
    ^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser7)/(3.3e-5))
    ^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser8)/(3.3e-5))
    ^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser9)/(3.3e-5))
    ^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser10)/(3.3e-5)
    )^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser11)/(3.3e-5)
    )^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser12)/(3.3e-5)
    )^2));
```

```
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser13)/(3.3e-5)
    )^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser14)/(3.3e-5)
    )^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser15)/(3.3e-5)
    )^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser16)/(3.3e-5)
    )^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser17)/(3.3e-5)
    )^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser18)/(3.3e-5)
    )^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser19)/(3.3e-5)
    )^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser20)/(3.3e-5)
    )^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser21)/(3.3e-5)
    )^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser22)/(3.3e-5)
    )^2));
Bmlaser(row,row)+=-P*g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser23)/(3.3e
    -5))^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser24)/(3.3e-5)
    )^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser25)/(3.3e-5)
    )^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser26)/(3.3e-5)
    )^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser27)/(3.3e-5)
    )^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser28)/(3.3e-5)
    )^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser29)/(3.3e-5)
    )^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser30)/(3.3e-5)
    )^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser31)/(3.3e-5)
    )^2));
%Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser32)/(3.3e-5)
    )^2));
```

```
        %Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser33)/(3.3e-5)
            )^2));
        %Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser34)/(3.3e-5)
            )^2));
        %Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser35)/(3.3e-5)
            )^2));
        %Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser36)/(3.3e-5)
            )^2));
        %Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser37)/(3.3e-5)
            )^2));
        %Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser38)/(3.3e-5)
            )^2));
        %Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser39)/(3.3e-5)
            )^2));
        %Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser40)/(3.3e-5)
            )^2));
        %Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser51)/(3.3e-5)
            )^2));
        %Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser52)/(3.3e-5)
            )^2));
        %Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser53)/(3.3e-5)
            )^2));
        %Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser54)/(3.3e-5)
            )^2));
        %Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser55)/(3.3e-5)
            )^2));
        %Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser56)/(3.3e-5)
            )^2));
        %Bmlaser(row,row)+=-g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser57)/(3.3e-5)
            )^2));
        Bmlaser(row,row)+=-P*g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser61)/(3.3e
            -5))^2));
        %Bmlaser(row,row)+=-P*g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser62)/(3.3e
            -5))^2));
        %Bmlaser(row,row)+=-P*g*IM(i,12)*(1/(1+(2*(IM(i,11)-Elaser63)/(3.3e
            -5))^2));

    endif
endfor

dlmwrite("laser_output.txt", Bmlaser);
```

```
soln=(Am+Bmrho+Bmlaser);         %the total interaction matrix is the sum of
     spontaneous and stimulated processes
endfunction
```

D.2.2.4. AlH$^+$ Rate Equation Simulations Input Data from JHVN. The data below, in comma delimited format, is used for the rate equation modeling. Information calculated by JHVN.

```
state',v',K',J',parity',"state""""",v'',K'',J'',"parity""""",E (cm-1),A (s-1)
1,0,1,0.5,-1,0,0,0,0.5,1,27672.34738,10297702.14
1,0,1,0.5,-1,0,0,2,1.5,1,27633.07319,5126959.531
1,0,1,0.5,1,0,0,1,0.5,-1,27659.24289,10283079.35
1,0,1,0.5,1,0,0,1,1.5,-1,27659.24289,5141539.676
1,0,1,1.5,1,0,0,1,0.5,-1,27677.73327,1599197.755
1,0,1,1.5,1,0,0,1,1.5,-1,27677.73327,9892383.119
1,0,1,1.5,1,0,0,3,2.5,-1,27612.35062,3935967.291
1,0,1,1.5,-1,0,0,0,0.5,1,27690.83776,3702314.509
1,0,1,1.5,-1,0,0,2,1.5,1,27651.56357,6517280.826
1,0,1,1.5,-1,0,0,2,2.5,1,27651.56357,5207910.536
1,0,2,2.5,-1,0,0,2,1.5,1,27682.47762,1594056.938
1,0,2,2.5,-1,0,0,2,2.5,1,27682.47762,10676247.96
1,0,2,2.5,-1,0,0,4,3.5,1,27591.04139,3161680.899
1,0,2,2.5,1,0,0,1,1.5,-1,27708.64731,4821255.725
1,0,2,2.5,1,0,0,3,2.5,-1,27643.26467,5145375.21
1,0,2,2.5,1,0,0,3,3.5,-1,27643.26467,5465320.934
1,0,3,3.5,1,0,0,3,2.5,-1,27686.72681,1412139.727
1,0,3,3.5,1,0,0,3,3.5,-1,27686.72681,11448195.4
1,0,3,3.5,1,0,0,5,4.5,-1,27569.3138,2577241.607
1,0,3,3.5,-1,0,0,2,2.5,1,27725.93975,5507590.904
1,0,3,3.5,-1,0,0,4,3.5,1,27634.50352,4185106.41
1,0,3,3.5,-1,0,0,4,4.5,1,27634.50352,5744880.083
1,0,4,4.5,-1,0,0,4,3.5,1,27690.65719,1211083.174
1,0,4,4.5,-1,0,0,4,4.5,1,27690.65719,12114001.63
1,0,4,4.5,-1,0,0,6,5.5,1,27547.36616,2118912.356
1,0,4,4.5,1,0,0,3,3.5,-1,27742.88047,6001888.383
1,0,4,4.5,1,0,0,5,4.5,-1,27625.46747,3440541.91
1,0,4,4.5,1,0,0,5,5.5,-1,27625.46747,6001632.756
1,0,5,5.5,1,0,0,5,4.5,-1,27694.46075,1028244.951
1,0,5,5.5,1,0,0,5,5.5,-1,27694.46075,12667103.46
1,0,5,5.5,1,0,0,7,6.5,-1,27525.4124,1755717.587
1,0,5,5.5,-1,0,0,4,4.5,1,27759.65047,6377024.45
1,0,5,5.5,-1,0,0,6,5.5,1,27616.35945,2850924.183
1,0,5,5.5,-1,0,0,6,6.5,1,27616.35945,6223280.441
```

```
1,0,6,6.5,-1,0,0,6,5.5,1,27698.33712,872136.1649
1,0,6,6.5,-1,0,0,6,6.5,1,27698.33712,13119963.62
1,0,6,6.5,-1,0,0,8,7.5,1,27503.6741,1466653.731
1,0,6,6.5,1,0,0,5,5.5,-1,27776.43843,6668444.01
1,0,6,6.5,1,0,0,7,6.5,-1,27607.39008,2381283.603
1,0,6,6.5,1,0,0,7,7.5,-1,27607.39008,6409317.939
1,0,7,7.5,1,0,0,7,6.5,-1,27702.49099,742211.6908
1,0,7,7.5,1,0,0,7,7.5,-1,27702.49099,13489344.9
1,0,7,7.5,1,0,0,9,8.5,-1,27482.37788,1235602.469
1,0,7,7.5,-1,0,0,6,6.5,1,27793.43804,6898343.682
1,0,7,7.5,-1,0,0,8,7.5,1,27598.77501,2005689.503
1,0,7,7.5,-1,0,0,8,8.5,1,27598.77501,6563577.73
1,0,8,8.5,-1,0,0,8,7.5,1,27707.13251,635136.7821
1,0,8,8.5,-1,0,0,8,8.5,1,27707.13251,13791446.38
1,0,8,8.5,-1,0,0,10,9.5,1,27461.75586,1049898.488
1,0,8,8.5,1,0,0,7,7.5,-1,27810.84849,7082253.482
1,0,8,8.5,1,0,0,9,8.5,-1,27590.73538,1703775.309
1,0,8,8.5,1,0,0,9,9.5,-1,27590.73538,6691094.193
1,0,9,9.5,1,0,0,9,8.5,-1,27712.47923,547067.6387
1,0,9,9.5,1,0,0,9,9.5,-1,27712.47923,14040286.72
1,0,9,9.5,1,0,0,11,10.5,-1,27442.04751,899640.5989
1,0,9,9.5,-1,0,0,8,8.5,1,27828.87636,7231520.638
1,0,9,9.5,-1,0,0,10,9.5,1,27583.4997,1459540.793
1,0,9,9.5,-1,0,0,10,10.5,1,27583.4997,6796792.716
1,0,10,10.5,-1,0,0,10,9.5,1,27718.75831,474469.3644
1,0,10,10.5,-1,0,0,10,10.5,1,27718.75831,14247405.4
1,0,10,10.5,-1,0,0,12,11.5,1,27423.50196,777154.7939
1,0,10,10.5,1,0,0,9,9.5,-1,27847.73783,7354588.793
1,0,10,10.5,1,0,0,11,10.5,-1,27577.30611,1260532.008
1,0,10,10.5,1,0,0,11,11.5,-1,27577.30611,6885012.944
1,0,2,1.5,-1,0,0,0,0.5,1,27788.33039,6686170.592
1,0,2,1.5,-1,0,0,2,1.5,1,27749.0562,7950548.126
1,0,2,1.5,-1,0,0,2,2.5,1,27749.0562,966955.5539
1,0,2,1.5,1,0,0,1,0.5,-1,27775.22589,8796825.61
1,0,2,1.5,1,0,0,1,1.5,-1,27775.22589,4580887.766
1,0,2,1.5,1,0,0,3,2.5,-1,27709.84325,2225965.767
1,0,3,2.5,1,0,0,1,1.5,-1,27810.50683,4532820.321
1,0,3,2.5,1,0,0,3,2.5,-1,27745.12419,9921459.695
1,0,3,2.5,1,0,0,3,3.5,-1,27745.12419,1146359.781
1,0,3,2.5,-1,0,0,2,1.5,1,27784.33713,7769191.558
1,0,3,2.5,-1,0,0,2,2.5,1,27784.33713,4393605.384
```

```
1,0,3,2.5,-1,0,0,4,3.5,1,27692.90091,3437886.323
1,0,4,3.5,-1,0,0,2,2.5,1,27833.58027,3409800.381
1,0,4,3.5,-1,0,0,4,3.5,1,27742.14404,11082483.3
1,0,4,3.5,-1,0,0,4,4.5,1,27742.14404,1104939.305
1,0,4,3.5,1,0,0,3,2.5,-1,27794.36732,7515204.725
1,0,4,3.5,1,0,0,3,3.5,-1,27794.36732,3821106.888
1,0,4,3.5,1,0,0,5,4.5,-1,27676.95432,4261017.028
1,0,5,4.5,1,0,0,3,3.5,-1,27857.4615,2678227.967
1,0,5,4.5,1,0,0,5,4.5,-1,27740.04849,11919497.91
1,0,5,4.5,1,0,0,5,5.5,-1,27740.04849,996197.9715
1,0,5,4.5,-1,0,0,4,3.5,1,27805.23822,7479465.206
1,0,5,4.5,-1,0,0,4,4.5,1,27805.23822,3246871.837
1,0,5,4.5,-1,0,0,6,5.5,1,27661.94719,4867784.348
1,0,6,5.5,-1,0,0,4,4.5,1,27882.07687,2156647.586
1,0,6,5.5,-1,0,0,6,5.5,1,27738.78584,12560271.64
1,0,6,5.5,-1,0,0,6,6.5,1,27738.78584,874237.6815
1,0,6,5.5,1,0,0,5,4.5,-1,27816.88714,7516165.679
1,0,6,5.5,1,0,0,5,5.5,-1,27816.88714,2744308.139
1,0,6,5.5,1,0,0,7,6.5,-1,27647.83879,5331003.896
1,0,7,6.5,1,0,0,5,5.5,-1,27907.37623,1767910.927
1,0,7,6.5,1,0,0,7,6.5,-1,27738.32787,13062198.87
1,0,7,6.5,1,0,0,7,7.5,-1,27738.32787,759129.9575
1,0,7,6.5,-1,0,0,6,5.5,1,27829.27492,7575116.927
1,0,7,6.5,-1,0,0,6,6.5,1,27829.27492,2323356.367
1,0,7,6.5,-1,0,0,8,7.5,1,27634.6119,5691241.775
1,0,8,7.5,-1,0,0,6,6.5,1,27933.33557,1470421.706
1,0,8,7.5,-1,0,0,8,7.5,1,27738.67254,13460530.59
1,0,8,7.5,-1,0,0,8,8.5,1,27738.67254,657462.0396
1,0,8,7.5,1,0,0,7,6.5,-1,27842.38852,7637449.389
1,0,8,7.5,1,0,0,7,7.5,-1,27842.38852,1976614.192
1,0,8,7.5,1,0,0,9,8.5,-1,27622.27541,5975010.905
1,0,9,8.5,1,0,0,7,7.5,-1,27959.95657,1238444.537
1,0,9,8.5,1,0,0,9,8.5,-1,27739.84346,13780136.11
1,0,9,8.5,1,0,0,9,9.5,-1,27739.84346,570295.3852
1,0,9,8.5,-1,0,0,8,7.5,1,27856.24059,7696256.075
1,0,9,8.5,-1,0,0,8,8.5,1,27856.24059,1692460.818
1,0,9,8.5,-1,0,0,10,9.5,1,27610.86393,6201033.207
1,0,10,9.5,-1,0,0,8,8.5,1,27987.26469,1054791.375
1,0,10,9.5,-1,0,0,10,9.5,1,27741.88804,14039463.88
1,0,10,9.5,-1,0,0,10,10.5,1,27741.88804,496543.423
1,0,10,9.5,1,0,0,9,8.5,-1,27870.86756,7749489.257
```

```
1,0,10,9.5,1,0,0,9,9.5,-1,27870.86756,1459426.634
1,0,10,9.5,1,0,0,11,10.5,-1,27600.43585,6382998.512
1,0,11,10.5,1,0,0,9,9.5,-1,28015.30696,907472.5324
1,0,11,10.5,1,0,0,11,10.5,-1,27744.87524,14252443.04
1,0,11,10.5,1,0,0,11,11.5,-1,27744.87524,434436.8427
1,0,11,10.5,-1,0,0,10,9.5,1,27886.32743,7797075.687
1,0,11,10.5,-1,0,0,10,10.5,1,27886.32743,1267569.041
1,0,11,10.5,-1,0,0,12,11.5,1,27591.07108,6531091.52
1,0,1,0.5,-1,0,1,0,0.5,1,26149.507,119645.3963
1,0,1,0.5,-1,0,1,2,1.5,1,26110.23281,59553.55795
1,0,1,0.5,1,0,1,1,0.5,-1,26136.4025,119465.6101
1,0,1,0.5,1,0,1,1,1.5,-1,26136.4025,59732.80503
1,0,1,1.5,1,0,1,1,0.5,-1,26154.88712,18581.13829
1,0,1,1.5,1,0,1,1,1.5,-1,26154.88712,114939.968
1,0,1,1.5,1,0,1,3,2.5,-1,26089.50447,45713.23822
1,0,1,1.5,-1,0,1,0,0.5,1,26167.99161,43020.88621
1,0,1,1.5,-1,0,1,2,1.5,1,26128.71742,75712.01123
1,0,1,1.5,-1,0,1,2,2.5,1,26128.71742,60500.90391
1,0,2,2.5,-1,0,1,2,1.5,1,26159.60648,18521.90833
1,0,2,2.5,-1,0,1,2,2.5,1,26159.60648,124051.0809
1,0,2,2.5,-1,0,1,4,3.5,1,26068.17025,36715.42475
1,0,2,2.5,1,0,1,1,1.5,-1,26185.77618,56029.10698
1,0,2,2.5,1,0,1,3,2.5,-1,26120.39354,59771.1146
1,0,2,2.5,1,0,1,3,3.5,-1,26120.39354,63487.75561
1,0,3,3.5,1,0,1,3,2.5,-1,26163.7884,16408.4615
1,0,3,3.5,1,0,1,3,3.5,-1,26163.7884,133023.1491
1,0,3,3.5,1,0,1,5,4.5,-1,26046.37539,29924.18312
1,0,3,3.5,-1,0,1,2,2.5,1,26203.00135,64011.6643
1,0,3,3.5,-1,0,1,4,3.5,1,26111.56512,48613.10605
1,0,3,3.5,-1,0,1,4,4.5,1,26111.56512,66731.03079
1,0,4,4.5,-1,0,1,4,3.5,1,26167.57694,14072.39003
1,0,4,4.5,-1,0,1,4,4.5,1,26167.57694,140760.7335
1,0,4,4.5,-1,0,1,6,5.5,1,26024.28591,24598.71191
1,0,4,4.5,1,0,1,3,3.5,-1,26219.80022,69762.90398
1,0,4,4.5,1,0,1,5,4.5,-1,26102.38722,39961.50006
1,0,4,4.5,1,0,1,5,5.5,-1,26102.38722,69708.27679
1,0,5,5.5,1,0,1,5,4.5,-1,26171.12257,11947.80301
1,0,5,5.5,1,0,1,5,5.5,-1,26171.12257,147186.7736
1,0,5,5.5,1,0,1,7,6.5,-1,26002.07422,20378.87896
1,0,5,5.5,-1,0,1,4,4.5,1,26236.31229,74128.91263
1,0,5,5.5,-1,0,1,6,5.5,1,26093.02126,33110.26414
```

```
1,0,5,5.5,-1,0,1,6,6.5,1,26093.02126,72276.37286
1,0,6,6.5,-1,0,1,6,5.5,1,26174.57418,10133.63444
1,0,6,6.5,-1,0,1,6,6.5,1,26174.57418,152445.1347
1,0,6,6.5,-1,0,1,8,7.5,1,25979.91115,17020.47161
1,0,6,6.5,1,0,1,5,5.5,-1,26252.67548,77520.88054
1,0,6,6.5,1,0,1,7,6.5,-1,26083.62713,27652.99465
1,0,6,6.5,1,0,1,7,7.5,-1,26083.62713,74429.1165
1,0,7,7.5,1,0,1,7,6.5,-1,26178.07649,8623.581535
1,0,7,7.5,1,0,1,7,7.5,-1,26178.07649,156729.4978
1,0,7,7.5,1,0,1,9,8.5,-1,25957.96338,14336.0942
1,0,7,7.5,-1,0,1,6,6.5,1,26269.02353,80196.0454
1,0,7,7.5,-1,0,1,8,7.5,1,26074.36051,23288.3347
1,0,7,7.5,-1,0,1,8,8.5,1,26074.36051,76210.59729
1,0,8,8.5,-1,0,1,8,7.5,1,26181.77046,7378.91836
1,0,8,8.5,-1,0,1,8,8.5,1,26181.77046,160226.8358
1,0,8,8.5,-1,0,1,10,9.5,1,25936.39381,12178.51494
1,0,8,8.5,1,0,1,7,7.5,-1,26285.48644,82334.14084
1,0,8,8.5,1,0,1,9,8.5,-1,26065.37333,19779.59954
1,0,8,8.5,1,0,1,9,9.5,-1,26065.37333,77678.76602
1,0,9,9.5,1,0,1,9,8.5,-1,26185.79523,6354.996837
1,0,9,9.5,1,0,1,9,9.5,-1,26185.79523,163098.6214
1,0,9,9.5,1,0,1,11,10.5,-1,25915.36351,10432.64858
1,0,9,9.5,-1,0,1,8,8.5,1,26302.19236,84066.239
1,0,9,9.5,-1,0,1,10,9.5,1,26056.8157,16940.85452
1,0,9,9.5,-1,0,1,10,10.5,1,26056.8157,78890.20792
1,0,10,10.5,-1,0,1,10,9.5,1,26190.29031,5510.753072
1,0,10,10.5,-1,0,1,10,10.5,1,26190.29031,165477.35
1,0,10,10.5,-1,0,1,12,11.5,1,25895.03395,9009.307265
1,0,10,10.5,1,0,1,9,9.5,-1,26319.26983,85489.60165
1,0,10,10.5,1,0,1,11,10.5,-1,26048.83811,14627.38127
1,0,10,10.5,1,0,1,11,11.5,-1,26048.83811,79894.60698
1,0,2,1.5,-1,0,1,0,0.5,1,26334.09039,78351.66153
1,0,2,1.5,-1,0,1,2,1.5,1,26294.8162,93146.38245
1,0,2,1.5,-1,0,1,2,2.5,1,26294.8162,11328.57891
1,0,2,1.5,1,0,1,1,0.5,-1,26320.98589,103077.2429
1,0,2,1.5,1,0,1,1,1.5,-1,26320.98589,53676.78094
1,0,2,1.5,1,0,1,3,2.5,-1,26255.60325,26072.66535
1,0,3,2.5,1,0,1,1,1.5,-1,26356.26683,53124.71745
1,0,3,2.5,1,0,1,3,2.5,-1,26290.88419,116234.2942
1,0,3,2.5,1,0,1,3,3.5,-1,26290.88419,13430.11251
1,0,3,2.5,-1,0,1,2,1.5,1,26330.09713,91040.83795
```

```
1,0,3,2.5,-1,0,1,2,2.5,1,26330.09713,51485.08861
1,0,3,2.5,-1,0,1,4,3.5,1,26238.66091,40263.7584
1,0,4,3.5,-1,0,1,2,2.5,1,26379.34027,39968.39202
1,0,4,3.5,-1,0,1,4,3.5,1,26287.90404,129833.8854
1,0,4,3.5,-1,0,1,4,4.5,1,26287.90404,12944.62254
1,0,4,3.5,1,0,1,3,2.5,-1,26340.12732,88069.83863
1,0,4,3.5,1,0,1,3,3.5,-1,26340.12732,44779.1217
1,0,4,3.5,1,0,1,5,4.5,-1,26222.71432,49899.3011
1,0,5,4.5,1,0,1,3,3.5,-1,26403.2215,31397.6276
1,0,5,4.5,1,0,1,5,4.5,-1,26285.80849,139637.9568
1,0,5,4.5,1,0,1,5,5.5,-1,26285.80849,11670.54606
1,0,5,4.5,-1,0,1,4,3.5,1,26350.99822,87656.68725
1,0,5,4.5,-1,0,1,4,4.5,1,26350.99822,38052.18973
1,0,5,4.5,-1,0,1,6,5.5,1,26207.70719,56999.79886
1,0,6,5.5,-1,0,1,4,4.5,1,26427.83687,25286.68092
1,0,6,5.5,-1,0,1,6,5.5,1,26284.54584,147143.5652
1,0,6,5.5,-1,0,1,6,6.5,1,26284.54584,10241.69325
1,0,6,5.5,1,0,1,5,4.5,-1,26362.64714,88092.91125
1,0,6,5.5,1,0,1,5,5.5,-1,26362.64714,32164.55087
1,0,6,5.5,1,0,1,7,6.5,-1,26193.59879,62418.6111
1,0,7,6.5,1,0,1,5,5.5,-1,26453.13623,20731.84803
1,0,7,6.5,1,0,1,7,6.5,-1,26284.08787,153023.2227
1,0,7,6.5,1,0,1,7,7.5,-1,26284.08787,8893.182053
1,0,7,6.5,-1,0,1,6,5.5,1,26375.03492,88790.38733
1,0,7,6.5,-1,0,1,6,6.5,1,26375.03492,27232.8089
1,0,7,6.5,-1,0,1,8,7.5,1,26180.3719,66631.1815
1,0,8,7.5,-1,0,1,6,6.5,1,26479.09557,17245.90953
1,0,8,7.5,-1,0,1,8,7.5,1,26284.43254,157689.9907
1,0,8,7.5,-1,0,1,8,8.5,1,26284.43254,7702.161681
1,0,8,7.5,1,0,1,7,6.5,-1,26388.14852,89527.98076
1,0,8,7.5,1,0,1,7,7.5,-1,26388.14852,23170.33715
1,0,8,7.5,1,0,1,9,8.5,-1,26168.03541,69948.25078
1,0,9,8.5,1,0,1,7,7.5,-1,26505.71657,14527.43318
1,0,9,8.5,1,0,1,9,8.5,-1,26285.60346,161435.2975
1,0,9,8.5,1,0,1,9,9.5,-1,26285.60346,6681.051944
1,0,9,8.5,-1,0,1,8,7.5,1,26402.00059,90224.74383
1,0,9,8.5,-1,0,1,8,8.5,1,26402.00059,19841.05548
1,0,9,8.5,-1,0,1,10,9.5,1,26156.62393,72589.24644
1,0,10,9.5,-1,0,1,8,8.5,1,26533.02469,12375.09783
1,0,10,9.5,-1,0,1,10,9.5,1,26287.64804,164475.353
1,0,10,9.5,-1,0,1,10,10.5,1,26287.64804,5817.113495
```

```
1,0,10,9.5,1,0,1,9,8.5,-1,26416.62756,90856.68576
1,0,10,9.5,1,0,1,9,9.5,-1,26416.62756,17110.63306
1,0,10,9.5,1,0,1,11,10.5,-1,26146.19585,74714.62232
1,0,11,10.5,1,0,1,9,9.5,-1,26561.06696,10648.46569
1,0,11,10.5,1,0,1,11,10.5,-1,26290.63524,166973.4334
1,0,11,10.5,1,0,1,11,11.5,-1,26290.63524,5089.612429
1,0,11,10.5,-1,0,1,10,9.5,1,26432.08743,91422.9693
1,0,11,10.5,-1,0,1,10,10.5,1,26432.08743,14862.61391
1,0,11,10.5,-1,0,1,12,11.5,1,26136.83108,76443.75921
0,1,0,0.5,1,0,0,1,0.5,-1,1640.840571,16.66666667
0,1,0,0.5,1,0,0,1,1.5,-1,1640.840571,33.33333333
0,1,1,0.5,-1,0,0,0,0.5,1,1666.363786,35.71428571
0,1,1,0.5,-1,0,0,2,1.5,1,1627.114802,14.28571429
0,1,1,1.5,-1,0,0,0,0.5,1,1666.363786,35.71428571
0,1,1,1.5,-1,0,0,2,1.5,1,1627.114802,1.428571429
0,1,1,1.5,-1,0,0,2,2.5,1,1627.114802,12.85714286
0,1,2,1.5,1,0,0,1,0.5,-1,1678.176664,25.36231884
0,1,2,1.5,1,0,0,1,1.5,-1,1678.176664,5.072463768
0,1,2,1.5,1,0,0,3,2.5,-1,1612.730955,19.56521739
0,1,2,2.5,1,0,0,1,1.5,-1,1678.176664,30.43478261
0,1,2,2.5,1,0,0,3,2.5,-1,1612.730955,0.931677019
0,1,2,2.5,1,0,0,3,3.5,-1,1612.730955,18.63354037
0,1,3,2.5,-1,0,0,2,1.5,1,1689.362328,26.80851064
0,1,3,2.5,-1,0,0,2,2.5,1,1689.362328,1.914893617
0,1,3,2.5,-1,0,0,4,3.5,1,1597.67379,21.27659574
0,1,3,3.5,-1,0,0,2,2.5,1,1689.362328,28.72340426
0,1,3,3.5,-1,0,0,4,3.5,1,1597.67379,0.591016548
0,1,3,3.5,-1,0,0,4,4.5,1,1597.67379,20.6855792
0,1,4,3.5,1,0,0,3,2.5,-1,1699.92097,26.85352622
0,1,4,3.5,1,0,0,3,3.5,-1,1699.92097,0.994575045
0,1,4,3.5,1,0,0,5,4.5,-1,1581.925057,22.15189873
0,1,4,4.5,1,0,0,3,3.5,-1,1699.92097,27.84810127
0,1,4,4.5,1,0,0,5,4.5,-1,1581.925057,0.402761795
0,1,4,4.5,1,0,0,5,5.5,-1,1581.925057,21.74913694
0,1,5,4.5,-1,0,0,4,3.5,1,1709.849772,26.70401494
0,1,5,4.5,-1,0,0,4,4.5,1,1709.849772,0.60690943
0,1,5,4.5,-1,0,0,6,5.5,1,1565.463497,22.68907563
0,1,5,5.5,-1,0,0,4,4.5,1,1709.849772,27.31092437
0,1,5,5.5,-1,0,0,6,5.5,1,1565.463497,0.290885585
0,1,5,5.5,-1,0,0,6,6.5,1,1565.463497,22.39819005
0,1,6,5.5,1,0,0,5,4.5,-1,1719.142907,26.53783342
```

```
0,1,6,5.5,1,0,0,5,5.5,-1,1719.142907,0.40827436
0,1,6,5.5,1,0,0,7,6.5,-1,1548.264841,23.05389222
0,1,6,6.5,1,0,0,5,5.5,-1,1719.142907,26.94610778
0,1,6,6.5,1,0,0,7,6.5,-1,1548.264841,0.219560878
0,1,6,6.5,1,0,0,7,7.5,-1,1548.264841,22.83433134
0,1,7,6.5,-1,0,0,6,5.5,1,1727.791538,26.3884098
0,1,7,6.5,-1,0,0,6,6.5,1,1727.791538,0.293204553
0,1,7,6.5,-1,0,0,8,7.5,1,1530.301811,23.31838565
0,1,7,7.5,-1,0,0,6,6.5,1,1727.791538,26.68161435
0,1,7,7.5,-1,0,0,8,7.5,1,1530.301811,0.171458718
0,1,7,7.5,-1,0,0,8,8.5,1,1530.301811,23.14692693
0,1,8,7.5,1,0,0,7,6.5,-1,1735.783819,26.2601626
0,1,8,7.5,1,0,0,7,7.5,-1,1735.783819,0.220673635
0,1,8,7.5,1,0,0,9,8.5,-1,1511.544119,23.51916376
0,1,8,8.5,1,0,0,7,7.5,-1,1735.783819,26.48083624
0,1,8,8.5,1,0,0,9,8.5,-1,1511.544119,0.137538969
0,1,8,8.5,1,0,0,9,9.5,-1,1511.544119,23.38162479
0,1,9,8.5,-1,0,0,8,7.5,1,1743.104894,26.15107324
0,1,9,8.5,-1,0,0,8,8.5,1,1743.104894,0.172046534
0,1,9,8.5,-1,0,0,10,9.5,1,1491.958467,23.67688022
0,1,9,9.5,-1,0,0,8,8.5,1,1743.104894,26.32311978
0,1,9,9.5,-1,0,0,10,9.5,1,1491.958467,0.112747049
0,1,9,9.5,-1,0,0,10,10.5,1,1491.958467,23.56413317
0,1,10,9.5,1,0,0,9,8.5,-1,1749.736897,26.05802662
0,1,10,9.5,1,0,0,9,9.5,-1,1749.736897,0.137873157
0,1,10,9.5,1,0,0,11,10.5,-1,1471.508548,23.80410023
0,1,10,10.5,1,0,0,9,9.5,-1,1749.736897,26.19589977
0,1,10,10.5,1,0,0,11,10.5,-1,1471.508548,0.094087353
0,1,10,10.5,1,0,0,11,11.5,-1,1471.508548,23.71001288
```

D.2.3. Thermal Rotational State Distribution

The code below outputs the exptected Maxwell-Boltzmann distribution population fractions of AlH^+ at a specified temperature.

```
%This script calculates the Thermal distribution at a given temperature T.
%Resets and format for long number strings
clear all
close all
format long g

%Temperature Guess in K and number of rotational states to consider.
T = 293;
%numstates = 20;
numstates = 100;

%Molecule Properties and constants.
hc = 1.23984193E-4; %eV*cm
%B = 6.7636; %cm^-1 from Muller and Ottinger (Nguyen) paper
B = 6.5632; %cm^-1 from Szajna paper
k = 8.61733E-5; %ev/K
kT = k*T;

%Calculating2
ntexp = ntAl + ntAlH;
npexp = npAl + npAlH;
nppexp = nppAl + nppAlH;
ptexp = ntAl ./ ntexp;
ppexp = npAl ./ npexp;
pppexp = nppAl ./ nppexp;

%Set up Partition Function with initialization of Z.
Z = 0;
numstates = numstates + 1;
for i = 1:numstates
       N(i) = i-1;
       g(i) = 2*(2*N(i)+1);
       E(i) = hc*B*N(i)*(N(i)+1);
       Z = Z + g(i)*exp(-E(i)/kT);
endfor
```

```
%Map out thermal distribution probabilities.
for i = 1:numstates
       pt(i) = 1/Z*g(i)*exp(-E(i)/kT);
endfor

%Generate fit curve for plotting.
for i = 1:1001
       fit1(i,1) = (i-1)*(numstates)/1000;
       fit1(i,2) = 1/Z*2*(2*fit1(i,1)+1)*exp(-hc*B*fit1(i,1)*(fit1(i,1)+1)/kT)
           ;;
endfor

%Generate matrix for experimental data
for i = 1:length(ptexp)
       fit2(i,1) = i-1;
       fit2(i,2) = ptexp(i);
endfor

%Plot histogram and curve fit.
figure(1)
hold on
bar(N,pt)
plot(fit1(:,1),fit1(:,2),’b’,’linewidth’,2)
plot(fit2(:,1),fit2(:,2),’r.’,’markersize’,10)
legend("Thermal distribution","Maxwell-Boltzmann fit","Experimental results")
xlabel("Rotational State, N")
ylabel("Population Fraction")
title(["AlH Rotational Thermal Distribution at ",num2str(T),"K"]);
hold off

display(" ")
display(["First column is N and the second column is the calculated thermal
   probability at ",num2str(T)," K"])
[N’ pt’]

sm = sum(pt)

s = 0;
for i = 11:length(pt)
       s = s + pt(i);
```

```
endfor
s
```

D.2.4. Time-of-flight Mass Spectrometry Analysis

The code below inputs a binary file from the Agilent DSO7014A digital oscilloscope with the MCP connected to Channel 2. It then asks for user input to pick t_0 before subtracting the DC background, applying a discriminator level, and integrating signal windows for total charge.

```
%Resets and format for long number strings
clear all
close all
%format long g

%Filenames to import and analyze.
file1 = input("Type file name to run:","s");
filename = strcat(file1,".bin");

%Resistance Load on CEM.
R = 50;
%Time widths to integrate over in us.
%Noise refers to time before/after signal to integrate for a background.
AlNoiseLow = 24;
AlHNoiseHigh = 38;
AlWidth = 2;
AlHWidth = 2;
%Time to split Al and AlH+ in us. Set to zero to put in after plotting.
Split = 0;
BaStart = 60;
%Discriminator Level Above Background in mV. Set to zero to put in after
   plotting.
Disc = -10.0;


%_____________________________________________________________________
%_________________________End User Input______________________________
%_____________________________________________________________________
%Import .bin file data from Scope with time in us and voltage in mV using
   Agilent's
```

```
%modified importer (fixed short-circuit errors by "&", "|" to "&&" and "||" on
   lines
%29 and 72.
[t,mcp] = importAgilentBin(filename,2);
CEM(:,1) = t*1E6;
CEM(:,2) = mcp*1E3;
%Plot data for visual checks & determine Al/AlH split Ba integration start.
figure(1)
plot(CEM(:,1),CEM(:,2))
xlabel("Time (us)")
ylabel("CEM Signal (mV)")
title({"CEM Signal vs. Time",file1});
if (Split == 0)
       display(" ")
       Split = input("Pick the time in us to separate Al+ and AlH+ (usually 31
          us): ");
endif
if (BaStart == 0)
       display(" ")
       BaStart = input("Pick the time in us to start integrating Ba+ (usually
          60 us): ");
endif
if (Disc == 0)
       display(" ")
       Disc = input("Pick discriminator level for signals (usually -1.5 mV): ")
          ;
endif
%Define numerical integration ranges for Al, AlH, and Ba in us.
AlLow = Split - AlWidth;
AlHigh = Split;
AlHLow = Split;
AlHHigh = Split + AlHWidth;
BaLow = AlHNoiseHigh;
BaHigh = 80;
%AlNoiseLow = 24;
AlNoiseHigh = AlLow;
AlHNoiseLow = AlHHigh;
%AlHNoiseHigh = 38;
%Pick out portions of the data for a matrix of only Al, AlH, and Ba.
j1 = 0;
j2 = 0;
```

```
j3 = 0;
j4 = 0;
j5 = 0;
for i = 1:rows(CEM)
       if ([CEM(i,1) >= AlNoiseLow, CEM(i,1) <=AlNoiseHigh])
                     j1 = j1 + 1;
                     AlNoisedata(j1,1) = CEM(i,1);
                     AlNoisedata(j1,2) = CEM(i,2);
              elseif(CEM(1,1) >= AlNoiseHigh)
                     AlNoisedata = zeros(1,2);
       endif
       if ([CEM(i,1) >= AlLow, CEM(i,1) <=AlHigh])
              j2 = j2 + 1;
              Aldata(j2,1) = CEM(i,1);
              Aldata(j2,2) = CEM(i,2);
       endif
       if ([CEM(i,1) >= AlHLow, CEM(i,1) <=AlHHigh])
              j3 = j3 + 1;
              AlHdata(j3,1) = CEM(i,1);
              AlHdata(j3,2) = CEM(i,2);
       endif
       if ([CEM(i,1) >= AlHNoiseLow, CEM(i,1) <=AlHNoiseHigh])
                     j4 = j4 + 1;
                     AlHNoisedata(j4,1) = CEM(i,1);
                     AlHNoisedata(j4,2) = CEM(i,2);
              elseif(CEM(rows(CEM),1) <= AlHNoiseLow)
                     AlHNoisedata = zeros(1,2);
       endif
       if ([CEM(i,1) >= BaLow, CEM(i,1) <=BaHigh])
                     j5 = j5 + 1;
                     Badata(j5,1) = CEM(i,1);
                     Badata(j5,2) = CEM(i,2);
              elseif(CEM(rows(CEM),1) <= BaLow)
                     Badata = zeros(1,2);
       endif
endfor
%Calculate average background voltage before/after Al/AlH.
mVAlNoise = mean(AlNoisedata(:,2));
mVAlHNoise = mean(AlHNoisedata(:,2));
mVNoise = mean([mVAlNoise, mVAlHNoise]);
if (mVAlNoise == 0)
```

```
       mVNoise = mVAlHNoise;
endif
if (mVAlHNoise == 0)
       mVNoise = 0;
endif
%Zero any data that's not greater in magnitude than discriminator level.
%Subtract off background!
DiscLevel = mVNoise + Disc;
for k1 = 1:rows(AlNoisedata)
       if (AlNoisedata(k1,2) <= DiscLevel)
              AlNoisedata(k1,3) = AlNoisedata(k1,2) - mVNoise;
              else
              AlNoisedata(k1,3) = 0;
       endif
endfor
for k2 = 1:rows(Aldata)
       if (Aldata(k2,2) <= DiscLevel)
              Aldata(k2,3) = Aldata(k2,2) - mVNoise;
              else
              Aldata(k2,3) = 0;
       endif
endfor
for k3 = 1:rows(AlHdata)
       if (AlHdata(k3,2) <= DiscLevel)
              AlHdata(k3,3) = AlHdata(k3,2) - mVNoise;
              else
              AlHdata(k3,3) = 0;
       endif
endfor
for k4 = 1:rows(AlHNoisedata)
       if (AlHNoisedata(k4,2) <= DiscLevel)
              AlHNoisedata(k4,3) = AlHNoisedata(k4,2) - mVNoise;
              else
              AlHNoisedata(k4,3) = 0;
       endif
endfor
for k5 = 1:rows(Badata)
       if (Badata(k5,2) <= DiscLevel)
              Badata(k5,3) = Badata(k5,2) - mVNoise;
              else
              Badata(k5,3) = 0;
       endif
endfor
%Combine all processed signal matrices into one for later use.
FullSignal = zeros(2,3);
FullSignal = [AlNoisedata; Aldata; AlHdata; AlHNoisedata; Badata];
%Plot Results of counting and background subtraction:
figure(2)
plot(AlNoisedata(:,1),AlNoisedata(:,3))
xlabel("Time (us)")
ylabel("Counting Signal(mV)")
title({"Noise Before Al with Backgrounds Removed vs. Time",file1});
figure(3)
plot(Aldata(:,1),Aldata(:,3))
xlabel("Time (us)")
ylabel("Counting Signal(mV)")
title({"Al+ Signal with Backgrounds Removed vs. Time",file1});
figure(4)
plot(AlHdata(:,1),AlHdata(:,3))
xlabel("Time (us)")
ylabel("Counting Signal(mV)")
title({"AlH+ Signal with Backgrounds Removed vs. Time",file1});
figure(5)
plot(AlHNoisedata(:,1),AlHNoisedata(:,3))
xlabel("Time (us)")
ylabel("Counting Signal(mV)")
title({"Noise After AlH+ with Backgrounds Removed vs. Time",file1});
figure(6)
plot(Badata(:,1),Badata(:,3))
xlabel("Time (us)")
ylabel("Counting Signal(mV)")
title({"Ba+ Signal with Backgrounds Removed vs. Time",file1});
%Plot combined signal output for visual check.
figure(7)
plot(FullSignal(:,1),FullSignal(:,3))
xlabel("Time (us)")
ylabel("Counting Signal(mV)")
title({"Full Signal with Backgrounds Removed vs. Time",file1});
%Define the third column as current (mA).
AlNoisedata(:,4) = AlNoisedata(:,3)/R;
Aldata(:,4) = Aldata(:,3)/R;
AlHdata(:,4) = AlHdata(:,3)/R;
```

```
AlHNoisedata(:,4) = AlHNoisedata(:,3)/R;
Badata(:,4) = Badata(:,3)/R;
%Numerically integrate to obtain charge in units of mA*us (nC)
CAlNoise = -trapz(AlNoisedata(:,1),AlNoisedata(:,4));
CAl = -trapz(Aldata(:,1),Aldata(:,4));
CAlH = -trapz(AlHdata(:,1),AlHdata(:,4));
CAlHNoise = -trapz(AlHNoisedata(:,1),AlHNoisedata(:,4));
CBa  = -trapz(Badata(:,1),Badata(:,4));
%Change to pC.
CAlNoisedisp = CAlNoise*1E3;
CAldisp = CAl*1E3;
CAlHdisp = CAlH*1E3;
CAlHNoisedisp = CAlHNoise*1E3;
CBadisp = CBa*1E3;
%Display results!
display(" ")
display(["Background voltage removed is ",num2str(mVNoise,6)," mV."])
display(["Total charge before Al+ window is ",num2str(CAlNoisedisp,6)," pC."])
display(["Total charge within Al+ window is ",num2str(CAldisp,6)," pC."])
display(["Total charge within AlH+ window is ",num2str(CAlHdisp,6)," pC."])
display(["Total charge after AlH+ window is ",num2str(CAlHNoisedisp,6)," pC."])
display(["Total charge within Ba+ window is ",num2str(CBadisp,6)," pC."])

%Output data to import for plotting. This is the noise and Al, AlH signal
   windows raw data and processed.
%output = [AlNoisedata; Aldata; AlHdata; AlHNoisedata];
%dlmwrite(strcat(file1,"_Processed",".txt"), output, ",");
```

D.2.4.1. Agilent Binary File Extractor. The code below unpacks the Agilent binary file from the Agilent DSO7014A digital oscilloscope. The original code is sourced from Agilent, but modified to fix short-circuit errors on lines 29 and 72 by replacing “&” and “|” to “&&” and “||”. This file must me named “importAgilentBin.m”.

```
function [timeVector, voltageVector] = importAgilentBin(inputFilename, varargin
    )
% ImportAgilentBin reads the Agilent Binary Waveform filetype.
% [timeVector, voltageVector] = importAgilentBin(inputFilename)
% [timeVector, voltageVector] = importAgilentBin(inputFilename, waveform_index)
% if waveformIndex is not provided, the first waveform will be read
% voltageVector may contain two columns [MIN, MAX]

if (~exist(inputFilename))
    error(’inputFilename missing.’);
end

fileId = fopen(inputFilename, ’r’);

% read file header
fileCookie = fread(fileId, 2, ’char’);
fileVersion = fread(fileId, 2, ’char’);
fileSize = fread(fileId, 1, ’int32’);
nWaveforms = fread(fileId, 1, ’int32’);

% verify cookie
fileCookie = char(fileCookie’);
if (~strcmp(fileCookie, ’AG’))
    fclose(fileId);
    error(’Unrecognized file format.’);
end

% determine which waveform to read
waveformSelect = 1;
if (size(varargin) == 1 && varargin{1} <= nWaveforms)
    waveformSelect = varargin{1};
end
```

```
for waveformIndex = 1:nWaveforms
   % read waveform header
   headerSize = fread(fileId, 1, 'int32'); bytesLeft = headerSize - 4;
   waveformType = fread(fileId, 1, 'int32'); bytesLeft = bytesLeft - 4;
   nWaveformBuffers = fread(fileId, 1, 'int32'); bytesLeft = bytesLeft - 4;
   nPoints = fread(fileId, 1, 'int32'); bytesLeft = bytesLeft - 4;
   count = fread(fileId, 1, 'int32'); bytesLeft = bytesLeft - 4;
   xDisplayRange = fread(fileId, 1, 'float32'); bytesLeft = bytesLeft - 4;
   xDisplayOrigin = fread(fileId, 1, 'double'); bytesLeft = bytesLeft - 8;
   xIncrement = fread(fileId, 1, 'double'); bytesLeft = bytesLeft - 8;
   xOrigin = fread(fileId, 1, 'double'); bytesLeft = bytesLeft - 8;
   xUnits = fread(fileId, 1, 'int32'); bytesLeft = bytesLeft - 4;
   yUnits = fread(fileId, 1, 'int32'); bytesLeft = bytesLeft - 4;
   dateString = fread(fileId, 16, 'char'); bytesLeft = bytesLeft - 16;
   timeString = fread(fileId, 16, 'char'); bytesLeft = bytesLeft - 16;
   frameString = fread(fileId, 24, 'char'); bytesLeft = bytesLeft - 24;
   waveformString = fread(fileId, 16, 'char'); bytesLeft = bytesLeft - 16;
   timeTag = fread(fileId, 1, 'double'); bytesLeft = bytesLeft - 8;
   segmentIndex = fread(fileId, 1, 'uint32'); bytesLeft = bytesLeft - 4;

   % skip over any remaining data in the header
   fseek(fileId, bytesLeft, 'cof');

   % generate time vector from xIncrement and xOrigin values
   if (waveformIndex == waveformSelect)
      timeVector = (xIncrement * [0:(nPoints-1)]') + xOrigin;
   end

   for bufferIndex = 1:nWaveformBuffers
      % read waveform buffer header
      headerSize = fread(fileId, 1, 'int32'); bytesLeft = headerSize - 4;
      bufferType = fread(fileId, 1, 'int16'); bytesLeft = bytesLeft - 2;
      bytesPerPoint = fread(fileId, 1, 'int16'); bytesLeft = bytesLeft - 2;
      bufferSize = fread(fileId, 1, 'int32'); bytesLeft = bytesLeft - 4;

      % skip over any remaining data in the header
      fseek(fileId, bytesLeft, 'cof');

      if (waveformIndex == waveformSelect)
         if ((bufferType == 1) || (bufferType == 2) || (bufferType == 3))
```

```
                % bufferType is PB_DATA_NORMAL, PB_DATA_MIN, or PB_DATA_MAX (
                    float)
                voltageVector(:, bufferIndex) = fread(fileId, nPoints, 'float');
            elseif (bufferType == 4)
                % bufferType is PB_DATA_COUNTS (int32)
                voltageVector(:, bufferIndex) = fread(fileId, nPoints, '*int32')
                    ;
            elseif (bufferType == 5)
                % bufferType is PB_DATA_LOGIC (int8)
                voltageVector(:, bufferIndex) = fread(fileId, nPoints, '*uint8')
                    ;
            else
                % unrecognized bufferType read as unformated bytes
                voltageVector(:, bufferIndex) = fread(fileId, bufferSize, '*
                    uint8');
            end
        else
            fseek(fileId, bufferSize, 'cof');
        end
    end
end
fclose(fileId);
```

D.2.4.2. Agilent Binary File to CSV. The code below unpacks the Agilent binary file from the Agilent DSO7014A digital oscilloscope and saves it to a .CSV file.

```
%Resets and format for long number strings
clear all
close all
%format long g

%Filenames to import and convert.
file1 = input("Type .bin file name to run:","s");
filename = strcat(file1,".bin");

%Scope Channel to Convert.
channel = 1;
%____________________________________________________________________
%_________________________End User Input_____________________________
%____________________________________________________________________
%Import .bin file data from Scope with time in s and voltage in V using Agilent
    's
%modified importer (fixed short-circuit errors by "&", "|" to "&&" and "||" on
    lines
%29 and 72.
[t,V] = importAgilentBin(filename,channel);
data = [t,V];
%If Subset is on (= 1), only the subset of the data is saved. Else the entire
    dataset is saved.
subset = 0;
if subset == 1
        for j = 1:10000
                data1(j,1) = t(j);
                data1(j,2) = V(j);
        endfor
        %Output data to import for plotting.
        dlmwrite(strcat(file1,"_Processed_1",".txt"), data1, ",");
        else
                %Output data to import for plotting.
                dlmwrite(strcat(file1,"_Processed",".txt"), data, ",");
                %csvwrite(strcat(file1,"_Processed",".csv"), data);
endif
```

D.2.4.3. Time-of-flight Mass Spectrometry Peak Analysis. The code below runs after the Time-of-flight Mass Spectrometry Analysis to numerically integrate the signal around individual peaks for histogram analysis.

```
%Run this to numerically integrate a subset of the data after running
   AlHCountingv3.
%Format for long number strings and zero data matrices
format long g
Signal = zeros(2,3);
%Upper and lower numerical integration windows
Lower = 32.7
Upper = 32.9


%___________________________________________________________________
%_________________________End User Input____________________________
%___________________________________________________________________
%Select out the region of interest.
j6 = 0;
for i = 1:rows(FullSignal)
       if ([FullSignal(i,1) >= Lower, FullSignal(i,1) <=Upper])
                      j6 = j6 + 1;
                      Signal(j6,1) = FullSignal(i,1);
                      Signal(j6,2) = FullSignal(i,3);
       endif
endfor
%Plot window of interest signal output for visual check.
figure(8)
plot(Signal(:,1),Signal(:,2),’.-’)
xlabel("Time (us)")
ylabel("Counting Signal(mV)")
title({"Window of Interest Signal with Backgrounds Removed vs. Time",file1});
%Define the third column as current (mA).
Signal(:,3) = Signal(:,2)/R;
%Numerically integrate to obtain charge in units of mA*us (nC)
CSignal = -trapz(Signal(:,1),Signal(:,3));
%Change to pC.
CSignaldisp = CSignal*1E3;
%Display results!
```

```
display(" ")
display(["Total charge within window of interest ",num2str(CSignaldisp,6)," pC
    ."])
```

APPENDIX E

Python Code for Motional Ground State Cooling Project

E.1. iPython Notebooks

The code below is for the iPython notebooks used for Part 3. Note that an empty line corresponds to a new cell. Notebooks that simply generate plots from data are omitted.

E.1.1. Doppler Fluorescence Analysis

The code below performs data analysis of the interleaved and randomized Doppler curve data.

```
%pylab inline
import scipy.optimize as sc
import scipy.special as scs

data0 = load('20151215_DopplerScanPMT_9.npy')
data1 = load('20160111_DopplerScanPMT_0.npy')
data2 = load('20160111_DopplerScanPMT_1.npy')
data3 = load('20160111_DopplerScanPMT_2.npy')

expose_t0 = 100.0e-3
expose_t1 = 100.0e-3
expose_t2 = 100.0e-3
expose_t3 = 100.0e-3

data0 = data0[data0[:,0].argsort()]
data1 = data1[data1[:,0].argsort()]
data2 = data2[data2[:,0].argsort()]
data3 = data3[data3[:,0].argsort()]
```

```
pruned0 = data0[:,1:]/expose_t0
pruned1 = data1[:,1:]/expose_t1
pruned2 = data2[:,1:]/expose_t2
pruned3 = data3[:,1:]/expose_t3
hist(pruned0.ravel(),30)
xlabel(r'PMT Count Rate (s$^{-1}$)')
plt.grid(True)

#Add a zero row to the bottom of each data matrix to fix a boundary
#issue in a for loop in data analysis.
data0 = np.vstack((data0,np.zeros((data0.shape[1]))))
data1 = np.vstack((data1,np.zeros((data1.shape[1]))))
data2 = np.vstack((data2,np.zeros((data2.shape[1]))))
data3 = np.vstack((data3,np.zeros((data3.shape[1]))))

#DO WORK. This section extracts frequencies and runs the loops to calculate
   mean
#fluorescence and number of experiments with sigma given by standard dev of the
    mean.
#It uses a clever indexing method for interleaved datapoints.
v0 = np.zeros((np.unique(data0[:,0]).shape[0]))
p0 = np.zeros((np.unique(data0[:,0]).shape[0]))
err0 = np.zeros((np.unique(data0[:,0]).shape[0]))
n0 = np.zeros((np.unique(data0[:,0]).shape[0]))
j = 0
for i in range(data0.shape[0]-1):
    index = (np.argwhere(data0[:,0]==data0[i,0]).transpose()[0])
    v0[j] = data0[i,0]
    p0[j] = np.average(data0[index,1:])/expose_t0
    err0[j] = np.std(data0[index,1:])/expose_t0/sqrt(data0[index,1:].size)
    n0[j] = data0[index,1:].size
    if data0[i,0] != data0[i+1,0]:
        j = j + 1
v1 = np.zeros((np.unique(data1[:,0]).shape[0]))
p1 = np.zeros((np.unique(data1[:,0]).shape[0]))
err1 = np.zeros((np.unique(data1[:,0]).shape[0]))
n1 = np.zeros((np.unique(data1[:,0]).shape[0]))
j = 0
for i in range(data1.shape[0]-1):
    index = (np.argwhere(data1[:,0]==data1[i,0]).transpose()[0])
    v1[j] = data1[i,0]
    p1[j] = np.average(data1[index,1:])/expose_t1
    err1[j] = np.std(data1[index,1:])/expose_t1/sqrt(data1[index,1:].size)
    n1[j] = data1[index,1:].size
    if data1[i,0] != data1[i+1,0]:
        j = j + 1
v2 = np.zeros((np.unique(data2[:,0]).shape[0]))
p2 = np.zeros((np.unique(data2[:,0]).shape[0]))
err2 = np.zeros((np.unique(data2[:,0]).shape[0]))
n2 = np.zeros((np.unique(data2[:,0]).shape[0]))
j = 0
for i in range(data2.shape[0]-1):
    index = (np.argwhere(data2[:,0]==data2[i,0]).transpose()[0])
    v2[j] = data2[i,0]
    p2[j] = np.average(data2[index,1:])/expose_t2
    err2[j] = np.std(data2[index,1:])/expose_t2/sqrt(data2[index,1:].size)
    n2[j] = data2[index,1:].size
    if data2[i,0] != data2[i+1,0]:
        j = j + 1
v3 = np.zeros((np.unique(data3[:,0]).shape[0]))
p3 = np.zeros((np.unique(data3[:,0]).shape[0]))
err3 = np.zeros((np.unique(data3[:,0]).shape[0]))
n3 = np.zeros((np.unique(data3[:,0]).shape[0]))
j = 0
for i in range(data3.shape[0]-1):
    index = (np.argwhere(data3[:,0]==data3[i,0]).transpose()[0])
    v3[j] = data3[i,0]
    p3[j] = np.average(data3[index,1:])/expose_t3
    err3[j] = np.std(data3[index,1:])/expose_t3/sqrt(data3[index,1:].size)
    n3[j] = data3[index,1:].size
    if data3[i,0] != data3[i+1,0]:
        j = j + 1

#Fix the boundary condition.
v0 = np.delete(v0,v0.shape[0]-1)
p0 = np.delete(p0,p0.shape[0]-1)
err0 = np.delete(err0,err0.shape[0]-1)
n0 = np.delete(n0,n0.shape[0]-1)
v1 = np.delete(v1,v1.shape[0]-1)
p1 = np.delete(p1,p1.shape[0]-1)
err1 = np.delete(err1,err1.shape[0]-1)
n1 = np.delete(n1,n1.shape[0]-1)
```

```
v2 = np.delete(v2,v2.shape[0]-1)
p2 = np.delete(p2,p2.shape[0]-1)
err2 = np.delete(err2,err2.shape[0]-1)
n2 = np.delete(n2,n2.shape[0]-1)
v3 = np.delete(v3,v3.shape[0]-1)
p3 = np.delete(p3,p3.shape[0]-1)
err3 = np.delete(err3,err3.shape[0]-1)
n3 = np.delete(n3,n3.shape[0]-1)

#Provision to offset data runs to deal with any background changes.
bg = False
normalizedata = False
if bg == True:
    bg0 = min(p0)
    bg1 = min(p1)
    bg2 = min(p2)
    bg3 = min(p3)
    p0 = p0 - bg0
    p1 = p1 - bg1
    p2 = p2 - bg2
    p3 = p3 - bg3
if normalizedata == True:
    err0 = err0/(max(p0))
    p0 = (p0)/(max(p0))
    err1 = err1/(max(p1))
    p1 = (p1)/(max(p1))
    err2 = err2/(max(p2))
    p2 = (p2)/(max(p2))
    err3 = err3/(max(p3))
    p3 = (p3)/(max(p3))

#Scale all results
#scale = 1.0E5
scale = 0
if scale != 0:
    p0 = p0/scale
    err0 = err0/scale
    p1 = p1/scale
    err1 = err1/scale
    p2 = p2/scale
    err2 = err2/scale
    p3 = p3/scale
    err3 = err3/scale

ZeroOffset = False
if ZeroOffset == True:
    p0 = p0 - mean(p0)
    p1 = p1 - mean(p1)
    p2 = p2 - mean(p2)
    p3 = p3 - mean(p3)

BadPoints = False
if BadPoints == True:
    for i in range(p0.shape[0]):
        if p0[i] > 500:
            p0[i] = p0[i]
        else:
            p0[i] = NaN
    for i in range(p1.shape[0]):
        if p1[i] > 500:
            p1[i] = p1[i]
        else:
            p1[i] = NaN
    for i in range(p2.shape[0]):
        if p2[i] > 500:
            p2[i] = p2[i]
        else:
            p2[i] = NaN
    for i in range(p3.shape[0]):
        if p3[i] > 500:
            p3[i] = p3[i]
        else:
            p3[i] = NaN

#Find the applied voltage to AOM drive frequency. Done 20151215.
#aomdata = np.array([[-0.50,59.89],[-0.25,70.11],[-0.10,76.09],[0.00,80.11],
#                [0.10,84.13],[0.25,90.22],[0.50,100.32]])
#aomfit = polyfit(aomdata[:,0],aomdata[:,1], 1)
#aomfit_fn = poly1d(aomfit)
#plt.scatter(aomdata[:,0],aomdata[:,1],color='blue')
#aomx = linspace(-1,1,10)
#plt.plot(aomx,aomfit_fn(aomx), 'k')
```

```
#plt.title('20151215 493 nm 2x Doppler AOM Driver FM Calibration\nSlope = '+
    array_str(aomfit_fn[1])+r' MHz/V, Intercept = '+array_str(aomfit_fn[0])+'
    MHz')
#plt.xlabel('AOM Driver FM Input Voltage (V)')
#plt.ylabel('Output Frequency (MHz))')
#axis([-1.0,1.0,55,105])
#plt.grid(True)

#Define AOM fit function from the results in the figure above.
#Note that the (-1) order switching AOM is exactly 80 MHz.
def testfunc(x):
    return 2*(x*40.38+80.12)-80.00

plt.errorbar(testfunc(v0),p0,yerr=err0,color='blue',fmt='-o',elinewidth=3,
    capthick=3,linewidth=3,ms=12)
plt.errorbar(testfunc(v1),p1,yerr=err1,color='green',fmt='-o',elinewidth=3,
    capthick=3,linewidth=3,ms=12)
plt.errorbar(testfunc(v2),p2,yerr=err2,color='red',fmt='-o',elinewidth=3,
    capthick=3,linewidth=3,ms=12)
plt.errorbar(testfunc(v3),p3,yerr=err3,color='purple',fmt='-o',elinewidth=3,
    capthick=3,linewidth=3,ms=12)
plt.xlabel('Net 493 nm Doppler AOM Frequency (MHz)',fontsize='xx-large')
plt.ylabel('Normalized Counts',fontsize='xx-large')
#plt.ylabel(r'Average PMT Count Rate (s$^{-1}$)',fontsize='xx-large')
#plt.title('Doppler Fluorescence vs 650 nm Power, ' + r'5.0 $\mu$W 493 nm',
    fontsize='xx-large')
#plt.title(r'Doppler Fluorescence, 5 $\mu$W 493 nm, 11 $\mu$W 650 nm',fontsize
    ='xx-large')
plt.legend((r'Sacher 30 $\mu$W',r'30 $\mu$W',r'13 $\mu$W',r'7.0 $\mu$W'),loc='
    upper left',fontsize='xx-large')
axis([45,115,-0.05,1.1])
#axis([45,115,1000,30000])
#plt.grid(True)
figsize(11,8.5)
xticks(size = 'xx-large')
yticks(size = 'xx-large')
tick_params('both', length=12, width=2, which='major')
minorticks_on()
tick_params('both', length=5, width=2, which='minor')
#savefig('650DopplerLaserNorm.pdf', dpi=None, facecolor='w', edgecolor='w',
    format='pdf', pad_inches=0.1, frameon=None)
```

```
plt.errorbar(v0,p0,yerr=err0,color='blue',fmt='-o')
plt.errorbar(v1,p1,yerr=err1,color='green',fmt='-o')
#plt.errorbar(v2,p2,yerr=err2,color='red',fmt='-o')
#plt.errorbar(v3,p3,yerr=err3,color='purple',fmt='-o')
plt.xlabel('FM Voltage (V)')
#plt.ylabel('Normalized Counts')
plt.ylabel(r'Average PMT Count Rate (s$^{-1}$)')
plt.title(r'Doppler Fluorescence, 5 $\mu$W 493 nm, 11 $\mu$W 650 nm')
#plt.legend((r'Old 30 $\mu$W',r'30 $\mu$W',r'13 $\mu$W',r'7.0 $\mu$W'),loc='
   upper left')
#axis([-.4,.3,-0.05,1.05])
axis([-.2,.4,3000,23000])
plt.grid(True)

try:
   del temp
except NameError:
   print
temp = np.zeros([v2.shape[0],3])
temp[:,0] = v2
temp[:,1] = p2
temp[:,2] = err2
#temp

Gamma1 = 2*pi*(1.0/3)*15.2
Gamma2 = 2*pi*(2.0/3)*15.2
GammaT = Gamma1+Gamma2
#Delta1 = 2*pi*(-80.0)
Delta2 = 2*pi*(80-10-2*(80.0-11.0/2)) #accounts for Doppler switching AOM and
   -10 MHz detuning of Doppler light from resonance.
def fitfn(x,Omega1,Omega2,dark_l,c_A,omegaz,bsb_A,rsb_A,D):
   Delta1 = Delta2 + x
   R = Omega2**2/(4*Delta1**2)*GammaT
   OmegaE = Omega1*Omega2/(2*Delta1)
   DeltaP = Omega2**2/(4*Delta1)
   return \
   OmegaE**2*((x-dark_l)/DeltaP)**2*R/(4*Gamma1)/(((x-dark_l)-DeltaP)**2+R
      **2/4+OmegaE**2*GammaT/(2*Gamma1))*GammaT*c_A+\
   OmegaE**2*((x-dark_l-omegaz)/DeltaP)**2*R/(4*Gamma1)/(((x-dark_l-omegaz)-
      DeltaP)**2+R**2/4+OmegaE**2*GammaT/(2*Gamma1))*GammaT*bsb_A+\
    OmegaE**2*((x-dark_l+omegaz)/DeltaP)**2*R/(4*Gamma1)/(((x-dark_l+omegaz)-
       DeltaP)**2+R**2/4+OmegaE**2*GammaT/(2*Gamma1))*GammaT*rsb_A+\
    D

try:
    del (popt0,pcov0,perr0)
except NameError:
    print
#Perform curve fit.
popt0, pcov0 = sc.curve_fit(fitfn, 2*pi*f0, p0,\
               #For background-subtracted data!
               #p0
                  =(800,0.5,10,-11.7,-10.6,1.14,70,53,39,0.13,73,1.0,-2.2,-13.1)
                  ,\
               #For regular data.
                p0=(2*pi*0.6,2*pi*16.0,-2*pi*11.0,1.6E5,1.14*2*pi,7E5,9E5,0.0)
                   ,\
               sigma=err0,absolute_sigma=’true’,maxfev=100000)
perr0 = np.sqrt(np.diag(pcov0))
tempfit = np.empty((popt0.shape[0],3),dtype=’|S8’)
tempfit[:,0] = [’Omega1’,’Omega2’,’dark_l’,’c_A’,’omegaz’,’bsb_A’,’rsb_A’,’D’]
tempfit[:,1] = (popt0/(2*pi)).astype(’|S8’)
tempfit[:,2] = (perr0/(2*pi)).astype(’|S8’)
tempfit

try:
    del (popt1,pcov1,perr1)
except NameError:
    print
#Perform curve fit.
popt1, pcov1 = sc.curve_fit(fitfn, 2*pi*f1, p1,\
               #For background-subtracted data!
               #p0
                  =(800,0.5,10,-11.7,-10.6,1.14,70,53,39,0.13,73,1.0,-2.2,-13.1)
                  ,\
               #For regular data.
                p0=(2*pi*0.6,2*pi*16.0,-2*pi*11.0,1.6E5,1.14*2*pi,7E5,9E5,0.0)
                   ,\
               sigma=err1,absolute_sigma=’true’,maxfev=100000)
perr1 = np.sqrt(np.diag(pcov1))
tempfit = np.empty((popt1.shape[0],3),dtype=’|S8’)
```

```
tempfit[:,0] = ['Omega1','Omega2','dark_l','c_A','omegaz','bsb_A','rsb_A','D']
tempfit[:,1] = (popt1/(2*pi)).astype('|S8')
tempfit[:,2] = (perr1/(2*pi)).astype('|S8')
tempfit

try:
   del (popt2,pcov2,perr2)
except NameError:
   print
#Perform curve fit.
popt2, pcov2 = sc.curve_fit(fitfn, 2*pi*f2, p2,\
            #For background-subtracted data!
            #p0
               =(800,0.5,10,-11.7,-10.6,1.14,70,53,39,0.13,73,1.0,-2.2,-13.1)
               ,\
            #For regular data.
             p0=(2*pi*0.6,2*pi*16.0,-2*pi*11.0,1.6E5,1.14*2*pi,7E5,9E5,0.0)
                ,\
            sigma=err2,absolute_sigma='true',maxfev=100000)
perr2 = np.sqrt(np.diag(pcov2))
tempfit = np.empty((popt2.shape[0],3),dtype='|S8')
tempfit[:,0] = ['Omega1','Omega2','dark_l','c_A','omegaz','bsb_A','rsb_A','D']
tempfit[:,1] = (popt2/(2*pi)).astype('|S8')
tempfit[:,2] = (perr2/(2*pi)).astype('|S8')
tempfit

try:
   del (popt3,pcov3,perr3)
except NameError:
   print
#Perform curve fit.
popt3, pcov3 = sc.curve_fit(fitfn, 2*pi*f3, p3,\
            #For background-subtracted data!
            #p0
               =(800,0.5,10,-11.7,-10.6,1.14,70,53,39,0.13,73,1.0,-2.2,-13.1)
               ,\
            #For regular data.
             p0=(2*pi*0.6,2*pi*16.0,-2*pi*11.0,1.6E5,1.14*2*pi,7E5,9E5,0.0)
                ,\
            sigma=err3,absolute_sigma='true',maxfev=100000)
perr3 = np.sqrt(np.diag(pcov3))
```

```
tempfit = np.empty((popt3.shape[0],3),dtype='|S8')
tempfit[:,0] = ['Omega1','Omega2','dark_l','c_A','omegaz','bsb_A','rsb_A','D']
tempfit[:,1] = (popt3/(2*pi)).astype('|S8')
tempfit[:,2] = (perr3/(2*pi)).astype('|S8')
tempfit

try:
   del (nbar0,delta_nbar0,nbar1,delta_nbar1,nbar2,delta_nbar2,nbar3,
       delta_nbar3)
except NameError:
   print
rsb_Amp0 = popt0[6]
delta_rsb_Amp0 = perr0[6]
bsb_Amp0 = popt0[5]
delta_bsb_Amp0 = perr0[5]
r0 = rsb_Amp0/bsb_Amp0
nbar0 = r0/(1-r0)
delta_nbar0 = nbar0*sqrt(((1/(bsb_Amp0-rsb_Amp0))*delta_bsb_Amp0)**2+\
                    ((bsb_Amp0/(rsb_Amp0*(bsb_Amp0-rsb_Amp0)))*delta_rsb_Amp0)
                        **2)
rsb_Amp1 = popt1[6]
delta_rsb_Amp1 = perr1[6]
bsb_Amp1 = popt1[5]
delta_bsb_Amp1 = perr1[5]
r1 = rsb_Amp1/bsb_Amp1
nbar1 = r1/(1-r1)
delta_nbar1 = nbar1*sqrt(((1/(bsb_Amp1-rsb_Amp1))*delta_bsb_Amp1)**2+\
                    ((bsb_Amp1/(rsb_Amp1*(bsb_Amp1-rsb_Amp1)))*delta_rsb_Amp1)
                        **2)
rsb_Amp2 = popt2[6]
delta_rsb_Amp2 = perr2[6]
bsb_Amp2 = popt2[5]
delta_bsb_Amp2 = perr2[5]
r2 = rsb_Amp2/bsb_Amp2
nbar2 = r2/(1-r2)
delta_nbar2 = nbar2*sqrt(((1/(bsb_Amp2-rsb_Amp2))*delta_bsb_Amp2)**2+\
                    ((bsb_Amp2/(rsb_Amp2*(bsb_Amp2-rsb_Amp2)))*delta_rsb_Amp2)
                        **2)
rsb_Amp3 = popt3[6]
delta_rsb_Amp3 = perr3[6]
bsb_Amp3 = popt3[5]
```

```
delta_bsb_Amp3 = perr3[5]
r3 = rsb_Amp3/bsb_Amp3
nbar3 = r3/(1-r3)
delta_nbar3 = nbar3*sqrt(((1/(bsb_Amp3-rsb_Amp3))*delta_bsb_Amp3)**2+\
                      ((bsb_Amp3/(rsb_Amp3*(bsb_Amp3-rsb_Amp3)))*delta_rsb_Amp3)
                         **2)
np.array([['nbar0',np.around(nbar0,decimals=3),np.around(delta_nbar0,decimals
    =3)],\
          ['nbar1',np.around(nbar1,decimals=3),np.around(delta_nbar1,decimals=3)
              ],\
          ['nbar2',np.around(nbar2,decimals=3),np.around(delta_nbar2,decimals=3)
              ],\
          ['nbar3',np.around(nbar3,decimals=3),np.around(delta_nbar3,decimals=3)
              ]])

cl0 = (popt0[2]-(popt0[1])**2/(4*2*pi*80.0))/(2*pi)
dcl0 = sqrt((perr0[2])**2+(2*perr0[1]*popt0[1]/(4*2*pi*80.0))**2)/(2*pi)
rsbl0 = (popt0[2]-(popt0[1])**2/(4*2*pi*80.0)-popt0[4])/(2*pi)
drsbl0 = sqrt((perr0[2])**2+(2*perr0[1]*popt0[1]/(4*2*pi*80.0))**2+(perr0[4])
    **2)/(2*pi)

cl1 = (popt1[2]-(popt1[1])**2/(4*2*pi*80.0))/(2*pi)
dcl1 = sqrt((perr1[2])**2+(2*perr1[1]*popt1[1]/(4*2*pi*80.0))**2)/(2*pi)
rsbl1 = (popt1[2]-(popt1[1])**2/(4*2*pi*80.0)-popt1[4])/(2*pi)
drsbl1 = sqrt((perr1[2])**2+(2*perr1[1]*popt1[1]/(4*2*pi*80.0))**2+(perr1[4])
    **2)/(2*pi)

cl2 = (popt2[2]-(popt2[1])**2/(4*2*pi*80.0))/(2*pi)
dcl2 = sqrt((perr2[2])**2+(2*perr2[1]*popt2[1]/(4*2*pi*80.0))**2)/(2*pi)
rsbl2 = (popt2[2]-(popt2[1])**2/(4*2*pi*80.0)-popt2[4])/(2*pi)
drsbl2 = sqrt((perr2[2])**2+(2*perr2[1]*popt2[1]/(4*2*pi*80.0))**2+(perr2[4])
    **2)/(2*pi)

cl3 = (popt3[2]-(popt3[1])**2/(4*2*pi*80.0))/(2*pi)
dcl3 = sqrt((perr3[2])**2+(2*perr3[1]*popt3[1]/(4*2*pi*80.0))**2)/(2*pi)
rsbl3 = (popt3[2]-(popt3[1])**2/(4*2*pi*80.0)-popt3[4])/(2*pi)
drsbl3 = sqrt((perr3[2])**2+(2*perr3[1]*popt3[1]/(4*2*pi*80.0))**2+(perr3[4])
    **2)/(2*pi)

np.array([[' Run ','Carrier','dCarr',' RSB ','dRSB '],\
         [' blue ',np.around(cl0,decimals=3),np.around(dcl0,decimals=3),np.
            around(rsbl0,decimals=3),np.around(drsbl0,decimals=3)],\
         ['green ',np.around(cl1,decimals=3),np.around(dcl1,decimals=3),np.
            around(rsbl1,decimals=3),np.around(drsbl1,decimals=3)],\
         [' red ',np.around(cl2,decimals=3),np.around(dcl2,decimals=3),np.
            around(rsbl2,decimals=3),np.around(drsbl2,decimals=3)],\
         ['purple',np.around(cl3,decimals=3),np.around(dcl3,decimals=3),np.
            around(rsbl3,decimals=3),np.around(drsbl3,decimals=3)],\
        ])

plt.errorbar(f0,p0,yerr=err0,color='blue',fmt='o')
plt.errorbar(f1,p1,yerr=err1,color='green',fmt='o')
plt.errorbar(f2,p2,yerr=err2,color='red',fmt='o')
plt.errorbar(f3,p3,yerr=err3,color='purple',fmt='o')
x = 2*pi*linspace(-13.5,-9.5,1000)
plot(x/(2*pi),fitfn(x,popt0[0],popt0[1],popt0[2],popt0[3],popt0[4],popt0[5],
   popt0[6],popt0[7]),color='blue',linewidth=2)
plot(x/(2*pi),fitfn(x,popt1[0],popt1[1],popt1[2],popt1[3],popt1[4],popt1[5],
   popt1[6],popt1[7]),color='green',linewidth=2)
plot(x/(2*pi),fitfn(x,popt2[0],popt2[1],popt2[2],popt2[3],popt2[4],popt2[5],
   popt2[6],popt2[7]),color='red',linewidth=2)
plot(x/(2*pi),fitfn(x,popt3[0],popt3[1],popt3[2],popt3[3],popt3[4],popt3[5],
   popt3[6],popt3[7]),color='purple',linewidth=2)
#plot(x,fitfn(x,800,0.5,10,-11.7,-10.6,1.14,70,53,39,0.13,73,-2.2,-13.1),color
   ='black',linewidth=3)
plt.xlabel('(Probe - Pump) Frequency (MHz)')
plt.ylabel(r'Average PMT Count Rate (s$^{-1}$)')
plt.title('20160331 Near-resonant Raman Spectrum and Fit')
         #\nRSB @ '+array_str(np.around(rsb_l,decimals=3))+\
#         '('+array_str(np.around(drsb_l,decimals=3))+') MHz')
#axis([-15.1,-7.9,0,700])
#axis([-15.1,-7.9,400,1200])
axis([-13.6,-9.4,200,700])
#axis([-13.6,-9.4,-30,510])
plt.grid(True)
```

E.1.2. FOR Raman Spectrum Analysis

The code below analyzes the FOR frequency spectrum and performs a fit to measure $\bar{n}$ supplying that the correct frequency ranges are included.

```
%pylab inline
import scipy.optimize as sc

file0 = r'C:\data\Rabi\Temperature\20160208
    _CWRamancool_FOR_Temperature_NS_scanPMT_0.npy'
file1 = r'C:\data\Rabi\Temperature\20160205
    _CWRamancool_FOR_Temperature_NS_scanPMT_1.npy'
file2 = r'C:\data\Rabi\Temperature\20160205
    _CWRamancool_FOR_Temperature_NS_scanPMT_2.npy'
file3 = r'C:\data\Rabi\Temperature\20160205
    _CWRamancool_FOR_Temperature_NS_scanPMT_4.npy'
data0 = load(file0)
data1 = load(file1)
data2 = load(file2)
data3 = load(file3)

expose_t0 = 5.0e-3
expose_t1 = 5.0e-3
expose_t2 = 5.0e-3
expose_t3 = 5.0e-3

#Histogram discriminator check if needed. Only runs on data0 unless code below
    changed.
pruned0 = data0[:,1:]/expose_t0
pruned1 = data1[:,1:]/expose_t1
pruned2 = data2[:,1:]/expose_t2
pruned3 = data3[:,1:]/expose_t3
hist(pruned1.ravel(),50)
xlabel(r'PMT Count Rate (s$^{-1}$)')
#axis([0,10000,0,100])

#Define ion bight/dark discriminator value and number of points in
#each scan range.
discr0 = 20000
discrh0 = 60000
```

```
discr1 = 20000
discrh1 = 60000

discr2 = 20000
discrh2 = 60000

discr3 = 20000
discrh3 = 60000

data0 = data0[data0[:,0].argsort()]
data1 = data1[data1[:,0].argsort()]
data2 = data2[data2[:,0].argsort()]
data3 = data3[data3[:,0].argsort()]

#DO WORK. This section extracts frequencies and runs the loops to calculate
   bright probability and number of experiments.
#define number of events and event probability with sigma giving stat error.
res0 = zeros(data0.shape[0])
f0 = zeros(data0.shape[0])
n0 = zeros(data0.shape[0])
p0 = zeros(data0.shape[1])
for i in range(data0.shape[0]):
    f0[i] = data0[i,0]
    res0[i] = sum(np.logical_and(data0[i,1:]/expose_t0 > discr0, data0[i,1:]/
       expose_t0 < discrh0))
    n0[i] = sum(np.logical_and(data0[i,1:]/expose_t0 > 0, data0[i,1:]/expose_t0
        < discrh0))
p0 = res0/n0
sigma0 = sqrt(p0*(1-p0)/n0)

res1 = zeros(data1.shape[0])
f1 = zeros(data1.shape[0])
n1 = zeros(data1.shape[0])
p1 = zeros(data1.shape[1])
for i in range(data1.shape[0]):
    f1[i] = data1[i,0]
    res1[i] = sum(np.logical_and(data1[i,1:]/expose_t1 > discr1, data1[i,1:]/
       expose_t1 < discrh1))
    n1[i] = sum(np.logical_and(data1[i,1:]/expose_t1 > 0, data1[i,1:]/expose_t1
        < discrh1))
```

```
p1 = res1/n1
sigma1 = sqrt(p1*(1-p1)/n1)

res2 = zeros(data2.shape[0])
f2 = zeros(data2.shape[0])
n2 = zeros(data2.shape[0])
p2 = zeros(data2.shape[1])
for i in range(data2.shape[0]):
    f2[i] = data2[i,0]
    res2[i] = sum(np.logical_and(data2[i,1:]/expose_t2 > discr2, data2[i,1:]/
        expose_t2 < discrh2))
    n2[i] = sum(np.logical_and(data2[i,1:]/expose_t2 > 0, data2[i,1:]/expose_t2
         < discrh2))
p2 = res2/n2
sigma2 = sqrt(p2*(1-p2)/n2)

res3 = zeros(data3.shape[0])
f3 = zeros(data3.shape[0])
n3 = zeros(data3.shape[0])
p3 = zeros(data3.shape[1])
for i in range(data3.shape[0]):
    f3[i] = data3[i,0]
    res3[i] = sum(np.logical_and(data3[i,1:]/expose_t3 > discr3, data3[i,1:]/
        expose_t3 < discrh3))
    n3[i] = sum(np.logical_and(data3[i,1:]/expose_t3 > 0, data3[i,1:]/expose_t3
         < discrh3))
p3 = res3/n3
sigma3 = sqrt(p3*(1-p3)/n3)

#Combine same frequency points!
temp = np.zeros((np.unique(f0).shape[0],3))
j = 0
temp[0,0] = f0[0]
temp[0,1] = p0[0]*n0[0]
temp[0,2] = n0[0]
for i in range(f0.shape[0]-1):
    if f0[i] == f0[i+1]:
        temp[j,0] = f0[i]
        temp[j,1] = temp[j,1]+p0[i+1]*n0[i+1]
        temp[j,2] = temp[j,2]+n0[i+1]
    else:
        j = j + 1
        temp[j,0] = f0[i]
        temp[j,1] = p0[i]*n0[i]
        temp[j,2] = n0[i]
        if i == (f0.shape[0]-2):
            temp[j,0] = f0[i+1]
            temp[j,1] = p0[i+1]*n0[i+1]
            temp[j,2] = n0[i+1]
del (f0,p0,n0,sigma0)
f0 = temp[:,0]
p0 = temp[:,1]/temp[:,2]
n0 = temp[:,2]
sigma0 = sqrt(p0*(1-p0)/n0)

temp = np.zeros((np.unique(f1).shape[0],3))
j = 0
temp[0,0] = f1[0]
temp[0,1] = p1[0]*n1[0]
temp[0,2] = n1[0]
for i in range(f1.shape[0]-1):
    if f1[i] == f1[i+1]:
        temp[j,0] = f1[i]
        temp[j,1] = temp[j,1]+p1[i+1]*n1[i+1]
        temp[j,2] = temp[j,2]+n1[i+1]
    else:
        j = j + 1
        temp[j,0] = f1[i]
        temp[j,1] = p1[i]*n1[i]
        temp[j,2] = n1[i]
        if i == (f1.shape[0]-2):
            temp[j,0] = f1[i+1]
            temp[j,1] = p1[i+1]*n1[i+1]
            temp[j,2] = n1[i+1]
del (f1,p1,n1,sigma1)
f1 = temp[:,0]
p1 = temp[:,1]/temp[:,2]
n1 = temp[:,2]
sigma1 = sqrt(p1*(1-p1)/n1)

temp = np.zeros((np.unique(f2).shape[0],3))
j = 0
```

```
temp[0,0] = f2[0]
temp[0,1] = p2[0]*n2[0]
temp[0,2] = n2[0]
for i in range(f2.shape[0]-1):
    if f2[i] == f2[i+1]:
        temp[j,0] = f2[i]
        temp[j,1] = temp[j,1]+p2[i+1]*n2[i+1]
        temp[j,2] = temp[j,2]+n2[i+1]
    else:
        j = j + 1
        temp[j,0] = f2[i]
        temp[j,1] = p2[i]*n2[i]
        temp[j,2] = n2[i]
        if i == (f2.shape[0]-2):
            temp[j,0] = f2[i+1]
            temp[j,1] = p2[i+1]*n2[i+1]
            temp[j,2] = n2[i+1]
del (f2,p2,n2,sigma2)
f2 = temp[:,0]
p2 = temp[:,1]/temp[:,2]
n2 = temp[:,2]
sigma2 = sqrt(p2*(1-p2)/n2)

temp = np.zeros((np.unique(f3).shape[0],3))
j = 0
temp[0,0] = f3[0]
temp[0,1] = p3[0]*n3[0]
temp[0,2] = n3[0]
for i in range(f3.shape[0]-1):
    if f3[i] == f3[i+1]:
        temp[j,0] = f3[i]
        temp[j,1] = temp[j,1]+p3[i+1]*n3[i+1]
        temp[j,2] = temp[j,2]+n3[i+1]
    else:
        j = j + 1
        temp[j,0] = f3[i]
        temp[j,1] = p3[i]*n3[i]
        temp[j,2] = n3[i]
        if i == (f3.shape[0]-2):
            temp[j,0] = f3[i+1]
            temp[j,1] = p3[i+1]*n3[i+1]
            temp[j,2] = n3[i+1]
del (f3,p3,n3,sigma3)
f3 = temp[:,0]
p3 = temp[:,1]/temp[:,2]
n3 = temp[:,2]
sigma3 = sqrt(p3*(1-p3)/n3)

#Provision to offset data runs to deal with any background changes.
bg = False
normalizedata = False
if bg == True:
    bg0 = np.nanmin(p0)
    bg1 = np.nanmin(p1)
    bg2 = np.nanmin(p2)
    bg3 = np.nanmin(p3)
    p0 = p0 - bg0
    p1 = p1 - bg1
    p2 = p2 - bg2
    p3 = p3 - bg3
if normalizedata == True:
    err0 = err0/(np.nanmax(p0))
    p0 = (p0)/(np.nanmax(p0))
    err1 = err1/(np.nanmax(p1))
    p1 = (p1)/(np.nanmax(p1))
    err2 = err2/(np.nanmax(p2))
    p2 = (p2)/(np.nanmax(p2))
    err3 = err3/(np.nanmax(p3))
    p3 = (p3)/(np.nanmax(p3))

RemovePoints = False
if RemovePoints == True:
    for i in range(p0.shape[0]):
        if p0[i] < 0.09:
            p0[i] = np.nan
    for i in range(p1.shape[0]):
        if p1[i] < 0.09:
            p1[i] = np.nan
    for i in range(p2.shape[0]):
        if p2[i] < 0.09:
            p2[i] = np.nan
    for i in range(p3.shape[0]):
        if p3[i] < 0.04:
            p3[i] = np.nan

plt.scatter(f0,p0,color='blue')
#plt.plot(f0,p0,color='blue')
plt.errorbar(f0,p0,yerr=sigma0,color='blue')
#plt.scatter(f1,p1,color='green')
#plt.plot(f1,p1,color='green')
#plt.errorbar(f1,p1,yerr=sigma1,color='green')
plt.scatter(f2,p2,color='red')
#plt.plot(f2,p2,color='red')
plt.errorbar(f2,p2,yerr=sigma2,color='red')
plt.scatter(f3,p3,color='purple')
#plt.plot(f3,p3,color='purple')
plt.errorbar(f3,p3,yerr=sigma3,color='purple')
plt.xlabel('(Probe - Pump) Frequency (MHz)')
plt.ylabel('Bright Fraction')
#plt.legend((r'-10 $\mu$s',r'-12 $\mu$s',r'-15 $\mu$s',r'-20 $\mu$s'),loc='
    upper right')
#plt.legend(('RF=-3.4','RF=-3.2','RF=-3.0','RF=-2.8'),loc='upper right')
plt.title(r'FOR Raman Pi Pulses')
plt.axis([-12.3,-9.6,0.05,0.6])
#plt.axis([-12.3,-11.9,0.04,0.57])
#plt.axis([-11.3,-10.65,0.05,0.6])
#plt.axis([-10.2,-9.5,0.05,0.32])
plt.grid(True)

#To fit the data above, define a fit function
def fitfn(x, c_A, c_l, c_w, omega_z, bsb_A, sb_w, rsb_A, D):
    #return c_A*np.exp(-(x-c_l)**2/(2*c_w**2))+bsb_A*np.exp(-(x-(c_l+omega_z))
        **2/(2*sb_w**2))+\
    #       rsb_A*np.exp(-(x-(c_l-omega_z))**2/(2*sb_w**2))+D
    return c_A*((1/2.0*c_w)**2)/((x-(c_l))**2+(1/2.0*c_w)**2)+\
       bsb_A*((1/2.0*sb_w)**2)/((x-(c_l+omega_z))**2+(1/2.0*sb_w)**2)+\
       rsb_A*((1/2.0*sb_w)**2)/((x-(c_l-omega_z))**2+(1/2.0*sb_w)**2)+\
       D

del (ffit,pfit,sigmafit)
#Do any combination of the fits:
ffit = f0
#ffit = np.concatenate((f0,f1))
```

```
pfit = p0
#pfit = np.concatenate((p0,p1))
sigmafit = sigma0
#sigmafit = np.concatenate((sigma0,sigma1))
#Perform curve fit.
popt, pcov = sc.curve_fit(fitfn, ffit, pfit,p0
   =(0.50,-10.9,0.1,1.14,0.15,0.02,0.05,0.12),sigma=sigmafit,absolute_sigma='
   true')
perr = np.sqrt(np.diag(pcov))
tempfit = np.empty((popt.shape[0],3),dtype='|S8')
tempfit[:,0] = ['c_A', 'c_l', 'c_w', 'omega_z', 'bsb_A', 'sb_w', 'rsb_A','D']
tempfit[:,1] = popt.astype('|S8')
tempfit[:,2] = perr.astype('|S8')
bsb_Amp = popt[4]
delta_bsb_Amp = perr[4]
rsb_Amp = popt[6]
delta_rsb_Amp = perr[6]
tempfit

#Delta_nbar definition fixed on 20160202... the deltas were backwards.
del (nbar,delta_nbar)
r = rsb_Amp/bsb_Amp
nbar = r/(1-r)
delta_nbar = nbar*sqrt(((1/(bsb_Amp-rsb_Amp))*delta_bsb_Amp)**2+\
                    ((bsb_Amp/(rsb_Amp*(bsb_Amp-rsb_Amp)))*delta_rsb_Amp)**2)
nbar_s = np.around([nbar],decimals=3)
delta_nbar_s = np.around([delta_nbar],decimals=3)
np.concatenate((['nbar'],nbar_s,delta_nbar_s))

plt.scatter(ffit,pfit,color='blue')
plt.errorbar(ffit,pfit,yerr=sigmafit,color='blue')
x = linspace(-13,-8,1000)
plot(x,fitfn(x,popt[0],popt[1],popt[2],popt[3],popt[4],popt[5],popt[6],popt[7])
   ,color='black',linewidth=3)
#plot(x,fitfn(x,0.12,-9.1,0.02,0.32),color='black',linewidth=3)
plt.xlabel(r'FOR Raman Pulse Time ($\mu$s)')
plt.ylabel('Bright Fraction')
plt.title('20160208 FOR Raman Spectrum, Lorentzian Fit, Software Triggered\
   nnbar = '+array_str(nbar_s[0])+', dnbar = '+array_str(delta_nbar_s[0]))
#plt.axis([-12.18,-12.04,0.05,0.25])
#plt.axis([-11.3,-10.65,0.05,0.6])
```

```
#plt.axis([-9.9,-9.75,0.05,0.4])
plt.axis([-12.4,-9.5,0.05,0.6])
plt.grid(True)

del temp
temp = np.zeros((f3.shape[0],4))
temp[:,0] = f3
temp[:,1] = p3
temp[:,2] = sigma3
temp[:,3] = n3
temp
#np.save(r'C:\data\Rabi\Temperature\20160205
    _CWRamancool_FOR_Temperature_NS_scanPMT_2_processed.npy',temp)

#del (nbarq,delta_nbarq)
rsb_Ampq = p3[0] - p3[1]
bsb_Ampq = p3[3] - p3[2]
delta_rsb_Ampq = rsb_Ampq*sqrt((sigma3[0]/p3[0])**2+(sigma3[1]/p3[1])**2)
delta_bsb_Ampq = bsb_Ampq*sqrt((sigma3[3]/p3[3])**2+(sigma3[2]/p3[2])**2)
rq = rsb_Ampq/bsb_Ampq
if rq > 1.0:
    print('Nbar cannot be calculated! Amplitude ratio >1!')
else:
    nbarq = rq/(1-rq)
    delta_nbarq = nbarq*sqrt(((1/(bsb_Ampq-rsb_Ampq))*delta_bsb_Ampq)**2+\
                        ((bsb_Ampq/(rsb_Ampq*(bsb_Ampq-rsb_Ampq)))*
                            delta_rsb_Ampq)**2)
    nbar_sq = np.around([nbarq],decimals=3)
    delta_nbar_sq = np.around([delta_nbarq],decimals=3)
    print(np.concatenate((['quick nbar'],nbar_sq,delta_nbar_sq)))
```

E.1.3. FOR Raman Time Analysis

The code below analyzes the FOR Raman pulse time scan and performs a fit to measure the effective π pulse time on either the carrier, RSB, or BSB.

```
%pylab inline
import scipy.optimize as sc

file0 = r'C:\data\Rabi\Time\20160224_CWRamancool_FOR_Time_NS_scanPMT_0.npy'
file1 = r'C:\data\Rabi\Time\20160224_CWRamancool_FOR_Time_NS_scanPMT_1.npy'
file2 = r'C:\data\Rabi\Time\20160224_CWRamancool_FOR_Time_NS_scanPMT_2.npy'
file3 = r'C:\data\Rabi\Time\20160224_CWRamancool_FOR_Time_NS_scanPMT_3.npy'
data0 = load(file0)
data1 = load(file1)
data2 = load(file2)
data3 = load(file3)

expose_t0 = 5.0e-3
expose_t1 = 5.0e-3
expose_t2 = 5.0e-3
expose_t3 = 5.0e-3

#Histogram discriminator check if needed. Only runs on data0 unless code below
   changed.
pruned0 = data0[:,1:]/expose_t0
pruned1 = data1[:,1:]/expose_t1
pruned2 = data2[:,1:]/expose_t2
pruned3 = data3[:,1:]/expose_t3
hist(pruned1.ravel(),50)
xlabel(r'PMT Count Rate (s$^{-1}$)')
#axis([0,10000,0,100])

#Define ion bight/dark discriminator value and number of points in
#each scan range.
discr0 = 20000
discrh0 = 60000

discr1 = 20000
discrh1 = 60000
```

```
discr2 = 20000
discrh2 = 60000

discr3 = 20000
discrh3 = 60000

data0 = data0[data0[:,0].argsort()]
data1 = data1[data1[:,0].argsort()]
data2 = data2[data2[:,0].argsort()]
data3 = data3[data3[:,0].argsort()]

#Add a zero row to the bottom of each data matrix to fix a boundary
#issue in a for loop in data analysis.
data0 = np.vstack((data0,np.zeros((data0.shape[1]))))
data1 = np.vstack((data1,np.zeros((data1.shape[1]))))
data2 = np.vstack((data2,np.zeros((data2.shape[1]))))
data3 = np.vstack((data3,np.zeros((data3.shape[1]))))

#DO WORK. This section extracts frequencies and runs the loops to calculate
   bright
#probability and number of experiments with sigma giving stat error. It uses a
   clever
#indexing method for interleaved datapoints.
t0 = np.zeros((np.unique(data0[:,0]).shape[0]))
res0 = np.zeros((np.unique(data0[:,0]).shape[0]))
n0 = np.zeros((np.unique(data0[:,0]).shape[0]))
j = 0
for i in range(data0.shape[0]-1):
    index = (np.argwhere(data0[:,0]==data0[i,0]).transpose()[0])
    t0[j] = data0[i,0]
    res0[j] = sum(np.logical_and(data0[index,1:]/expose_t0 > discr0, data0[
       index,1:]/expose_t0 < discrh0))
    n0[j] = sum(np.logical_and(data0[index,1:]/expose_t0 > 0, data0[index,1:]/
       expose_t0 < discrh0))
    if data0[i,0] != data0[i+1,0]:
        j = j + 1
j0 = j
t1 = np.zeros((np.unique(data1[:,0]).shape[0]))
res1 = np.zeros((np.unique(data1[:,0]).shape[0]))
n1 = np.zeros((np.unique(data1[:,0]).shape[0]))
j = 0
```

```
for i in range(data1.shape[0]-1):
    index = (np.argwhere(data1[:,0]==data1[i,0]).transpose()[0])
    t1[j] = data1[i,0]
    res1[j] = sum(np.logical_and(data1[index,1:]/expose_t1 > discr1, data1[
        index,1:]/expose_t1 < discrh1))
    n1[j] = sum(np.logical_and(data1[index,1:]/expose_t1 > 0, data1[index,1:]/
        expose_t1 < discrh1))
    if data1[i,0] != data1[i+1,0]:
        j = j + 1
t2 = np.zeros((np.unique(data2[:,0]).shape[0]))
res2 = np.zeros((np.unique(data2[:,0]).shape[0]))
n2 = np.zeros((np.unique(data2[:,0]).shape[0]))
j = 0
for i in range(data2.shape[0]-1):
    index = (np.argwhere(data2[:,0]==data2[i,0]).transpose()[0])
    t2[j] = data2[i,0]
    res2[j] = sum(np.logical_and(data2[index,1:]/expose_t2 > discr2, data2[
        index,1:]/expose_t2 < discrh2))
    n2[j] = sum(np.logical_and(data2[index,1:]/expose_t2 > 0, data2[index,1:]/
        expose_t2 < discrh2))
    if data2[i,0] != data2[i+1,0]:
        j = j + 1
t3 = np.zeros((np.unique(data3[:,0]).shape[0]))
res3 = np.zeros((np.unique(data3[:,0]).shape[0]))
n3 = np.zeros((np.unique(data3[:,0]).shape[0]))
j = 0
for i in range(data3.shape[0]-1):
    index = (np.argwhere(data3[:,0]==data3[i,0]).transpose()[0])
    t3[j] = data3[i,0]
    res3[j] = sum(np.logical_and(data3[index,1:]/expose_t3 > discr3, data3[
        index,1:]/expose_t3 < discrh3))
    n3[j] = sum(np.logical_and(data3[index,1:]/expose_t3 > 0, data3[index,1:]/
        expose_t3 < discrh3))
    if data3[i,0] != data3[i+1,0]:
        j = j + 1

#Fix the boundary condition.
t0 = np.delete(t0,t0.shape[0]-1)
res0 = np.delete(res0,res0.shape[0]-1)
n0 = np.delete(n0,n0.shape[0]-1)
t1 = np.delete(t1,t1.shape[0]-1)
```

```
res1 = np.delete(res1,res1.shape[0]-1)
n1 = np.delete(n1,n1.shape[0]-1)
t2 = np.delete(t2,t2.shape[0]-1)
res2 = np.delete(res2,res2.shape[0]-1)
n2 = np.delete(n2,n2.shape[0]-1)
t3 = np.delete(t3,t3.shape[0]-1)
res3 = np.delete(res3,res3.shape[0]-1)
n3 = np.delete(n3,n3.shape[0]-1)

#Calculate the bright fraction and uncertainty using binomial statistics.
p0 = np.zeros((np.unique(data0[:,0]).shape[0]-1))
sigma0 = np.zeros((np.unique(data0[:,0]).shape[0]-1))
p0 = res0/n0
sigma0 = sqrt(p0*(1-p0)/n0)
p1 = np.zeros((np.unique(data1[:,0]).shape[0]-1))
sigma1 = np.zeros((np.unique(data1[:,0]).shape[0]-1))
p1 = res1/n1
sigma1 = sqrt(p1*(1-p1)/n1)
p2 = np.zeros((np.unique(data2[:,0]).shape[0]-1))
sigma2 = np.zeros((np.unique(data2[:,0]).shape[0]-1))
p2 = res2/n2
sigma2 = sqrt(p2*(1-p2)/n2)
p3 = np.zeros((np.unique(data3[:,0]).shape[0]-1))
sigma3 = np.zeros((np.unique(data3[:,0]).shape[0]-1))
p3 = res3/n3
sigma3 = sqrt(p3*(1-p3)/n3)

RemovePoints = False
if RemovePoints == True:
    for i in range(p0.shape[0]):
        if p0[i] < 0.09:
            p0[i] = np.nan
    for i in range(p1.shape[0]):
        if p1[i] < 0.09:
            p1[i] = np.nan
    for i in range(p2.shape[0]):
        if p2[i] < 0.09:
            p2[i] = np.nan
    for i in range(p3.shape[0]):
        if p3[i] < 0.09:
            p3[i] = np.nan
```

```
plt.errorbar(t0*1e6,p0,yerr=sigma0,color=’blue’,fmt=’-o’)
plt.errorbar(t1*1e6,p1,yerr=sigma1,color=’green’,fmt=’-o’)
plt.errorbar(t2*1e6,p2,yerr=sigma2,color=’red’,fmt=’-o’)
plt.errorbar(t3*1e6,p3,yerr=sigma3,color=’purple’,fmt=’-o’)
plt.xlabel(r’FOR Raman Pulse Time ($\mu$s)’)
plt.ylabel(’Bright Fraction’)
#plt.legend((r’-10 $\mu$s’,r’-12 $\mu$s’,r’-15 $\mu$s’,r’-20 $\mu$s’),loc=’
    upper right’)
#plt.legend((’Software’,’Line’),loc=’upper right’)
plt.title(r’FOR Raman Pulse Time’)
plt.axis([-5,95,0.20,0.70])
plt.grid(True)

#To fit the data above, define a fit function
def fitfn(t, A, B, P, D):
    return A/2*np.exp(-t/B)*sin(2*pi*t/P-pi/2)+D

try:
    del (tfit,pfit,sigmafit)
except NameError:
    print
#Do any combination of the fits:
tfit = 1E6*t3
#np.concatenate((t2,t3))*1E6
pfit = p3
#= np.concatenate((p2,p3))
sigmafit = sigma3
#np.concatenate((sigma2,sigma3))
#Perform curve fit.
popt, pcov = sc.curve_fit(fitfn, tfit, pfit,p0=(0.35,200.0,20,0.50),sigma=
    sigmafit,\
                        absolute_sigma=’true’,maxfev=100000)
perr = np.sqrt(np.diag(pcov))
tempfit = np.empty((popt.shape[0],3),dtype=’|S8’)
tempfit[:,0] = [’A’,’B’,’P’,’D’]
tempfit[:,1] = popt.astype(’|S8’)
tempfit[:,2] = perr.astype(’|S8’)
tempfit

#plt.scatter(tfit,pfit,color=’green’)
```

```
plt.errorbar(tfit,pfit,yerr=sigmafit,color='purple',fmt='o')
x = linspace(0,210,2100)
plot(x,fitfn(x,popt[0],popt[1],popt[2],popt[3]),color='black',linewidth=3)
#plot(x,fitfn(x,0.45,160.0,19.8,0.33),color='black',linewidth=3)
plt.xlabel(r'FOR Raman Pulse Time ($\mu$s)')
plt.ylabel('Bright Fraction')
plt.title('20160216 FOR Raman Carrier Rabi Oscillations\nPi Pulse Time = '\
         +array_str(np.around(popt[2]/2,decimals=3))+\
         '('+array_str(np.around(perr[2]/2,decimals=3))+r') $\mu$s')
plt.axis([-1,95,0.30,0.70])
plt.grid(True)

try:
   del temp
except NameError:
   print
temp = np.zeros((t1.shape[0],4))
temp[:,0] = 1E6*t1
temp[:,1] = p1
temp[:,2] = sigma1
temp[:,3] = n1
#temp
#np.save(r'C:\data\Rabi\Time\20160204
   _CWRamancool_FOR_Time_NS_scanPMT_3_processed.npy',temp)
```

E.1.4. Heating Rate Analysis

The code below analyzes the heating rate data using the fast, four-point temperature measurement.

```
%pylab inline
import scipy.optimize as sc

file0 = r'C:\data\Rabi\Heating\20160212_CWRamancool_FOR_Heating_NS_scanPMT_0.
    npy'
file1 = r'C:\data\Rabi\Heating\20160212_CWRamancool_FOR_Heating_NS_scanPMT_1.
    npy'
file2 = r'C:\data\Rabi\Heating\20160211_CWRamancool_FOR_Heating_NS_scanPMT_3.
    npy'
file3 = r'C:\data\Rabi\Temperature Fast\20160208
    _CWRamancool_FOR_FastTemperature_NS_scanPMT_3.npy'
data0 = load(file0)
data1 = load(file1)
data2 = load(file2)
data3 = load(file3)

expose_t0 = 5.0e-3
expose_t1 = 5.0e-3
expose_t2 = 5.0e-3
expose_t3 = 5.0e-3

#Histogram discriminator check if needed. Only runs on data0 unless code below
    changed.
pruned0 = data0[:,2:]/expose_t0
pruned1 = data1[:,2:]/expose_t1
pruned2 = data2[:,2:]/expose_t2
pruned3 = data3[:,2:]/expose_t3
hist(pruned1.ravel(),50)
xlabel(r'PMT Count Rate (s$^{-1}$)')
#axis([0,10000,0,100])

#Define ion bight/dark discriminator value and number of points in
#each scan range.
discr0 = 20000
discrh0 = 60000
```

```
discr1 = 20000
discrh1 = 60000

discr2 = 20000
discrh2 = 60000

discr3 = 20000
discrh3 = 60000

# Combine Datasets here if desired.
data3 = np.zeros(((data0.shape[0]+data1.shape[0]),data1.shape[1]))
data3 = np.vstack((data0,data1))

data0 = data0[np.lexsort((data0[:,1],data0[:,0]))]
data1 = data1[np.lexsort((data1[:,1],data1[:,0]))]
data2 = data2[np.lexsort((data2[:,1],data2[:,0]))]
data3 = data3[np.lexsort((data3[:,1],data3[:,0]))]

#Add a zero row to the bottom of each data matrix to fix a boundary
#issue in a for loop in data analysis.
data0 = np.vstack((data0,np.array(np.repeat(nan,data0.shape[1]))))
data1 = np.vstack((data1,np.array(np.repeat(nan,data1.shape[1]))))
data2 = np.vstack((data2,np.array(np.repeat(nan,data2.shape[1]))))
data3 = np.vstack((data3,np.array(np.repeat(nan,data3.shape[1]))))

#DO WORK. This section extracts frequencies and runs the loops to calculate
   bright
#probability and number of experiments with sigma giving stat error. It uses a
   clever
#indexing method for interleaved datapoints.
htt0 = np.zeros(((np.unique(data0[:,0]).shape[0]-1)*(np.unique(data0[:,1]).
   shape[0]-1)))
f0 = np.zeros(((np.unique(data0[:,0]).shape[0]-1)*(np.unique(data0[:,1]).shape
   [0]-1)))
res0 = np.zeros(((np.unique(data0[:,0]).shape[0]-1)*(np.unique(data0[:,1]).
   shape[0]-1)))
n0 = np.zeros(((np.unique(data0[:,0]).shape[0]-1)*(np.unique(data0[:,1]).shape
   [0]-1)))
j = 0
for i in range(data0.shape[0]-1):
   index = np.argwhere(np.logical_and(data0[i,0]==data0[:,0],data0[i,1]==data0
       [:,1])).transpose()[0]
   htt0[j] = data0[i,0]*1E3
   f0[j] = data0[i,1]
   res0[j] = sum(np.logical_and(data0[index,2:]/expose_t0 > discr0, data0[
       index,2:]/expose_t0 < discrh0))
   n0[j] = sum(np.logical_and(data0[index,2:]/expose_t0 > 0, data0[index,2:]/
       expose_t0 < discrh0))
   if np.logical_or(data0[i,0] != data0[i+1,0],data0[i,1] != data0[i+1,1]):
       j = j + 1
   j0 = j
htt1 = np.zeros(((np.unique(data1[:,0]).shape[0]-1)*(np.unique(data1[:,1]).
   shape[0]-1)))
f1 = np.zeros(((np.unique(data1[:,0]).shape[0]-1)*(np.unique(data1[:,1]).shape
   [0]-1)))
res1 = np.zeros(((np.unique(data1[:,0]).shape[0]-1)*(np.unique(data1[:,1]).
   shape[0]-1)))
n1 = np.zeros(((np.unique(data1[:,0]).shape[0]-1)*(np.unique(data1[:,1]).shape
   [0]-1)))
j = 0
for i in range(data1.shape[0]-1):
   index = np.argwhere(np.logical_and(data1[i,0]==data1[:,0],data1[i,1]==data1
       [:,1])).transpose()[0]
   htt1[j] = data1[i,0]*1E3
   f1[j] = data1[i,1]
   res1[j] = sum(np.logical_and(data1[index,2:]/expose_t1 > discr1, data1[
       index,2:]/expose_t1 < discrh1))
   n1[j] = sum(np.logical_and(data1[index,2:]/expose_t1 > 0, data1[index,2:]/
       expose_t1 < discrh1))
   if np.logical_or(data1[i,0] != data1[i+1,0],data1[i,1] != data1[i+1,1]):
       j = j + 1
htt2 = np.zeros(((np.unique(data2[:,0]).shape[0]-1)*(np.unique(data2[:,1]).
   shape[0]-1)))
f2 = np.zeros(((np.unique(data2[:,0]).shape[0]-1)*(np.unique(data2[:,1]).shape
   [0]-1)))
res2 = np.zeros(((np.unique(data2[:,0]).shape[0]-1)*(np.unique(data2[:,1]).
   shape[0]-1)))
n2 = np.zeros(((np.unique(data2[:,0]).shape[0]-1)*(np.unique(data2[:,1]).shape
   [0]-1)))
j = 0
for i in range(data2.shape[0]-1):
   index = np.argwhere(np.logical_and(data2[i,0]==data2[:,0],data2[i,1]==data2
       [:,1])).transpose()[0]
   htt2[j] = data2[i,0]*1E3
   f2[j] = data2[i,1]
   res2[j] = sum(np.logical_and(data2[index,2:]/expose_t2 > discr2, data2[
       index,2:]/expose_t2 < discrh2))
   n2[j] = sum(np.logical_and(data2[index,2:]/expose_t2 > 0, data2[index,2:]/
       expose_t2 < discrh2))
   if np.logical_or(data2[i,0] != data2[i+1,0],data2[i,1] != data2[i+1,1]):
       j = j + 1
htt3 = np.zeros(((np.unique(data3[:,0]).shape[0]-1)*(np.unique(data3[:,1]).
   shape[0]-1)))
f3 = np.zeros(((np.unique(data3[:,0]).shape[0]-1)*(np.unique(data3[:,1]).shape
   [0]-1)))
res3 = np.zeros(((np.unique(data3[:,0]).shape[0]-1)*(np.unique(data3[:,1]).
   shape[0]-1)))
n3 = np.zeros(((np.unique(data3[:,0]).shape[0]-1)*(np.unique(data3[:,1]).shape
   [0]-1)))
j = 0
for i in range(data3.shape[0]-1):
   index = np.argwhere(np.logical_and(data3[i,0]==data3[:,0],data3[i,1]==data3
       [:,1])).transpose()[0]
   htt3[j] = data3[i,0]*1E3
   f3[j] = data3[i,1]
   res3[j] = sum(np.logical_and(data3[index,2:]/expose_t3 > discr3, data3[
       index,2:]/expose_t3 < discrh3))
   n3[j] = sum(np.logical_and(data3[index,2:]/expose_t3 > 0, data3[index,2:]/
       expose_t3 < discrh3))
   if np.logical_or(data3[i,0] != data3[i+1,0],data3[i,1] != data3[i+1,1]):
       j = j + 1

p0 = np.zeros((np.unique(data0[:,0]).shape[0]-1))
sigma0 = np.zeros((np.unique(data0[:,0]).shape[0]-1))
p0 = res0/n0
sigma0 = sqrt(p0*(1-p0)/n0)
p1 = np.zeros((np.unique(data1[:,0]).shape[0]-1))
sigma1 = np.zeros((np.unique(data1[:,0]).shape[0]-1))
p1 = res1/n1
sigma1 = sqrt(p1*(1-p1)/n1)
p2 = np.zeros((np.unique(data2[:,0]).shape[0]-1))
sigma2 = np.zeros((np.unique(data2[:,0]).shape[0]-1))
```

```
p2 = res2/n2
sigma2 = sqrt(p2*(1-p2)/n2)
p3 = np.zeros((np.unique(data3[:,0]).shape[0]-1))
sigma3 = np.zeros((np.unique(data3[:,0]).shape[0]-1))
p3 = res3/n3
sigma3 = sqrt(p3*(1-p3)/n3)

#plt.scatter(f0,p0,color='blue')
#plt.plot(f0,p0,color='blue')
plt.errorbar(f0,p0,yerr=sigma0,color='blue',fmt='o')
#plt.scatter(f1,p1,color='green')
#plt.plot(f1,p1,color='green')
plt.errorbar(f1,p1,yerr=sigma1,color='green',fmt='o')
#plt.scatter(f2,p2,color='red')
#plt.plot(f2,p2,color='red')
plt.errorbar(f2,p2,yerr=sigma2,color='red',fmt='o')
#plt.scatter(f3,p3,color='purple')
#plt.plot(f3,p3,color='purple')
#plt.errorbar(f3,p3,yerr=sigma3,color='purple')
plt.xlabel('FOR (Probe - Pump) Frequency (MHz)')
plt.ylabel('Bright Fraction')
#plt.legend((r'-10 $\mu$s',r'-12 $\mu$s',r'-15 $\mu$s',r'-20 $\mu$s'),loc='
    upper right')
#plt.legend(('RF=-3.4','RF=-3.2','RF=-3.0','RF=-2.8'),loc='upper right')
plt.title(r'FOR Raman Pi Pulses')
plt.axis([-12.3,-9.6,0.05,0.6])
#plt.axis([-12.3,-11.9,0.04,0.57])
#plt.axis([-11.3,-10.65,0.05,0.6])
#plt.axis([-10.2,-9.5,0.05,0.32])
plt.grid(True)

try:
    del temp
except NameError:
    print
temp = np.zeros((htt0.shape[0],4))
temp[:,0] = htt0
temp[:,1] = f0
temp[:,2] = p0
temp[:,3] = sigma0
#temp
```

```
reps0 = 1.0*htt0.shape[0]/(np.unique(htt0).shape[0])
ht0 = np.zeros((np.unique(htt0).shape[0]))
rsb_Amp0 = np.zeros((np.unique(htt0).shape[0]))
delta_rsb_Amp0 = np.zeros((np.unique(htt0).shape[0]))
bsb_Amp0 = np.zeros((np.unique(htt0).shape[0]))
delta_bsb_Amp0 = np.zeros((np.unique(htt0).shape[0]))
if reps0.is_integer():
   for i in range((np.unique(htt0).shape[0])):
      ht0[i] = htt0[reps0*i]
      rsb_Amp0[i] = p0[reps0*i+0]-p0[reps0*i+1]
      delta_rsb_Amp0[i] = sqrt(sigma0[reps0*i+0]**2+sigma0[reps0*i+1]**2)
      bsb_Amp0[i] = p0[reps0*i+3]-p0[reps0*i+2]
      delta_bsb_Amp0[i] = sqrt(sigma0[reps0*i+3]**2+sigma0[reps0*i+2]**2)
else:
   print '0 FAIL'

reps1 = 1.0*htt1.shape[0]/(np.unique(htt1).shape[0])
ht1 = np.zeros((np.unique(htt1).shape[0]))
rsb_Amp1 = np.zeros((np.unique(htt1).shape[0]))
delta_rsb_Amp1 = np.zeros((np.unique(htt1).shape[0]))
bsb_Amp1 = np.zeros((np.unique(htt1).shape[0]))
delta_bsb_Amp1 = np.zeros((np.unique(htt1).shape[0]))
if reps1.is_integer():
   for i in range((np.unique(htt1).shape[0])):
      ht1[i] = htt1[reps1*i]
      rsb_Amp1[i] = p1[reps1*i+0]-p1[reps1*i+1]
      delta_rsb_Amp1[i] = sqrt(sigma1[reps1*i+0]**2+sigma1[reps1*i+1]**2)
      bsb_Amp1[i] = p1[reps1*i+3]-p1[reps1*i+2]
      delta_bsb_Amp1[i] = sqrt(sigma1[reps1*i+3]**2+sigma1[reps1*i+2]**2)
else:
   print '1 FAIL'

reps2 = 1.0*htt2.shape[0]/(np.unique(htt2).shape[0])
ht2 = np.zeros((np.unique(htt2).shape[0]))
rsb_Amp2 = np.zeros((np.unique(htt2).shape[0]))
delta_rsb_Amp2 = np.zeros((np.unique(htt2).shape[0]))
bsb_Amp2 = np.zeros((np.unique(htt2).shape[0]))
delta_bsb_Amp2 = np.zeros((np.unique(htt2).shape[0]))
if reps2.is_integer():
   for i in range((np.unique(htt2).shape[0])):
        ht2[i] = htt2[reps2*i]
        rsb_Amp2[i] = p2[reps2*i+0]-p2[reps2*i+1]
        delta_rsb_Amp2[i] = sqrt(sigma2[reps2*i+0]**2+sigma2[reps2*i+1]**2)
        bsb_Amp2[i] = p2[reps2*i+3]-p2[reps2*i+2]
        delta_bsb_Amp2[i] = sqrt(sigma2[reps2*i+3]**2+sigma2[reps2*i+2]**2)
else:
    print ’2 FAIL’

reps3 = 1.0*htt3.shape[0]/(np.unique(htt3).shape[0])
ht3 = np.zeros((np.unique(htt3).shape[0]))
rsb_Amp3 = np.zeros((np.unique(htt3).shape[0]))
delta_rsb_Amp3 = np.zeros((np.unique(htt3).shape[0]))
bsb_Amp3 = np.zeros((np.unique(htt3).shape[0]))
delta_bsb_Amp3 = np.zeros((np.unique(htt3).shape[0]))
if reps0.is_integer():
    for i in range((np.unique(htt3).shape[0])):
        ht3[i] = htt3[reps3*i]
        rsb_Amp3[i] = p3[reps3*i+0]-p3[reps3*i+1]
        delta_rsb_Amp3[i] = sqrt(sigma3[reps3*i+0]**2+sigma3[reps3*i+1]**2)
        bsb_Amp3[i] = p3[reps3*i+3]-p3[reps3*i+2]
        delta_bsb_Amp3[i] = sqrt(sigma3[reps3*i+3]**2+sigma3[reps3*i+2]**2)
else:
    print ’3 FAIL’

r0 = rsb_Amp0/bsb_Amp0
delta_r0 = r0*sqrt((delta_rsb_Amp0/rsb_Amp0)**2+(delta_bsb_Amp0/bsb_Amp0)**2)
nbar0 = r0/(1-r0)
delta_nbar0 = nbar0*sqrt(((1/(bsb_Amp0-rsb_Amp0))*delta_bsb_Amp0)**2+\
                     ((bsb_Amp0/(rsb_Amp0*(bsb_Amp0-rsb_Amp0)))*delta_rsb_Amp0)
                        **2)
r1 = rsb_Amp1/bsb_Amp1
delta_r1 = r1*sqrt((delta_rsb_Amp1/rsb_Amp1)**2+(delta_bsb_Amp1/bsb_Amp1)**2)
nbar1 = r1/(1-r1)
delta_nbar1 = nbar1*sqrt(((1/(bsb_Amp1-rsb_Amp1))*delta_bsb_Amp1)**2+\
                     ((bsb_Amp1/(rsb_Amp1*(bsb_Amp1-rsb_Amp1)))*delta_rsb_Amp1)
                        **2)
r2 = rsb_Amp2/bsb_Amp2
delta_r2 = r2*sqrt((delta_rsb_Amp2/rsb_Amp2)**2+(delta_bsb_Amp2/bsb_Amp2)**2)
nbar2 = r2/(1-r2)
delta_nbar2 = nbar2*sqrt(((1/(bsb_Amp2-rsb_Amp2))*delta_bsb_Amp2)**2+\
                    ((bsb_Amp2/(rsb_Amp2*(bsb_Amp2-rsb_Amp2)))*delta_rsb_Amp2)
                        **2)
r3 = rsb_Amp3/bsb_Amp3
delta_r3 = r3*sqrt((delta_rsb_Amp3/rsb_Amp3)**2+(delta_bsb_Amp3/bsb_Amp3)**2)
nbar3 = r3/(1-r3)
delta_nbar3 = nbar3*sqrt(((1/(bsb_Amp3-rsb_Amp3))*delta_bsb_Amp3)**2+\
                    ((bsb_Amp3/(rsb_Amp3*(bsb_Amp3-rsb_Amp3)))*delta_rsb_Amp3)
                        **2)

RemovePoints = False
if RemovePoints == True:
    for i in range(r0.shape[0]):
        if np.logical_or(r0[i] < 0.0, r0[i] > 1.0):
            r0[i] = np.nan
            delta_r0[i] = np.nan
            nbar0[i] = np.nan
            delta_nbar0[i] = np.nan
    for i in range(r1.shape[0]):
        if np.logical_or(r1[i] < 0.0, r1[i] > 1.0):
            r1[i] = np.nan
            delta_r1[i] = np.nan
            nbar1[i] = np.nan
            delta_nbar1[i] = np.nan
    for i in range(r2.shape[0]):
        if np.logical_or(r2[i] < 0.0, r2[i] > 1.0):
            r2[i] = np.nan
            delta_r2[i] = np.nan
            nbar2[i] = np.nan
            delta_nbar2[i] = np.nan
    for i in range(r3.shape[0]):
        if np.logical_or(r3[i] < 0.0, r3[i] > 1.0):
            r3[i] = np.nan
            delta_r3[i] = np.nan
            nbar3[i] = np.nan
            delta_nbar3[i] = np.nan
    for i in range(r4.shape[0]):
        if np.logical_or(r4[i] < 0.0, r4[i] > 1.0):
            r4[i] = np.nan
            delta_r4[i] = np.nan
            nbar4[i] = np.nan
            delta_nbar4[i] = np.nan
```

```
RemovePoints2 = True
try:
    del index
except NameError:
    print
if RemovePoints2 == True:
    index = np.argwhere(np.logical_or(r0 < (0.0-delta_r0),r0 > (1.0-delta_r0+
        delta_r0))).transpose()[0]
    ht0 = np.delete(ht0,index)
    r0 = np.delete(r0,index)
    delta_r0 = np.delete(delta_r0,index)
    nbar0 = np.delete(nbar0,index)
    delta_nbar0 = np.delete(delta_nbar0,index)

    index = np.argwhere(np.logical_or(r1 < (0.0-delta_r1),r1 > (1.0-delta_r1+
        delta_r1))).transpose()[0]
    ht1 = np.delete(ht1,index)
    r1 = np.delete(r1,index)
    delta_r1 = np.delete(delta_r1,index)
    nbar1 = np.delete(nbar1,index)
    delta_nbar1 = np.delete(delta_nbar1,index)

    index = np.argwhere(np.logical_or(r2 < (0.0-delta_r2),r2 > (1.0-delta_r2+
        delta_r2))).transpose()[0]
    ht2 = np.delete(ht2,index)
    r2 = np.delete(r2,index)
    delta_r2 = np.delete(delta_r2,index)
    nbar2 = np.delete(nbar2,index)
    delta_nbar2 = np.delete(delta_nbar2,index)

    index = np.argwhere(np.logical_or(r3 < (0.0-delta_r3),r3 > (1.0-delta_r3+
        delta_r3))).transpose()[0]
    ht3 = np.delete(ht3,index)
    r3 = np.delete(r3,index)
    delta_r3 = np.delete(delta_r3,index)
    nbar3 = np.delete(nbar3,index)
    delta_nbar3 = np.delete(delta_nbar3,index)

    index = np.argwhere(np.logical_or(r4 < (0.0-delta_r4),r4 > (1.0-delta_r4+
        delta_r4))).transpose()[0]
    ht4 = np.delete(ht4,index)
    r4 = np.delete(r4,index)
    delta_r4 = np.delete(delta_r4,index)
    nbar4 = np.delete(nbar4,index)
    delta_nbar4 = np.delete(delta_nbar4,index)

ht5 = np.array([15.0,5.0,25.0,10.0])
              #\,20.0,0.0,30.0,35.0,40.0,0.0,\
              #5.0,10.0,15.0,20.0,25.0,30.0,35.0,40.0,0.0])
nbar5 = np.array([0.178,0.0315,0.635,0.212])
#,0.0016,-0.0306,-0.0459,0.0495,0.122,-0.0167,\
#                0.012,-0.0443,-0.0511,0.071,0.055,0.444,-0.1240,0.005,0.005])
delta_nbar5 = np.array([0.138,0.0680,0.300,0.122])
#,0.0631,0.0660,0.0788,0.0953,0.136,0.0860,\
#
    0.106,0.0929,0.0880,0.130,0.105,0.339,0.0983,0.121,0.105])

#plt.errorbar(ht0,r0,yerr=delta_r0,color=’blue’,fmt=’o’)
#plt.errorbar(ht1,r1,yerr=delta_r1,color=’green’,fmt=’o’)
#plt.errorbar(ht2,r2,yerr=delta_r2,color=’red’,fmt=’o’)
plt.errorbar(ht3,r3,yerr=delta_r3,color=’purple’,fmt=’o’)
#plt.errorbar(ht4,r4,yerr=delta_r4,color=’black’,fmt=’o’)
plt.xlabel(’Heating Time (ms)’)
plt.ylabel(’RSB/BSB Amplitude Ratio’)
#plt.legend((r’-10 $\mu$s’,r’-12 $\mu$s’,r’-15 $\mu$s’,r’-20 $\mu$s’),loc=’
    upper right’)
#plt.legend((’RF=-3.4’,’RF=-3.2’,’RF=-3.0’,’RF=-2.8’),loc=’upper right’)
plt.title(r’Heating Rate Scan’)
#plt.axis([-12.3,-9.6,0.05,0.6])
#plt.axis([-12.3,-11.9,0.04,0.57])
#plt.axis([-11.3,-10.65,0.05,0.6])
plt.axis([-10,260,-1,2])
plt.grid(True)

#plt.errorbar(ht0,nbar0,yerr=delta_nbar0,color=’blue’,fmt=’s’)
#plt.errorbar(ht1,nbar1,yerr=delta_nbar1,color=’green’,fmt=’s’)
#plt.errorbar(ht2,nbar2,yerr=delta_nbar2,color=’red’,fmt=’s’)
plt.errorbar(ht3,nbar3,yerr=delta_nbar3,color=’purple’,fmt=’s’)
#plt.errorbar(ht4,nbar4,yerr=delta_nbar4,color=’black’,fmt=’s’)
plt.xlabel(’Heating Time (ms)’)
plt.ylabel(r’$\overline{n}$’)
```

```
#plt.legend((r'-10 $\mu$s',r'-12 $\mu$s',r'-15 $\mu$s',r'-20 $\mu$s'),loc='
   upper right')
#plt.legend(('RF=-3.4','RF=-3.2','RF=-3.0','RF=-2.8'),loc='upper right')
plt.title(r'Heating Rate Scan')
#plt.axis([-12.3,-9.6,0.05,0.6])
#plt.axis([-12.3,-11.9,0.04,0.57])
#plt.axis([-11.3,-10.65,0.05,0.6])
plt.axis([-1,260,-0.75,2])
plt.grid(True)

#To fit the data above, define a fit function
def fitfn(x, heating_rate, intercept):
   return heating_rate*x+intercept

try:
   del (ffit,pfit,sigmafit)
except NameError:
   print
#Do any combination of the fits:
htfit = ht5
#ffit = np.concatenate((f0,f1))
rfit = nbar5
#pfit = np.concatenate((p0,p1))
sigmafit = delta_nbar5
#sigmafit = np.concatenate((sigma0,sigma1))
#Perform curve fit.
popt, pcov = sc.curve_fit(fitfn, htfit, rfit,p0=(0.002,0.1),sigma=sigmafit,
   absolute_sigma='true',\
                      maxfev=10000)
perr = np.sqrt(np.diag(pcov))
tempfit = np.empty((popt.shape[0],3),dtype='|S32')
tempfit[:,0] = ['heating rate', 'intercept']
tempfit[:,1] = popt.astype('|S32')
tempfit[:,2] = perr.astype('|S32')
tempfit

plt.errorbar(htfit,rfit,yerr=sigmafit,color='black',fmt='o')
x = linspace(-5,100,100)
plot(x,fitfn(x,popt[0],popt[1]),color='black',linewidth=3)
#plot(x,fitfn(x,0.003,0.1),color='black',linewidth=3)
#plt.ylabel('RSB/BSB Amplitude Ratio')
```

```
#plt.xlabel('Heating Time (ms)')
plt.ylabel(r'$\overline{n}$')
plt.title('Heating Rate Scan and Fit\nHeating Rate = '+array_str(np.around(popt
    [0],decimals=4))+'('+array_str(np.around(perr[0],decimals=4))+') quanta/ms
    ')
#plt.title('Heating Rate Scan and Fit\nHeating Rate r = '+array_str(np.around(
    popt[0],decimals=3))+'('+array_str(np.around(perr[0],decimals=3))+') /ms')
#lt.title('Heating Rate Scan and Fit')
plt.axis([-1,45,-0.4,1.0])
plt.grid(True)
```

E.1.5. Histogram Analysis

The code below analyzes verifies efficient shelving and deshelving by displaying a histogram and calculating the bright and dark ion fractions with binomial counting uncertainties.

```
%pylab inline

#file0 = r'C:\data\Shelving\20160224_shelvingPMT_0.npy'
file0 = r'C:\data\Deshelving\20160224_deshelvingPMT_0.npy'
data = load(file0)

expose_t = 5.0e-3
pruned = data/expose_t
#pruned = data[:,1:]

hist(pruned.ravel(),30)
xlabel(r'PMT Count Rate (s$^{-1}$)')
#axis([0,200,0,200])
plt.title(file0)

disc = 20000
disch = 60000

#Calculate shelved percentage
res0 = float(sum(np.logical_and(pruned < disc, pruned < disch)))
n0 = float(sum(np.logical_and(pruned >= 0, pruned < disch)))
sp = res0/n0
sp

#Calculate error of this using binomial variance
sqrt(sp*(1-sp)/n0)

#Number Bright
sum(np.logical_and(pruned > disc, pruned < disch))

#Number Dark
sum(np.logical_and(pruned < disc, pruned < disch))
```

```
#Total Experiments
sum(np.logical_and(pruned > 1000, pruned < disch))
```

E.1.6. Magnetic Field Trimming Analysis

The code below analyzes the magnetic field trimming data to produce a contour and heat map. The data is fed in by polarity on the coil breakout box and is self-explanatory.

```
%pylab inline

#Fluorescence Files
neg_neg = load('20151216_BTrim_1.npy')
pos_pos = load('20151216_BTrim_2.npy')
neg_pos = load('20151216_BTrim_3.npy')
pos_neg = load('20151216_BTrim_4.npy')
#Error Files
neg_neg_err = load('20151216_BTrim_1_err.npy')
pos_pos_err = load('20151216_BTrim_2_err.npy')
neg_pos_err = load('20151216_BTrim_3_err.npy')
pos_neg_err = load('20151216_BTrim_4_err.npy')

neg = np.concatenate((flipud(neg_neg[1:,1:]),pos_neg[:,1:]),axis=0)
pos = np.concatenate((flipud(neg_pos[1:,:]),pos_pos),axis=0)
tot = np.concatenate((fliplr(neg),pos),axis=1)
neg_err = np.concatenate((flipud(neg_neg_err[1:,1:]),pos_neg_err[:,1:]),axis=0)
pos_err = np.concatenate((flipud(neg_pos_err[1:,:]),pos_pos_err),axis=0)
tot_err = np.concatenate((fliplr(neg_err),pos_err),axis=1)

xpts = linspace(0,6,7)
ypts = linspace(0,0.6,7)
xpts_neg_neg,ypts_neg_neg = np.meshgrid(-xpts,-ypts,indexing='ij')
xpts_neg_pos,ypts_neg_pos = np.meshgrid(-xpts,ypts,indexing='ij')
xpts_pos_neg,ypts_pos_neg = np.meshgrid(xpts,-ypts,indexing='ij')
xpts_pos_pos,ypts_pos_pos = np.meshgrid(xpts,ypts,indexing='ij')
xneg = np.concatenate((flipud(xpts_neg_neg[1:,1:]),xpts_pos_neg[:,1:]),axis=0)
yneg = np.concatenate((flipud(ypts_neg_neg[1:,1:]),ypts_pos_neg[:,1:]),axis=0)
xpos = np.concatenate((flipud(xpts_neg_pos[1:,:]),xpts_pos_pos),axis=0)
ypos = np.concatenate((flipud(ypts_neg_pos[1:,:]),ypts_pos_pos),axis=0)
xtot = np.concatenate((fliplr(xneg),xpos),axis=1)
ytot = np.concatenate((fliplr(yneg),ypos),axis=1)
```

Some of the fluorescence values may be bad due to bad camera exposures. This will show up in large error values in a contour plot of such. These entries can then be removed if they are too large.

```
cs = contour(xtot,ytot,tot_err,10)
grid(True)
clabel(cs, inline=1, fontsize=10)#, fmt=’%1.0e’)
xlabel(’X Current Set Point’)
ylabel(’Z Current Set Point’)
title(’Fluorescence Error Contour Map’)

imshow(tot_err.T,interpolation=’bilinear’, origin=’lower’)
xlabel(’X Current Set Point’)
ylabel(’Z Current Set Point’)
title(’Fluorescence Error Contour Image’)

cs = contour(xtot,ytot,tot,15,linewidth=3)
grid(True)
clabel(cs, inline=1, fontsize=15, fmt=’%1.0f’)
xlabel(’X Current Set Point’,fontsize=’xx-large’)
ylabel(’Z Current Set Point’,fontsize=’xx-large’)
#title(’Fluorescence Contour Map’,fontsize=’xx-large’)
figsize(11,8.5)
xticks(size = ’xx-large’)
yticks(size = ’xx-large’)
tick_params(’both’, length=12, width=2, which=’major’)
minorticks_on()
tick_params(’both’, length=5, width=2, which=’minor’)
axis([-5.999,5.999,-.5999,0.5999])
#savefig(’B_Trim_Contour_v0.pdf’, dpi=None, facecolor=’w’, edgecolor=’w’,
   format=’pdf’, pad_inches=0.1, frameon=None)

imshow(tot.T,interpolation=’bilinear’, origin=’lower’,extent=[-6,6,-0.6,0.6],
   aspect=’auto’)
colorbar()
xlabel(’X Current Set Point’,fontsize=’xx-large’)
ylabel(’Z Current Set Point’,fontsize=’xx-large’)
#title(’Fluorescence Heat Map’,fontsize=’xx-large’)
figsize(11,8.5)
xticks(size = ’xx-large’)
yticks(size = ’xx-large’)
```

```
tick_params('both', length=12, width=2, which='major')
minorticks_on()
tick_params('both', length=5, width=2, which='minor')
axis([-5.999,5.999,-.5999,0.5999])
#savefig('B_Trim_Heatmap_v0.pdf', dpi=None, facecolor='w', edgecolor='w',
   format='pdf', pad_inches=0.1, frameon=None)
```

E.1.7. NR Raman Cooling Frequency Analysis

The code below analyzes the fast, four-point temperature measurements as the NR Raman cooling frequency is scanned.

```
%pylab inline
import scipy.optimize as sc

file0 = r’C:\data\Rabi\Cool Freq\20160209_CWRamancool_FOR_CoolFreq_NS_scanPMT_1
    .npy’
file1 = r’C:\data\Rabi\Cool Freq\20160209_CWRamancool_FOR_CoolFreq_NS_scanPMT_2
    .npy’
file2 = r’C:\data\Rabi\Cool Freq\20160209_CWRamancool_FOR_CoolFreq_NS_scanPMT_3
    .npy’
file3 = r’C:\data\Rabi\Temperature Fast\20160208
    _CWRamancool_FOR_FastTemperature_NS_scanPMT_3.npy’
data0 = load(file0)
data1 = load(file1)
data2 = load(file2)
data3 = load(file3)

expose_t0 = 5.0e-3
expose_t1 = 5.0e-3
expose_t2 = 5.0e-3
expose_t3 = 5.0e-3

#Histogram discriminator check if needed. Only runs on data0 unless code below
    changed.
pruned0 = data0[:,2:]/expose_t0
pruned1 = data1[:,2:]/expose_t1
pruned2 = data2[:,2:]/expose_t2
pruned3 = data3[:,2:]/expose_t3
hist(pruned2.ravel(),50)
xlabel(r’PMT Count Rate (s$^{-1}$)’)
#axis([0,10000,0,100])

#Define ion bight/dark discriminator value and number of points in
#each scan range.
discr0 = 20000
discrh0 = 60000
```

```
discr1 = 20000
discrh1 = 60000

discr2 = 20000
discrh2 = 60000

discr3 = 20000
discrh3 = 60000

# Combine Datasets here if desired.
data3 = np.zeros(((data0.shape[0]+data1.shape[0]+data2.shape[0]),data0.shape
    [1]))
data3 = np.vstack((data0,data1,data2))

data0 = data0[np.lexsort((data0[:,1],data0[:,0]))]
data1 = data1[np.lexsort((data1[:,1],data1[:,0]))]
data2 = data2[np.lexsort((data2[:,1],data2[:,0]))]
data3 = data3[np.lexsort((data3[:,1],data3[:,0]))]

#Add a zero row to the bottom of each data matrix to fix a boundary
#issue in a for loop in data analysis.
data0 = np.vstack((data0,np.zeros((data0.shape[1]))))
data1 = np.vstack((data1,np.zeros((data1.shape[1]))))
data2 = np.vstack((data2,np.zeros((data2.shape[1]))))
data3 = np.vstack((data3,np.zeros((data3.shape[1]))))

#DO WORK. This section extracts frequencies and runs the loops to calculate
    bright
#probability and number of experiments with sigma giving stat error. It uses a
    clever
#indexing method for interleaved datapoints.
cft0 = np.zeros(((np.unique(data0[:,0]).shape[0]-1)*(np.unique(data0[:,1]).
    shape[0]-1)))
f0 = np.zeros(((np.unique(data0[:,0]).shape[0]-1)*(np.unique(data0[:,1]).shape
    [0]-1)))
res0 = np.zeros(((np.unique(data0[:,0]).shape[0]-1)*(np.unique(data0[:,1]).
    shape[0]-1)))
n0 = np.zeros(((np.unique(data0[:,0]).shape[0]-1)*(np.unique(data0[:,1]).shape
    [0]-1)))
j = 0
```

```
for i in range(data0.shape[0]-1):
    index = np.argwhere(np.logical_and(data0[i,0]==data0[:,0],data0[i,1]==data0
        [:,1])).transpose()[0]
    cft0[j] = data0[i,0]
    f0[j] = data0[i,1]
    res0[j] = sum(np.logical_and(data0[index,2:]/expose_t0 > discr0, data0[
        index,2:]/expose_t0 < discrh0))
    n0[j] = sum(np.logical_and(data0[index,2:]/expose_t0 > 0, data0[index,2:]/
        expose_t0 < discrh0))
    if np.logical_or(data0[i,0] != data0[i+1,0],data0[i,1] != data0[i+1,1]):
        j = j + 1
cft1 = np.zeros(((np.unique(data1[:,0]).shape[0]-1)*(np.unique(data1[:,1]).
    shape[0]-1)))
f1 = np.zeros(((np.unique(data1[:,0]).shape[0]-1)*(np.unique(data1[:,1]).shape
    [0]-1)))
res1 = np.zeros(((np.unique(data1[:,0]).shape[0]-1)*(np.unique(data1[:,1]).
    shape[0]-1)))
n1 = np.zeros(((np.unique(data1[:,0]).shape[0]-1)*(np.unique(data1[:,1]).shape
    [0]-1)))
j = 0
for i in range(data1.shape[0]-1):
    index = np.argwhere(np.logical_and(data1[i,0]==data1[:,0],data1[i,1]==data1
        [:,1])).transpose()[0]
    cft1[j] = data1[i,0]
    f1[j] = data1[i,1]
    res1[j] = sum(np.logical_and(data1[index,2:]/expose_t1 > discr1, data1[
        index,2:]/expose_t1 < discrh1))
    n1[j] = sum(np.logical_and(data1[index,2:]/expose_t1 > 0, data1[index,2:]/
        expose_t1 < discrh1))
    if np.logical_or(data1[i,0] != data1[i+1,0],data1[i,1] != data1[i+1,1]):
        j = j + 1
cft2 = np.zeros(((np.unique(data2[:,0]).shape[0]-1)*(np.unique(data2[:,1]).
    shape[0]-1)))
f2 = np.zeros(((np.unique(data2[:,0]).shape[0]-1)*(np.unique(data2[:,1]).shape
    [0]-1)))
res2 = np.zeros(((np.unique(data2[:,0]).shape[0]-1)*(np.unique(data2[:,1]).
    shape[0]-1)))
n2 = np.zeros(((np.unique(data2[:,0]).shape[0]-1)*(np.unique(data2[:,1]).shape
    [0]-1)))
j = 0
for i in range(data2.shape[0]-1):
    index = np.argwhere(np.logical_and(data2[i,0]==data2[:,0],data2[i,1]==data2
        [:,1])).transpose()[0]
    cft2[j] = data2[i,0]
    f2[j] = data2[i,1]
    res2[j] = sum(np.logical_and(data2[index,2:]/expose_t2 > discr2, data2[
        index,2:]/expose_t2 < discrh2))
    n2[j] = sum(np.logical_and(data2[index,2:]/expose_t2 > 0, data2[index,2:]/
        expose_t2 < discrh2))
    if np.logical_or(data2[i,0] != data2[i+1,0],data2[i,1] != data2[i+1,1]):
        j = j + 1
cft3 = np.zeros(((np.unique(data3[:,0]).shape[0]-1)*(np.unique(data3[:,1]).
    shape[0]-1)))
f3 = np.zeros(((np.unique(data3[:,0]).shape[0]-1)*(np.unique(data3[:,1]).shape
    [0]-1)))
res3 = np.zeros(((np.unique(data3[:,0]).shape[0]-1)*(np.unique(data3[:,1]).
    shape[0]-1)))
n3 = np.zeros(((np.unique(data3[:,0]).shape[0]-1)*(np.unique(data3[:,1]).shape
    [0]-1)))
j = 0
for i in range(data3.shape[0]-1):
    index = np.argwhere(np.logical_and(data3[i,0]==data3[:,0],data3[i,1]==data3
        [:,1])).transpose()[0]
    cft3[j] = data3[i,0]
    f3[j] = data3[i,1]
    res3[j] = sum(np.logical_and(data3[index,2:]/expose_t3 > discr3, data3[
        index,2:]/expose_t3 < discrh3))
    n3[j] = sum(np.logical_and(data3[index,2:]/expose_t3 > 0, data3[index,2:]/
        expose_t3 < discrh3))
    if np.logical_or(data3[i,0] != data3[i+1,0],data3[i,1] != data3[i+1,1]):
        j = j + 1

p0 = np.zeros((np.unique(data0[:,0]).shape[0]-1))
sigma0 = np.zeros((np.unique(data0[:,0]).shape[0]-1))
p0 = res0/n0
sigma0 = sqrt(p0*(1-p0)/n0)
p1 = np.zeros((np.unique(data1[:,0]).shape[0]-1))
sigma1 = np.zeros((np.unique(data1[:,0]).shape[0]-1))
p1 = res1/n1
sigma1 = sqrt(p1*(1-p1)/n1)
p2 = np.zeros((np.unique(data2[:,0]).shape[0]-1))
sigma2 = np.zeros((np.unique(data2[:,0]).shape[0]-1))
```

```
p2 = res2/n2
sigma2 = sqrt(p2*(1-p2)/n2)
p3 = np.zeros((np.unique(data3[:,0]).shape[0]-1))
sigma3 = np.zeros((np.unique(data3[:,0]).shape[0]-1))
p3 = res3/n3
sigma3 = sqrt(p3*(1-p3)/n3)

plt.errorbar(f0,p0,yerr=sigma0,color=’blue’,fmt=’o’)
plt.errorbar(f1,p1,yerr=sigma1,color=’green’,fmt=’o’)
plt.errorbar(f2,p2,yerr=sigma2,color=’red’,fmt=’o’)
#plt.errorbar(f3,p3,yerr=sigma3,color=’purple’,fmt=’o’)
#plt.scatter(f0,p0,color=’blue’)
#plt.plot(f0,p0,color=’blue’)
#plt.scatter(f1,p1,color=’green’)
#plt.plot(f1,p1,color=’green’)
#plt.scatter(f2,p2,color=’red’)
#plt.plot(f2,p2,color=’red’)
#plt.scatter(f3,p3,color=’purple’)
#plt.plot(f3,p3,color=’purple’)
plt.xlabel(’(Probe - Pump) Frequency (MHz)’)
plt.ylabel(’Bright Fraction’)
#plt.legend((r’-10 $\mu$s’,r’-12 $\mu$s’,r’-15 $\mu$s’,r’-20 $\mu$s’),loc=’
   upper right’)
#plt.legend((’RF=-3.4’,’RF=-3.2’,’RF=-3.0’,’RF=-2.8’),loc=’upper right’)
plt.title(r’FOR Raman Pi Pulses’)
plt.axis([-12.3,-9.6,0.05,0.6])
#plt.axis([-12.3,-11.9,0.04,0.57])
#plt.axis([-11.3,-10.65,0.05,0.6])
#plt.axis([-10.2,-9.5,0.05,0.32])
plt.grid(True)

try:
   del temp
except NameError:
   print
temp = np.zeros((cft0.shape[0],4))
temp[:,0] = cft0
temp[:,1] = f0
temp[:,2] = p0
temp[:,3] = sigma0
#temp
```

```
reps0 = 1.0*cft0.shape[0]/(np.unique(cft0).shape[0])
cf0 = np.zeros((np.unique(cft0).shape[0]))
rsb_Amp0 = np.zeros((np.unique(cft0).shape[0]))
delta_rsb_Amp0 = np.zeros((np.unique(cft0).shape[0]))
bsb_Amp0 = np.zeros((np.unique(cft0).shape[0]))
delta_bsb_Amp0 = np.zeros((np.unique(cft0).shape[0]))
if reps0.is_integer():
    for i in range((np.unique(cft0).shape[0])):
        cf0[i] = cft0[reps0*i]
        rsb_Amp0[i] = p0[reps0*i+0]-p0[reps0*i+1]
        delta_rsb_Amp0[i] = sqrt(sigma0[reps0*i+0]**2+sigma0[reps0*i+1]**2)
        bsb_Amp0[i] = p0[reps0*i+3]-p0[reps0*i+2]
        delta_bsb_Amp0[i] = sqrt(sigma0[reps0*i+3]**2+sigma0[reps0*i+2]**2)
else:
    print '0 FAIL'

reps1 = 1.0*cft1.shape[0]/(np.unique(cft1).shape[0])
cf1 = np.zeros((np.unique(cft1).shape[0]))
rsb_Amp1 = np.zeros((np.unique(cft1).shape[0]))
delta_rsb_Amp1 = np.zeros((np.unique(cft1).shape[0]))
bsb_Amp1 = np.zeros((np.unique(cft1).shape[0]))
delta_bsb_Amp1 = np.zeros((np.unique(cft1).shape[0]))
if reps1.is_integer():
    for i in range((np.unique(cft1).shape[0])):
        cf1[i] = cft1[reps1*i]
        rsb_Amp1[i] = p1[reps1*i+0]-p1[reps1*i+1]
        delta_rsb_Amp1[i] = sqrt(sigma1[reps1*i+0]**2+sigma1[reps1*i+1]**2)
        bsb_Amp1[i] = p1[reps1*i+3]-p1[reps1*i+2]
        delta_bsb_Amp1[i] = sqrt(sigma1[reps1*i+3]**2+sigma1[reps1*i+2]**2)
else:
    print '1 FAIL'

reps2 = 1.0*cft2.shape[0]/(np.unique(cft2).shape[0])
cf2 = np.zeros((np.unique(cft2).shape[0]))
rsb_Amp2 = np.zeros((np.unique(cft2).shape[0]))
delta_rsb_Amp2 = np.zeros((np.unique(cft2).shape[0]))
bsb_Amp2 = np.zeros((np.unique(cft2).shape[0]))
delta_bsb_Amp2 = np.zeros((np.unique(cft2).shape[0]))
if reps2.is_integer():
    for i in range((np.unique(cft2).shape[0])):
        cf2[i] = cft2[reps2*i]
        rsb_Amp2[i] = p2[reps2*i+0]-p2[reps2*i+1]
        delta_rsb_Amp2[i] = sqrt(sigma2[reps2*i+0]**2+sigma2[reps2*i+1]**2)
        bsb_Amp2[i] = p2[reps2*i+3]-p2[reps2*i+2]
        delta_bsb_Amp2[i] = sqrt(sigma2[reps2*i+3]**2+sigma2[reps2*i+2]**2)
else:
    print ’2 FAIL’

reps3 = 1.0*cft3.shape[0]/(np.unique(cft3).shape[0])
cf3 = np.zeros((np.unique(cft3).shape[0]))
rsb_Amp3 = np.zeros((np.unique(cft3).shape[0]))
delta_rsb_Amp3 = np.zeros((np.unique(cft3).shape[0]))
bsb_Amp3 = np.zeros((np.unique(cft3).shape[0]))
delta_bsb_Amp3 = np.zeros((np.unique(cft3).shape[0]))
if reps0.is_integer():
    for i in range((np.unique(cft3).shape[0])):
        cf3[i] = cft3[reps3*i]
        rsb_Amp3[i] = p3[reps3*i+0]-p3[reps3*i+1]
        delta_rsb_Amp3[i] = sqrt(sigma3[reps3*i+0]**2+sigma3[reps3*i+1]**2)
        bsb_Amp3[i] = p3[reps3*i+3]-p3[reps3*i+2]
        delta_bsb_Amp3[i] = sqrt(sigma3[reps3*i+3]**2+sigma3[reps3*i+2]**2)
else:
    print ’3 FAIL’

r0 = rsb_Amp0/bsb_Amp0
delta_r0 = r0*sqrt((delta_rsb_Amp0/rsb_Amp0)**2+(delta_bsb_Amp0/bsb_Amp0)**2)
nbar0 = r0/(1-r0)
delta_nbar0 = nbar0*sqrt(((1/(bsb_Amp0-rsb_Amp0))*delta_bsb_Amp0)**2+\
                     ((bsb_Amp0/(rsb_Amp0*(bsb_Amp0-rsb_Amp0)))*delta_rsb_Amp0)
                        **2)
r1 = rsb_Amp1/bsb_Amp1
delta_r1 = r1*sqrt((delta_rsb_Amp1/rsb_Amp1)**2+(delta_bsb_Amp1/bsb_Amp1)**2)
nbar1 = r1/(1-r1)
delta_nbar1 = nbar1*sqrt(((1/(bsb_Amp1-rsb_Amp1))*delta_bsb_Amp1)**2+\
                     ((bsb_Amp1/(rsb_Amp1*(bsb_Amp1-rsb_Amp1)))*delta_rsb_Amp1)
                        **2)
r2 = rsb_Amp2/bsb_Amp2
delta_r2 = r1*sqrt((delta_rsb_Amp2/rsb_Amp2)**2+(delta_bsb_Amp2/bsb_Amp2)**2)
nbar2 = r2/(1-r2)
delta_nbar2 = nbar2*sqrt(((1/(bsb_Amp2-rsb_Amp2))*delta_bsb_Amp2)**2+\
                     ((bsb_Amp2/(rsb_Amp2*(bsb_Amp2-rsb_Amp2)))*delta_rsb_Amp2)
                        **2)
r3 = rsb_Amp3/bsb_Amp3
delta_r3 = r1*sqrt((delta_rsb_Amp3/rsb_Amp3)**2+(delta_bsb_Amp3/bsb_Amp3)**2)
nbar3 = r3/(1-r3)
delta_nbar3 = nbar3*sqrt(((1/(bsb_Amp3-rsb_Amp3))*delta_bsb_Amp3)**2+\
                     ((bsb_Amp3/(rsb_Amp3*(bsb_Amp3-rsb_Amp3)))*delta_rsb_Amp3)
                        **2)

RemovePoints = False
if RemovePoints == True:
    for i in range(r0.shape[0]):
        if r0[i] < 0.0:
            r0[i] = np.nan
    for i in range(r1.shape[0]):
        if r1[i] < 0.0:
            r1[i] = np.nan
    for i in range(r2.shape[0]):
        if r2[i] < 0.0:
            r2[i] = np.nan
    for i in range(r3.shape[0]):
        if r3[i] < 0.0:
            r3[i] = np.nan

#plt.errorbar(cf0,r0,yerr=delta_r0,color=’blue’,fmt=’o’)
#plt.errorbar(cf1,r1,yerr=delta_r1,color=’green’,fmt=’o’)
#plt.errorbar(cf2,r2,yerr=delta_r2,color=’red’,fmt=’o’)
plt.errorbar(cf3,r3,yerr=delta_r3,color=’purple’,fmt=’o’)
#plt.scatter(cf0,r0,color=’blue’)
#plt.plot(cf0,r0,color=’blue’)
#plt.scatter(cf1,r1,color=’green’)
#plt.plot(cf1,r1,color=’green’)
#plt.scatter(cf2,r2,color=’red’)
#plt.plot(cf2,r2,color=’red’)
#plt.scatter(cf3,r3,color=’purple’)
#plt.plot(cf3,r3,color=’purple’)
plt.xlabel(’CW Raman Cooling (Probe - Pump) Frequency (MHz)’)
plt.ylabel(’RSB/BSB Amplitude Ratio’)
#plt.legend((r’-10 $\mu$s’,r’-12 $\mu$s’,r’-15 $\mu$s’,r’-20 $\mu$s’),loc=’
    upper right’)
#plt.legend((’RF=-3.4’,’RF=-3.2’,’RF=-3.0’,’RF=-2.8’),loc=’upper right’)
```

```
plt.title(r'CW Raman Cooling Frequency Scan')
#plt.axis([-12.3,-9.6,0.05,0.6])
#plt.axis([-12.3,-11.9,0.04,0.57])
#plt.axis([-11.3,-10.65,0.05,0.6])
plt.axis([-13.3,-12.2,-0.5,1.5])
plt.grid(True)

try:
    del temp
except NameError:
    print
temp = np.zeros((nbar3.shape[0],3))
temp[:,0] = cf3
temp[:,1] = r3
temp[:,2] = delta_r3
#temp
#np.save(r'C:\data\Rabi\Cool Freq\20160209
    _CWRamancool_FOR_CoolFreq_NS_scanPMT_1-3_Combined.npy',temp)
```

E.1.8. NR Raman Cooling Rate Analysis

The code below analyzes the NR Raman cooling rate data.

```
%pylab inline
import scipy.optimize as sc

file0 = r'C:\data\Rabi\Cool Time\20160210_CWRamancool_FOR_CoolTime_NS_scanPMT_0
    .npy'
file1 = r'C:\data\Rabi\Cool Time\20160210_CWRamancool_FOR_CoolTime_NS_scanPMT_1
    .npy'
file2 = r'C:\data\Rabi\Cool Time\20160210_CWRamancool_FOR_CoolTime_NS_scanPMT_2
    .npy'
file3 = r'C:\data\Rabi\Cool Time\20160210_CWRamancool_FOR_CoolTime_NS_scanPMT_3
    .npy'
data0 = load(file0)
data1 = load(file1)
data2 = load(file2)
data3 = load(file3)

expose_t0 = 5.0e-3
expose_t1 = 5.0e-3
expose_t2 = 5.0e-3
expose_t3 = 5.0e-3

#Histogram discriminator check if needed. Only runs on data0 unless code below
    changed.
pruned0 = data0[:,2:]/expose_t0
pruned1 = data1[:,2:]/expose_t1
pruned2 = data2[:,2:]/expose_t2
pruned3 = data3[:,2:]/expose_t3
hist(pruned0.ravel(),50)
xlabel(r'PMT Count Rate (s$^{-1}$)')
#axis([0,10000,0,100])

#Define ion bight/dark discriminator value and number of points in
#each scan range.
discr0 = 20000
discrh0 = 60000
```

```
discr1 = 20000
discrh1 = 60000

discr2 = 20000
discrh2 = 60000

discr3 = 20000
discrh3 = 60000

# Combine Datasets here if desired.
data3 = np.zeros(((data0.shape[0]+data1.shape[0]+data2.shape[0]),data0.shape
    [1]))
data3 = np.vstack((data0,data1,data2))

data0 = data0[np.lexsort((data0[:,1],data0[:,0]))]
data1 = data1[np.lexsort((data1[:,1],data1[:,0]))]
data2 = data2[np.lexsort((data2[:,1],data2[:,0]))]
data3 = data3[np.lexsort((data3[:,1],data3[:,0]))]

#Add a zero row to the bottom of each data matrix to fix a boundary
#issue in a for loop in data analysis.
data0 = np.vstack((data0,np.zeros((data0.shape[1]))))
data1 = np.vstack((data1,np.zeros((data1.shape[1]))))
data2 = np.vstack((data2,np.zeros((data2.shape[1]))))
data3 = np.vstack((data3,np.zeros((data3.shape[1]))))

#DO WORK. This section extracts frequencies and runs the loops to calculate
    bright
#probability and number of experiments with sigma giving stat error. It uses a
    clever
#indexing method for interleaved datapoints.
ctt0 = np.zeros(((np.unique(data0[:,0]).shape[0]-1)*(np.unique(data0[:,1]).
    shape[0]-1)))
f0 = np.zeros(((np.unique(data0[:,0]).shape[0]-1)*(np.unique(data0[:,1]).shape
    [0]-1)))
res0 = np.zeros(((np.unique(data0[:,0]).shape[0]-1)*(np.unique(data0[:,1]).
    shape[0]-1)))
n0 = np.zeros(((np.unique(data0[:,0]).shape[0]-1)*(np.unique(data0[:,1]).shape
    [0]-1)))
j = 0
for i in range(data0.shape[0]-1):
    index = np.argwhere(np.logical_and(data0[i,0]==data0[:,0],data0[i,1]==data0
        [:,1])).transpose()[0]
    ctt0[j] = data0[i,0]*1E0
    f0[j] = data0[i,1]
    res0[j] = sum(np.logical_and(data0[index,2:]/expose_t0 > discr0, data0[
        index,2:]/expose_t0 < discrh0))
    n0[j] = sum(np.logical_and(data0[index,2:]/expose_t0 > 0, data0[index,2:]/
        expose_t0 < discrh0))
    if np.logical_or(data0[i,0] != data0[i+1,0],data0[i,1] != data0[i+1,1]):
        j = j + 1
ctt1 = np.zeros(((np.unique(data1[:,0]).shape[0]-1)*(np.unique(data1[:,1]).
    shape[0]-1)))
f1 = np.zeros(((np.unique(data1[:,0]).shape[0]-1)*(np.unique(data1[:,1]).shape
    [0]-1)))
res1 = np.zeros(((np.unique(data1[:,0]).shape[0]-1)*(np.unique(data1[:,1]).
    shape[0]-1)))
n1 = np.zeros(((np.unique(data1[:,0]).shape[0]-1)*(np.unique(data1[:,1]).shape
    [0]-1)))
j = 0
for i in range(data1.shape[0]-1):
    index = np.argwhere(np.logical_and(data1[i,0]==data1[:,0],data1[i,1]==data1
        [:,1])).transpose()[0]
    ctt1[j] = data1[i,0]*1E0
    f1[j] = data1[i,1]
    res1[j] = sum(np.logical_and(data1[index,2:]/expose_t1 > discr1, data1[
        index,2:]/expose_t1 < discrh1))
    n1[j] = sum(np.logical_and(data1[index,2:]/expose_t1 > 0, data1[index,2:]/
        expose_t1 < discrh1))
    if np.logical_or(data1[i,0] != data1[i+1,0],data1[i,1] != data1[i+1,1]):
        j = j + 1
ctt2 = np.zeros(((np.unique(data2[:,0]).shape[0]-1)*(np.unique(data2[:,1]).
    shape[0]-1)))
f2 = np.zeros(((np.unique(data2[:,0]).shape[0]-1)*(np.unique(data2[:,1]).shape
    [0]-1)))
res2 = np.zeros(((np.unique(data2[:,0]).shape[0]-1)*(np.unique(data2[:,1]).
    shape[0]-1)))
n2 = np.zeros(((np.unique(data2[:,0]).shape[0]-1)*(np.unique(data2[:,1]).shape
    [0]-1)))
j = 0
for i in range(data2.shape[0]-1):
    index = np.argwhere(np.logical_and(data2[i,0]==data2[:,0],data2[i,1]==data2
        [:,1])).transpose()[0]
    ctt2[j] = data2[i,0]*1E0
    f2[j] = data2[i,1]
    res2[j] = sum(np.logical_and(data2[index,2:]/expose_t2 > discr2, data2[
        index,2:]/expose_t2 < discrh2))
    n2[j] = sum(np.logical_and(data2[index,2:]/expose_t2 > 0, data2[index,2:]/
        expose_t2 < discrh2))
    if np.logical_or(data2[i,0] != data2[i+1,0],data2[i,1] != data2[i+1,1]):
        j = j + 1
ctt3 = np.zeros(((np.unique(data3[:,0]).shape[0]-1)*(np.unique(data3[:,1]).
    shape[0]-1)))
f3 = np.zeros(((np.unique(data3[:,0]).shape[0]-1)*(np.unique(data3[:,1]).shape
    [0]-1)))
res3 = np.zeros(((np.unique(data3[:,0]).shape[0]-1)*(np.unique(data3[:,1]).
    shape[0]-1)))
n3 = np.zeros(((np.unique(data3[:,0]).shape[0]-1)*(np.unique(data3[:,1]).shape
    [0]-1)))
j = 0
for i in range(data3.shape[0]-1):
    index = np.argwhere(np.logical_and(data3[i,0]==data3[:,0],data3[i,1]==data3
        [:,1])).transpose()[0]
    ctt3[j] = data3[i,0]*1E0
    f3[j] = data3[i,1]
    res3[j] = sum(np.logical_and(data3[index,2:]/expose_t3 > discr3, data3[
        index,2:]/expose_t3 < discrh3))
    n3[j] = sum(np.logical_and(data3[index,2:]/expose_t3 > 0, data3[index,2:]/
        expose_t3 < discrh3))
    if np.logical_or(data3[i,0] != data3[i+1,0],data3[i,1] != data3[i+1,1]):
        j = j + 1

p0 = np.zeros((np.unique(data0[:,0]).shape[0]-1))
sigma0 = np.zeros((np.unique(data0[:,0]).shape[0]-1))
p0 = res0/n0
sigma0 = sqrt(p0*(1-p0)/n0)
p1 = np.zeros((np.unique(data1[:,0]).shape[0]-1))
sigma1 = np.zeros((np.unique(data1[:,0]).shape[0]-1))
p1 = res1/n1
sigma1 = sqrt(p1*(1-p1)/n1)
p2 = np.zeros((np.unique(data2[:,0]).shape[0]-1))
sigma2 = np.zeros((np.unique(data2[:,0]).shape[0]-1))
```

```
p2 = res2/n2
sigma2 = sqrt(p2*(1-p2)/n2)
p3 = np.zeros((np.unique(data3[:,0]).shape[0]-1))
sigma3 = np.zeros((np.unique(data3[:,0]).shape[0]-1))
p3 = res3/n3
sigma3 = sqrt(p3*(1-p3)/n3)

#plt.scatter(f0,p0,color='blue')
#plt.plot(f0,p0,color='blue')
plt.errorbar(f0,p0,yerr=sigma0,color='blue',fmt='o')
#plt.scatter(f1,p1,color='green')
#plt.plot(f1,p1,color='green')
plt.errorbar(f1,p1,yerr=sigma1,color='green',fmt='o')
#plt.scatter(f2,p2,color='red')
#plt.plot(f2,p2,color='red')
plt.errorbar(f2,p2,yerr=sigma2,color='red',fmt='o')
#plt.scatter(f3,p3,color='purple')
#plt.plot(f3,p3,color='purple')
#plt.errorbar(f3,p3,yerr=sigma3,color='purple')
plt.xlabel('FOR (Probe - Pump) Frequency (MHz)')
plt.ylabel('Bright Fraction')
#plt.legend((r'-10 $\mu$s',r'-12 $\mu$s',r'-15 $\mu$s',r'-20 $\mu$s'),loc='
   upper right')
#plt.legend(('RF=-3.4','RF=-3.2','RF=-3.0','RF=-2.8'),loc='upper right')
plt.title(r'FOR Raman Pi Pulses')
plt.axis([-12.3,-9.6,0.05,0.6])
#plt.axis([-12.3,-11.9,0.04,0.57])
#plt.axis([-11.3,-10.65,0.05,0.6])
#plt.axis([-10.2,-9.5,0.05,0.32])
plt.grid(True)

try:
   del temp
except NameError:
   print
temp = np.zeros((ctt0.shape[0],4))
temp[:,0] = ctt0
temp[:,1] = f0
temp[:,2] = p0
temp[:,3] = sigma0
#temp
```

```
reps0 = 1.0*ctt0.shape[0]/(np.unique(ctt0).shape[0])
ct0 = np.zeros((np.unique(ctt0).shape[0]))
rsb_Amp0 = np.zeros((np.unique(ctt0).shape[0]))
delta_rsb_Amp0 = np.zeros((np.unique(ctt0).shape[0]))
bsb_Amp0 = np.zeros((np.unique(ctt0).shape[0]))
delta_bsb_Amp0 = np.zeros((np.unique(ctt0).shape[0]))
if reps0.is_integer():
    for i in range((np.unique(ctt0).shape[0])):
        ct0[i] = ctt0[reps0*i]
        rsb_Amp0[i] = p0[reps0*i+0]-p0[reps0*i+1]
        delta_rsb_Amp0[i] = sqrt(sigma0[reps0*i+0]**2+sigma0[reps0*i+1]**2)
        bsb_Amp0[i] = p0[reps0*i+3]-p0[reps0*i+2]
        delta_bsb_Amp0[i] = sqrt(sigma0[reps0*i+3]**2+sigma0[reps0*i+2]**2)
else:
    print '0 FAIL'

reps1 = 1.0*ctt1.shape[0]/(np.unique(ctt1).shape[0])
ct1 = np.zeros((np.unique(ctt1).shape[0]))
rsb_Amp1 = np.zeros((np.unique(ctt1).shape[0]))
delta_rsb_Amp1 = np.zeros((np.unique(ctt1).shape[0]))
bsb_Amp1 = np.zeros((np.unique(ctt1).shape[0]))
delta_bsb_Amp1 = np.zeros((np.unique(ctt1).shape[0]))
if reps1.is_integer():
    for i in range((np.unique(ctt1).shape[0])):
        ct1[i] = ctt1[reps1*i]
        rsb_Amp1[i] = p1[reps1*i+0]-p1[reps1*i+1]
        delta_rsb_Amp1[i] = sqrt(sigma1[reps1*i+0]**2+sigma1[reps1*i+1]**2)
        bsb_Amp1[i] = p1[reps1*i+3]-p1[reps1*i+2]
        delta_bsb_Amp1[i] = sqrt(sigma1[reps1*i+3]**2+sigma1[reps1*i+2]**2)
else:
    print '1 FAIL'

reps2 = 1.0*ctt2.shape[0]/(np.unique(ctt2).shape[0])
ct2 = np.zeros((np.unique(ctt2).shape[0]))
rsb_Amp2 = np.zeros((np.unique(ctt2).shape[0]))
delta_rsb_Amp2 = np.zeros((np.unique(ctt2).shape[0]))
bsb_Amp2 = np.zeros((np.unique(ctt2).shape[0]))
delta_bsb_Amp2 = np.zeros((np.unique(ctt2).shape[0]))
if reps2.is_integer():
    for i in range((np.unique(ctt2).shape[0])):
        ct2[i] = ctt2[reps2*i]
        rsb_Amp2[i] = p2[reps2*i+0]-p2[reps2*i+1]
        delta_rsb_Amp2[i] = sqrt(sigma2[reps2*i+0]**2+sigma2[reps2*i+1]**2)
        bsb_Amp2[i] = p2[reps2*i+3]-p2[reps2*i+2]
        delta_bsb_Amp2[i] = sqrt(sigma2[reps2*i+3]**2+sigma2[reps2*i+2]**2)
else:
    print '2 FAIL'

reps3 = 1.0*ctt3.shape[0]/(np.unique(ctt3).shape[0])
ct3 = np.zeros((np.unique(ctt3).shape[0]))
rsb_Amp3 = np.zeros((np.unique(ctt3).shape[0]))
delta_rsb_Amp3 = np.zeros((np.unique(ctt3).shape[0]))
bsb_Amp3 = np.zeros((np.unique(ctt3).shape[0]))
delta_bsb_Amp3 = np.zeros((np.unique(ctt3).shape[0]))
if reps0.is_integer():
    for i in range((np.unique(ctt3).shape[0])):
        ct3[i] = ctt3[reps3*i]
        rsb_Amp3[i] = p3[reps3*i+0]-p3[reps3*i+1]
        delta_rsb_Amp3[i] = sqrt(sigma3[reps3*i+0]**2+sigma3[reps3*i+1]**2)
        bsb_Amp3[i] = p3[reps3*i+3]-p3[reps3*i+2]
        delta_bsb_Amp3[i] = sqrt(sigma3[reps3*i+3]**2+sigma3[reps3*i+2]**2)
else:
    print '3 FAIL'

r0 = rsb_Amp0/bsb_Amp0
delta_r0 = r0*sqrt((delta_rsb_Amp0/rsb_Amp0)**2+(delta_bsb_Amp0/bsb_Amp0)**2)
nbar0 = r0/(1-r0)
delta_nbar0 = nbar0*sqrt(((1/(bsb_Amp0-rsb_Amp0))*delta_bsb_Amp0)**2+\
                    ((bsb_Amp0/(rsb_Amp0*(bsb_Amp0-rsb_Amp0)))*delta_rsb_Amp0)
                        **2)
r1 = rsb_Amp1/bsb_Amp1
delta_r1 = r1*sqrt((delta_rsb_Amp1/rsb_Amp1)**2+(delta_bsb_Amp1/bsb_Amp1)**2)
nbar1 = r1/(1-r1)
delta_nbar1 = nbar1*sqrt(((1/(bsb_Amp1-rsb_Amp1))*delta_bsb_Amp1)**2+\
                    ((bsb_Amp1/(rsb_Amp1*(bsb_Amp1-rsb_Amp1)))*delta_rsb_Amp1)
                        **2)
r2 = rsb_Amp2/bsb_Amp2
delta_r2 = r2*sqrt((delta_rsb_Amp2/rsb_Amp2)**2+(delta_bsb_Amp2/bsb_Amp2)**2)
nbar2 = r2/(1-r2)
delta_nbar2 = nbar2*sqrt(((1/(bsb_Amp2-rsb_Amp2))*delta_bsb_Amp2)**2+\
                    ((bsb_Amp2/(rsb_Amp2*(bsb_Amp2-rsb_Amp2)))*delta_rsb_Amp2)
                        **2)
r3 = rsb_Amp3/bsb_Amp3
delta_r3 = r3*sqrt((delta_rsb_Amp3/rsb_Amp3)**2+(delta_bsb_Amp3/bsb_Amp3)**2)
nbar3 = r3/(1-r3)
delta_nbar3 = nbar3*sqrt(((1/(bsb_Amp3-rsb_Amp3))*delta_bsb_Amp3)**2+\
                    ((bsb_Amp3/(rsb_Amp3*(bsb_Amp3-rsb_Amp3)))*delta_rsb_Amp3)
                        **2)

RemovePoints = False
if RemovePoints == True:
    for i in range(nbar0.shape[0]):
        if r0[i] < 0.0:
            r0[i] = np.nan
    for i in range(nbar1.shape[0]):
        if r1[i] < 0.0:
            r1[i] = np.nan
    for i in range(nbar2.shape[0]):
        if r2[i] < 0.0:
            r2[i] = np.nan
    for i in range(nbar3.shape[0]):
        if r3[i] < 0.0:
            r3[i] = np.nan

#plt.errorbar(ct0,r0,yerr=delta_r0,color='blue',fmt='o')
#plt.errorbar(ct1,r1,yerr=delta_r1,color='green',fmt='o')
#plt.errorbar(ct2,r2,yerr=delta_r2,color='red',fmt='o')
plt.errorbar(ct3,r3,yerr=delta_r3,color='purple',fmt='o')
#plt.scatter(ct0,nbar0,color='blue')
#plt.plot(ct0,nbar0,color='blue')
#plt.scatter(ct1,nbar1,color='green')
#plt.plot(ct1,nbar1,color='green')
#plt.scatter(ct2,nbar2,color='red')
#plt.plot(ct2,nbar2,color='red')
#plt.scatter(ct3,nbar3,color='purple')
#plt.plot(ct3,nbar3,color='purple')
plt.semilogx(0,0)
plt.xlabel('CW Raman Cooling Time (s)')
plt.ylabel('RSB/BSB Amplitude Ratio')
#plt.legend((r'-10 $\mu$s',r'-12 $\mu$s',r'-15 $\mu$s',r'-20 $\mu$s'),loc='
    upper right')
```

```
#plt.legend(('RF=-3.4','RF=-3.2','RF=-3.0','RF=-2.8'),loc='upper right')
plt.title(r'CW Raman Cooling Time Scan')
#plt.axis([-1,31,-1,2.5])
plt.axis([8e-5,40e-3,-0.1,1.1])
#plt.axis([-1e-3,11e-3,-.5,1.5])
plt.grid(True)
```

E.1.9. NR Raman Sideband Cooling Model

The code below models the NR Raman sideband cooling equilibrium $\bar{n}$ and $\frac{\mathrm{d}}{\mathrm{d}t}\bar{n}$ as a function of two-photon detuning δ.

```
%pylab inline

#Define Constants and Parameters with frequencies in MHz and wavelength in nm
#Additionally, parameters are for 138-Ba+ and 493 nm.
hbar = (1.0546E-34)*(1.0/(1.66E-27))*(1.0/(1.0E-9))**2*(1.0/(1.0E6))
m = 138
Gamma = 2*pi*15.2
k = 2*pi/493.0
Omega1 = 0.05*Gamma
Omega2 = Gamma
Delta1 = -2.0*pi*80.0
Deltap = Omega2**2/(4.0*Delta1)
Gamma1 = Gamma/3.0
Gamma2 = 2.0*Gamma/3.0
alpha = 1.0
omegaz = 2.0*pi*1.15
eta = sqrt(hbar*(sqrt(2)*k)**2/(2*m*omegaz))

#Define the steady-state excited state population
def p33(x):
    return Omega1**2*x**2*(Gamma/Gamma1)/(4*Delta1**2*(x-Deltap)**2+x**2*Gamma
       **2+4*x**2*Delta1**2*(Omega1**2/Omega2**2)*(Gamma2/Gamma1)+(Omega1**2*
       Omega2**2/4)*((Gamma2/Gamma1)+2))

#Define the steady-state excited state population
def nzbar(x):
    return (p33(x-omegaz)+alpha*p33(x))/(p33(x+omegaz)-p33(x-omegaz))

def rate(x):
    return eta**2*Gamma1*(p33(x+omegaz)-p33(x-omegaz))

figsize(18/1.5,8.5/1.5)
delta = 2*pi*linspace(-2.5,0,1000)
plt.plot(delta/(2*pi),nzbar(delta),linewidth=3)
plt.legend(loc="upper right", fontsize='xx-large')
plt.xlabel('$\delta/(2\pi)$ (MHz))',fontsize='xx-large')
plt.ylabel(r'$\bar{n_z}$',fontsize='xx-large')
plt.axis([-2.5,-1.6,0,2])
xticks(size = 'xx-large')
yticks(size = 'xx-large')
tick_params('both', length=12, width=2, which='major')
minorticks_on()
tick_params('both', length=5, width=2, which='minor')
#savefig('201308225_Laser.pdf', dpi=None, facecolor='w', edgecolor='w', format
    ='pdf', pad_inches=0.1, frameon=None,bbox_inches='tight')

figsize(18/1.5,8.5/1.5)
delta = 2*pi*linspace(-2.5,0,1000)
plt.plot(delta/(2*pi),rate(delta)*1E3,linewidth=3)
plt.legend(loc="upper right", fontsize='xx-large')
plt.xlabel('$\delta/(2\pi)$ (MHz)',fontsize='xx-large')
plt.ylabel(r'$\frac{\mathrm{d}}{\mathrm{d}t}\bar{n_z}$',fontsize='xx-large')
xticks(size = 'xx-large')
yticks(size = 'xx-large')
tick_params('both', length=12, width=2, which='major')
minorticks_on()
tick_params('both', length=5, width=2, which='minor')
#savefig('201308225_Laser.pdf', dpi=None, facecolor='w', edgecolor='w', format
    ='pdf', pad_inches=0.1, frameon=None,bbox_inches='tight')

figsize(8.5,11)
f, (ax1, ax2) = plt.subplots(2, 1, sharex=True)
delta = 2*pi*linspace(-2.5,0,1000)
ax1.plot(delta/(2*pi),nzbar(delta),linewidth=4)
ax1.plot([-1.88,-1.88],[-1,3],color='black',linestyle='--',linewidth=3)
ax1.axis([-2.5,-1.6,-0.1,2])
ax1.set_ylabel(r'$\bar{n_z}$',fontsize='xx-large')
ax1.tick_params(axis='x', labelsize='xx-large')
ax1.tick_params(axis='y', labelsize='xx-large')
ax1.tick_params('both', length=12, width=2, which='major')
ax1.minorticks_on()
ax1.tick_params('both', length=5, width=2, which='minor')
ax1.text(-2.05,0.75,'$\delta/(2\pi)=-1.88$ MHz\n'+r'$\bar{n_z}_\mathrm{min
    }=0.11$',color='blue', horizontalalignment='center',verticalalignment='
    center',size=18)
ax2.plot(delta/(2*pi),rate(delta)*1E6,color='red',linewidth=4)
```

```
ax2.plot([-1.88,-1.88],[-10,1200],color='black',linestyle='--',linewidth=3)
ax2.axis([-2.5,-1.6,-5,1100])
ax2.set_xlabel('$\delta/(2\pi)$ (MHz))',fontsize='xx-large')
ax2.set_ylabel(r'$\frac{\mathrm{d}}{\mathrm{d}t}\bar{n_z}$ (quanta/s)',fontsize
    ='xx-large')
ax2.tick_params(axis='x', labelsize='xx-large')
ax2.tick_params(axis='y', labelsize='xx-large')
ax2.tick_params('both', length=12, width=2, which='major')
ax2.minorticks_on()
ax2.tick_params('both', length=5, width=2, which='minor')
ax2.text(-2.05,900,'$\delta/(2\pi)=-1.88$ MHz\n'+r'$\frac{\mathrm{d}}{\mathrm{d
    }t}\bar{n_z}=890$ s$^{-1}$',color='red', horizontalalignment='center',
    verticalalignment='center',size=18)
f.subplots_adjust(hspace=0.02)
#savefig('NRRamanModeling.pdf', dpi=None, facecolor='w', edgecolor='w', format
    ='pdf', pad_inches=0.1, frameon=None,bbox_inches='tight')

delta = 2*pi*linspace(-2.5,-1.6,1000)
m = np.min(nzbar(delta))
l = [i for i, j in enumerate(nzbar(delta)) if j == m]
[m,delta[l][0]/(2*pi)]

delta = 2*pi*linspace(-2.5,-1.6,1000)
m = np.max(rate(delta))
l = [i for i, j in enumerate(rate(delta)) if j == m]
[m*1E6,delta[l][0]/(2*pi)]

#Update with NR Raman results.
Omega1 = 2.0*pi*1.07
Omega2 = 2.0*pi*14.9
Delta1 = -2.0*pi*79.0
Deltap = Omega2**2/(4.0*Delta1)
omegaz = 2.0*pi*1.149
eta = sqrt(hbar*(sqrt(2)*k)**2/(2*m*omegaz))

figsize(8.5,11)
f, (ax1, ax2) = plt.subplots(2, 1, sharex=True)
delta = 2*pi*linspace(-2.5,0,1000)
ax1.plot(delta/(2*pi),nzbar(delta),linewidth=4)
ax1.plot([-1.87,-1.87],[-1,3],color='black',linestyle='--',linewidth=3)
ax1.axis([-2.5,-1.6,0.0,2])
```

```
ax1.set_ylabel(r’$\bar{n_z}$’,fontsize=’xx-large’)
ax1.tick_params(axis=’x’, labelsize=’xx-large’)
ax1.tick_params(axis=’y’, labelsize=’xx-large’)
ax1.tick_params(’both’, length=12, width=2, which=’major’)
ax1.minorticks_on()
ax1.tick_params(’both’, length=5, width=2, which=’minor’)
ax1.text(-2.05,0.75,’$\delta/(2\pi)=-1.87$ MHz\n’+r’$\bar{n_z}_\mathrm{min
    }=0.17$’,color=’blue’, horizontalalignment=’center’,verticalalignment=’
    center’,size=18)

ax2.plot(delta/(2*pi),rate(delta)*1E6,color=’red’,linewidth=4)
ax2.plot([-1.87,-1.87],[-10,1250],color=’black’,linestyle=’--’,linewidth=3)
ax2.axis([-2.5,-1.6,-5,1250])
ax2.set_xlabel(’$\delta/(2\pi)$ (MHz))’,fontsize=’xx-large’)
ax2.set_ylabel(r’$\frac{\mathrm{d}}{\mathrm{d}t}\bar{n_z}$ (quanta/s)’,fontsize
    =’xx-large’)
ax2.tick_params(axis=’x’, labelsize=’xx-large’)
ax2.tick_params(axis=’y’, labelsize=’xx-large’)
ax2.tick_params(’both’, length=12, width=2, which=’major’)
ax2.minorticks_on()
ax2.tick_params(’both’, length=5, width=2, which=’minor’)
ax2.text(-2.05,1050,’$\delta/(2\pi)=-1.87$ MHz\n’+r’$\frac{\mathrm{d}}{\mathrm{
    d}t}\bar{n_z}=1070$ s$^{-1}$’,color=’red’, horizontalalignment=’center’,
    verticalalignment=’center’,size=18)

f.subplots_adjust(hspace=0.02)
#savefig(’NRRamanModelingwNRResults.pdf’, dpi=None, facecolor=’w’, edgecolor=’w
    ’, format=’pdf’, pad_inches=0.1, frameon=None,bbox_inches=’tight’)

delta = 2*pi*linspace(-2.5,-1.6,1000)
m = np.min(nzbar(delta))
l = [i for i, j in enumerate(nzbar(delta)) if j == m]
[m,delta[l][0]/(2*pi)]

delta = 2*pi*linspace(-2.5,-1.6,1000)
m = np.max(rate(delta))
l = [i for i, j in enumerate(rate(delta)) if j == m]
[m*1E6,delta[l][0]/(2*pi)]
```

E.1.10. NR Raman Spectra Analysis

The code below analyzes the NR Raman spectra taken by the code found in App. E.3.1.

```
%pylab inline
import scipy.optimize as sc

#data0 = load(r'C:\data\CW Raman Fluorescence\20160124_CWRamanscanPMT_0.npy')
#CW Raman
data0 = load(r'C:\data\CW Raman Fluorescence\20160221_CWRamanscanPMT_NS_0.npy')
data1 = load(r'C:\data\CW Raman Fluorescence\20160401_CWRamanscanPMT_NS_3.npy')
data2 = load(r'C:\data\CW Raman Fluorescence\20160401_CWRamanscanPMT_NS_0.npy')
data3 = load(r'C:\data\CW Raman Fluorescence\20160401_CWRamanscanPMT_NS_1.npy')

#data0t = np.vstack((data0,data1,data2,data3))
#data1t = np.vstack((data5,data6,data7,data8))
data2t = np.vstack((data0,data1))
#data3t = np.vstack((data13,data14,data15,data16))

#del (data0,data1,data2,data3)
#data0 = data0t
#data1 = data1t
data2 = data2t
#data3 = data3t

#Processed Data
data4 = load(r'C:\data\CW Raman Fluorescence\20160223_CWRamanscanPMT_Processed.
   npy')
f4 = data4[:,0]
p4 = data4[:,1]
err4 = data4[:,2]
n4 = data4[:,3]

expose_t0 = 1.0e-3
expose_t1 = 1.0e-3
expose_t2 = 1.0e-3
expose_t3 = 1.0e-3

#Sort data by first column, i.e. frequency.
data0 = data0[data0[:,0].argsort()]
```

```
data1 = data1[data1[:,0].argsort()]
data2 = data2[data2[:,0].argsort()]
data3 = data3[data3[:,0].argsort()]

#Add a zero row to the bottom of each data matrix to fix a boundary
#issue in a for loop in data analysis.
#data0 = np.vstack((data0,np.zeros((data0.shape[1]))))
#data1 = np.vstack((data1,np.zeros((data1.shape[1]))))
#data2 = np.vstack((data2,np.zeros((data2.shape[1]))))
#data3 = np.vstack((data3,np.zeros((data3.shape[1]))))

#DO WORK. This section extracts frequencies and runs the loops to calculate
    mean
#fluorescence and number of experiments with sigma given by standard dev of the
     mean.
#It uses a clever indexing method for interleaved datapoints.
f0 = np.zeros((np.unique(data0[:,0]).shape[0]))
p0 = np.zeros((np.unique(data0[:,0]).shape[0]))
err0 = np.zeros((np.unique(data0[:,0]).shape[0]))
n0 = np.zeros((np.unique(data0[:,0]).shape[0]))
j = 0
for i in range(data0.shape[0]-1):
    index = (np.argwhere(data0[:,0]==data0[i,0]).transpose()[0])
    f0[j] = data0[i,0]
    p0[j] = np.average(data0[index,1:])/expose_t0
    err0[j] = np.std(data0[index,1:])/expose_t0/sqrt(data0[index,1:].size)
    n0[j] = data0[index,1:].size
    if data0[i,0] != data0[i+1,0]:
        j = j + 1
f1 = np.zeros((np.unique(data1[:,0]).shape[0]))
p1 = np.zeros((np.unique(data1[:,0]).shape[0]))
err1 = np.zeros((np.unique(data1[:,0]).shape[0]))
n1 = np.zeros((np.unique(data1[:,0]).shape[0]))
j = 0
for i in range(data1.shape[0]-1):
    index = (np.argwhere(data1[:,0]==data1[i,0]).transpose()[0])
    f1[j] = data1[i,0]
    p1[j] = np.average(data1[index,1:])/expose_t1
    err1[j] = np.std(data1[index,1:])/expose_t1/sqrt(data1[index,1:].size)
    n1[j] = data1[index,1:].size
    if data1[i,0] != data1[i+1,0]:
        j = j + 1
f2 = np.zeros((np.unique(data2[:,0]).shape[0]))
p2 = np.zeros((np.unique(data2[:,0]).shape[0]))
err2 = np.zeros((np.unique(data2[:,0]).shape[0]))
n2 = np.zeros((np.unique(data2[:,0]).shape[0]))
j = 0
for i in range(data2.shape[0]-1):
    index = (np.argwhere(data2[:,0]==data2[i,0]).transpose()[0])
    f2[j] = data2[i,0]
    p2[j] = np.average(data2[index,1:])/expose_t2
    err2[j] = np.std(data2[index,1:])/expose_t2/sqrt(data2[index,1:].size)
    n2[j] = data2[index,1:].size
    if data2[i,0] != data2[i+1,0]:
        j = j + 1
f3 = np.zeros((np.unique(data3[:,0]).shape[0]))
p3 = np.zeros((np.unique(data3[:,0]).shape[0]))
err3 = np.zeros((np.unique(data3[:,0]).shape[0]))
n3 = np.zeros((np.unique(data3[:,0]).shape[0]))
j = 0
for i in range(data3.shape[0]-1):
    index = (np.argwhere(data3[:,0]==data3[i,0]).transpose()[0])
    f3[j] = data3[i,0]
    p3[j] = np.average(data3[index,1:])/expose_t3
    err3[j] = np.std(data3[index,1:])/expose_t3/sqrt(data3[index,1:].size)
    n3[j] = data3[index,1:].size
    if data3[i,0] != data3[i+1,0]:
        j = j + 1

#Fix the boundary condition.
f0 = np.delete(f0,f0.shape[0]-1)
p0 = np.delete(p0,p0.shape[0]-1)
err0 = np.delete(err0,err0.shape[0]-1)
n0 = np.delete(n0,n0.shape[0]-1)
f1 = np.delete(f1,f1.shape[0]-1)
p1 = np.delete(p1,p1.shape[0]-1)
err1 = np.delete(err1,err1.shape[0]-1)
n1 = np.delete(n1,n1.shape[0]-1)
f2 = np.delete(f2,f2.shape[0]-1)
p2 = np.delete(p2,p2.shape[0]-1)
err2 = np.delete(err2,err2.shape[0]-1)
n2 = np.delete(n2,n2.shape[0]-1)
```

```
f3 = np.delete(f3,f3.shape[0]-1)
p3 = np.delete(p3,p3.shape[0]-1)
err3 = np.delete(err3,err3.shape[0]-1)
n3 = np.delete(n3,n3.shape[0]-1)

showhist = False
if showhist == True:
    pruned0 = data0[:,1:]/expose_t0
    pruned1 = data1[:,1:]/expose_t1
    pruned2 = data2[:,1:]/expose_t2
    pruned3 = data3[:,1:]/expose_t3
    hist(pruned0.ravel(),30)
    xlabel(r'PMT Count Rate (s$^{-1}$)')

#Provision to offset data runs to deal with any background changes.
bg = False
normalizedata = False
if bg == True:
    bg0 = np.nanmin(p0)
    bg1 = np.nanmin(p1)
    bg2 = np.nanmin(p2)
    bg3 = np.nanmin(p3)
    p0 = p0 - bg0
    p1 = p1 - bg1
    p2 = p2 - bg2
    p3 = p3 - bg3
if normalizedata == True:
    err0 = err0/(np.nanmax(p0))
    p0 = (p0)/(np.nanmax(p0))
    err1 = err1/(np.nanmax(p1))
    p1 = (p1)/(np.nanmax(p1))
    err2 = err2/(np.nanmax(p2))
    p2 = (p2)/(np.nanmax(p2))
    err3 = err3/(np.nanmax(p3))
    p3 = (p3)/(np.nanmax(p3))

#Scale all results
#scale = 1E5
scale = 0
if scale != 0:
    p0 = p0/scale
    err0 = err0/scale
    p1 = p1/scale
    err1 = err1/scale
    p2 = p2/scale
    err2 = err2/scale
    p3 = p3/scale
    err3 = err3/scale

ZeroOffset = False
if ZeroOffset == True:
    p0 = p0 - np.nanmean(p0)
    p1 = p1 - np.nanmean(p1)
    p2 = p2 - np.nanmean(p2)
    p3 = p3 - np.nanmean(p3)

BadPoints = False
if BadPoints == True:
    for i in range(p0.shape[0]):
        if err0[i] < 200:
            p0[i] = p0[i]
        else:
            p0[i] = NaN
    for i in range(p1.shape[0]):
        if p1[i] > 300:
            p1[i] = p1[i]
        else:
            p1[i] = NaN
    for i in range(p2.shape[0]):
        if err2[i] < 200:
            p2[i] = p2[i]
        else:
            p2[i] = NaN
    for i in range(p3.shape[0]):
        if err3[i] < 200:
            p3[i] = p3[i]
        else:
            p3[i] = NaN

#try:
#   del(f4,p4,err4,n4)
#except NameError:
```

```
#   print
#f4 = f0
#p4 = p0-p1-p2+p3
#err4 = sqrt(err0**2+err1**2+err2**2+err3**2)
#n4 = n0+n1+n2+n3

#ChangeLog, 20150603: f0 --> -2*f0 to reflect real probe-pump frequency
#plt.plot(f0,p0,color='blue',linewidth=2)
plt.errorbar(f0,p0,yerr=err0,color='blue',fmt='-o')
#plt.errorbar(f1,p1,yerr=err1,color='green',fmt='-o')
#plt.errorbar(f2,p2,yerr=err2,color='red',fmt='-o')
#plt.errorbar(f3,p3,yerr=err3,color='purple',fmt='-o')
#plt.errorbar(f4,p4,yerr=err4,color='black',fmt='-o')
#plt.scatter(f0,p0,color='blue')
#plt.scatter(f1,p1,color='green')
#plt.scatter(f2,p2,color='red')
#plt.scatter(f3,p3,color='purple')
plt.xlabel('(Probe - Pump) Frequency (MHz)')
plt.ylabel(r'Average PMT Count Rate (s$^{-1}$)')
#plt.ylabel('Normalized PMT Count Rate')
plt.title('CW Raman Spectrum')
#plt.title(r'20151229 CW Raman Spectrum, 1.3 $\mu$W Pump')
#plt.legend((r'16 $\mu$W Probe',r'1.28 $\mu$W Probe',r'0.89 $\mu$W Probe',r
    '0.53 $\mu$W Probe'),loc='upper Left')
#plt.legend(('NR On','NR $\sigma$ Bkgnd','NR $\pi$ Bkgnd','PMT Bkgnd'),loc='
    upper right')
#axis([-50,0,400,950])
#axis([-15.1,-7.9,400,1200])
plt.grid(True)

m = max(p0)
k = [i for i, j in enumerate(p0) if j == m]
f0[k]
#p3[k]
#err3[k]
```

E.1.11. Optical Pumping and Protect Time Analysis

The code below analyzes and plots the bright fraction as a function of optical pumping time.

```
%pylab inline
import scipy.optimize as sc

file0 = r'C:\data\Rabi\OP\20160202_CWRamancool_FOR_OP_NS_scanPMT_0.npy'
file1 = r'C:\data\Rabi\OP\20160202_CWRamancool_FOR_OP_NS_scanPMT_1.npy'
file2 = r'C:\data\Rabi\OP\20160202_CWRamancool_FOR_OP_NS_scanPMT_2.npy'
file3 = r'C:\data\Rabi\OP\20160201_CWRamancool_FOR_OP_NS_scanPMT_4.npy'
data0 = load(file0)
data1 = load(file1)
data2 = load(file2)
data3 = load(file3)

expose_t0 = 5.0e-3
expose_t1 = 5.0e-3
expose_t2 = 5.0e-3
expose_t3 = 5.0e-3

#Histogram discriminator check if needed. Only runs on data0 unless code below
    changed.
pruned0 = data0[:,1:]/expose_t0
pruned1 = data1[:,1:]/expose_t1
pruned2 = data2[:,1:]/expose_t2
pruned3 = data3[:,1:]/expose_t3
hist(pruned1.ravel(),50)
xlabel(r'PMT Count Rate (s$^{-1}$)')
#axis([0,20000,0,100])

#Define ion bight/dark discriminator value and number of points in
#each scan range.
discr0 = 20000
discrh0 = 60000

discr1 = 20000
discrh1 = 60000
```

```
discr2 = 20000
discrh2 = 60000

discr3 = 20000
discrh3 = 60000

data0 = data0[data0[:,0].argsort()]
data1 = data1[data1[:,0].argsort()]
data2 = data2[data2[:,0].argsort()]
data3 = data3[data3[:,0].argsort()]

#DO WORK. This section extracts frequencies and runs the loops to calculate
   bright probability and number of experiments.
#define number of events and event probability with sigma giving stat error.
res0 = zeros(data0.shape[0])
pt0 = zeros(data0.shape[0])
n0 = zeros(data0.shape[0])
p0 = zeros(data0.shape[1])
for i in range(data0.shape[0]):
   pt0[i] = data0[i,0]
   res0[i] = sum(np.logical_and(data0[i,1:]/expose_t0 > discr0, data0[i,1:]/
      expose_t0 < discrh0))
   n0[i] = sum(np.logical_and(data0[i,1:]/expose_t0 > 0, data0[i,1:]/expose_t0
       < discrh0))
p0 = res0/n0
sigma0 = sqrt(p0*(1-p0)/n0)

res1 = zeros(data1.shape[0])
pt1 = zeros(data1.shape[0])
n1 = zeros(data1.shape[0])
p1 = zeros(data1.shape[1])
for i in range(data1.shape[0]):
   pt1[i] = data1[i,0]
   res1[i] = sum(np.logical_and(data1[i,1:]/expose_t1 > discr1, data1[i,1:]/
      expose_t1 < discrh1))
   n1[i] = sum(np.logical_and(data1[i,1:]/expose_t1 > 0, data1[i,1:]/expose_t1
       < discrh1))
p1 = res1/n1
sigma1 = sqrt(p1*(1-p1)/n1)

res2 = zeros(data2.shape[0])
```

```
pt2 = zeros(data2.shape[0])
n2 = zeros(data2.shape[0])
p2 = zeros(data2.shape[1])
for i in range(data2.shape[0]):
    pt2[i] = data2[i,0]
    res2[i] = sum(np.logical_and(data2[i,1:]/expose_t2 > discr2, data2[i,1:]/
        expose_t2 < discrh2))
    n2[i] = sum(np.logical_and(data2[i,1:]/expose_t2 > 0, data2[i,1:]/expose_t2
         < discrh2))
p2 = res2/n2
sigma2 = sqrt(p2*(1-p2)/n2)

res3 = zeros(data3.shape[0])
pt3 = zeros(data3.shape[0])
n3 = zeros(data3.shape[0])
p3 = zeros(data3.shape[1])
for i in range(data3.shape[0]):
    pt3[i] = data3[i,0]
    res3[i] = sum(np.logical_and(data3[i,1:]/expose_t3 > discr3, data3[i,1:]/
        expose_t3 < discrh3))
    n3[i] = sum(np.logical_and(data3[i,1:]/expose_t3 > 0, data3[i,1:]/expose_t3
         < discrh3))
p3 = res3/n3
sigma3 = sqrt(p3*(1-p3)/n3)

#Combine same frequency points!
temp = np.zeros((np.unique(pt0).shape[0],3))
j = 0
temp[0,0] = pt0[0]
temp[0,1] = p0[0]*n0[0]
temp[0,2] = n0[0]
for i in range(pt0.shape[0]-1):
    if pt0[i] == pt0[i+1]:
        temp[j,0] = pt0[i]
        temp[j,1] = temp[j,1]+p0[i+1]*n0[i+1]
        temp[j,2] = temp[j,2]+n0[i+1]
    else:
        j = j + 1
        temp[j,0] = pt0[i]
        temp[j,1] = p0[i]*n0[i]
        temp[j,2] = n0[i]
        if i == (pt0.shape[0]-2):
            temp[j,0] = pt0[i+1]
            temp[j,1] = p0[i+1]*n0[i+1]
            temp[j,2] = n0[i+1]
del (pt0,p0,n0,sigma0)
pt0 = temp[:,0]
p0 = temp[:,1]/temp[:,2]
n0 = temp[:,2]
sigma0 = sqrt(p0*(1-p0)/n0)

temp = np.zeros((np.unique(pt1).shape[0],3))
j = 0
temp[0,0] = pt1[0]
temp[0,1] = p1[0]*n1[0]
temp[0,2] = n1[0]
for i in range(pt1.shape[0]-1):
    if pt1[i] == pt1[i+1]:
        temp[j,0] = pt1[i]
        temp[j,1] = temp[j,1]+p1[i+1]*n1[i+1]
        temp[j,2] = temp[j,2]+n1[i+1]
    else:
        j = j + 1
        temp[j,0] = pt1[i]
        temp[j,1] = p1[i]*n1[i]
        temp[j,2] = n1[i]
        if i == (pt1.shape[0]-2):
            temp[j,0] = pt1[i+1]
            temp[j,1] = p1[i+1]*n1[i+1]
            temp[j,2] = n1[i+1]
del (pt1,p1,n1,sigma1)
pt1 = temp[:,0]
p1 = temp[:,1]/temp[:,2]
n1 = temp[:,2]
sigma1 = sqrt(p1*(1-p1)/n1)

temp = np.zeros((np.unique(pt2).shape[0],3))
j = 0
temp[0,0] = pt2[0]
temp[0,1] = p2[0]*n2[0]
temp[0,2] = n2[0]
for i in range(pt2.shape[0]-1):
    if pt2[i] == pt2[i+1]:
        temp[j,0] = pt2[i]
        temp[j,1] = temp[j,1]+p2[i+1]*n2[i+1]
        temp[j,2] = temp[j,2]+n2[i+1]
    else:
        j = j + 1
        temp[j,0] = pt2[i]
        temp[j,1] = p2[i]*n2[i]
        temp[j,2] = n2[i]
        if i == (pt2.shape[0]-2):
            temp[j,0] = pt2[i+1]
            temp[j,1] = p2[i+1]*n2[i+1]
            temp[j,2] = n2[i+1]
del (pt2,p2,n2,sigma2)
pt2 = temp[:,0]
p2 = temp[:,1]/temp[:,2]
n2 = temp[:,2]
sigma2 = sqrt(p2*(1-p2)/n2)

temp = np.zeros((np.unique(pt3).shape[0],3))
j = 0
temp[0,0] = pt3[0]
temp[0,1] = p3[0]*n3[0]
temp[0,2] = n3[0]
for i in range(pt3.shape[0]-1):
    if pt3[i] == pt3[i+1]:
        temp[j,0] = pt3[i]
        temp[j,1] = temp[j,1]+p3[i+1]*n3[i+1]
        temp[j,2] = temp[j,2]+n3[i+1]
    else:
        j = j + 1
        temp[j,0] = pt3[i]
        temp[j,1] = p3[i]*n3[i]
        temp[j,2] = n3[i]
        if i == (pt3.shape[0]-2):
            temp[j,0] = pt3[i+1]
            temp[j,1] = p3[i+1]*n3[i+1]
            temp[j,2] = n3[i+1]
del (pt3,p3,n3,sigma3)
pt3 = temp[:,0]
p3 = temp[:,1]/temp[:,2]
```

```
n3 = temp[:,2]
sigma3 = sqrt(p3*(1-p3)/n3)

RemovePoints = False
if RemovePoints == True:
    for i in range(p0.shape[0]):
        if p0[i] < 0.09:
            p0[i] = np.nan
    for i in range(p1.shape[0]):
        if p1[i] < 0.09:
            p1[i] = np.nan
    for i in range(p2.shape[0]):
        if p2[i] < 0.09:
            p2[i] = np.nan
    for i in range(p3.shape[0]):
        if p3[i] < 0.09:
            p3[i] = np.nan

pt4 = pt2
p4=p1-p2
sigma4 = sqrt((sigma1)**2+(sigma2)**2)
#p4 = p4+abs(min(p4))+0.12

# plt.scatter(pt0,p0,color='blue')
plt.semilogx(pt0,p0,color='blue')
plt.errorbar(pt0,p0,yerr=sigma0,color='blue')
#plt.scatter(pt1,p1,color='green')
#plt.semilogx(pt1,p1,color='green')
plt.errorbar(pt1,p1,yerr=sigma1,color='green')
plt.scatter(pt2,p2,color='red')
#plt.semilogx(pt2,p2,color='red')
plt.errorbar(pt2,p2,yerr=sigma2,color='red')
#plt.scatter(pt3,p3,color='purple')
#plt.semilogx(pt3,p3,color='purple')
#plt.errorbar(pt3,p3,yerr=sigma3,color='purple')
plt.scatter(pt4,p4,color='black')
#plt.semilogx(pt4,p4,color='black')
plt.errorbar(pt4,p4,yerr=sigma4,color='black')
plt.semilogx([10,11],[0.5,0.5],color='black')
plt.xlabel(r'Protect Pulse Time (s)')
plt.ylabel('Bright Fraction')
plt.legend(('No OP w/ FOR','No OP w/o FOR','Offset Signal'),loc='upper left')
plt.title(r'20160202 Optical Pumping Time vs Bright Fraction')
plt.axis([5e-8,2e-4,0.05,0.7])
plt.grid(True)
```

E.1.12. Single-ion Trap Radial Secular Frequencies Calculations

The code below calculates the single-ion trap radial secular frequencies.

```
%pylab inline

#Directional coupler power detector voltages and Q-factor and axial trap
   frequency.
#20160201, -5 dBm on function generator, RF AM = -1.6.
cpl_fwd = 0.7214
cpl_rev = 1.1792
Omega = (2*pi)*23.420E6

#Things needed to calculate omega_r_0, then actual omega_r.
q = 1.6e-19 #C
m = 138*1.6605e-27 #kg
#R = 0.46E-3 #m
R = 1.26E-3 #m
z0 = 0.95E-3 #m
omega_z = (2*pi)*1.2E6
Q = 414
alpha_ac = 1.16 #Determined experimentally from 20160120 data
alpha_z = 4.4E-3 #calculated with 200 V DC and 1.2 MHz.
V_z = 200.

#Power to voltage calibration from the Minicircuits power detectors (ZX47-40LN
   +) with output
#voltage being the horizontal axis and power the vertical. Use only linear
   portion.
p_det_pts = np.array
   ([[1.9,-35],[1.78,-30],[1.65,-25],[1.55,-20],[1.42,-15],[1.3,-10],[1.2,-5],
                    [1.05,0],[0.92,5],[0.8,10],[0.68,15]])
pfit = polyfit(p_det_pts[:,0], p_det_pts[:,1], 1)
pfit_fun = poly1d(pfit)
x = linspace(-40,20,10)
#plt.scatter(p_det_pts[:,0],p_det_pts[:,1],color='blue')
#plt.plot(x,pfit_fun(x),color='black')
```

```
#plt.title('Power Detector Calibration\nSlope = '+array_str(pfit_fun[1])+r' V/
   dBm, Intercept = '+array_str(pfit_fun[0])+' V')#+' MHz\n'+'Slope = '+
   array_str(fit_fnl[1])+r' MHz/$\mu$W, Intercept = '+array_str(fit_fnl[0])+'
   MHz')
#plt.xlabel('Volts')
#plt.ylabel('Power (dBm)')
#axis([0.5,2.1,-40,20])
#plt.grid(True)

#Calculate power of each in dBm and W. Remember the ZFBDC20-61HP+ bidrectional
   coupler!
dBm_fwd = pfit_fun(cpl_fwd)+20
dBm_rev = pfit_fun(cpl_rev)+20
pwr_fwd = 10**((dBm_fwd-30)/10)
pwr_rev = 10**((dBm_rev-30)/10)
trap_pwr = pwr_fwd - pwr_rev
trap_pwr

#Calculate Trap voltage in RMS, then PP. Everything is impedance matched on
   resonance!
trap_vrms = sqrt(trap_pwr*50)*Q
trap_vpp = sqrt(2)*trap_vrms
trap_vpp

#Calculate omega_r_0, the unperturbed radial frequency.
omega_r_0 = alpha_ac*q*trap_vpp/(sqrt(2)*m*(R**2)*Omega)
omega_r_0/(2*pi)/1E6

#Calculate omega_r, the modified radial frequency induced by the
   destabilization along z.
omega_r = sqrt(omega_r_0**2-omega_z**2/2)
omega_r/(2*pi)/1E6
```

E.1.13. STIRAP Frequency Scan Analysis

The code below analyzes STIRAP frequency scans.

```
%pylab inline

file0 = '20151016_CWRamancool_STIRAPscanPMT_0.npy'
file1 = '20151016_CWRamancool_STIRAPscanPMT_1.npy'
file2 = '20151016_CWRamancool_STIRAPscanPMT_1.npy'
file3 = '20151015_CWRamancool_STIRAPscanPMT_2.npy'
data0 = load(file0)
data1 = load(file1)
data2 = load(file2)
data3 = load(file3)

expose_t0 = 75.0e-3
expose_t1 = 75.0e-3
expose_t2 = 75.0e-3
expose_t3 = 75.0e-3

#Histogram discriminator check if needed. Only runs on data0 unless code below
   changed.
pruned0 = data0[:,1:]/expose_t0
pruned1 = data1[:,1:]/expose_t1
pruned2 = data2[:,1:]/expose_t2
pruned3 = data3[:,1:]/expose_t3
hist(pruned2.ravel(),50)
xlabel(r'PMT Count Rate (s$^{-1}$)')
axis([0,14000,0,300])

#Define ion bight/dark discriminator value and number of points in
#each scan range.
discr0 = 3000
discrh0 = 14000

discr1 = 3000
discrh1 = 14000

discr2 = 3000
discrh2 = 14000
```

```
discr3 = 3000
discrh3 = 14000

data0 = data0[data0[:,0].argsort()]
data1 = data1[data1[:,0].argsort()]
data2 = data2[data2[:,0].argsort()]
data3 = data3[data3[:,0].argsort()]

#DO WORK. This section extracts frequencies and runs the loops to calculate
    bright probability and number of experiments.
#define number of events and event probability with sigma giving stat error.
res0 = zeros(data0.shape[0])
f0 = zeros(data0.shape[0])
n0 = zeros(data0.shape[0])
p0 = zeros(data0.shape[1])
for i in range(data0.shape[0]):
    f0[i] = data0[i,0]
    res0[i] = sum(np.logical_and(data0[i,1:]/expose_t0 > discr0, data0[i,1:]/
       expose_t0 < discrh0))
    n0[i] = sum(np.logical_and(data0[i,1:]/expose_t0 > 0, data0[i,1:]/expose_t0
        < discrh0))
p0 = res0/n0
sigma0 = sqrt(p0*(1-p0)/n0)

res1 = zeros(data1.shape[0])
f1 = zeros(data1.shape[0])
n1 = zeros(data1.shape[0])
p1 = zeros(data1.shape[1])
for i in range(data1.shape[0]):
    f1[i] = data1[i,0]
    res1[i] = sum(np.logical_and(data1[i,1:]/expose_t1 > discr1, data1[i,1:]/
       expose_t1 < discrh1))
    n1[i] = sum(np.logical_and(data1[i,1:]/expose_t1 > 0, data1[i,1:]/expose_t1
        < discrh1))
p1 = res1/n1
sigma1 = sqrt(p1*(1-p1)/n1)

res2 = zeros(data2.shape[0])
f2 = zeros(data2.shape[0])
n2 = zeros(data2.shape[0])
```

```
p2 = zeros(data2.shape[1])
for i in range(data2.shape[0]):
    f2[i] = data2[i,0]
    res2[i] = sum(np.logical_and(data2[i,1:]/expose_t2 > discr2, data2[i,1:]/
       expose_t2 < discrh2))
    n2[i] = sum(np.logical_and(data2[i,1:]/expose_t2 > 0, data2[i,1:]/expose_t2
        < discrh2))
p2 = res2/n2
sigma2 = sqrt(p2*(1-p2)/n2)

res3 = zeros(data3.shape[0])
f3 = zeros(data3.shape[0])
n3 = zeros(data3.shape[0])
p3 = zeros(data3.shape[1])
for i in range(data3.shape[0]):
    f3[i] = data3[i,0]
    res3[i] = sum(np.logical_and(data3[i,1:]/expose_t3 > discr3, data3[i,1:]/
       expose_t3 < discrh3))
    n3[i] = sum(np.logical_and(data3[i,1:]/expose_t3 > 0, data3[i,1:]/expose_t3
        < discrh3))
p3 = res3/n3
sigma3 = sqrt(p3*(1-p3)/n3)

RemovePoints = False
if RemovePoints == True:
    for i in range(p0.shape[0]):
        if p0[i] < 0.1:
            p0[i] = np.nan
    for i in range(p1.shape[0]):
        if p1[i] < 0.1:
            p1[i] = np.nan
    for i in range(p2.shape[0]):
        if p2[i] < 0.1:
            p2[i] = np.nan
    for i in range(p3.shape[0]):
        if p3[i] < 0.1:
            p3[i] = np.nan

plt.scatter(-2*f0,p0,color='blue')
#plt.plot(-2*f0,p0,color='blue')
plt.errorbar(-2*f0,p0,yerr=sigma0,color='blue')
plt.scatter(-2*f1,p1,color='green')
#plt.plot(-2*f1,p1,color='green')
plt.errorbar(-2*f1,p1,yerr=sigma1,color='green')
#plt.scatter(-2*f2,p2,color='red')
#plt.plot(-2*f2,p2,color='red')
#plt.errorbar(-2*f2,p2,yerr=sigma2,color='red')
#plt.scatter(-2*f3,p3,color='purple')
#plt.plot(-2*f3,p3,color='purple')
#plt.errorbar(-2*f3,p3,yerr=sigma3,color='purple')
plt.xlabel('(Probe - Pump) Frequency (MHz)')
plt.ylabel('Bright Fraction')
#plt.legend((r'-10 $\mu$s',r'-12 $\mu$s',r'-15 $\mu$s',r'-20 $\mu$s'),loc='
    upper right')
#plt.legend(('20150921','20150917'))#,r'-2 $\mu$s'),loc='upper right')
#plt.title(r'20150918 STIRAP, Shorter Delay Times 2')
#plt.axis([-14.6,-6.4,0.1,0.33])
plt.axis([-13.1,-8.9,.39,0.84])
#plt.axis([-11.1,-6.9,0.04,0.35])
plt.grid(True)
```

E.1.14. Temperature and Humidity Monitor Analysis

The code below analyzes temperature and humidity stability as a function of time and calculates the Allan variance of the parameters. The input files are data saved from the ThorLabs monitor software.

```
%pylab inline
import numpy as np
import allantools as at
import time

filename0 = '20160620_Temp_Humitidy.txt'
data0 = loadtxt('20160620_Temp_Humitidy.txt',skiprows=58)
data1 = loadtxt('20160627_Temp_Humitidy.txt',skiprows=58)
data2 = loadtxt('20160704_Temp_Humitidy.txt',skiprows=58)
data3 = loadtxt('20160711_Temp_Humitidy.txt',skiprows=58)
data4 = loadtxt('20160718_Temp_Humitidy.txt',skiprows=58)
data = np.vstack((data0,data1,data2,data3,data4))
#data = data3
#data

temp = np.empty((data0.shape[0]-1))
for i in range(data0.shape[0]-1):
    temp[i] = data0[i+1,0]-data0[i,0]
reptime0 = np.nanmean(temp)
del temp
temp = np.empty((data1.shape[0]-1))
for i in range(data1.shape[0]-1):
    temp[i] = data1[i+1,0]-data1[i,0]
reptime1 = np.nanmean(temp)
del temp
temp = np.empty((data2.shape[0]-1))
for i in range(data2.shape[0]-1):
    temp[i] = data2[i+1,0]-data2[i,0]
reptime2 = np.nanmean(temp)
del temp
reptime=np.mean([reptime0,reptime1,reptime2])
reptime
```

```
etime = np.zeros(data.shape[0])
for i in range(data.shape[0]):
   etime[i] = 60.0*i
#etime = data[:,0]
temp0 = data[:,1]
hum = data[:,2]
temp1 = data[:,3]
temp2 = data[:,4]

figsize(11,8.5)
plt.errorbar(etime,temp0,color='blue',fmt='o')
plt.xlabel('Elapsed Time (s)')
plt.ylabel('Temperature ($^{\circ}$C)')
plt.title('Temperature\n'+filename0)
#plt.axis([0.0,1e6,600,700])
plt.grid(True)

[[np.max(temp0)-np.min(temp0),np.average(temp0)],
 [np.max(temp1)-np.min(temp1),np.average(temp1)],
 [np.max(temp2)-np.min(temp2),np.average(temp2)]]

(taus_used_temp0, adev_temp0, adeverror_temp0, adev_n_temp0) = at.adev(temp0,
   1.0/reptime, np.linspace(0,10000000.0,250000))
(taus_used_temp1, adev_temp1, adeverror_temp1, adev_n_temp1) = at.adev(temp1,
   1.0/reptime, np.linspace(0,10000000.0,250000))
(taus_used_temp2, adev_temp2, adeverror_temp2, adev_n_temp2) = at.adev(temp2,
   1.0/reptime, np.linspace(0,10000000.0,250000))

plt.loglog(taus_used_temp0,adev_temp0,color='blue',marker='s',linewidth=0)
plt.loglog(taus_used_temp1,adev_temp1,color='red',marker='s',linewidth=0)
plt.loglog(taus_used_temp2,adev_temp2,color='black',marker='s',linewidth=0)
plt.legend(('Temperature',''),loc='upper left')
plt.xlabel('Elapsed Time (s)')
plt.ylabel('Temperature Allan Deviation ($^{\circ}$C)')
plt.title('Temperature Allan Deviation\n'+filename0)
#plt.axis([9e0,1.1e5,2e-2,2e-1])
plt.grid(True)

plt.errorbar(etime/60,hum,color='blue',fmt='o')
plt.xlabel('Elapsed Time (s)')
plt.ylabel('Relative Humidity (%)')
plt.title(’Relative Humidity\n’+filename0)
#plt.axis([0.0,1e6,600,700])
plt.grid(True)

(taus_used_hum, adev_hum, adeverror_hum, adev_n_hum) = at.adev(hum, 1.0/reptime
    , np.linspace(0,10000000.0,250000))\

plt.loglog(taus_used_hum,adev_hum,color=’blue’,marker=’s’,linewidth=0)
plt.legend((’Temperature’,’’),loc=’upper left’)
plt.xlabel(’Elapsed Time (s)’)
plt.ylabel(’Relative Humidity Allan Deviation (%)’)
plt.title(’Relative Humidity Allan Deviation\n’+filename0)
#plt.axis([9e0,1.1e5,2e-2,2e-1])
plt.grid(True)

figsize(18,18)
f, ([ax1, ax2],[ax3, ax4]) = plt.subplots(2, 2)
ax1.errorbar(etime/60,temp0,color=’black’,fmt=’o’,elinewidth=2,capthick=2,
    linewidth=2,ms=8,label=’Lab Space’)
ax1.errorbar(etime/60,temp1,color=’blue’,fmt=’s’,elinewidth=2,capthick=2,
    linewidth=2,ms=8,label=’Over Optics’)
ax1.errorbar(etime/60,temp2,color=’red’,fmt=’^’,elinewidth=2,capthick=2,
    linewidth=2,ms=8,label=’Resonator’)
ax1.axis([-3000,57000,19,29.2])
ax1.set_xlabel(’Elapsed Time (min)’,fontsize=’xx-large’)
#ax1.ticklabel_format(style=’sci’, axis=’x’, scilimits=(2,1))
ax1.set_ylabel(’Temperature ($^{\circ}$C)’,fontsize=’xx-large’)
ax1.tick_params(axis=’x’, labelsize=’xx-large’)
ax1.tick_params(axis=’y’, labelsize=’xx-large’)
ax1.ticklabel_format(useOffset=False)
ax1.tick_params(’both’, length=12, width=2, which=’major’)
ax1.minorticks_on()
ax1.tick_params(’both’, length=5, width=2, which=’minor’)
ax2.loglog(taus_used_temp0/60,adev_temp0,color=’black’,marker=’o’,linewidth=2,
    ms=8,label=’Lab Space’)
ax2.loglog(taus_used_temp1/60,adev_temp1,color=’blue’,marker=’s’,linewidth=2,ms
    =8,label=’Over Optics’)
ax2.loglog(taus_used_temp2/60,adev_temp2,color=’red’,marker=’^’,linewidth=2,ms
    =8,label=’Resonator’)
ax2.set_xlabel(’Time Interval (min)’,fontsize=’xx-large’)
ax2.set_ylabel(’Temperature Allan Deviation ($^{\circ}$C)’,fontsize=’xx-large’)
```

```
ax2.axis([6E-1,3.0E4,1.5E-2,2.5E0])
ax2.legend(bbox_to_anchor=(0.42,0.98), fontsize='xx-large')
ax2.tick_params(axis='x', labelsize='xx-large')
ax2.tick_params(axis='y', labelsize='xx-large')
ax2.tick_params('both', length=12, width=2, which='major')
ax2.minorticks_on()
ax2.tick_params('both', length=5, width=2, which='minor')
ax3.errorbar(etime/60,hum,color='blue',fmt='D',elinewidth=2,capthick=2,
   linewidth=2,ms=8,label='Relative Humidity')
ax3.axis([-3000,57000,24.5,51])
ax3.set_xlabel('Elapsed Time (min)',fontsize='xx-large')
#ax4.ticklabel_format(style='sci', axis='x', scilimits=(2,1))
ax3.set_ylabel('Relative Humidity (%)',fontsize='xx-large')
ax3.tick_params(axis='x', labelsize='xx-large')
ax3.tick_params(axis='y', labelsize='xx-large')
#ax3.ticklabel_format(useOffset=False)
ax3.tick_params('both', length=12, width=2, which='major')
ax3.minorticks_on()
ax3.tick_params('both', length=5, width=2, which='minor')
ax4.loglog(taus_used_hum/60,adev_hum,color='blue',marker='D',linewidth=2,ms=8)
ax4.set_xlabel('Time Interval (min)',fontsize='xx-large')
ax4.set_ylabel('Relative Humidity Allan Deviation (%)',fontsize='xx-large')
ax4.axis([6E-1,3.0E4,2.0E-1,5.0E0])
ax4.tick_params(axis='x', labelsize='xx-large')
ax4.tick_params(axis='y', labelsize='xx-large')
ax4.tick_params('both', length=12, width=2, which='major')
ax4.minorticks_on()
ax4.tick_params('both', length=5, width=2, which='minor')

#f.subplots_adjust(wspace=0.08)
#savefig('TempHumidMon.pdf', dpi=None, facecolor='w', edgecolor='w', format='
   pdf', pad_inches=0.1, frameon=None,bbox_inches='tight')
#savefig('TempHumidMon.png', dpi=None, facecolor='w', edgecolor='w', format='
   png', pad_inches=0.1, frameon=None,bbox_inches='tight')
```

E.1.15. Voltages Monitor Analysis

The code below analyzes magnetic field stability and secular frequencies as a function of time and calculates the Allan variance of the parameters.

```
%pylab inline
import allantools as at
import time

file0 = r'C:\data\Voltages Monitor\20160229_Voltages_Monitor_0.npy'
filename0 = '20160229_Voltages_Monitor_0.npy'
data0 = load(file0)

#Start of data run.
time.localtime(np.real(data0[0,0]))

#Time of interest.
time.localtime(1.45595e9+50000)

#Split up the data into relevant bins.
time0 = np.real(data0[:,0])
elapsedtime0 = time0[:] - time0[0]
fluxgate0 = np.real(data0[:,1])
delta_fluxgate0 = np.imag(data0[:,1])
cpl_fwd0 = np.real(data0[:,2])
delta_cpl_fwd0 = np.imag(data0[:,2])
cpl_rev0 = np.real(data0[:,3])
delta_cpl_rev0 = np.imag(data0[:,3])
endcap10 = np.real(data0[:,4])
delta_endcap10 = np.imag(data0[:,4])
endcap20 = np.real(data0[:,5])
delta_endcap20 = np.imag(data0[:,5])

#Delete any "bad points, i.e. where the elapsedtime0 value is < 0.0.
index = (np.argwhere(elapsedtime0[:] < 0.0).transpose()[0])
elapsedtime0 = np.delete(elapsedtime0,index)
fluxgate0 = np.delete(fluxgate0,index)
delta_fluxgate0 = np.delete(delta_fluxgate0,index)
cpl_fwd0 = np.delete(cpl_fwd0,index)
delta_cpl_fwd0 = np.delete(delta_cpl_fwd0,index)
```

```
cpl_rev0 = np.delete(cpl_rev0,index)
delta_cpl_rev0 = np.delete(delta_cpl_rev0,index)
endcap10 = np.delete(endcap10,index)
delta_endcap10 = np.delete(delta_endcap10,index)
endcap20 = np.delete(endcap20,index)
delta_endcap20 = np.delete(delta_endcap20,index)

#Delete last entry just in case it catches the zero.
elapsedtime0 = np.delete(elapsedtime0,elapsedtime0.shape[0]-1)
fluxgate0 = np.delete(fluxgate0,fluxgate0.shape[0]-1)
delta_fluxgate0 = np.delete(delta_fluxgate0,delta_fluxgate0.shape[0]-1)
cpl_fwd0 = np.delete(cpl_fwd0,cpl_fwd0.shape[0]-1)
delta_cpl_fwd0 = np.delete(delta_cpl_fwd0,delta_cpl_fwd0.shape[0]-1)
cpl_rev0 = np.delete(cpl_rev0,cpl_rev0.shape[0]-1)
delta_cpl_rev0 = np.delete(delta_cpl_rev0,delta_cpl_rev0.shape[0]-1)
endcap10 = np.delete(endcap10,endcap10.shape[0]-1)
delta_endcap10 = np.delete(delta_endcap10,delta_endcap10.shape[0]-1)
endcap20 = np.delete(endcap20,endcap20.shape[0]-1)
delta_endcap20 = np.delete(delta_endcap20,delta_endcap20.shape[0]-1)

#Calculate the average time between points for the Allan deviations.
temp = np.empty((elapsedtime0.shape[0]-1))
for i in range(elapsedtime0.shape[0]-1):
    temp[i] = elapsedtime0[i+1]-elapsedtime0[i]
reptime = np.nanmean(temp)
del temp
reptime

FluxgateGain = 2e-3 #V/mG
plt.errorbar(elapsedtime0,fluxgate0/FluxgateGain,\
            yerr=delta_fluxgate0/FluxgateGain,color='blue',fmt='o')
plt.xlabel('Elapsed Time (s)')
plt.ylabel('Detected B-Field (mG)')
plt.title('Fluxgate Monitor\n'+filename0)
#plt.axis([0.0,1e6,600,700])
plt.grid(True)

np.max(fluxgate0/FluxgateGain)-np.min(fluxgate0/FluxgateGain)
np.average(fluxgate0/FluxgateGain)
#np.std(fluxgate0/FluxgateGain)
```

```
(taus_used_fluxgate0, adev_fluxgate0, adeverror_fluxgate0, adev_n_fluxgate0) =\
    at.adev(fluxgate0/FluxgateGain, 1/reptime, np.linspace(0,10000000,2500000))

#plt.errorbar(taus_used_fluxgate0,adev_fluxgate0,yerr = adeverror_fluxgate0,
    color='blue',fmt='-s')
plt.loglog(taus_used_fluxgate0,adev_fluxgate0,color='blue',marker='s',linewidth
    =0)
#plt.loglog(taus_used_fluxgate0[1],adev_fluxgate0[1])
plt.legend(('Fluxgate',''),loc='upper left')
plt.xlabel('Elapsed Time (s)')
plt.ylabel('Fluxgate Detected B-Field Allan Deviation (mG)')
plt.title('Fluxgate Detected B-Field Allan Deviation\n'+filename0)
#plt.axis([9e0,1.1e5,2e-2,2e-1])
plt.grid(True)

(planck*10.982E6/(2*muB))/(np.average(fluxgate0)/FluxgateGain*1E-3)
Bgain

muB = 5.788382E-9 #eV/Gauss
planck = 4.135668E-15 #eVs
Bgain = (planck*10.94E6/(2*muB))/(np.average(fluxgate0)/FluxgateGain*1E-3) #
    Bfield gain between the measured above and the ion splitting of approx 10
    MHz.
BtoFreq = 2*muB*(Bgain/FluxgateGain)/planck/1E9
plt.errorbar(elapsedtime0,fluxgate0*BtoFreq,\
            yerr=delta_fluxgate0*BtoFreq,color='blue',fmt='o')
plt.xlabel('Elapsed Time (s)')
plt.ylabel('Calculated Zeeman Splitting (MHz)')
plt.title('Fluxgate Monitor Zeeman Splitting\n'+filename0)
#plt.axis([0.0,1e6,600,700])
plt.grid(True)
ax = plt.gca()
ax.ticklabel_format(useOffset=False)

(taus_used_zeeman0, adev_zeeman0, adeverror_zeeman0, adev_n_zeeman0) =\
    at.adev(fluxgate0*BtoFreq, 1/reptime, np.linspace(0,10000000,2500000))

#plt.errorbar(taus_used_zeeman0,adev_zeeman0*1E3,yerr = adeverror_zeeman0*1E3,
    color='blue',fmt='-s')
plt.loglog(taus_used_zeeman0,adev_zeeman0*1E6,color='blue',marker='s',linewidth
    =0)
```

```
plt.legend(('Zeeman Splitting',''),loc='upper left')
plt.xlabel('Elapsed Time (s)')
plt.ylabel('Calculated Zeeman Splitting Allan Deviation (Hz)')
plt.title('Fluxgate Monitor Zeeman Splitting Allan Deviation\n'+filename0)
#plt.axis([9e0,1.1e5,2e-2,2e-1])
plt.grid(True)

plt.errorbar(elapsedtime0,cpl_fwd0,yerr=delta_cpl_fwd0,color='green',fmt='o')
plt.xlabel('Elapsed Time (s)')
plt.ylabel('CPL FWD Power Detector Voltage (V)')
plt.title('CPL FWD Power Monitor\n'+filename0)
#plt.axis([-0.05*max(elapsedtime0),1.05*max(elapsedtime0),0.719,0.7196])
plt.grid(True)

plt.errorbar(elapsedtime0,cpl_rev0,yerr=delta_cpl_rev0,color='red',fmt='o')
plt.xlabel('Elapsed Time (s)')
plt.ylabel('CPL REV Power Detector Voltage (V)')
plt.title('CPL REV Power Monitor\n'+filename0)
#plt.axis([-12.3,-9.6,0.05,0.6])
plt.grid(True)

(taus_used_cpl_fwd0, adev_cpl_fwd0, adeverror_cpl_fwd0, adev_n_cpl_fwd0) =\
    at.adev(cpl_fwd0, 1/reptime, np.linspace(0,10000000,2500000))
(taus_used_cpl_rev0, adev_cpl_rev0, adeverror_cpl_rev0, adev_n_cpl_rev0) =\
    at.adev(cpl_rev0, 1/reptime, np.linspace(0,10000000,2500000))

#plt.errorbar(taus_used_cpl_fwd0,adev_cpl_fwd0,yerr = adeverror_cpl_fwd0,color
    ='green',fmt='-s')
#plt.errorbar(taus_used_cpl_rev0,adev_cpl_rev0,yerr = adeverror_cpl_rev0,color
    ='red',fmt='-s')
plt.loglog(taus_used_cpl_fwd0,adev_cpl_fwd0,color='green',marker='s',linewidth
    =0)
plt.loglog(taus_used_cpl_rev0,adev_cpl_rev0,color='red',marker='s',linewidth=0)
#plt.loglog(taus_used_cpl_fwd0[1],adev_cpl_fwd0[1])
plt.legend(('CPL FWD','CPL REV'),loc='upper left')
plt.xlabel('Elapsed Time (s)')
plt.ylabel('Directional Coupler Allan Deviation (V)')
plt.title('Directional Coupler Allan Deviation\n'+filename0)
#plt.axis([9e0,2e5,2e-5,8e-4])
plt.grid(True)
```

```
ECG = 36.15 #Endcap gain.
plt.errorbar(elapsedtime0,endcap10*ECG,yerr=delta_endcap10*ECG,color='purple',
    fmt='o')
plt.xlabel('Elapsed Time (s)')
plt.ylabel('Endcap 1 Voltage (V)')
plt.title('Endcap 1 Votlage via Monitor\n'+filename0)
#plt.axis([-12.3,-9.6,0.05,0.6])
plt.grid(True)

plt.errorbar(elapsedtime0,endcap20*ECG,yerr=delta_endcap20*ECG,color='black',
    fmt='o')
plt.xlabel('Elapsed Time (s)')
plt.ylabel('Endcap 2 Voltage (V)')
plt.title('Endcap 2 Voltage via Monitor\n'+filename0)
#plt.axis([-12.3,-9.6,0.05,0.6])
plt.grid(True)

(taus_used_endcap10, adev_endcap10, adeverror_endcap10, adev_n_endcap10) =\
    at.adev(endcap10*ECG, 1/reptime, np.linspace(0,10000000,2500000))
(taus_used_endcap20, adev_endcap20, adeverror_endcap20, adev_n_endcap20) =\
    at.adev(endcap20*ECG, 1/reptime, np.linspace(0,10000000,2500000))

#plt.errorbar(taus_used_endcap10,adev_endcap10,yerr = adeverror_endcap10,color
    ='purple',fmt='-s')
#plt.errorbar(taus_used_endcap20,adev_endcap20,yerr = adeverror_endcap20,color
    ='black',fmt='-s')
plt.loglog(taus_used_endcap10,adev_endcap10,color='purple',marker='s',linewidth
    =0)
plt.loglog(taus_used_endcap20,adev_endcap20,color='black',marker='s',linewidth
    =0)
plt.legend(('Endcap 1','Endcap 2'),loc='upper left')
plt.xlabel('Elapsed Time (s)')
plt.ylabel('Endcap Voltage Allan Deviation (V)')
plt.title('Endcap Voltage Allan Deviation via Monitors\n'+filename0)
#plt.axis([9e0,1.1e5,5e-4,8e-3])
plt.grid(True)

#Things needed to calculate omega_r_0, then actual omega_r.
Omega = (2*pi)*23.420E6
q = 1.6e-19 #C
m = 138*1.6605e-27 #kg
```

```
#R = 0.46E-3 #m
R = 1.26E-3 #m
z0 = 0.95E-3 #m
omega_z = (2*pi)*1.1377E6 #Measured via FOR.
Q = 414.0
alpha_ac = 1.16 #Determined experimentally from 20160120 data
#alpha_z = 4.4E-3 #calculated with 200 V DC and 1.2 MHz.
V_z = 200.0 #alpha_z calibration voltage.
alpha_z = omega_z**2*z0**2*m/(2*V_z*q)
alpha_z

#Power to voltage calibration from the Minicircuits power detectors (ZX47-40LN
    +) with output
#voltage being the horizontal axis and power the vertical. Use only linear
    portion.
p_det_pts = np.array
    ([[1.9,-35],[1.78,-30],[1.65,-25],[1.55,-20],[1.42,-15],[1.3,-10],[1.2,-5],
                    [1.05,0],[0.92,5],[0.8,10],[0.68,15]])
pfit = polyfit(p_det_pts[:,0], p_det_pts[:,1], 1)
pfit_fun = poly1d(pfit)
x = linspace(-40,20,10)
#plt.scatter(p_det_pts[:,0],p_det_pts[:,1],color=’blue’)
#plt.plot(x,pfit_fun(x),color=’black’)
#plt.title(’Power Detector Calibration\nSlope = ’+array_str(pfit_fun[1])+r’ V/
    dBm, Intercept = ’+array_str(pfit_fun[0])+’ V’)#+’ MHz\n’+’Slope = ’+
    array_str(fit_fnl[1])+r’ MHz/$\mu$W, Intercept = ’+array_str(fit_fnl[0])+’
    MHz’)
#plt.xlabel(’Volts’)
#plt.ylabel(’Power (dBm)’)
#axis([0.5,2.1,-40,20])
#plt.grid(True)

#Calculate power of each in dBm and W. Remember the ZFBDC20-61HP+ bidrectional
    coupler!
dBm_fwd0 = pfit_fun(cpl_fwd0)+20
dBm_rev0 = pfit_fun(cpl_rev0)+20
pwr_fwd0 = 10**((dBm_fwd0-30)/10)
pwr_rev0 = 10**((dBm_rev0-30)/10)
trap_pwr0 = pwr_fwd0 - pwr_rev0
#Calculate Trap voltage in RMS, then PP. Everything is impedance matched on
    resonance!
```

```
trap_vrms0 = sqrt(trap_pwr0*50)*Q
trap_vpp0 = sqrt(2)*trap_vrms0
#Calculate omega_z, the axial frequency.
omega_z0 = sqrt(2*q*alpha_z*((endcap10+endcap20)/2*ECG)/(m*z0**2))
f_z0 = omega_z0/(2*pi)/1E6
#Calculate omega_r_0, the unperturbed radial frequency.
omega_r_00 = alpha_ac*q*trap_vpp0/(sqrt(2)*m*(R**2)*Omega)
f_r_00 = omega_r_00/(2*pi)/1E6
#Calculate omega_r, the modified radial frequency induced by the
   destabilization along z.
omega_r0 = sqrt(omega_r_00**2-omega_z**2/2)
f_r0 = omega_r0/(2*pi)/1E6
#np.average(trap_vpp0)

plt.errorbar(elapsedtime0,f_z0*1E6,color=’blue’,fmt=’-o’)
plt.xlabel(’Elapsed Time (s)’)
plt.ylabel(’$\omega_{axial}$/2$\pi$ (Hz)’)
plt.title(’Axial Secular Frequency Monitor\n’+filename0)
#plt.axis([-12.3,-9.6,0.05,0.6])
plt.grid(True)

(taus_used_z0, adev_z0, adeverror_z0, adev_n_z0) = at.adev(f_z0, 1/reptime, np.
   linspace(0,100000000,25000000))

#plt.errorbar(taus_used_z0,adev_z0,yerr = adeverror_z0,color=’blue’,fmt=’-s’)
plt.loglog(taus_used_z0,adev_z0*1E6,color=’blue’,marker=’s’,linewidth=0)
#plt.loglog(taus_used_z0[1],adev_z0[1])
plt.legend((’$\omega_{axial}$’,’’),loc=’upper left’)
plt.xlabel(’Elapsed Time (s)’)
plt.ylabel(’Axial Allan Deviation (Hz)’)
plt.title(’Axial Secular Frequency Allan Deviation\n’+filename0)
#plt.axis([9e0,1.1e5,1e-6,3e-5])
plt.grid(True)

plt.errorbar(elapsedtime0,f_r0,color=’green’,fmt=’-o’)
plt.xlabel(’Elapsed Time (s)’)
plt.ylabel(’$\omega_{radial}$/2$\pi$ (MHz)’)
plt.title(’Radial Secular Frequency Monitor\n’+filename0)
#plt.axis([-12.3,-9.6,0.05,0.6])
plt.grid(True)
```

```
(taus_used_r0, adev_r0, adeverror_r0, adev_n_r0) = at.adev(f_r0, 1/reptime, np.
   linspace(0,10000000,2500000))

#plt.errorbar(taus_used_r0,adev_r0,yerr = adeverror_r0,color=’green’,fmt=’-s’)
plt.loglog(taus_used_r0,adev_r0*1E6,color=’green’,marker=’s’,linewidth=0)
#plt.loglog(taus_used_r0[1],adev_r0[1])
plt.legend((’$\omega_{radial}$’,’’),loc=’upper left’)
plt.xlabel(’Elapsed Time (s)’)
plt.ylabel(’Radial Allan Deviation (Hz)’)
plt.title(’Radial Secular Frequency Allan Deviation\n’+filename0)
#plt.axis([9e0,1.1e5,3e-4,3e-3])
plt.grid(True)
```

E.2. Python Backend

The Python environment is used for data collection and analysis in the single-ion motional ground state cooling project. The section below contains the backend scripts that are used to interface Python with the National Instruments hardware and the DDS.

E.2.1. 6602 Timing Backend

The code below provides the class for setting up 6602 counter pulse sequences. Created by MRD.

```
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 16 14:38:50 2015

A class for setting up a oneshot counter pulse to be used in experiments. Uses
    the C interface for DAQmx that comes with labview. Note that DAQmx can be
    downloaded for free from NI, so that in principle this code can be run
    without labview installed.

@author: matt
"""

import ctypes
import pdb

# This needs to point ctypes to the local DAQmx drivers
ni = ctypes.windll.nicaiu

int32 = ctypes.c_long
uInt32 = ctypes.c_ulong
uInt64 = ctypes.c_ulonglong
float64 = ctypes.c_double

# Select configuration constants from the file NIDaqmx.h
config_default = ctypes.c_int32(-1)
```

```
# To set the pulse duration units
val_seconds = int32(10364)

# To set the resting state for the counter
val_high = int32(10192)
val_low = int32(10214)

# To set rising or falling edge for trigger
val_rising = int32(10280)
val_falling = int32(10171)

def CHK(err):
    """a simple error checking routine"""
    if err < 0:
        buf_size = 4000
        buf = ctypes.create_string_buffer(’\000’ * buf_size)
        ni.DAQmxGetErrorString(err,ctypes.byref(buf),buf_size)
        raise RuntimeError(’nidaq call failed with error %d: %s’%(err,repr(buf.
            value)))

class OneShot(object):
    def __init__(self,chan,trigchan,highlow,start,end):
        """
        __init__(self,chan,trigchan,highlow,start,end,cycletime):

        chan is the counter device to be used for this oneshot
        trigchan is the channel used as its trigger
        highlow is True if you would like this channel to be high at the
            beginning
        of a cycle, and False if you would like it to be low
        start is the beginning of the time at which the state toggles
        end is the end of the time at which the state toggles back to its
            original
        state
        """
        safe = 1e-5
        self.th = uInt32(0)
        self.chan = chan
        self.trigchan = trigchan
        if end <= start:
            raise ValueError
        if highlow:
            self.idle = val_high

            if start < safe:
                self.high_time = float64(safe)
            else:
                self.high_time = float64(start)
            self.low_time = float64(end-start)
            self.initial_delay = self.high_time
        else:
            self.idle = val_low
            self.high_time = float64(end-start)
            if start < safe:
                self.low_time = float64(safe)
            else:
                self.low_time = float64(start)
            self.initial_delay = self.low_time
        CHK(ni.DAQmxCreateTask("",ctypes.byref(self.th)))

    #def setTimes(self,start,end):
    #    self.starttime = float64(start)
    #    self.gotime = float64(end-start)

    def begin(self):
        """ Begin accepting triggers. No parameters are accepted at this stage
            ."""
        CHK(ni.DAQmxCreateCOPulseChanTime(self.th,self.chan,"",val_seconds,self.
            idle,
                                  self.initial_delay,self.low_time,self.
                                      high_time))
        CHK(ni.DAQmxCfgDigEdgeStartTrig(self.th,self.trigchan,val_rising))
        CHK(ni.DAQmxSetStartTrigRetriggerable(self.th,uInt32(1)))

        #CHK(ni.DAQmxSetCOEnableInitialDelayOnRetrigger(self.th,self.chan,uInt32
            (1)))
        CHK(ni.DAQmxStartTask(self.th))

    def stop(self):
        if self.th.value != 0:
            ni.DAQmxStopTask(self.th)
            ni.DAQmxClearTask(self.th)
```

```
def __del__(self):
    if self.th.value != 0:
        ni.DAQmxStopTask(self.th)
        ni.DAQmxClearTask(self.th)
```

E.2.2. 6602 Timing Plotter

The code below generates a HTML-format, interactive plot for displaying the experimental timing diagram for a given experiment. Created by MRD.

```
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 24 10:52:09 2015

@author: lynx
"""

import bokeh.plotting as bk

def plotOneShots(ctimes, filename="timing.html", show=True):
    """
    def plotOneShots(ctimes, filename="timing.html", show=True):

    ctimes is a dict whose keys are various channel names, and whose values
    are [start_time, end_time] lists. It then plots using Bokeh these start
    and end times in a way understandable to the user.

    filename is the name of the html file to output to
    if show is True, then we go ahead and show the plot. If show is false,
    then not.
    The figure object is returned.
    """
    bk.output_file(filename, title="Timing sequence")

    keys = list(ctimes.keys())

    start_times = [ctimes[k][0] for k in keys]
    end_times = [ctimes[k][1] for k in keys]

    p1 = bk.figure(title="One Shot Timings", tools="save,box_zoom,reset",
        y_range=keys, x_range=[0,max(end_times)])

    p1.segment(start_times, keys, end_times, keys, line_width=2, line_color='
        green')
```

```
p1.circle(start_times, keys, size=15, fill_color="orange", line_color="
    green", line_width=3)
p1.circle(end_times, keys, size=15, fill_color="orange", line_color="green
    ", line_width=3)

p1.xaxis.axis_label = "Times (s)"

if show:
    bk.show(p1)
return p1
```

E.2.3. AI Read Backend

The code below generates a class for reading in voltages from the NI AI channels. Created by MRD and modified by CMS.

```
# -*- coding: utf-8 -*-
"""
Created on Tue Jun 16 14:30:27 2015

@author: matt
"""

import ctypes
import numpy as np

int32 = ctypes.c_long
uInt32 = ctypes.c_ulong
uInt64 = ctypes.c_ulonglong
float64 = ctypes.c_double

# This needs to point ctypes to the local DAQmx drivers
# In priciple, this is the only system dependant line in this file
ni = ctypes.windll.nicaiu

# Select configuration constants from the file NIDaqmx.h
config_default = ctypes.c_int32(-1)
val_rising = int32(10280)
val_volts = int32(10348)
val_finitesamps = int32(10178)
val_contsamps = int32(10123)
val_groupbychannel = int32(0)
val_groupbyscannum = int32(1)
val_rse = int32(10083)

class Reserved(Exception):
    pass

class MissedData(Exception):
    pass
```

```
class InvalidTask(Exception):
    pass

def CHK(err):
    """a simple error checking routine"""
    if err < 0:
        if err == -50103:
            raise Reserved
        elif err == -200279:
            raise MissedData
        elif err == -200088:
            raise InvalidTask
        else:
            buf_size = 100
            buf = ctypes.create_string_buffer('\000' * buf_size)
            ni.DAQmxGetErrorString(err,ctypes.byref(buf),buf_size)
            raise RuntimeError('nidaq call failed with error %d: %s'%(err,repr(
                buf.value)))

class AIRead():
    def __init__(self,chan):
        """
        __init__(self,chan):

        chan is the NI name of the channel
        """
        self.th = uInt32(0)
        self.chan = chan
        self.trig = False
        CHK(ni.DAQmxCreateTask("",ctypes.byref(self.th)))
        CHK(ni.DAQmxCreateAIVoltageChan(self.th,chan,"",
                          val_rse,float64(-10.0),
                          float64(10.0),val_volts,None))

    def __del__(self):
        if self.th.value != 0:
            ni.DAQmxStopTask(self.th)
            ni.DAQmxClearTask(self.th)
        self.th = uInt32(0)

    def stop(self):
```

```
    if self.th.value != 0:
        ni.DAQmxStopTask(self.th)

def start(self):
    if self.th.value != 0:
        ni.DAQmxStartTask(self.th)

def getVoltage(self):
    """
    getVoltage():

    Retrieves the current value of the analog in channel, and returns that
        number.

    Basically, just get a single data point, right now.
    """
    val = float64(0)
    if self.trig:
        raise RuntimeError('You may not getVoltage after setTrig')
    else:
        CHK(ni.DAQmxReadAnalogScalarF64(self.th,float64(-1),ctypes.byref(val
            ),None))
        return val.value

def setTrig(self,samplerate,bufsize,trig):
    """
    setTrig(samplerate,trig)
    Prepare to collect a number of data points equal to 'bufsize', at a
    speed given by 'samplerate', upon reciept of a trigger on NI channel
    'trig'. Although this command sets the trigger and buffer, it does not
    actually start the task, so the method start() has to be run before
    the system will respond to a trigger.

    To actually retrieve the data output, execute the method getData().

    This version of setTrig does not implement retriggering, which, for many
         NI
    card version, would require a separate counter to control the sampling
    clock. Instead, this version requires you to stop() then start() the
    task to ready the card for a new trigger.
    """
```

```
        CHK(ni.DAQmxCfgSampClkTiming(self.th,"OnBoardClock",float64(samplerate),
                                     val_rising,val_finitesamps,uInt64(bufsize)
                                         ))
        CHK(ni.DAQmxCfgDigEdgeStartTrig(self.th,trig,val_rising))
        #CHK(ni.DAQmxSetStartTrigRetriggerable(self.th,uInt32(1)))
        self.trig = True
        self.data = np.zeros((bufsize,))
        self.bufsize = bufsize

    def getData(self):
        """
        getData()

        Waits for a trigger event and then collects and returns data, all as
        determined by setTrig(). Returns numpy array.

        Order of operations: setTrig, start, getData, stop.
        """
        read = int32()

        if self.trig:
            CHK(ni.DAQmxReadAnalogF64(self.th,self.bufsize,float64(1),
                val_groupbyscannum,
                      self.data.ctypes.data,
                      self.bufsize,ctypes.byref(read),None))

            return self.data
        else:
            raise RuntimeError('You must run setTrig before getData')
```

E.2.4. AO Write Backend

The code below generates a class for writing to the NI AO channels. Created by MRD and modified by CMS.

```
# -*- coding: utf-8 -*-
"""
Created on Thu Mar 19 10:04:13 2015

@author: lynx
"""

import ctypes
import numpy as np

int32 = ctypes.c_long
uInt32 = ctypes.c_ulong
uInt64 = ctypes.c_ulonglong
float64 = ctypes.c_double

# This needs to point ctypes to the local DAQmx drivers
# In priciple, this is the only system dependant line in this file
ni = ctypes.windll.nicaiu

# Select configuration constants from the file NIDaqmx.h
config_default = ctypes.c_int32(-1)
val_rising = int32(10280)
val_volts = int32(10348)
val_finitesamps = int32(10178)
val_contsamps = int32(10123)
val_groupbychannel = int32(0)
val_groupbyscannum = int32(1)

class Reserved(Exception):
    pass

class MissedData(Exception):
    pass

class InvalidTask(Exception):
   pass

def CHK(err):
   """a simple error checking routine"""
   if err < 0:
       if err == -50103:
           raise Reserved
       elif err == -200279:
           raise MissedData
       elif err == -200088:
           raise InvalidTask
       else:
           buf_size = 100
           buf = ctypes.create_string_buffer('\000' * buf_size)
           ni.DAQmxGetErrorString(err,ctypes.byref(buf),buf_size)
           raise RuntimeError('nidaq call failed with error %d: %s'%(err,repr(
               buf.value)))

class AOWrite():
   def __init__(self,chan):
       """
       __init__(self,chan):

       chan is the NI name of the channel
       """
       self.th = uInt32(0)
       self.chan = chan
       self.trig = False
       CHK(ni.DAQmxCreateTask("",ctypes.byref(self.th)))
       CHK(ni.DAQmxCreateAOVoltageChan(self.th, chan, "", float64(-10),
                                    float64(10), val_volts, "",))
       #Original code line is below, which set a mimimum AO value of 0.
       #CHK(ni.DAQmxCreateAOVoltageChan(self.th, chan, "", float64(0),
       #                              float64(10), val_volts, "",))

   def __del__(self):
       if self.th.value != 0:
           ni.DAQmxStopTask(self.th)
           ni.DAQmxClearTask(self.th)
       self.th = uInt32(0)
```

```
def stop(self):
    if self.th.value != 0:
        ni.DAQmxStopTask(self.th)

def clear(self):
            if self.th.value != 0:
                    ni.DAQmxClearTask(self.th)

def start(self):
    if self.th.value != 0:
        ni.DAQmxStartTask(self.th)

def setVoltage(self,volt):
    """
    setVoltage(volt):

    Set the voltage of the AO channel to volt, which should be a single
        floating
    point number.
    """
    if self.trig:
        raise RuntimeError('You may not setVoltage after voltSequence')
    else:
        CHK(ni.DAQmxWriteAnalogScalarF64(self.th,uInt32(1),float64(1),
            float64(volt),None))

def setTrig(self,samplerate,bufsize,trig):
    """
    setTrig(samplerate,trig)
    Prepare the AO channel for a voltage sequence. The points are sequenced
    through at samplerate speed (Hz) and triggered by the channel found in
    trig. To load a voltage sequence into the buffer, use the method
    voltSequence. bufsize should be set to the length of the array to be
    provided to voltSequence.

            CMS (20160118): I changed voltsequence such that multiple
                channels can operate
            on the on-board clock sicne they must be in a single task.
                Bufsize is then
            the buffer size PER CHANNEL.
    This version of setTrig does not implement retriggering, which, for many
        NI
    card version, would require a separate counter to control the sampling
    clock. Instead, this version requires you to stop() then start() the
    task to ready the card for a new trigger.
    """
    CHK(ni.DAQmxCfgSampClkTiming(self.th,"OnBoardClock",float64(samplerate),
                                val_rising,val_finitesamps,uInt64(bufsize)
                                    ))
    CHK(ni.DAQmxCfgDigEdgeStartTrig(self.th,trig,val_rising))
    #CHK(ni.DAQmxSetStartTrigRetriggerable(self.th,uInt32(1)))
    self.trig = True

def voltSequence(self,num_chan,volts):
    """
    voltSequence(volts)
    Load the AO channel with a sequence of voltages found in numpy array
    volts. The points are sequenced through at samplerate speed (Hz) and
    triggered by the channel found in trig, both set with the setTrig method
        .
    volts must have dtypes float64, which can be made, for example,

    volts = linspace(0,1,100,dtype=np.float64)

    although, incidently, float64 is the default dtype for many types of
    numpy arrays, such as that produced with linspace.

            CMS (20160118): I changed voltsequence such that multiple
                channels can operate
            on the on-board clock since they must be in a single task.
                Bufsize set in setTrig
            is then the buffer size PER CHANNEL, and the num_chan is the
                number of AO channels
            that run in the task. The change to the code below is adding the
                 divide in
            uInt32(len(volts)/num_chan). Note that input data is
                noninterleaved.
    """
    ret = int32(0)
    if self.trig:
        if volts.dtype == np.float64:
```

```
        CHK(ni.DAQmxWriteAnalogF64(self.th,uInt32(len(volts)/num_chan),
           uInt32(1),float64(-1),
                                val_groupbychannel,volts.ctypes.data,ctypes.
                                   byref(ret),None))
    else:
        raise RuntimeError('volts must have dtype of float64')
else:
    raise RuntimeError('You must run setTrig before voltSequence')
```

E.2.5. Data Server

The code below provides interfacing between LabVIEW and Python using a TCP/IP server. This is used for reading the camera ROI counts data (App. C.3.3) before moving fluorescence detection to the PMT. Created by MRD.

```
# -*- coding: utf-8 -*-
"""
Created on Thu Mar 26 13:40:49 2015

This is a very simple server intended to accept integer data from some external
source, originally intended to be a camera operating program but could in
principle be anything. The external source should make a TCP/IP connection,
deposit a single integer number less than 16 digits, and then close the
connection. The server will then deposit that number into its queue, and then
will be picked up by the main program the next time getData() is called. Basic
operation:

ds = dataServer('127.0.0.1',4444)
ds.startServer()
data = ds.getData()
ds.stopServer()

@author: lynx
"""

import multiprocessing as mp
import socket

def handleTCP(addr, q, r):
    s = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
    s.bind(addr)
    s.listen(5)
    while r.value:
        client, cl_addr = s.accept()
        data = client.recv(16).strip()
        if data:
            try:
                res = int(data)
                q.put(res)
            except ValueError:
                q.put(-1)
        client.close()
    s.close()

class dataServer(object):
    def __init__(self,ip,port):
        self.ip = ip
        self.port = port
        self.addr = (ip,port)

    def client(self,message):
        """
        Send a message to the server specified when the object was created.
        Notice that if you don’t run startServer(), this method can be used
        to communicate with a server started by another python script.
        """
        sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        try:
            sock.connect(self.addr)
        except:
            return
        try:
            sock.sendall(str(message))
        finally:
            sock.close()

    def startServer(self):
        """
        Starts the server. Server will run in the background until stopServer
        is called. Data can be retrieved using the getData method.
        """
        # Setup the TCP server from which we collect data
        self.q = mp.Queue()
        self.r = mp.Value(’i’,1)
        self.proc = mp.Process(target=handleTCP,args=(self.addr,self.q,self.r))
        self.proc.start()
        self.started = True

    def getData(self):
        """
        Returns the next data point in the queue. If no new data point arrives
        within the next second, fail and return a -1
        """
        try:
            ret = self.q.get(True,1)
        except:
            ret = -1
        return ret

    def stopServer(self):
        """
        Stop the server and shut down the parallel process.
        """
        self.r.value = 0
        self.client('0')
        self.proc.join(5)
        self.started = False

    def __del__(self):
        if self.started:
            self.stopServer()
```

E.2.6. DDS Control Backend

The code below controls the DDS frequencies over USB. Modifications to this code will permit full programming of the DDS. Created by MRD by reverse engineering the DDS GUI interface.

```
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 16 12:55:21 2015

This is a class for controlling an AD9959 DDS evaluation board through its
   onboard USB interface. I hijack the proprietary driver that comes with the
   evaluation software to send commands to the Cypress EZUSB controller that
   comes with the eval board. I had to reverse engineer the API from the VB
   source code of the eval software - notes on this process can be found in
   the Wiki notes from March 2015, AlH+ single ion experiment.

http://odomlab.com/AlH/SingleIon/2015/March

@author: matt
"""


import ctypes
import ctypes.wintypes as wintypes
from ctypes import windll


LPDWORD = ctypes.POINTER(wintypes.DWORD)
LPOVERLAPPED = wintypes.LPVOID
LPSECURITY_ATTRIBUTES = wintypes.LPVOID

GENERIC_READ = 0x80000000
GENERIC_WRITE = 0x40000000
GENERIC_EXECUTE = 0x20000000
GENERIC_ALL = 0x10000000

CREATE_NEW = 1
CREATE_ALWAYS = 2
```

```
OPEN_EXISTING = 3
OPEN_ALWAYS = 4
TRUNCATE_EXISTING = 5

FILE_ATTRIBUTE_NORMAL = 0x00000080

INVALID_HANDLE_VALUE = -1

NULL = 0
FALSE = wintypes.BOOL(0)
TRUE = wintypes.BOOL(1)

def _CreateFile(filename, access, mode, creation, flags):
    """See: CreateFile function

    http://msdn.microsoft.com/en-us/library/windows/desktop/aa363858(v=vs.85).
        aspx

    """
    CreateFile_Fn = windll.kernel32.CreateFileW
    CreateFile_Fn.argtypes = [
            wintypes.LPWSTR,                # _In_        LPCTSTR lpFileName
            wintypes.DWORD,                 # _In_        DWORD dwDesiredAccess
            wintypes.DWORD,                 # _In_        DWORD dwShareMode
            LPSECURITY_ATTRIBUTES,          # _In_opt_    LPSECURITY_ATTRIBUTES
                 lpSecurityAttributes
            wintypes.DWORD,                 # _In_        DWORD
                dwCreationDisposition
            wintypes.DWORD,                 # _In_        DWORD
                dwFlagsAndAttributes
            wintypes.HANDLE]                # _In_opt_    HANDLE hTemplateFile
    CreateFile_Fn.restype = wintypes.HANDLE

    return wintypes.HANDLE(CreateFile_Fn(filename,
                       access,
                       mode,
                       NULL,
                       creation,
                       flags,
                       NULL))
```

```
def _DeviceIoControl(devhandle, ioctl, inbuf, inbufsiz, outbuf, outbufsiz):
    """See: DeviceIoControl function

    http://msdn.microsoft.com/en-us/library/aa363216(v=vs.85).aspx

    """
    DeviceIoControl_Fn = windll.kernel32.DeviceIoControl
    DeviceIoControl_Fn.argtypes = [
            wintypes.HANDLE,                 # _In_          HANDLE hDevice
            wintypes.DWORD,                  # _In_          DWORD dwIoControlCode
            wintypes.LPVOID,                 # _In_opt_      LPVOID lpInBuffer
            wintypes.DWORD,                  # _In_          DWORD nInBufferSize
            wintypes.LPVOID,                 # _Out_opt_     LPVOID lpOutBuffer
            wintypes.DWORD,                  # _In_          DWORD nOutBufferSize
            LPDWORD,                         # _Out_opt_     LPDWORD
                lpBytesReturned
            LPOVERLAPPED]                    # _Inout_opt_ LPOVERLAPPED
                lpOverlapped
    DeviceIoControl_Fn.restype = wintypes.BOOL

    # allocate a DWORD, and take its reference
    dwBytesReturned = wintypes.DWORD(0)
    lpBytesReturned = ctypes.byref(dwBytesReturned)

    status = DeviceIoControl_Fn(devhandle,
                 ioctl,
                 inbuf,
                 inbufsiz,
                 outbuf,
                 outbufsiz,
                 lpBytesReturned,
                 None)

    return status, dwBytesReturned


class DeviceIoControl(object):

    def __init__(self, path):
        self.path = path
        self._fhandle = None
    def _validate_handle(self):
        if self._fhandle is None:
            raise Exception('No file handle')
        if self._fhandle.value == wintypes.HANDLE(INVALID_HANDLE_VALUE).value:
            raise Exception('Failed to open %s. GetLastError(): %d' %
                    (self.path, windll.kernel32.GetLastError()))

    def ioctl(self, ctl, inbuf, inbufsiz, outbuf, outbufsiz):
        self._validate_handle()
        return _DeviceIoControl(self._fhandle, ctl, inbuf, inbufsiz, outbuf,
            outbufsiz)

    def __enter__(self):
        self._fhandle = _CreateFile(
                self.path,
                GENERIC_READ | GENERIC_WRITE,
                0,
                OPEN_EXISTING,
                FILE_ATTRIBUTE_NORMAL)
        self._validate_handle()
        return self

    def __exit__(self, typ, val, tb):
        try:
            self._validate_handle()
        except Exception:
            pass
        else:
            windll.kernel32.CloseHandle(self._fhandle)

class DDSControl(DeviceIoControl):
    #def __init__(self,path,clk):
    #    self.path = path
    #    self.dctl = None
    #    self.clk = clk

    def __init__(self,path,clk):
        self.clk = clk
        super(DDSControl, self).__init__(path)
    #def __enter__(self):
    #    self.dctl = DeviceIoControl(self.path)
    #    self.dctl.__enter__()

    #def __exit__(self, typ, val, tb):
    #    self.dctl.__exit__(typ, val, tb)

    def BulkXfer(self,stri):
        """
        BulkXfer(stri)

        Uses the EZUSB interface to send a bulk transfer command to the DDS.
        stri should be a string of numbers, each of which is converted into a
        byte and delivered to the DDS.
        """
        # From clsEZUSB.cls
        Ezusb_IOCTL_INDEX = 0x800
        METHOD_BUFFERED = 0
        METHOD_IN_DIRECT = 1
        METHOD_OUT_DIRECT = 2

        IOCTL_EZUSB_BULK_READ = 0x220000 + METHOD_OUT_DIRECT + (
           Ezusb_IOCTL_INDEX + 19)*4
        IOCTL_EZUSB_BULK_WRITE = 0x220000 + METHOD_IN_DIRECT + (
           Ezusb_IOCTL_INDEX + 20)*4

        data = [int(i) for i in stri]
        buf = (ctypes.c_byte*len(data))(*data)
        lpbuf = ctypes.pointer(buf)

        ep = (ctypes.c_long)(3)
        lpep = ctypes.pointer(ep)

#        status, junk = self.dctl.ioctl(IOCTL_EZUSB_BULK_WRITE,lpep,ctypes.
    sizeof(ep),lpbuf,ctypes.sizeof(buf))
        status, junk = self.ioctl(IOCTL_EZUSB_BULK_WRITE,lpep,ctypes.sizeof(ep),
           lpbuf,ctypes.sizeof(buf))
        return status

    def BulkRead(self):
            # From clsEZUSB.cls
        Ezusb_IOCTL_INDEX = 0x800
        METHOD_BUFFERED = 0
        METHOD_IN_DIRECT = 1
        METHOD_OUT_DIRECT = 2

        IOCTL_EZUSB_BULK_READ = 0x220000 + METHOD_OUT_DIRECT + (
           Ezusb_IOCTL_INDEX + 19)*4
        IOCTL_EZUSB_BULK_WRITE = 0x220000 + METHOD_IN_DIRECT + (
           Ezusb_IOCTL_INDEX + 20)*4

        buf = ctypes.c_byte(0)
        lpbuf = ctypes.pointer(buf)

        ep = (ctypes.c_long)(0x80)
        lpep = ctypes.pointer(ep)

#        status, junk = self.dctl.ioctl(IOCTL_EZUSB_BULK_WRITE,lpep,ctypes.
    sizeof(ep),lpbuf,ctypes.sizeof(buf))
        status, junk = self.ioctl(IOCTL_EZUSB_BULK_READ,lpep,ctypes.sizeof(ep),
           lpbuf,ctypes.sizeof(buf))
        return status, buf.value

    def PortDUpdate(self,byte):
        """
        PortDUpdate(byte)

        Updates port D with the contents of byte. Usually used to commit
        changes, such as to frequency.
        """
        # From clsEZUSB.cls
        Ezusb_IOCTL_INDEX = 0x800
        METHOD_BUFFERED = 0
        METHOD_IN_DIRECT = 1
        METHOD_OUT_DIRECT = 2

        IOCTL_ESUSB_BULK_READ = 0x220000 + METHOD_OUT_DIRECT + (
           Ezusb_IOCTL_INDEX + 19)*4
        IOCTL_EZUSB_BULK_WRITE = 0x220000 + METHOD_IN_DIRECT + (
           Ezusb_IOCTL_INDEX + 20)*4

        buf = (ctypes.c_byte*2)(0x0C, byte)
        lpbuf = ctypes.pointer(buf)

        # EP is the endpoint, minus 1 actually. So to issue a port update
           command,
        # we go to ep = 1-1. To read back from the firmware, we use ep=2-1. To
        # write to a register we use ep=4-1=3, to read a register we use ep
           =6-1=5.
        ep = (ctypes.c_long)(0)
        lpep = ctypes.pointer(ep)

        #status, junk = self.dctl.ioctl(IOCTL_EZUSB_BULK_WRITE,lpep,ctypes.
           sizeof(ep),lpbuf,ctypes.sizeof(buf))
        status, junk = self.ioctl(IOCTL_EZUSB_BULK_WRITE,lpep,ctypes.sizeof(ep),
           lpbuf,ctypes.sizeof(buf))
        return status

    def SelectChannel(self,chan):
        """
        SelectChannel(chan)

        Sets the current channel to chan, which should be a number 0-3,
           inclusive.
        """
        chlist = [’0001’, ’0010’, ’0100’, ’1000’]
        # The proper command is 3 zeros, then the register as a 5 ’bit’ string.
        # Since the register for select channel is zero, this is just 8 zeros.
        # Next is the value to place in this register, which for us will just be
        # the value for chlist followed by 4 zeros.
        stri = ’’.zfill(8) + chlist[chan] + ’’.zfill(4)
        return self.BulkXfer(stri)

    def SetFrequency(self,freq):
        """
        SetFrequency(freq)

        Changes the register for the current channel (set by SetChannel) to freq
           ,
        given in absolute units. Then issues commands to commit this change,
        thereby immediately changing the frequency of the DDS channel in
           question.
        """
        assert freq <= self.clk
        norm_freq = int(round(0xFFFFFFFF*(float(freq)/float(self.clk))))
        # The register for controlling the frequency is 4, and the data is the
            the
        # (normalized) frequency, padded to 32 'bits'
        stri = bin(4)[2:].zfill(8) + bin(norm_freq)[2:].zfill(32)
        # The norm_freq is calculated as 0xFFFFFFFF*(abs_freq/clk_freq). See AN
            -621.
        ret = self.BulkXfer(stri)
        thirdret = self.PortDUpdate(0)
        stat1, readone = self.BulkRead()
        fourthret = self.PortDUpdate(0x10)
        stat2, readtwo = self.BulkRead()
        return ret + thirdret + fourthret + stat1 + stat2, readone, readtwo

    def SetAmplitude(self,amp):
        """
        SetFrequency(freq)

        Changes the register for the current channel (set by SetChannel) to freq
            ,
        given in absolute units. Then issues commands to commit this change,
        thereby immediately changing the frequency of the DDS channel in
            question.
        """
        assert amp <= 1.0
        norm_amp = int(round(1023*float(amp)))
        # The register for controlling the frequency is 4, and the data is the
            the
        # (normalized) frequency, padded to 32 'bits'
        stri = bin(6)[2:].zfill(8) + bin(norm_amp)[2:].zfill(24)
        # The norm_freq is calculated as 0xFFFFFFFF*(abs_freq/clk_freq). See AN
            -621.
        ret = self.BulkXfer(stri)
        thirdret = self.PortDUpdate(0)
        stat1, readone = self.BulkRead()
        fourthret = self.PortDUpdate(0x10)
        stat2, readtwo = self.BulkRead()
        return ret + thirdret + fourthret + stat1 + stat2, readone, readtwo
```

E.2.7. PMT Pulse Counting Backend

The code below generates a class for reading data from the PMT using a 6602 counter input channel. Created by MRD.

```
# -*- coding: utf-8 -*-
"""
Created on Thu Jun 25 16:06:03 2015

A class for reading data from the PMT.

@author: lynx
"""

import ctypes
import numpy as np

# This needs to point ctypes to the local DAQmx drivers
ni = ctypes.windll.nicaiu

int32 = ctypes.c_long
uInt32 = ctypes.c_ulong
uInt64 = ctypes.c_ulonglong
float64 = ctypes.c_double

# Select configuration constants from the file NIDaqmx.h
config_default = ctypes.c_int32(-1)

# To set the pulse duration units
val_seconds = int32(10364)

# To set the resting state for the counter
val_high = int32(10192)
val_low = int32(10214)

# To set rising or falling edge for trigger
val_rising = int32(10280)
val_falling = int32(10171)

val_count_up = int32(10128)
```

```
val_count_down = int32(10124)
val_ext_controlled = int32(10326)

val_seconds = int32(10364)
val_ticks = int32(10304)
val_sampclkperiods = int32(10286)

val_finite = int32(10178)
val_contsamps = int32(10123)
val_hwtimed = int32(12522)

val_LowFreq1Ctr = int32(10105)
val_HighFreq2Ctr = int32(10157)
val_LargeRng2Ctr = int32(10205)

val_dma = int32(10054)
val_interrupts = int32(10204)
val_programmedio = int32(10264)

class DataOverwritten(Exception):
    pass


def CHK(err):
    """a simple error checking routine"""
    if err < 0:
        if err == -200141:
            raise DataOverwritten
        buf_size = 4000
        buf = ctypes.create_string_buffer('\000' * buf_size)
        ni.DAQmxGetErrorString(err,ctypes.byref(buf),buf_size)
        raise RuntimeError('nidaq call failed with error %d: %s'%(err,repr(buf.
            value)))

def DAQmxVersion():
    """
    DAQmxVersion()

    Returns a tuple representing the version of DAQmx. (major version number,
        minor version number, update number).
    """
    major = uInt32(0)
    minor = uInt32(0)
    update = uInt32(0)
    CHK(ni.DAQmxGetSysNIDAQMajorVersion(ctypes.byref(major)))
    CHK(ni.DAQmxGetSysNIDAQMinorVersion(ctypes.byref(minor)))
    CHK(ni.DAQmxGetSysNIDAQUpdateVersion(ctypes.byref(update)))
    return (major.value,minor.value,update.value)

class PMT():
    def __init__(self,chan):
        """
        PMT(chan)

        chan is the name of a counter devoted to reading TTL events from
            something
        like a PMT. If you set chan to be /dev4/ctr0, for instance, the PMT
            output
        should be connected to the SRC of ctr0 on device 4. The object is
            intended
        to support several readout modes, the simplest for example is setRead()
            and
        getRead().
        """
        self.chan = chan
        self.reading = False
        self.sequence = False
        self.th = uInt32(0)

    def startRead(self):
        """
        startRead()

        Just start counting PMT events. Use software delay to set the length
        of time to read data, e.g.

        pmt.startRead()
        time.sleep(.4)
        data = pmt.getRead()
        """
        if not self.reading:
            CHK(ni.DAQmxCreateTask("",ctypes.byref(self.th)))
            CHK(ni.DAQmxCreateCICountEdgesChan(self.th,self.chan,"",val_rising,
               uInt32(0),val_count_up))
            CHK(ni.DAQmxStartTask(self.th))
            self.reading = True

    def getRead(self):
        """
        getRead()

        Returns the number of counts since last startRead(). Does nothing if
        startRead() not run.
        """
        if self.reading:
            value = uInt32(0)
            CHK(ni.DAQmxReadCounterScalarU32(self.th,float64(-1),ctypes.byref(
               value),None))
            CHK(ni.DAQmxStopTask(self.th))
            CHK(ni.DAQmxClearTask(self.th))
            self.reading = False
            return value.value

    def setSequenceRead(self,bufsize=1000):
        """
        setSequenceRead(bufsize=1000)

        Prepares to collect a sequence of time delays from the counter. For
        each start signal (recieved on Aux), this will detect exactly one end
        signal, recieved on Gate. The result is buffered, so if you use 1000
        starts and 1000 stops, there will be 1000 points waiting for you.

        Just like startRead and getRead, just start setSequence, wait for a
        trigger and then some time after that, and then use getSequenceRead()
        to retrieve the result.
        pmt.setSequenceRead()
        send trigger here...
        time.sleep(.4)
        data = pmt.getSequenceRead()

        The start signal should be on Aux, and the end signal should be on Gate.
        """
        self.bufsize = bufsize
        if not self.sequence:
            CHK(ni.DAQmxCreateTask("",ctypes.byref(self.th)))
            CHK(ni.DAQmxCreateCITwoEdgeSepChan(self.th,self.chan,"",float64(1e
               -6),
                                          float64(10),val_seconds,val_rising,
                                              val_rising,""))
            #CHK(ni.DAQmxCfgDigEdgeStartTrig(self.th,trig,val_rising))
            CHK(ni.DAQmxCfgImplicitTiming(self.th,val_contsamps,uInt64(self.
               bufsize)))
            CHK(ni.DAQmxStartTask(self.th))
            self.sequence = True

    def getSequenceRead(self):
        """
        getSequenceRead()

        Returns a numpy array with the requested data. See setSequenceRead().
        """
        if self.sequence:
            returned = int32(0)
            ret = np.zeros((self.bufsize,))
            CHK(ni.DAQmxReadCounterF64(self.th,int32(-1),float64(1),ret.ctypes.
               data,
                                  uInt32(self.bufsize),ctypes.byref(returned),
                                      None))
            CHK(ni.DAQmxStopTask(self.th))
            CHK(ni.DAQmxClearTask(self.th))
            self.sequence = False
            return ret

    def setMultiSequenceRead(self,bufsize=1000,timebase="80MHzTimeBase",
        sampclock=None,trigchan=None):
        """
        setMultiSequenceRead(bufsize=1000,timebase="80MHzTimeBase",sampclock=
            None,trigchan=None)

        Starts a Count Edges task, counting edges on the channel 'timebase', and
             sampling the
        counter on the events given by 'sampclock'. If trigchan is provided,
            this channel
        is used to reset the counter value to zero. trigchan and sampclock are
            incompatible
        at least on some software.
        """
        if not self.sequence:
            CHK(ni.DAQmxCreateTask("",ctypes.byref(self.th)))
            CHK(ni.DAQmxCreateCICountEdgesChan(self.th,self.chan,"",val_rising,
                uInt32(0),val_count_up))
            CHK(ni.DAQmxSetCICountEdgesTerm(self.th,self.chan,timebase))
            if trigchan:
                CHK(ni.DAQmxSetCICountEdgesCountResetTerm(self.th,self.chan,
                    trigchan))
                CHK(ni.DAQmxSetCICountEdgesCountResetEnable(self.th,self.chan,
                    uInt32(1)))
            if sampclock:
                CHK(ni.DAQmxCfgSampClkTiming(self.th,sampclock,float64(1e7),
                    val_rising,val_contsamps,uInt64(bufsize)))
            CHK(ni.DAQmxStartTask(self.th))
            self.sequence = True
            self.bufsize = bufsize

    def getMultiSequenceRead(self,num=False):
        if self.sequence:
            returned = int32(0)
            ret = np.zeros((self.bufsize,),dtype=np.dtype('uint32'))
            CHK(ni.DAQmxReadCounterU32(self.th,int32(-1),float64(-1),ret.ctypes.
                data,uInt32(self.bufsize),ctypes.byref(returned),None))
            #CHK(ni.DAQmxStopTask(self.th))
            #CHK(ni.DAQmxClearTask(self.th))
            #self.sequence = False
            if num:
                return (ret,returned.value)
            else:
                return ret

    def setPeriod(self,bufsize=1000):
        """
        setPeriod(bufsize=1000)

        Begins a period measurement. This returns the amount of time in between
            subsequent pulses arriving
        at the counter’s gate. Nothing synchronizes the first pulse, so I
            suggest you somehow set the absolute
        time of that TTL pulse.

        Use getPeriod to retrieve the data and stop the sequence.
        """
        if not self.sequence:
            CHK(ni.DAQmxCreateTask("",ctypes.byref(self.th)))
            CHK(ni.DAQmxCreateCIPeriodChan(self.th,self.chan,"",float64(2e-8),
                float64(10),val_seconds,val_rising,val_LowFreq1Ctr,float64(1),
                uInt32(1),""))
            CHK(ni.DAQmxCfgImplicitTiming(self.th,val_contsamps,uInt64(10*
                bufsize)))
            #CHK(ni.DAQmxSetCIPeriodDigSyncEnable(self.th,self.chan,uInt32(1)))
            CHK(ni.DAQmxSetCIDataXferMech(self.th,self.chan,val_dma))
            CHK(ni.DAQmxStartTask(self.th))
            self.sequence = True
            self.bufsize = bufsize

    def getPeriod(self,num=False):
        """
        getPeriod(num=False)

        Return data prepped by setPeriod. If num is True, returns a tuple
        whose second element is the number of points collected.
        """
        if self.sequence:
            returned = int32(0)
            ret = np.zeros((self.bufsize,),dtype=np.dtype(’float64’))
            CHK(ni.DAQmxReadCounterF64(self.th,int32(-1),float64(-1),ret.ctypes.
                data,uInt32(self.bufsize),ctypes.byref(returned),None))
            #CHK(ni.DAQmxStopTask(self.th))
            #CHK(ni.DAQmxClearTask(self.th))
            #self.sequence = False
            if num:
                return (ret, returned.value)
            else:
                return ret

    def stop(self):
        if self.th.value != 0:
            ni.DAQmxStopTask(self.th)
            ni.DAQmxClearTask(self.th)
            self.sequence = False
            self.reading = False

    def __del__(self):
        if self.th.value != 0:
            ni.DAQmxStopTask(self.th)
            ni.DAQmxClearTask(self.th)
```

E.2.8. Trigger Control

The code below defines a class for running an experimental trigger to initialize experimental cycles. Created by MRD.

```
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 16 17:33:34 2015

@author: matt
"""

import ctypes
import numpy as np

# This needs to point ctypes to the local DAQmx drivers
ni = ctypes.windll.nicaiu

int32 = ctypes.c_long
uInt32 = ctypes.c_ulong
uInt64 = ctypes.c_ulonglong
float64 = ctypes.c_double

# To set the pulse duration units
val_seconds = int32(10364)

# To set the resting state for the counter
val_high = int32(10192)
val_low = int32(10214)

# To set rising or falling edge for trigger
val_rising = int32(10280)
val_falling = int32(10171)

val_count_up = int32(10128)
val_count_down = int32(10124)
val_ext_controlled = int32(10326)

val_seconds = int32(10364)
val_ticks = int32(10304)
```

```
val_sampclkperiods = int32(10286)

val_finite = int32(10178)
val_contsamps = int32(10123)
val_hwtimed = int32(12522)

val_bychannel = int32(0)
val_byscannumber = int32(1)

# Select configuration constants from the file NIDaqmx.h
config_default = ctypes.c_int32(-1)
do_not_invert_polarity = ctypes.c_int32(0)
invert_polarity = ctypes.c_int32(1)
chan_per_line = ctypes.c_int32(0)
chan_for_all_lines = ctypes.c_int32(1)

def CHK(err):
    """a simple error checking routine"""
    if err < 0:
        buf_size = 100
        buf = ctypes.create_string_buffer(’\000’ * buf_size)
        ni.DAQmxGetErrorString(err,ctypes.byref(buf),buf_size)
        raise RuntimeError(’nidaq call failed with error %d: %s’%(err,repr(buf.
            value)))

class DOControl(object):
    def __init__(self,trig):
        #self.source = source
        self.trig = trig
        self.th = int32(0)
        #CHK(ni.DAQmxConnectTerms(source,trig,do_not_invert_polarity))
        CHK(ni.DAQmxCreateTask("",ctypes.byref(self.th)))
        CHK(ni.DAQmxCreateDOChan(self.th,trig,"",chan_per_line))
        CHK(ni.DAQmxStartTask(self.th))

    def setHigh(self):
        CHK(ni.DAQmxWriteDigitalScalarU32(self.th,int32(1),float64(0),uInt32(0
            xFFFFFFFF),None))

    def setLow(self):
        CHK(ni.DAQmxWriteDigitalScalarU32(self.th,int32(1),float64(0),uInt32(0),
           None))

    def writeSequence(self,data,samprate=1000):
        """
        writeSequence(data,samprate=1000)

        data must be a 1D numpy array filled with 1's and 0's. Points are output
             at
        a rate of samprate per second.
        """
        sendme = np.zeros(data.shape,dtype=np.uint8)
        sendme[data > 0] = 0xFF
        CHK(ni.DAQmxStopTask(self.th))
        CHK(ni.DAQmxCfgSampClkTiming(self.th,"OnboardClock",float64(samprate),
           val_rising,val_contsamps,uInt64(data.shape[0])))
        CHK(ni.DAQmxWriteDigitalLines(self.th,int32(data.shape[0]),uInt32(0.1),
           float64(-1),val_bychannel,sendme.ctypes.data,None))
        CHK(ni.DAQmxStartTask(self.th))

    def stop(self):     #CHK(ni.DAQmxDisconnectTerms(self.source,self.trig))
        if self.th.value != 0:
           ni.DAQmxStopTask(self.th)
           ni.DAQmxClearTask(self.th)

    def __del__(self):     #CHK(ni.DAQmxDisconnectTerms(self.source,self.trig))
        if self.th.value != 0:
           ni.DAQmxStopTask(self.th)
           ni.DAQmxClearTask(self.th)

class TrigControl(DOControl):
    def sendTriggers(self,period,number):
        self.writeSequence(data)
```

E.3. Python Frontend

The Python environment is used for data collection and analysis in the single-ion motional ground state cooling project. The section below contains the frontend scripts that are used to perform experiments and collect data. All scripts are run using Spyder, a GUI interface allowing multiple scripts to run simultaneously as long as the hardware for each is not shared.

E.3.1. Continuous NR Raman Spectrum

The code below records a NR Raman spectrum with data points randomized and interleaved. The version below uses no shutters. Additional features may be included with minor changes such as AI voltage monitoring.

```
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 16 15:09:02 2015

The main code for executing a frequency scan with STIRAP.

@author: matt
"""

from OneShot import OneShot
from DDSControl import DDSControl
from TrigControl import TrigControl
import time
import numpy as np
import os
import pdb
from plotoneshots import plotOneShots
from pmt import PMT

# How much data to take at each setting
```

```
reps = 50
scriptreps = 40

# The Zeeman splitting should be about 12.3 MHz at 1 A on the horizontal coils
# Divided by 2 due to the double pass.
# But I don’t know if the right frequency is positive or negative
#frequencies = np.linspace(0,-50,250)
frequencies1 = np.linspace(-11,-13.5,50)
frequencies2 = np.linspace(-9.5,-11,30)
frequencies12 = np.concatenate((frequencies1,frequencies2))
#frequencies12 = np.linspace(-8,-15,700)
frequencies = np.repeat(frequencies12,scriptreps)
frequencies = np.random.permutation(frequencies)

#Wait for line trigger time. Should be either zero or 40 ms.
line_t = 0.0
#line_t = 40e-3

# The path to save the data in
path = ’c:/data/CW Raman Fluorescence/’

# If this is true, then plot the ctimes sequence
plot_me = False
#PMT Setup
pmtc = ’/dev1/ctr7’
pmt = PMT(pmtc)

trigchan1 = ’/dev1/PFI14’
trigchan5 = ’/dev5/PFI3’
trigchan6 = ’/dev6/PFI14’
trigsource = ’/dev5/port0/line0’
r_doppler_c = ’/dev5/port0/line4’

# A ’safe interval’, which is a period added in to various places to ensure
# the light has actually turned off, or on. Also used as the duration of a
# trig pulse.
safe = 1e-5
# The amount of time to allow for the mechanical shutter to close or open
shutter_t = 10e-3
# The amount of time to optically pump for. Can safely be long.
op_t = 1e-5
```

```
#op_t = 0.0
# An interval set aside for the STIRAP pulses. Should be significantly longer
# than the STIRAP pulses
stirap_t = 2.0e-5
# A period after STIRAP where Raman A is left on, with no red, to pump
# any transfered atoms to the D(3/2) state. This interval is potentially
# sensitive, since if its too short, the ion won't be pumped, and if too long
# all atoms may be pumped by the imperfect B field trimming
protect_t = 1e-5
#stirap_protect_t = 10e-6
stirap_protect_t = 0.0
# The amount of time to use 455 nm light to shelve any remaining ions into
# D(5/2)
shelve_t = 5e-3
# Expose time of camera
expose_t = 1.0e-3
# Deshelving time
deshelve_t = 50e-3
# Recovery time between cycles to allow ion to cool
recover_t = 5e-3

#Wait time for equilibrium
equil_t = 0.0
#equil_t = 0


if __name__ == '__main__':
    # Setup trigger out
    trig = TrigControl(trigsource)

    # Manually set the shutter states. Only use the b_shutter in experiment.
    try:
        del clist
    except NameError:
        print
    oneshots = {}
    clist = {'b_shutter': ['/Dev5/Ctr2',trigchan5,False],
             'r_shutter': ['/Dev5/Ctr3',trigchan5,False],
             'shelve_shutter': ['/Dev1/Ctr3',trigchan1,True],
             'raman_shutter': ['/Dev1/Ctr2',trigchan1,True], #False due to using
                  OR + NOT
        'raman_shutter_protect': ['/Dev5/Ctr0',trigchan5,True], #False due
           to using OR + NOT
        'FOR_shutter': ['/Dev6/Ctr0',trigchan6,True],
        'deshelve_shutter': ['/Dev6/Ctr1',trigchan6,True]}

ctimes = {'b_shutter': [0.0, 2*shutter_t],
         'r_shutter': [0.0, 2*shutter_t],
         'shelve_shutter': [0.0, 2*shutter_t],
         'raman_shutter': [0.0, 2*shutter_t],
         'raman_shutter_protect': [0.0, 2*shutter_t],
         'FOR_shutter': [0.0, 2*shutter_t],
         'deshelve_shutter': [0.0,2*shutter_t]}
for k in ctimes.keys():
    oneshots[k] = OneShot(*(clist[k]+ctimes[k]))
for k in oneshots.keys():
    oneshots[k].begin()
trig.setHigh()
time.sleep(2*shutter_t)
trig.setLow()
time.sleep(recover_t)
for k in oneshots.keys():
    oneshots[k].stop()
del oneshots
del clist
oneshots = {}

# A dictionary of all the relevant channels used below.
# Doppler blue and red should be set to be ordinarily on, all others off
# False due to OR + NOT.
clist = {'b_doppler': ['/Dev5/Ctr6',trigchan5,True],
        'NR_sigma': ['/Dev5/Ctr4',trigchan5,False],
        'NR_pi': ['/Dev5/Ctr5',trigchan5,False],
        'camera': ['/Dev5/Ctr1',trigchan5,False],
        #'b_shutter': ['/Dev5/Ctr2',trigchan5,False],
        #'raman_shutter': ['/Dev1/Ctr2',trigchan1,False],
        'NR_sigma_protect': ['/Dev1/Ctr1',trigchan1,False],
        #'raman_shutter_protect': ['/Dev5/Ctr0',trigchan5,False]
        }

cwraman_b = safe
cwraman_e = cwraman_b + equil_t + expose_t
```

```
cooling_b = cwraman_e + safe

ctimes = {'b_doppler': [0.0,cooling_b],
          #'b_shutter': [0.0,cwraman_e],
          'NR_sigma': [cwraman_b,cwraman_e],
          #'raman_shutter': [shutter_t,cwraman_e],
          'NR_pi': [cwraman_b,cwraman_e],
          'camera': [cwraman_b + equil_t,cwraman_b + equil_t + expose_t],
          #'NR_sigma_protect': [0.0,safe],
          #'raman_shutter_protect': [shutter_t*2,shutter_t*2+safe]
          }
# First initialize all the oneshots
for k in ctimes.keys():
    oneshots[k] = OneShot(*(clist[k]+ctimes[k]))

# Then start all the oneshots
for k in oneshots.keys():
    try:
        oneshots[k].begin()
    except:
        pdb.post_mortem()
if plot_me:
    hold = plotOneShots(ctimes)

time.sleep(1) # Give the user some time to start the camera program

# The first column is the frequencies, all others are the data collected
# for that frequency
data = np.zeros((frequencies.shape[0],reps+1))
with DDSControl(r'\\.\Ezssp-0',500.0) as dds:
    for (i,f) in enumerate(frequencies):
        # Set the dds frequencies. I am making Raman A be the negative one.
        # Also, I am making them be symmetric around 80, because I can.
        dds.SelectChannel(2)
        #dds.SetFrequency(80.0-f/2.0)
        dds.SetFrequency(80.0-11/2.0)
        dds.SelectChannel(3)
        #dds.SetFrequency(80.0+f/2.0)
        dds.SetFrequency(80.0-11/2.0-f/2)
        data[i,0] = f
        print('% of Scan Completed:')
        print((i+1.0)/frequencies.shape[0]*100.0)
        for j in range(reps):
            pmt.startRead()
            trig.setHigh()
            time.sleep(cooling_b+line_t)
            res = pmt.getRead()
            trig.setLow()
            data[i,j+1] = res
            time.sleep(recover_t) # Make sure that the cycle has finished

# Build a suitable filename
t = time.localtime()
filename = "{}{:02g}{:02g}".format(t.tm_year,t.tm_mon,t.tm_mday)+ "
    _CWRamanscanPMT_NS"
i = 0
while os.path.exists(os.path.normpath(path + filename + "_{}.npy".format(i)
    )):
    i = i + 1
filenamen = os.path.normpath(path + filename + "_{}.npy".format(i))

# And save the data
np.save(filenamen,data)

for k in oneshots.keys():
    oneshots[k].stop()
del oneshots
del clist
del ctimes
oneshots = {}

clist = {’b_shutter’: [’/Dev5/Ctr2’,trigchan5,False],
        ’r_shutter’: [’/Dev5/Ctr3’,trigchan5,False],
        ’shelve_shutter’: [’/Dev1/Ctr3’,trigchan1,True],
        ’raman_shutter’: [’/Dev1/Ctr2’,trigchan1,False], #False due to
            using OR + NOT
        ’raman_shutter_protect’: [’/Dev5/Ctr0’,trigchan5,False], #False due
             to using OR + NOT
        ’FOR_shutter’: [’/Dev6/Ctr0’,trigchan6,True],
        ’deshelve_shutter’: [’/Dev6/Ctr1’,trigchan6,True]}

ctimes = {’b_shutter’: [0.0, 2*shutter_t],
         'r_shutter': [0.0, 2*shutter_t],
         'shelve_shutter': [0.0, 2*shutter_t],
         'raman_shutter': [0.0, 2*shutter_t],
         'raman_shutter_protect': [0.0, 2*shutter_t],
         'FOR_shutter': [0.0, 2*shutter_t],
         'deshelve_shutter': [0.0,2*shutter_t]}
for k in ctimes.keys():
    oneshots[k] = OneShot(*(clist[k]+ctimes[k]))
for k in oneshots.keys():
    oneshots[k].begin()
trig.setHigh()
time.sleep(2*shutter_t)
trig.setLow()
time.sleep(recover_t)
for k in oneshots.keys():
    oneshots[k].stop()
del oneshots
del clist

f = -11
with DDSControl(r'\\.\Ezssp-0',500.0) as dds:
    dds.SelectChannel(2)
    dds.SetFrequency(80.0-11/2.0)
    dds.SelectChannel(3)
    dds.SetFrequency(80.0-11/2-f/2)
trig.stop()
del trig
pmt.stop()
del pmt
print('Im Done')
```

E.3.2. DDS Frequency Reset

The code below resets the far-off-resonant Raman frequencies to $f = 0$ and the near-resonant Raman frequencies to the specified f. This is a good reset program to run to collect power measurements under the same operating AOM frequencies.

```
# -*- coding: utf-8 -*-
"""
Created on Tue Apr 28 19:15:10 2015

@author: lynx
"""

from DDSControl import DDSControl

f = -11
#f = 0

print(f)

with DDSControl(r'\\.\Ezssp-0',500.0) as dds:
    dds.SelectChannel(0)
    #dds.SetFrequency(80.0-f/2.0)
    dds.SetFrequency(80.0-11/2.0)
    dds.SelectChannel(2)
    #dds.SetFrequency(80.0-f/2.0)
    dds.SetFrequency(80.0-11/2.0)
    dds.SelectChannel(1)
    #dds.SetFrequency(80.0+f/2.0)
    dds.SetFrequency(80.0-11/2.0-f/2)
    dds.SelectChannel(3)
    #dds.SetFrequency(80.0+f/2.0)
    dds.SetFrequency(80.0-11/2.0-f/2)
```

E.3.3. Deshelving Test

The code below tests the 614 nm deshelve time for maximum deshelving efficiency.

```
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 16 15:09:02 2015

The main code for executing a frequency scan with STIRAP.

@author: matt
"""

from OneShot import OneShot
from TrigControl import TrigControl
import time
import numpy as np
import os
import pdb
from plotoneshots import plotOneShots
from pmt import PMT
from AIWrite import AOWrite

# How much data to take at each setting
reps = 200

# The path to save the data in
path = 'c:/data/Deshelving/'

# If this is true, then plot the ctimes sequence
plot_me = False

trigchan1 = '/dev1/PFI14'
trigchan5 = '/dev5/PFI3'
trigchan6 = '/dev6/PFI14'
trigsource = '/dev5/port0/line0'

# Expose time of camera
expose_t = 5e-3
```

```
# Change the Doppler cooling settings for ion detection after shelving. This
# sets up the AO task. It has to be defined as a single task to share the
# on-board clock. AO1 = b_FM, AO4 = b_AM, AO5 = r_FM.
trigchan2_camera = '/dev2/PFI7'
b_detect_FM_v = 0.25
b_cool_FM_v = 0.10
b_FM_voltages = np.array([b_detect_FM_v,b_cool_FM_v],dtype=np.float64)
b_detect_AM_v = -1.5
b_cool_AM_v = -0.15
b_AM_voltages = np.array([b_detect_AM_v,b_cool_AM_v],dtype=np.float64)
r_detect_AM_v = -0.10
r_cool_AM_v = -0.02
r_AM_voltages = np.array([r_detect_AM_v,r_cool_AM_v],dtype=np.float64)
AO_detect_voltages = np.concatenate((b_FM_voltages,b_AM_voltages,r_AM_voltages)
   )
AO_detect_control = AOWrite("/dev2/ao1,/dev2/ao4:5")
AO_detect_control.setTrig(1/expose_t,2,trigchan2_camera)
AO_detect_control.voltSequence(3,AO_detect_voltages)

#PMT info
pmtc = '/dev1/ctr7'
pmt = PMT(pmtc)

# A dictionary of all the relevant channels used below.
# Doppler blue and red should be set to be ordinarily on, all others off
clist = {'b_doppler': ['/Dev5/Ctr6',trigchan5,True],
        'r_doppler': ['/Dev1/Ctr4',trigchan1,False],
        'shelve': ['/Dev1/Ctr5',trigchan1,False],
        'deshelve': ['/Dev1/Ctr6',trigchan1,False],
        'deshelve_shutter': ['/Dev6/Ctr1',trigchan6,True],
        'camera': ['/Dev5/Ctr1',trigchan5,False],
        'b_shutter': ['/Dev5/Ctr2',trigchan5,False],
        'r_shutter': ['/Dev5/Ctr3',trigchan5,False],
        'shelve_shutter': ['/Dev1/Ctr3',trigchan1,True]}
#        'raman_shutter': ['/Dev1/Ctr2',trigchan1,False]}

# A 'safe interval', which is a period added in to various places to ensure
# the light has actually turned off, or on. Also used as the duration of a
# trig pulse.
safe = 1e-5
# Time to allow for the mechanical shutter to close
```

```
shutter_t = 10e-3

# Time to expose the Shelving laser for
shelve_t = 10e-3
# Deshelving time
deshelve_t = 15e-3
# Recovery time between cycles to allow ion to cool
recover_t = 20e-3

shutter_open_b = shelve_t+shutter_t*2
shutter_open_r = shutter_open_b-shutter_t*2
deshelve_b = shutter_open_b+shutter_t*2
camera_b = deshelve_b+deshelve_t+2*shutter_t

ctimes = {'b_doppler': [0.0,shutter_open_b+shutter_t],
         'b_shutter': [0.0,shutter_open_b],
         'r_doppler': [shutter_t,shutter_open_b+shutter_t],
         'r_shutter': [shutter_t,shutter_open_b],
         'shelve': [2*shutter_t,shutter_open_b],
         'shelve_shutter': [shutter_t, shutter_open_b],
         'deshelve': [deshelve_b+shutter_t,deshelve_b+deshelve_t+shutter_t],
         'deshelve_shutter': [deshelve_b,deshelve_b+deshelve_t+shutter_t],
         'camera': [camera_b+safe,camera_b+safe+expose_t]}

if __name__ == '__main__':
    # Setup trigger out
    trig = TrigControl(trigsource)

    # And now all the oneshots
    oneshots = {}
    # First initialize all the oneshots
    for k in ctimes.keys():
        oneshots[k] = OneShot(*(clist[k]+ctimes[k]))

    # Then start all the oneshots
    for k in oneshots.keys():
        try:
            oneshots[k].begin()
        except:
            pdb.post_mortem()
    if plot_me:
```

```
    tmp = plotOneShots(ctimes)

time.sleep(1) # Give the user some time to start the camera program

# The first column is the frequencies, all others are the data collected
# for that frequency
data = np.zeros((reps,))
for j in range(reps):
    AO_detect_control.start()
    pmt.startRead()
    trig.setHigh()
    time.sleep(camera_b+safe+expose_t)
    res = pmt.getRead()
    trig.setLow()
    data[j] = res
    time.sleep(recover_t) # Make sure that the cycle has finished
    AO_detect_control.stop()
    print('% of Scan Completed:')
    print((j+1.0)/reps*100.0)



# Build a suitable filename
t = time.localtime()
filename = "{}{:02g}{:02g}".format(t.tm_year,t.tm_mon,t.tm_mday)+ "
    _deshelvingPMT"
i = 0
while os.path.exists(os.path.normpath(path + filename + "_{}.npy".format(i)
    )):
    i = i + 1
filenamen = os.path.normpath(path + filename + "_{}.npy".format(i))

# And save the data
np.save(filenamen,data)

for k in oneshots.keys():
    oneshots[k].stop()
del oneshots
trig.stop()
del trig
pmt.stop()
```

```
del pmt
AO_detect_control.clear()
del AO_detect_control
print('Im Done')
```

E.3.4. Doppler Curve

The code below runs Doppler fluorescence spectra used to monitor/set Doppler cooling laser lock frequencies and powers. This iteration uses 100 ms exposure times with no randomization or interleaving of data points.

```
# -*- coding: utf-8 -*-
"""
Created on Wed Apr 15 10:12:33 2015

@author: lynx
"""

from OneShot import OneShot
from TrigControl import TrigControl
import time
import numpy as np
import os
import pdb
from AIWrite import AOWrite
import bokeh.plotting as bk
from pmt import PMT

cooling_v = 0.10
voltages = np.linspace(-0.4,0.4,50,dtype=np.float64)
voltages = np.random.permutation(voltages)
#voltages = np.array([0.0],dtype=np.float64)

#Wait for line trigger time. Should be either zero or 40 ms.
line_t = 0.0
#line_t = 40e-3

# How much data to take at each setting
reps = 10
# Recovery time between cycles to allow ion to cool
recover_t = 20e-3

#PMT Exposure time.
expose_t = 100e-3
```

```
safe = 1e-5

# The path to save the data in
path = 'c:/data/Fluorescence Curves/'

#PMT Setup
pmtc = '/dev1/ctr7'
pmt = PMT(pmtc)

trigchan1 = '/dev1/PFI14'
trigchan5 = '/dev5/PFI3'
trigchan2 = '/dev2/PFI6'
trigsource = '/dev5/port0/line0'

clist = {'b_doppler': ['/Dev5/Ctr6',trigchan5,True],
        'r_doppler': ['/Dev1/Ctr4',trigchan1,False],
        'shelve': ['/Dev1/Ctr5',trigchan1,False],
        'deshelve': ['/Dev1/Ctr6',trigchan1,False],
        'camera': ['/Dev5/Ctr1',trigchan5,False],
        'b_shutter': ['/Dev5/Ctr2',trigchan5,False],
        'r_shutter': ['/Dev5/Ctr3',trigchan5,False],
        'shelve_shutter': ['/Dev1/Ctr3',trigchan1,True]}

#The timings below are for equilibrium cooling behavior.
ctimes = {'camera': [.2,.2+expose_t]}

#The timings below are for non-equilibrium behavior.
#ctimes = {'camera': [.1+safe,.1+safe+expose_t]}

if __name__ == "__main__":
    freqcontrol = AOWrite("/dev2/ao1")
    #freqcontrol = AOWrite("/dev2/ao3")
    freqcontrol.setTrig(10,4,trigchan2)
    # Setup trigger out
    trig = TrigControl(trigsource)

    # And now all the oneshots
    oneshots = {}
    # First initialize all the oneshots
    for k in ctimes.keys():
```

```
    oneshots[k] = OneShot(*(clist[k]+ctimes[k]))

# Then start all the oneshots
for k in oneshots.keys():
    try:
        oneshots[k].begin()
    except:
        pdb.post_mortem()
time.sleep(1)
# The first column is the frequencies, all others are the data collected
# for that frequency
data = np.zeros((voltages.shape[0],reps+1))
for i,v in enumerate(voltages):
    volts = np.array([cooling_v, v, v, cooling_v])
    freqcontrol.voltSequence(1,volts)
    data[i,0] = v
    print('% of Scan Completed:')
    print((i+1.0)/voltages.shape[0]*100.0)
    for j in range(reps):
        pmt.startRead()
        freqcontrol.start()
        trig.setHigh()
        time.sleep(0.400+line_t)
        data[i,j+1] = pmt.getRead()
        trig.setLow()
        time.sleep(recover_t)
        freqcontrol.stop()

t = time.localtime()
filename = "{}{:02g}{:02g}".format(t.tm_year,t.tm_mon,t.tm_mday)+ "
    _DopplerScanPMT"
i = 0
while os.path.exists(os.path.normpath(path + filename + "_{}.npy".format(i)
    )):
    i = i + 1
filenamen = os.path.normpath(path + filename + "_{}.npy".format(i))

# And save the data
np.save(filenamen,data)

res = np.zeros(data.shape[0])
```

```
err = np.zeros(data.shape[0])
for i in range(data.shape[0]):
    res[i] = np.mean(data[i,1:])
    err[i] = np.std(data[i,1:])/np.sqrt(data.shape[1])

bk.output_file("fluorescence_curve.html", title="Fluorescence Curve")
p1 = bk.figure(title="Fluorescence Curve")
dataplot = np.zeros([data.shape[0],3])
dataplot[:,0] = data[:,0]
dataplot[:,1] = res
dataplot[:,2] = err
dataplot = dataplot[dataplot[:,0].argsort()]
p1.scatter(dataplot[:,0],dataplot[:,1])
p1.line(dataplot[:,0],dataplot[:,1])
p1.xaxis.axis_label = "Double-Pass Voltage"
p1.yaxis.axis_label = "Counts"

bk.show(p1)

for k in oneshots.keys():
    oneshots[k].stop()
del oneshots
trig.stop()
del trig
pmt.stop()
del pmt
freqcontrol.clear()
del freqcontrol
print('Im Done')
```

E.3.5. Doppler Curve Fast

The code below runs Doppler fluorescence spectra used to monitor/set Doppler cooling laser lock frequencies and powers. This iteration uses 10 ms exposure times with randomization and interleaving of data points.

```
# -*- coding: utf-8 -*-
"""
Created on Wed Apr 15 10:12:33 2015

@author: lynx
"""

from OneShot import OneShot
from TrigControl import TrigControl
import time
import numpy as np
import os
import pdb
from AIWrite import AOWrite
#import bokeh.plotting as bk
from pmt import PMT

# How much data to take at each setting
reps = 5
scriptreps = 4

cooling_v = 0.10
voltages1 = np.linspace(-0.4,0.4,50,dtype=np.float64)
voltages = np.repeat(voltages1,scriptreps)
voltages = np.random.permutation(voltages)
#voltages = np.array([0.0],dtype=np.float64)

#Wait for line trigger time. Should be either zero or 40 ms.
line_t = 0.0
#line_t = 40e-3

# Recovery time between cycles to allow ion to cool
recover_t = 20e-3
```

```
#PMT Exposure time.
expose_t = 10e-3

safe = 1e-5

# The path to save the data in
path = 'c:/data/Fluorescence Curves/'

#PMT Setup
pmtc = '/dev1/ctr7'
pmt = PMT(pmtc)

trigchan1 = '/dev1/PFI14'
trigchan5 = '/dev5/PFI3'
trigchan2 = '/dev2/PFI6'
trigsource = '/dev5/port0/line0'

clist = {'b_doppler': ['/Dev5/Ctr6',trigchan5,True],
        'r_doppler': ['/Dev1/Ctr4',trigchan1,False],
        'shelve': ['/Dev1/Ctr5',trigchan1,False],
        'deshelve': ['/Dev1/Ctr6',trigchan1,False],
        'camera': ['/Dev5/Ctr1',trigchan5,False],
        'b_shutter': ['/Dev5/Ctr2',trigchan5,False],
        'r_shutter': ['/Dev5/Ctr3',trigchan5,False],
        'shelve_shutter': ['/Dev1/Ctr3',trigchan1,True]}

#The timings below are for equilibrium cooling behavior.
ctimes = {'camera': [.02,.02+expose_t]}

#The timings below are for non-equilibrium behavior.
#ctimes = {'camera': [.1+safe,.1+safe+expose_t]}

if __name__ == "__main__":
    freqcontrol = AOWrite("/dev2/ao1")
    #freqcontrol = AOWrite("/dev2/ao3")
    freqcontrol.setTrig(100,4,trigchan2)
    # Setup trigger out
    trig = TrigControl(trigsource)

    # And now all the oneshots
```

```
oneshots = {}
# First initialize all the oneshots
for k in ctimes.keys():
    oneshots[k] = OneShot(*(clist[k]+ctimes[k]))

# Then start all the oneshots
for k in oneshots.keys():
    try:
        oneshots[k].begin()
    except:
        pdb.post_mortem()
time.sleep(1)
# The first column is the frequencies, all others are the data collected
# for that frequency
data = np.zeros((voltages.shape[0],reps+1))
for i,v in enumerate(voltages):
    volts = np.array([cooling_v, v, v, cooling_v])
    freqcontrol.voltSequence(1,volts)
    data[i,0] = v
    print('% of Scan Completed:')
    print((i+1.0)/voltages.shape[0]*100.0)
    for j in range(reps):
        pmt.startRead()
        freqcontrol.start()
        trig.setHigh()
        time.sleep(0.040+line_t)
        data[i,j+1] = pmt.getRead()
        trig.setLow()
        time.sleep(recover_t)
        freqcontrol.stop()

t = time.localtime()
filename = "{}{:02g}{:02g}".format(t.tm_year,t.tm_mon,t.tm_mday)+ "
    _DopplerScanPMT_Fast"
i = 0
while os.path.exists(os.path.normpath(path + filename + "_{}.npy".format(i)
    )):
    i = i + 1
filenamen = os.path.normpath(path + filename + "_{}.npy".format(i))

# And save the data
```

```
np.save(filenamen,data)

res = np.zeros(data.shape[0])
err = np.zeros(data.shape[0])
for i in range(data.shape[0]):
   res[i] = np.mean(data[i,1:])
   err[i] = np.std(data[i,1:])/np.sqrt(data.shape[1])

#bk.output_file("fluorescence_curve.html", title="Fluorescence Curve")
#p1 = bk.figure(title="Fluorescence Curve")
#dataplot = np.zeros([data.shape[0],3])
#dataplot[:,0] = data[:,0]
#dataplot[:,1] = res
#dataplot[:,2] = err
#dataplot = dataplot[dataplot[:,0].argsort()]
#p1.scatter(dataplot[:,0],dataplot[:,1])
#p1.line(dataplot[:,0],dataplot[:,1])
#p1.xaxis.axis_label = "Double-Pass Voltage"
#p1.yaxis.axis_label = "Counts"

#bk.show(p1)

for k in oneshots.keys():
   oneshots[k].stop()
del oneshots
trig.stop()
del trig
pmt.stop()
del pmt
freqcontrol.clear()
del freqcontrol
print('Im Done')
```

E.3.6. FOR Scan Master Control

The code below sets all operating conditions for a variety of FOR Raman experiments.

```
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 16 15:09:02 2015

The main code for executing a frequency scan with STIRAP.

@author: CMS
@secondauthor: MGK
"""

from OneShot import OneShot
from DDSControl import DDSControl
from TrigControl import TrigControl
import time
import numpy as np
import os
from plotoneshots import plotOneShots
from pmt import PMT
from AIWrite import AOWrite

# If this is true, then plot the ctimes sequence fopr the first run.
plot_me = False

# Wait for line trigger time. Should be either zero for software
# with the trigger box removed or 40 ms with the trigger box in place.
#line_t = 0.0
line_t = 40e-3

# Trap parameters.
#omega_z = 1.1329
#omega_z = 1.138883
omega_z = 1.1377

# CW Raman Cooling parameters
cw_carrier = -11.568
cw_rsb = cw_carrier - omega_z
```

```
cw_cool_freq = cw_rsb
cw_cool_freq = -12.649
cw_cool_t = 10e-3
#cw_cool_t = 0.1e-6

# Time to wait for heating beetween CW Raman end and FOR Pi-pulse beging.
heating_t = 0.0e-3

# FOR Parameters
for_carrier = -10.9544
for_rsb = for_carrier - omega_z
for_bsb = for_carrier + omega_z
#for_bsb = -9.80317
for_carrier_pi_pulse_t = 21.7e-6
for_bsb_pi_pulse_t = 60.0e-6

for_freq = for_carrier
for_pi_pulse_t = for_carrier_pi_pulse_t

# How much data to take at each setting
reps = 50
scriptreps = 10
carrierscriptreps = 4

# FOR Frequencies to run
for_frequencies = np.linspace(for_freq-0.3,for_freq+0.3,30)

# FOR Frequencies to run for temperature script.
for_frequencies_temp1 = np.linspace(for_bsb-0.075,for_bsb+0.075,15)
for_frequencies_temp2 = np.linspace(for_rsb-0.075,for_rsb+0.075,15)
for_frequencies_temp3 = np.linspace(for_rsb+0.45,for_rsb+0.35,4)
for_frequencies_temp4 = np.linspace(for_bsb-0.45,for_bsb-0.35,4)
for_frequencies_tempc = np.linspace(for_carrier-0.2,for_carrier+0.2,15)
for_frequencies_temps = np.concatenate((for_frequencies_temp1,\
    for_frequencies_temp2,for_frequencies_temp2,for_frequencies_temp3,
       for_frequencies_temp4))

# FOR Frequencies to run for fast temperature script.
for_frequencies_temp_fasts = np.array([for_bsb,for_rsb,for_bsb-0.40,for_rsb
   +0.40])
```

```
# Pi-Pulse times to run.
#for_pulse_times = np.linspace(0.1e-6,90e-6,30)
for_pulse_times = np.linspace(0.1e-6,90e-6,30)

# Protect times to run
protect_time_high = 1e-3
protect_time_low = 1e-7
log_protect_times = np.linspace(np.log10(protect_time_high),\
    np.log10(protect_time_low),25)
protect_times = 10**log_protect_times

# Optical Pumping times to run
op_time_high = 1.0e-4
op_time_low = 1.0e-7
log_op_times = np.linspace(np.log10(op_time_high),\
    np.log10(op_time_low),20)
op_times = 10**log_op_times

# CW Raman cooling frequencies to run:
cw_cool_frequencies = np.linspace(cw_rsb-0.5,cw_rsb+0.5,25)

# CW Raman cooling times to run:
#cw_cool_time_high = 30.0e-3
#cw_cool_time_low = 1.0e-5
#log_cw_cool_times = np.linspace(np.log10(cw_cool_time_high),\
#    np.log10(cw_cool_time_low),30)
#cw_cool_times = 10**log_cw_cool_times
cw_cool_times = np.linspace(1e-3,10.0e-3,10)

# CW Raman heating times to run:
heating_times = np.linspace(0.0,250.0e-3,25)

# What program to run?
FOR_Freq_Scan = False
FOR_Time_Scan = True
FOR_Protect_Scan = False
FOR_Temp_Measurement = False
FOR_Temp_Measurement_Fast = False
FOR_OP_Scan = False
NR_Cooling_Freq = False
```

```
NR_Cooling_Rate = False
Heating_Rate = False

# Repeats and Randomizations!
for_frequencies = np.repeat(for_frequencies,scriptreps)
for_frequencies = np.random.permutation(for_frequencies)

for_frequencies_temps = np.repeat(for_frequencies_temps,scriptreps)
for_frequencies_tempc = np.repeat(for_frequencies_tempc,carrierscriptreps)
for_frequencies_temp = np.concatenate((for_frequencies_temps,
   for_frequencies_tempc))
for_frequencies_temp = np.random.permutation(for_frequencies_temp)

# Set up array for full randomization of cooling frequencies.
cw_cool_frequencies2 = np.repeat(cw_cool_frequencies,for_frequencies_temp_fasts
    .shape[0])
cw_cool_frequencies_freqs = np.zeros((cw_cool_frequencies2.shape[0],2))
k = 0
for i in range(cw_cool_frequencies_freqs.shape[0]):
    if (1.0*i/for_frequencies_temp_fasts.shape[0]).is_integer():
        for j in range(for_frequencies_temp_fasts.shape[0]):
            cw_cool_frequencies_freqs[for_frequencies_temp_fasts.shape[0]*k+j,0]
                 = cw_cool_frequencies2[for_frequencies_temp_fasts.shape[0]*k+j]
            cw_cool_frequencies_freqs[for_frequencies_temp_fasts.shape[0]*k+j,1]
                 = for_frequencies_temp_fasts[j]
        k = k + 1
cw_cool_frequencies_freqs = np.repeat(cw_cool_frequencies_freqs,scriptreps,axis
    =0)
cw_cool_frequencies_freqs = np.random.permutation(cw_cool_frequencies_freqs)

# Set up array for full randomization of cooling times.
cw_cool_times2 = np.repeat(cw_cool_times,for_frequencies_temp_fasts.shape[0])
cw_cool_times_freqs = np.zeros((cw_cool_times2.shape[0],2))
k = 0
for i in range(cw_cool_times_freqs.shape[0]):
    if (1.0*i/for_frequencies_temp_fasts.shape[0]).is_integer():
        for j in range(for_frequencies_temp_fasts.shape[0]):
            cw_cool_times_freqs[for_frequencies_temp_fasts.shape[0]*k+j,0] =
                cw_cool_times2[for_frequencies_temp_fasts.shape[0]*k+j]
            cw_cool_times_freqs[for_frequencies_temp_fasts.shape[0]*k+j,1] =
                for_frequencies_temp_fasts[j]
        k = k + 1
cw_cool_times_freqs = np.repeat(cw_cool_times_freqs,scriptreps,axis=0)
cw_cool_times_freqs = np.random.permutation(cw_cool_times_freqs)

# Set up array for full randomization of heating times.
heating_times2 = np.repeat(heating_times,for_frequencies_temp_fasts.shape[0])
heating_times_freqs = np.zeros((heating_times2.shape[0],2))
k = 0
for i in range(heating_times_freqs.shape[0]):
    if (1.0*i/for_frequencies_temp_fasts.shape[0]).is_integer():
        for j in range(for_frequencies_temp_fasts.shape[0]):
            heating_times_freqs[for_frequencies_temp_fasts.shape[0]*k+j,0] =
                heating_times2[for_frequencies_temp_fasts.shape[0]*k+j]
            heating_times_freqs[for_frequencies_temp_fasts.shape[0]*k+j,1] =
                for_frequencies_temp_fasts[j]
        k = k + 1
heating_times_freqs = np.repeat(heating_times_freqs,scriptreps,axis=0)
heating_times_freqs = np.random.permutation(heating_times_freqs)

for_frequencies_temp_fast = np.repeat(for_frequencies_temp_fasts,scriptreps)
for_frequencies_temp_fast = np.random.permutation(for_frequencies_temp_fast)

for_pulse_times = np.repeat(for_pulse_times,scriptreps)
for_pulse_times = np.random.permutation(for_pulse_times)

protect_times = np.repeat(protect_times,scriptreps)
protect_times = np.random.permutation(protect_times)

op_times = np.repeat(op_times,scriptreps)
op_times = np.random.permutation(op_times)

cw_cool_frequencies = np.repeat(cw_cool_frequencies,scriptreps)
cw_cool_frequencies = np.random.permutation(cw_cool_frequencies)

cw_cool_times = np.repeat(cw_cool_times,scriptreps)
cw_cool_times = np.random.permutation(cw_cool_times)

heating_times = np.repeat(heating_times,scriptreps)
heating_times = np.random.permutation(heating_times)

# Times
# A 'safe interval', which is a period added in to various places to ensure
# the light has actually turned off, or on.
safe = 1e-5
# The amount of time to allow for the mechanical shutter to close or open
shutter_t = 5e-3
# The amount of time to optically pump for. Can safely be long.
#op_t = 5.0e-3
op_t = 1.0e-5
# A period after FOR pulse where NR Sigma is turned on, with no red, to pump
# any transfered ions to the D(3/2). This interval is sensitive! If it's too
#short, the ion won't be pumped. If too long all atoms may be pumped by the
#imperfect B field trimming
protect_t = 1e-5
# The 455 nm exposure time to shelve any remaining ions into D(5/2).
shelve_t = 10e-3
# Expose time of PMT
expose_t = 5e-3
# The 614 nm exposure time to deshelve any ions in D(5/2).
deshelve_t = 15e-3
# Recovery time between cycles to allow ion to Doppler cool to equilibrium
recover_t = 20e-3

# Change the Doppler cooling settings for ion detection after shelving. This
# sets up the AO task. It has to be defined as a single task to share the
# on-board clock. AO1 = b_FM, AO4 = b_AM, AO5 = r_FM.
trigchan2_camera = '/dev2/PFI7'
b_detect_FM_v = 0.25
b_cool_FM_v = 0.10
b_FM_voltages = np.array([b_detect_FM_v,b_cool_FM_v],dtype=np.float64)
b_detect_AM_v = -1.5
b_cool_AM_v = -0.15
b_AM_voltages = np.array([b_detect_AM_v,b_cool_AM_v],dtype=np.float64)
r_detect_AM_v = -0.10
r_cool_AM_v = -0.02
r_AM_voltages = np.array([r_detect_AM_v,r_cool_AM_v],dtype=np.float64)
AO_detect_voltages = np.concatenate((b_FM_voltages,b_AM_voltages,r_AM_voltages)
   )
AO_detect_control = AOWrite("/dev2/ao1,/dev2/ao4:5")
AO_detect_control.setTrig(1/expose_t,2,trigchan2_camera)
AO_detect_control.voltSequence(3,AO_detect_voltages)
```

```
# Hardware info
# PMT info
pmtc = '/dev1/ctr7'
# Start PMT task.
pmt = PMT(pmtc)

# Trigger info
trigchan1 = '/dev1/PFI14'
trigchan5 = '/dev5/PFI3'
trigchan6 = '/dev6/PFI14'
trigsource = '/dev5/port0/line0'
r_doppler_c = '/dev5/port0/line4'

#Counter card channel list.
clist = {'b_doppler': ['/Dev5/Ctr6',trigchan5,True],
        'r_doppler': ['/Dev1/Ctr4',trigchan1,False],
        'NR_sigma': ['/Dev5/Ctr4',trigchan5,False],
        'NR_pi' : ['/Dev5/Ctr5',trigchan5,False],
        'b_shutter': ['/Dev5/Ctr2',trigchan5,False],
        'r_shutter': ['/Dev5/Ctr3',trigchan5,False],
        'FOR_sigma' : ['/Dev5/Ctr7',trigchan5,False],
        'FOR_pi' : ['/Dev1/Ctr0',trigchan1,False],
        'NR_sigma_protect': ['/Dev1/Ctr1',trigchan1,False],
        'shelve': ['/Dev1/Ctr5',trigchan1,False],
        'deshelve': ['/Dev1/Ctr6',trigchan1,False],
        'camera': ['/Dev5/Ctr1',trigchan5,False],
        'shelve_shutter': ['/Dev1/Ctr3',trigchan1,True],
        'raman_shutter': ['/Dev1/Ctr2',trigchan1,False], #False due to using OR
             + NOT
        'raman_shutter_protect': ['/Dev5/Ctr0',trigchan5,False], #False due to
            using OR + NOT
        'FOR_shutter' : ['/Dev6/Ctr0',trigchan6,True],
        'deshelve_shutter': ['/Dev6/Ctr1',trigchan6,True]}

#Run specified program.
if sum([FOR_Freq_Scan,FOR_Time_Scan,FOR_Protect_Scan,FOR_Temp_Measurement,\
       FOR_Temp_Measurement_Fast,FOR_OP_Scan,NR_Cooling_Freq,NR_Cooling_Rate,\
       Heating_Rate]) != 1:
    FOR_Freq_Scan = False
    FOR_Time_Scan = False
    FOR_Protect_Scan = False
    FOR_Temp_Measurement = False
    FOR_Temp_Measurement_Fast = False
    FOR_OP_Scan = False
    NR_Cooling_Freq = False
    NR_Cooling_Rate = False
    Heating_Rate = False
    print('You messed up what to run! Give up while you are ahead.')

if scriptreps < 1:
    FOR_Freq_Scan = False
    FOR_Time_Scan = False
    FOR_Protect_Scan = False
    FOR_Temp_Measurement = False
    FOR_Temp_Measurement_Fast = False
    FOR_OP_Scan = False
    NR_Cooling_Freq = False
    NR_Cooling_Rate = False
    Heating_Rate = False
    print('You messed up scriptreps! Give up while you are ahead.')

if FOR_Freq_Scan == True:
    del for_freq
    #execfile('cwramancool_FOR_scanPMT_Master_Freq.py')
    execfile('cwramancool_FOR_scanPMT_Master_Freq_NS.py')

if FOR_Time_Scan == True:
    del for_pi_pulse_t
    #execfile('cwramancool_FOR_scanPMT_Master_Time.py')
    execfile('cwramancool_FOR_scanPMT_Master_Time_NS.py')

if FOR_Protect_Scan == True:
    del protect_t
    #execfile('cwramancool_FOR_scanPMT_Master_Protect.py')
    execfile('cwramancool_FOR_scanPMT_Master_Protect_NS.py')

if FOR_Temp_Measurement == True:
    del for_freq
    del for_pi_pulse_t
    #execfile('cwramancool_FOR_scanPMT_Master_Temperature.py')
    execfile('cwramancool_FOR_scanPMT_Master_Temperature_NS.py')
```

```
if FOR_Temp_Measurement_Fast == True:
    del for_freq
    del for_pi_pulse_t
    execfile('cwramancool_FOR_scanPMT_Master_FastTemperature_NS.py')

if FOR_OP_Scan == True:
    del op_t
    execfile('cwramancool_FOR_scanPMT_Master_OP_NS.py')

if NR_Cooling_Freq == True:
    del cw_cool_freq
    del for_freq
    execfile('cwramancool_FOR_scanPMT_Master_CoolFreq_NS.py')

if NR_Cooling_Rate == True:
    del cw_cool_t
    del for_freq
    execfile('cwramancool_FOR_scanPMT_Master_CoolTime_NS.py')

if Heating_Rate == True:
    del heating_t
    del for_freq
    execfile('cwramancool_FOR_scanPMT_Master_Heating_NS.py')

#Resets
del for_frequencies
del for_pulse_times
del protect_times
del for_frequencies_temp
AO_detect_control.clear()
del AO_detect_control
f = -11
with DDSControl(r'\\.\Ezssp-0',500.0) as dds:
    dds.SelectChannel(0)
    #dds.SetFrequency(80.0-f/2.0)
    dds.SetFrequency(80.0-11/2.0)
    dds.SelectChannel(2)
    #dds.SetFrequency(80.0-f/2.0)
    dds.SetFrequency(80.0-11/2.0)
    dds.SelectChannel(1)
    #dds.SetFrequency(80.0+f/2.0)
    dds.SetFrequency(80.0-11/2.0-f/2)
    dds.SelectChannel(3)
    #dds.SetFrequency(80.0+f/2.0)
    dds.SetFrequency(80.0-11/2.0-f/2)
print('Im Done')
```

E.3.6.1. FOR Cooling Frequency Scan. The code below scans the NR Raman cooling frequency while using a fast, four-point temperature measurement.

```
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 16 15:09:02 2015

The main code for executing a frequency scan with STIRAP.

@author: CMS
@secondauthor: MGK
"""
# The path to save the data in
path = 'c:/data/Rabi/Cool Freq/'

# Start PMT task.
pmt = PMT(pmtc)

if __name__ == '__main__':
    # Setup trigger out
    trig = TrigControl(trigsource)

    # And now all the oneshots
    oneshots = {}
    # First initialize all the oneshots

    time.sleep(1)
    # The first column is the frequencies, all others are the data collected
    # for that frequency

    # FOR beams for pi pulses

#       for (i,f) in enumerate(frequencies):
#           # Set the dds frequencies. I am making Raman A be the negative one.
#           # Also, I am making them be symmetric around 80, because I can.
#           dds.SelectChannel(0)
#           #dds.SetFrequency(80.0-f/2.0)
#           dds.SetFrequency(80.0-15/2.0)
#           dds.SelectChannel(1)
#           #dds.SetFrequency(80.0+f/2.0)
#           dds.SetFrequency(80.0-15/2.0+f)
#           data[i,0] = f

    # For a scan, this will be incorporated into a for loop
    # FOR_Sigma pulse length
#       FOR_sigma = OneShot('/Dev5/Ctr7',trigchan5,True,stirap_a_b-safe,
    protect_b+safe)
    # FOR_Pi pulse length
#       FOR_pi = OneShot('/Dev1/Ctr0',trigchan1,True,stirap_b_b-safe,protect_b+
    safe)

    #Manually set the default states of the shutters.
    try:
        del clist
    except NameError:
        print
    clist = {'b_shutter': ['/Dev5/Ctr2',trigchan5,False],
             'r_shutter': ['/Dev5/Ctr3',trigchan5,False],
             'shelve_shutter': ['/Dev1/Ctr3',trigchan1,False],
             'raman_shutter': ['/Dev1/Ctr2',trigchan1,True], #False due to using
                  OR + NOT
             'raman_shutter_protect': ['/Dev5/Ctr0',trigchan5,True], #False due
                 to using OR + NOT
             'FOR_shutter': ['/Dev6/Ctr0',trigchan6,False],
             'deshelve_shutter': ['/Dev6/Ctr1',trigchan6,False]}

    ctimes = {'b_shutter': [0.0, 2*shutter_t],
              'r_shutter': [0.0, 2*shutter_t],
              'shelve_shutter': [0.0, 2*shutter_t],
              'raman_shutter': [0.0, 2*shutter_t],
              'raman_shutter_protect': [0.0, 2*shutter_t],
              'FOR_shutter': [0.0, 2*shutter_t],
              'deshelve_shutter': [0.0,2*shutter_t]}
    for k in ctimes.keys():
        oneshots[k] = OneShot(*(clist[k]+ctimes[k]))
    for k in oneshots.keys():
        oneshots[k].begin()
    trig.setHigh()
    time.sleep(2*shutter_t)
    trig.setLow()
    time.sleep(recover_t)
```

```
for k in oneshots.keys():
    oneshots[k].stop()
del oneshots
del clist
oneshots = {}
# b is begin, s is stop

data = np.zeros((cw_cool_frequencies_freqs.shape[0],reps+2))
for i in range(cw_cool_frequencies_freqs.shape[0]):
    cw_cool_freq = cw_cool_frequencies_freqs[i,0]
    for_freq = cw_cool_frequencies_freqs[i,1]

    with DDSControl(r'\\.\Ezssp-0',500.0) as dds:
        # Set the dds frequencies. I am making Raman A be the negative one.
        # Also, I am making them be symmetric around 80, because I can.
        dds.SelectChannel(2)
        #dds.SetFrequency(80.0-f/2.0)
        dds.SetFrequency(80.0-11/2.0)
        dds.SelectChannel(3)
        #dds.SetFrequency(80.0+f/2.0)
        dds.SetFrequency(80.0-11/2.0-cw_cool_freq/2)
        dds.SelectChannel(0)
        #dds.SetFrequency(80.0-f/2.0)
        dds.SetFrequency(80.0-11/2.0)
        dds.SelectChannel(1)
        #dds.SetFrequency(80.0+f/2.0)
        dds.SetFrequency(80.0-11/2.0-for_freq/2)
        # And now all the oneshots
        oneshots = {}
        # First initialize all the oneshots

        if np.logical_and(for_freq <= for_carrier+0.5, for_freq >=
            for_carrier-0.5):
            for_pi_pulse_t = for_carrier_pi_pulse_t
        else:
            for_pi_pulse_t = for_bsb_pi_pulse_t

        # b is begin, s is stop
        # Raman Cooling
        ramancool_NR_sigma_b = safe
        ramancool_NR_pi_b = safe
```

```
ramancool_NR_sigma_s = ramancool_NR_sigma_b + cw_cool_t + op_t #
    optical pumping
ramancool_NR_pi_s = ramancool_NR_sigma_b + cw_cool_t
# Pi Pulse, the stop time is defined down below, since it changes
    for a scan
pi_pulse_b = ramancool_NR_sigma_s+2*safe +heating_t# The NR lights
    must be off
pi_pulse_s = pi_pulse_b + for_pi_pulse_t
# Protect
protect_b = pi_pulse_s + safe # Optical pumping shouldnt be on
    during shelving
protect_s = protect_b + protect_t
# Shelve
shelve_b = protect_s + safe
shelve_s = shelve_b + shelve_t
# Readout, the red and blue should be open at expose_s
expose_b = shelve_s + 2*safe # Doppler cooling sholdn't start until
    the shelving is off
expose_s = expose_b + expose_t
# Deshelve
deshelve_b = expose_s +safe
deshelve_s = deshelve_b + deshelve_t

clist = {'b_doppler': ['/Dev5/Ctr6',trigchan5,True],
        'r_doppler': ['/Dev1/Ctr4',trigchan1,False],
        'NR_sigma': ['/Dev5/Ctr4',trigchan5,False],
        'NR_pi' : ['/Dev5/Ctr5',trigchan5,False],
        'b_shutter': ['/Dev5/Ctr2',trigchan5,False],
        'r_shutter': ['/Dev5/Ctr3',trigchan5,False],
        'FOR_sigma' : ['/Dev5/Ctr7',trigchan5,False],
        'FOR_pi' : ['/Dev1/Ctr0',trigchan1,False],
        'NR_sigma_protect': ['/Dev1/Ctr1',trigchan1,False],
        'shelve': ['/Dev1/Ctr5',trigchan1,False],
        'deshelve': ['/Dev1/Ctr6',trigchan1,False],
        'camera': ['/Dev5/Ctr1',trigchan5,False],
        'shelve_shutter': ['/Dev1/Ctr3',trigchan1,True],
        'raman_shutter': ['/Dev1/Ctr2',trigchan1,False], #False due
            to using OR + NOT
        'raman_shutter_protect': ['/Dev5/Ctr0',trigchan5,False], #
            False due to using OR + NOT
        'FOR_shutter' : ['/Dev6/Ctr0',trigchan6,True],
                'deshelve_shutter': ['/Dev6/Ctr1',trigchan6,True]}

        ctimes = {'b_doppler': [0.0, shelve_s+safe],
                'r_doppler': [ramancool_NR_sigma_s+safe, shelve_s+safe],
                'NR_sigma': [ramancool_NR_sigma_b,ramancool_NR_sigma_s],
                'NR_pi' : [ramancool_NR_pi_b,ramancool_NR_pi_s],
                #'b_shutter': [0.0, shelve_s],
                #'r_shutter': [ramancool_NR_sigma_s+shutter_t, shelve_s+
                   shutter_t],
                'FOR_sigma' : [pi_pulse_b,pi_pulse_s],
                'FOR_pi' : [pi_pulse_b,pi_pulse_s],
                'NR_sigma_protect': [protect_b,protect_s],
                'shelve': [shelve_b, shelve_s],
                'deshelve': [deshelve_b, deshelve_s],
                'camera': [expose_b, expose_s],
                #'shelve_shutter': [shelve_b, shelve_s],
                #'raman_shutter': [0.0, ramancool_NR_sigma_s],
                #'raman_shutter_protect': [pi_pulse_s, protect_s],
                #'FOR_shutter': [ramancool_NR_sigma_s+shutter_t,pi_pulse_s],
                #'deshelve_shutter': [deshelve_b,deshelve_s]
                }

        for k in ctimes.keys():
            oneshots[k] = OneShot(*(clist[k]+ctimes[k]))

        # Then start all the oneshots
        for k in oneshots.keys():
            oneshots[k].begin()
        if np.logical_and(plot_me == True, i == 0):
            hold = plotOneShots(ctimes)

#         'raman_shutter': [shutter_t,stirap_protect_b],
#         'stirap_trig': [stirap_a_b+safe,stirap_a_b+2*safe],
#         #'shelve': [stirap_protect_b+shutter_t*2,stirap_protect_b+
   shutter_t*2+shelve_t],
#         #'shelve_shutter': [stirap_protect_b+shutter_t,stirap_protect_b+
   shutter_t*2+shelve_t],
#         'shelve': [stirap_protect_b+shutter_t,stirap_protect_b+shutter_t+
   shelve_t],
#         'shelve_shutter': [stirap_protect_b,stirap_protect_b+shutter_t+
   shelve_t],
#        ’deshelve’: [deshelve_b,deshelve_b+deshelve_t],
#        ’camera’: [cooling_s+shutter_t,cooling_s+shutter_t+expose_t]




        # Set the dds frequencies. I am making Raman A be the negative one.
        # Also, I am making them be symmetric around 80, because I can.

        data[i,0] = cw_cool_freq
        data[i,1] = for_freq
        print(’% of Scan Completed:’)
        print((i+1.0)/(cw_cool_frequencies_freqs.shape[0])*100.0)
        for j in range(reps):
            AO_detect_control.start()
            pmt.startRead()
            trig.setHigh()
            time.sleep(deshelve_b+deshelve_t+line_t)
            res = pmt.getRead()
            trig.setLow()
            data[i,j+2] = res
            time.sleep(recover_t) # Make sure that the cycle has finished
            AO_detect_control.stop()

        #clear the oneshots to start again!
        for k in oneshots.keys():
            oneshots[k].stop()
        del oneshots

# Build a suitable filename
t = time.localtime()
filename = "{}{:02g}{:02g}".format(t.tm_year,t.tm_mon,t.tm_mday)+ "
    _CWRamancool_FOR_CoolFreq_NS_scanPMT"
i = 0
while os.path.exists(os.path.normpath(path + filename + "_{}.npy".format(i)
    )):
    i = i + 1
filenamen = os.path.normpath(path + filename + "_{}.npy".format(i))

# And save the data
np.save(filenamen,data)
```

```
del clist
del ctimes
oneshots = {}

clist = {'b_shutter': ['/Dev5/Ctr2',trigchan5,False],
            'r_shutter': ['/Dev5/Ctr3',trigchan5,False],
            'shelve_shutter': ['/Dev1/Ctr3',trigchan1,True],
            'raman_shutter': ['/Dev1/Ctr2',trigchan1,False], #False due to
               using OR + NOT
            'raman_shutter_protect': ['/Dev5/Ctr0',trigchan5,False], #False
                due to using OR + NOT
            'FOR_shutter': ['/Dev6/Ctr0',trigchan6,True],
            'deshelve_shutter': ['/Dev6/Ctr1',trigchan6,True]}

ctimes = {'b_shutter': [0.0, 2*shutter_t],
         'r_shutter': [0.0, 2*shutter_t],
         'shelve_shutter': [0.0, 2*shutter_t],
         'raman_shutter': [0.0, 2*shutter_t],
         'raman_shutter_protect': [0.0, 2*shutter_t],
         'FOR_shutter': [0.0, 2*shutter_t],
         'deshelve_shutter': [0.0,2*shutter_t]}
for k in ctimes.keys():
    oneshots[k] = OneShot(*(clist[k]+ctimes[k]))
for k in oneshots.keys():
    oneshots[k].begin()
trig.setHigh()
time.sleep(2*shutter_t)
trig.setLow()
time.sleep(recover_t)
for k in oneshots.keys():
    oneshots[k].stop()
del oneshots
del clist

trig.stop()
del trig
pmt.stop()
del pmt
```

E.3.6.2. FOR Cooling Rate Scan. The code below scans the duration of NR RSC to quantify the cooling rate of the technique.

```
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 16 15:09:02 2015

The main code for executing a frequency scan with STIRAP.

@author: CMS
@secondauthor: MGK
"""
# The path to save the data in
path = 'c:/data/Rabi/Cool Time/'

# Start PMT task.
pmt = PMT(pmtc)

if __name__ == '__main__':
    # Setup trigger out
    trig = TrigControl(trigsource)

    # And now all the oneshots
    oneshots = {}
    # First initialize all the oneshots

    time.sleep(1)
    # The first column is the frequencies, all others are the data collected
    # for that frequency

    # FOR beams for pi pulses

#       for (i,f) in enumerate(frequencies):
#           # Set the dds frequencies. I am making Raman A be the negative one.
#           # Also, I am making them be symmetric around 80, because I can.
#           dds.SelectChannel(0)
#           #dds.SetFrequency(80.0-f/2.0)
#           dds.SetFrequency(80.0-15/2.0)
#           dds.SelectChannel(1)
#           #dds.SetFrequency(80.0+f/2.0)
#           dds.SetFrequency(80.0-15/2.0+f)
#           data[i,0] = f

    # For a scan, this will be incorporated into a for loop
    # FOR_Sigma pulse length
#       FOR_sigma = OneShot('/Dev5/Ctr7',trigchan5,True,stirap_a_b-safe,
    protect_b+safe)
    # FOR_Pi pulse length
#       FOR_pi = OneShot('/Dev1/Ctr0',trigchan1,True,stirap_b_b-safe,protect_b+
    safe)

    #Manually set the default states of the shutters.
    try:
        del clist
    except NameError:
        print
    clist = {'b_shutter': ['/Dev5/Ctr2',trigchan5,False],
             'r_shutter': ['/Dev5/Ctr3',trigchan5,False],
             'shelve_shutter': ['/Dev1/Ctr3',trigchan1,False],
             'raman_shutter': ['/Dev1/Ctr2',trigchan1,True], #False due to using
                  OR + NOT
             'raman_shutter_protect': ['/Dev5/Ctr0',trigchan5,True], #False due
                 to using OR + NOT
             'FOR_shutter': ['/Dev6/Ctr0',trigchan6,False],
             'deshelve_shutter': ['/Dev6/Ctr1',trigchan6,False]}

    ctimes = {'b_shutter': [0.0, 2*shutter_t],
              'r_shutter': [0.0, 2*shutter_t],
              'shelve_shutter': [0.0, 2*shutter_t],
              'raman_shutter': [0.0, 2*shutter_t],
              'raman_shutter_protect': [0.0, 2*shutter_t],
              'FOR_shutter': [0.0, 2*shutter_t],
              'deshelve_shutter': [0.0,2*shutter_t]}
    for k in ctimes.keys():
        oneshots[k] = OneShot(*(clist[k]+ctimes[k]))
    for k in oneshots.keys():
        oneshots[k].begin()
    trig.setHigh()
    time.sleep(2*shutter_t)
    trig.setLow()
    time.sleep(recover_t)
```

```
for k in oneshots.keys():
    oneshots[k].stop()
del oneshots
del clist
oneshots = {}
# b is begin, s is stop

data = np.zeros((cw_cool_times_freqs.shape[0],reps+2))
for i in range(cw_cool_times_freqs.shape[0]):
    cw_cool_t = cw_cool_times_freqs[i,0]
    for_freq = cw_cool_times_freqs[i,1]

    with DDSControl(r'\\.\Ezssp-0',500.0) as dds:
        # Set the dds frequencies. I am making Raman A be the negative one.
        # Also, I am making them be symmetric around 80, because I can.
        dds.SelectChannel(2)
        #dds.SetFrequency(80.0-f/2.0)
        dds.SetFrequency(80.0-11/2.0)
        dds.SelectChannel(3)
        #dds.SetFrequency(80.0+f/2.0)
        dds.SetFrequency(80.0-11/2.0-cw_cool_freq/2)
        dds.SelectChannel(0)
        #dds.SetFrequency(80.0-f/2.0)
        dds.SetFrequency(80.0-11/2.0)
        dds.SelectChannel(1)
        #dds.SetFrequency(80.0+f/2.0)
        dds.SetFrequency(80.0-11/2.0-for_freq/2)
        # And now all the oneshots
        oneshots = {}
        # First initialize all the oneshots

        if np.logical_and(for_freq <= for_carrier+0.5, for_freq >=
            for_carrier-0.5):
            for_pi_pulse_t = for_carrier_pi_pulse_t
        else:
            for_pi_pulse_t = for_bsb_pi_pulse_t

        # b is begin, s is stop
        # Raman Cooling
        ramancool_NR_sigma_b = safe
        ramancool_NR_pi_b = safe
```

```
ramancool_NR_sigma_s = ramancool_NR_sigma_b + cw_cool_t + op_t #
    optical pumping
ramancool_NR_pi_s = ramancool_NR_sigma_b + cw_cool_t
# Pi Pulse, the stop time is defined down below, since it changes
    for a scan
pi_pulse_b = ramancool_NR_sigma_s+2*safe +heating_t# The NR lights
    must be off
pi_pulse_s = pi_pulse_b + for_pi_pulse_t
# Protect
protect_b = pi_pulse_s + safe # Optical pumping shouldnt be on
    during shelving
protect_s = protect_b + protect_t
# Shelve
shelve_b = protect_s + safe
shelve_s = shelve_b + shelve_t
# Readout, the red and blue should be open at expose_s
expose_b = shelve_s + 2*safe # Doppler cooling sholdn’t start until
    the shelving is off
expose_s = expose_b + expose_t
# Deshelve
deshelve_b = expose_s +safe
deshelve_s = deshelve_b + deshelve_t

clist = {’b_doppler’: [’/Dev5/Ctr6’,trigchan5,True],
        ’r_doppler’: [’/Dev1/Ctr4’,trigchan1,False],
        ’NR_sigma’: [’/Dev5/Ctr4’,trigchan5,False],
        ’NR_pi’ : [’/Dev5/Ctr5’,trigchan5,False],
        ’b_shutter’: [’/Dev5/Ctr2’,trigchan5,False],
        ’r_shutter’: [’/Dev5/Ctr3’,trigchan5,False],
        ’FOR_sigma’ : [’/Dev5/Ctr7’,trigchan5,False],
        ’FOR_pi’ : [’/Dev1/Ctr0’,trigchan1,False],
        ’NR_sigma_protect’: [’/Dev1/Ctr1’,trigchan1,False],
        ’shelve’: [’/Dev1/Ctr5’,trigchan1,False],
        ’deshelve’: [’/Dev1/Ctr6’,trigchan1,False],
        ’camera’: [’/Dev5/Ctr1’,trigchan5,False],
        ’shelve_shutter’: [’/Dev1/Ctr3’,trigchan1,True],
        ’raman_shutter’: [’/Dev1/Ctr2’,trigchan1,False], #False due
            to using OR + NOT
        ’raman_shutter_protect’: [’/Dev5/Ctr0’,trigchan5,False], #
            False due to using OR + NOT
        ’FOR_shutter’ : [’/Dev6/Ctr0’,trigchan6,True],
                'deshelve_shutter': ['/Dev6/Ctr1',trigchan6,True]}

        ctimes = {'b_doppler': [0.0, shelve_s+safe],
                'r_doppler': [ramancool_NR_sigma_s+safe, shelve_s+safe],
                'NR_sigma': [ramancool_NR_sigma_b,ramancool_NR_sigma_s],
                'NR_pi' : [ramancool_NR_pi_b,ramancool_NR_pi_s],
                #'b_shutter': [0.0, shelve_s],
                #'r_shutter': [ramancool_NR_sigma_s+shutter_t, shelve_s+
                    shutter_t],
                'FOR_sigma' : [pi_pulse_b,pi_pulse_s],
                'FOR_pi' : [pi_pulse_b,pi_pulse_s],
                'NR_sigma_protect': [protect_b,protect_s],
                'shelve': [shelve_b, shelve_s],
                'deshelve': [deshelve_b, deshelve_s],
                'camera': [expose_b, expose_s],
                #'shelve_shutter': [shelve_b, shelve_s],
                #'raman_shutter': [0.0, ramancool_NR_sigma_s],
                #'raman_shutter_protect': [pi_pulse_s, protect_s],
                #'FOR_shutter': [ramancool_NR_sigma_s+shutter_t,pi_pulse_s],
                #'deshelve_shutter': [deshelve_b,deshelve_s]
                }

        for k in ctimes.keys():
            oneshots[k] = OneShot(*(clist[k]+ctimes[k]))

        # Then start all the oneshots
        for k in oneshots.keys():
            oneshots[k].begin()
        if np.logical_and(plot_me == True, i == 0):
            hold = plotOneShots(ctimes)

#        'raman_shutter': [shutter_t,stirap_protect_b],
#        'stirap_trig': [stirap_a_b+safe,stirap_a_b+2*safe],
#        #'shelve': [stirap_protect_b+shutter_t*2,stirap_protect_b+
    shutter_t*2+shelve_t],
#        #'shelve_shutter': [stirap_protect_b+shutter_t,stirap_protect_b+
    shutter_t*2+shelve_t],
#        'shelve': [stirap_protect_b+shutter_t,stirap_protect_b+shutter_t+
    shelve_t],
#        'shelve_shutter': [stirap_protect_b,stirap_protect_b+shutter_t+
    shelve_t],
#        'deshelve': [deshelve_b,deshelve_b+deshelve_t],
#        'camera': [cooling_s+shutter_t,cooling_s+shutter_t+expose_t]




        # Set the dds frequencies. I am making Raman A be the negative one.
        # Also, I am making them be symmetric around 80, because I can.

        data[i,0] = cw_cool_t
        data[i,1] = for_freq
        print('% of Scan Completed:')
        print((i+1.0)/(cw_cool_times_freqs.shape[0])*100.0)
        for j in range(reps):
            AO_detect_control.start()
            pmt.startRead()
            trig.setHigh()
            time.sleep(deshelve_b+deshelve_t+line_t)
            res = pmt.getRead()
            trig.setLow()
            data[i,j+2] = res
            time.sleep(recover_t) # Make sure that the cycle has finished
            AO_detect_control.stop()

        #clear the oneshots to start again!
        for k in oneshots.keys():
            oneshots[k].stop()
        del oneshots

# Build a suitable filename
t = time.localtime()
filename = "{}{:02g}{:02g}".format(t.tm_year,t.tm_mon,t.tm_mday)+ "
    _CWRamancool_FOR_CoolTime_NS_scanPMT"
i = 0
while os.path.exists(os.path.normpath(path + filename + "_{}.npy".format(i)
    )):
    i = i + 1
filenamen = os.path.normpath(path + filename + "_{}.npy".format(i))

# And save the data
np.save(filenamen,data)
```

```
del clist
del ctimes
oneshots = {}

clist = {'b_shutter': ['/Dev5/Ctr2',trigchan5,False],
            'r_shutter': ['/Dev5/Ctr3',trigchan5,False],
            'shelve_shutter': ['/Dev1/Ctr3',trigchan1,True],
            'raman_shutter': ['/Dev1/Ctr2',trigchan1,False], #False due to
               using OR + NOT
            'raman_shutter_protect': ['/Dev5/Ctr0',trigchan5,False], #False
                due to using OR + NOT
            'FOR_shutter': ['/Dev6/Ctr0',trigchan6,True],
            'deshelve_shutter': ['/Dev6/Ctr1',trigchan6,True]}

ctimes = {'b_shutter': [0.0, 2*shutter_t],
         'r_shutter': [0.0, 2*shutter_t],
         'shelve_shutter': [0.0, 2*shutter_t],
         'raman_shutter': [0.0, 2*shutter_t],
         'raman_shutter_protect': [0.0, 2*shutter_t],
         'FOR_shutter': [0.0, 2*shutter_t],
         'deshelve_shutter': [0.0,2*shutter_t]}
for k in ctimes.keys():
    oneshots[k] = OneShot(*(clist[k]+ctimes[k]))
for k in oneshots.keys():
    oneshots[k].begin()
trig.setHigh()
time.sleep(2*shutter_t)
trig.setLow()
time.sleep(recover_t)
for k in oneshots.keys():
    oneshots[k].stop()
del oneshots
del clist

trig.stop()
del trig
pmt.stop()
del pmt
```

E.3.6.3. FOR Fast Temperature Measurement. The code below performs a fast, four-point temperature measurement with a given set of operating conditions. This does not perform any scanning.

```
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 16 15:09:02 2015

The main code for executing a frequency scan with STIRAP.

@author: CMS
@secondauthor: MGK
"""
# The path to save the data in
path = 'c:/data/Rabi/Freq/'

# Start PMT task.
pmt = PMT(pmtc)

if __name__ == '__main__':
    # Setup trigger out
    trig = TrigControl(trigsource)

    # And now all the oneshots
    oneshots = {}
    # First initialize all the oneshots

    time.sleep(1)
    # The first column is the frequencies, all others are the data collected
    # for that frequency
    with DDSControl(r'\\.\Ezssp-0',500.0) as dds:
        # Set the dds frequencies. I am making Raman A be the negative one.
        # Also, I am making them be symmetric around 80, because I can.
        dds.SelectChannel(2)
        #dds.SetFrequency(80.0-f/2.0)
        dds.SetFrequency(80.0-11/2.0)
        dds.SelectChannel(3)
        #dds.SetFrequency(80.0+f/2.0)
        dds.SetFrequency(80.0-11/2.0-cw_cool_freq/2)
```

```
       # FOR beams for pi pulses

#       for (i,f) in enumerate(frequencies):
#           # Set the dds frequencies. I am making Raman A be the negative one.
#           # Also, I am making them be symmetric around 80, because I can.
#           dds.SelectChannel(0)
#           #dds.SetFrequency(80.0-f/2.0)
#           dds.SetFrequency(80.0-15/2.0)
#           dds.SelectChannel(1)
#           #dds.SetFrequency(80.0+f/2.0)
#           dds.SetFrequency(80.0-15/2.0+f)
#           data[i,0] = f

       # For a scan, this will be incorporated into a for loop
       # FOR_Sigma pulse length
#       FOR_sigma = OneShot('/Dev5/Ctr7',trigchan5,True,stirap_a_b-safe,
   protect_b+safe)
       # FOR_Pi pulse length
#       FOR_pi = OneShot('/Dev1/Ctr0',trigchan1,True,stirap_b_b-safe,protect_b+
   safe)

       #Manually set the default states of the shutters.
       try:
           del clist
       except NameError:
           print

       clist = {'b_shutter': ['/Dev5/Ctr2',trigchan5,False],
               'r_shutter': ['/Dev5/Ctr3',trigchan5,False],
               'shelve_shutter': ['/Dev1/Ctr3',trigchan1,False],
               'raman_shutter': ['/Dev1/Ctr2',trigchan1,True], #False due to
                   using OR + NOT
               'raman_shutter_protect': ['/Dev5/Ctr0',trigchan5,True], #False
                   due to using OR + NOT
               'FOR_shutter': ['/Dev6/Ctr0',trigchan6,False],
               'deshelve_shutter': ['/Dev6/Ctr1',trigchan6,False]}

       ctimes = {'b_shutter': [0.0, 2*shutter_t],
                'r_shutter': [0.0, 2*shutter_t],
                'shelve_shutter': [0.0, 2*shutter_t],
                'raman_shutter': [0.0, 2*shutter_t],
                'raman_shutter_protect': [0.0, 2*shutter_t],
                'FOR_shutter': [0.0, 2*shutter_t],
                'deshelve_shutter': [0.0,2*shutter_t]}
       for k in ctimes.keys():
           oneshots[k] = OneShot(*(clist[k]+ctimes[k]))
       for k in oneshots.keys():
           oneshots[k].begin()
       trig.setHigh()
       time.sleep(2*shutter_t)
       trig.setLow()
       time.sleep(recover_t)
       for k in oneshots.keys():
           oneshots[k].stop()
       del oneshots
       del clist
       oneshots = {}
       # b is begin, s is stop

       # Raman Cooling
       ramancool_NR_sigma_b = safe
       ramancool_NR_pi_b = safe
       ramancool_NR_sigma_s = ramancool_NR_sigma_b + cw_cool_t + op_t # optical
            pumping
       ramancool_NR_pi_s = ramancool_NR_sigma_b + cw_cool_t
       # Pi Pulse, the stop time is defined down below, since it changes for a
           scan
       pi_pulse_b = ramancool_NR_sigma_s+2*safe +heating_t# The NR lights must
           be off
       pi_pulse_s = pi_pulse_b + for_pi_pulse_t
       # Protect
       protect_b = pi_pulse_s + safe # Optical pumping shouldnt be on during
           shelving
       protect_s = protect_b + protect_t
       # Shelve
       shelve_b = protect_s + safe
       shelve_s = shelve_b + shelve_t
       # Readout, the red and blue should be open at expose_s
       expose_b = shelve_s + 2*safe # Doppler cooling sholdn't start until the
           shelving is off
       expose_s = expose_b + expose_t
       # Deshelve
```

```
deshelve_b = expose_s +safe
deshelve_s = deshelve_b + deshelve_t

clist = {'b_doppler': ['/Dev5/Ctr6',trigchan5,True],
        'r_doppler': ['/Dev1/Ctr4',trigchan1,False],
        'NR_sigma': ['/Dev5/Ctr4',trigchan5,False],
        'NR_pi' : ['/Dev5/Ctr5',trigchan5,False],
        'b_shutter': ['/Dev5/Ctr2',trigchan5,False],
        'r_shutter': ['/Dev5/Ctr3',trigchan5,False],
        'FOR_sigma' : ['/Dev5/Ctr7',trigchan5,False],
        'FOR_pi' : ['/Dev1/Ctr0',trigchan1,False],
        'NR_sigma_protect': ['/Dev1/Ctr1',trigchan1,False],
        'shelve': ['/Dev1/Ctr5',trigchan1,False],
        'deshelve': ['/Dev1/Ctr6',trigchan1,False],
        'camera': ['/Dev5/Ctr1',trigchan5,False],
        'shelve_shutter': ['/Dev1/Ctr3',trigchan1,True],
        'raman_shutter': ['/Dev1/Ctr2',trigchan1,False], #False due to
           using OR + NOT
        'raman_shutter_protect': ['/Dev5/Ctr0',trigchan5,False], #False
            due to using OR + NOT
        'FOR_shutter' : ['/Dev6/Ctr0',trigchan6,True],
        'deshelve_shutter': ['/Dev6/Ctr1',trigchan6,True]}

ctimes = {'b_doppler': [0.0, shelve_s+safe],
        'r_doppler': [ramancool_NR_sigma_s+safe, shelve_s+safe],
        'NR_sigma': [ramancool_NR_sigma_b,ramancool_NR_sigma_s],
        'NR_pi' : [ramancool_NR_pi_b,ramancool_NR_pi_s],
        #'b_shutter': [0.0, shelve_s],
        #'r_shutter': [ramancool_NR_sigma_s+shutter_t, shelve_s+
           shutter_t],
        'FOR_sigma' : [pi_pulse_b,pi_pulse_s],
        'FOR_pi' : [pi_pulse_b,pi_pulse_s],
        'NR_sigma_protect': [protect_b,protect_s],
        'shelve': [shelve_b, shelve_s],
        'deshelve': [deshelve_b, deshelve_s],
        'camera': [expose_b, expose_s],
        #'shelve_shutter': [shelve_b, shelve_s],
        #'raman_shutter': [0.0, ramancool_NR_sigma_s],
        #'raman_shutter_protect': [pi_pulse_s, protect_s],
        #'FOR_shutter': [ramancool_NR_sigma_s+shutter_t,pi_pulse_s],
        #'deshelve_shutter': [deshelve_b,deshelve_s]
```

```
                }

        for k in ctimes.keys():
            oneshots[k] = OneShot(*(clist[k]+ctimes[k]))

        # Then start all the oneshots
        for k in oneshots.keys():
            oneshots[k].begin()
        if plot_me:
            hold = plotOneShots(ctimes)

#        'raman_shutter': [shutter_t,stirap_protect_b],
#        'stirap_trig': [stirap_a_b+safe,stirap_a_b+2*safe],
#        #'shelve': [stirap_protect_b+shutter_t*2,stirap_protect_b+shutter_t*2+
   shelve_t],
#        #'shelve_shutter': [stirap_protect_b+shutter_t,stirap_protect_b+
   shutter_t*2+shelve_t],
#        'shelve': [stirap_protect_b+shutter_t,stirap_protect_b+shutter_t+
   shelve_t],
#        'shelve_shutter': [stirap_protect_b,stirap_protect_b+shutter_t+
   shelve_t],
#        'deshelve': [deshelve_b,deshelve_b+deshelve_t],
#        'camera': [cooling_s+shutter_t,cooling_s+shutter_t+expose_t]

        data = np.zeros((for_frequencies.shape[0],reps+1))
        for (i,for_freq) in enumerate(for_frequencies):
            # Set the dds frequencies. I am making Raman A be the negative one.
            # Also, I am making them be symmetric around 80, because I can.
            dds.SelectChannel(0)
            #dds.SetFrequency(80.0-f/2.0)
            dds.SetFrequency(80.0-11/2.0)
            dds.SelectChannel(1)
            #dds.SetFrequency(80.0+f/2.0)
            dds.SetFrequency(80.0-11/2.0-for_freq/2)
            data[i,0] = for_freq
            print('% of Scan Completed:')
            print((i+1.0)/for_frequencies.shape[0]*100.0)
            for j in range(reps):
                AO_detect_control.start()
                pmt.startRead()
                trig.setHigh()
```

```
           time.sleep(deshelve_b+deshelve_t+line_t)
           res = pmt.getRead()
           trig.setLow()
           data[i,j+1] = res
           time.sleep(recover_t) # Make sure that the cycle has finished
           AO_detect_control.stop()

# Build a suitable filename
t = time.localtime()
filename = "{}{:02g}{:02g}".format(t.tm_year,t.tm_mon,t.tm_mday)+ "
    _CWRamancool_FOR_Freq_NS_scanPMT"
i = 0
while os.path.exists(os.path.normpath(path + filename + "_{}.npy".format(i)
    )):
    i = i + 1
filenamen = os.path.normpath(path + filename + "_{}.npy".format(i))

# And save the data
np.save(filenamen,data)

for k in oneshots.keys():
    oneshots[k].stop()
del oneshots
del clist
del ctimes
oneshots = {}

clist = {'b_shutter': ['/Dev5/Ctr2',trigchan5,False],
            'r_shutter': ['/Dev5/Ctr3',trigchan5,False],
            'shelve_shutter': ['/Dev1/Ctr3',trigchan1,True],
            'raman_shutter': ['/Dev1/Ctr2',trigchan1,False], #False due to
               using OR + NOT
            'raman_shutter_protect': ['/Dev5/Ctr0',trigchan5,False], #False
                due to using OR + NOT
            'FOR_shutter': ['/Dev6/Ctr0',trigchan6,True],
            'deshelve_shutter': ['/Dev6/Ctr1',trigchan6,True]}

ctimes = {'b_shutter': [0.0, 2*shutter_t],
         'r_shutter': [0.0, 2*shutter_t],
         'shelve_shutter': [0.0, 2*shutter_t],
         'raman_shutter': [0.0, 2*shutter_t],
         'raman_shutter_protect': [0.0, 2*shutter_t],
         'FOR_shutter': [0.0, 2*shutter_t],
         'deshelve_shutter': [0.0,2*shutter_t]}
for k in ctimes.keys():
    oneshots[k] = OneShot(*(clist[k]+ctimes[k]))
for k in oneshots.keys():
    oneshots[k].begin()
trig.setHigh()
time.sleep(2*shutter_t)
trig.setLow()
time.sleep(recover_t)
for k in oneshots.keys():
    oneshots[k].stop()
del oneshots
del clist

#Clear the trigger and PMT tasks.
trig.stop()
del trig
pmt.stop()
del pmt
```

E.3.6.4. FOR Frequency Scan. The code below performs a FOR Raman two-photon detuning scan around a specified frequency range with single π pulse time.

```
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 16 15:09:02 2015

The main code for executing a frequency scan with STIRAP.

@author: CMS
@secondauthor: MGK
"""
# The path to save the data in
path = 'c:/data/Rabi/Freq/'

# Start PMT task.
pmt = PMT(pmtc)

if __name__ == '__main__':
    # Setup trigger out
    trig = TrigControl(trigsource)

    # And now all the oneshots
    oneshots = {}
    # First initialize all the oneshots

    time.sleep(1)
    # The first column is the frequencies, all others are the data collected
    # for that frequency
    with DDSControl(r'\\.\Ezssp-0',500.0) as dds:
        # Set the dds frequencies. I am making Raman A be the negative one.
        # Also, I am making them be symmetric around 80, because I can.
        dds.SelectChannel(2)
        #dds.SetFrequency(80.0-f/2.0)
        dds.SetFrequency(80.0-11/2.0)
        dds.SelectChannel(3)
        #dds.SetFrequency(80.0+f/2.0)
        dds.SetFrequency(80.0-11/2.0-cw_cool_freq/2)
        # FOR beams for pi pulses
#        for (i,f) in enumerate(frequencies):
#            # Set the dds frequencies. I am making Raman A be the negative one.
#            # Also, I am making them be symmetric around 80, because I can.
#            dds.SelectChannel(0)
#            #dds.SetFrequency(80.0-f/2.0)
#            dds.SetFrequency(80.0-15/2.0)
#            dds.SelectChannel(1)
#            #dds.SetFrequency(80.0+f/2.0)
#            dds.SetFrequency(80.0-15/2.0+f)
#            data[i,0] = f

        # For a scan, this will be incorporated into a for loop
        # FOR_Sigma pulse length
#        FOR_sigma = OneShot('/Dev5/Ctr7',trigchan5,True,stirap_a_b-safe,
   protect_b+safe)
        # FOR_Pi pulse length
#        FOR_pi = OneShot('/Dev1/Ctr0',trigchan1,True,stirap_b_b-safe,protect_b+
   safe)

        #Manually set the default states of the shutters.
        try:
            del clist
        except NameError:
            print

        clist = {'b_shutter': ['/Dev5/Ctr2',trigchan5,False],
                 'r_shutter': ['/Dev5/Ctr3',trigchan5,False],
                 'shelve_shutter': ['/Dev1/Ctr3',trigchan1,False],
                 'raman_shutter': ['/Dev1/Ctr2',trigchan1,True], #False due to
                     using OR + NOT
                 'raman_shutter_protect': ['/Dev5/Ctr0',trigchan5,True], #False
                     due to using OR + NOT
                 'FOR_shutter': ['/Dev6/Ctr0',trigchan6,False],
                 'deshelve_shutter': ['/Dev6/Ctr1',trigchan6,False]}

        ctimes = {'b_shutter': [0.0, 2*shutter_t],
                  'r_shutter': [0.0, 2*shutter_t],
                  'shelve_shutter': [0.0, 2*shutter_t],
                  'raman_shutter': [0.0, 2*shutter_t],
                  'raman_shutter_protect': [0.0, 2*shutter_t],
                  'FOR_shutter': [0.0, 2*shutter_t],
                  'deshelve_shutter': [0.0,2*shutter_t]}
        for k in ctimes.keys():
            oneshots[k] = OneShot(*(clist[k]+ctimes[k]))
        for k in oneshots.keys():
            oneshots[k].begin()
        trig.setHigh()
        time.sleep(2*shutter_t)
        trig.setLow()
        time.sleep(recover_t)
        for k in oneshots.keys():
            oneshots[k].stop()
        del oneshots
        del clist
        oneshots = {}
        # b is begin, s is stop

        # Raman Cooling
        ramancool_NR_sigma_b = safe
        ramancool_NR_pi_b = safe
        ramancool_NR_sigma_s = ramancool_NR_sigma_b + cw_cool_t + op_t # optical
             pumping
        ramancool_NR_pi_s = ramancool_NR_sigma_b + cw_cool_t
        # Pi Pulse, the stop time is defined down below, since it changes for a
            scan
        pi_pulse_b = ramancool_NR_sigma_s+2*safe +heating_t# The NR lights must
            be off
        pi_pulse_s = pi_pulse_b + for_pi_pulse_t
        # Protect
        protect_b = pi_pulse_s + safe # Optical pumping shouldnt be on during
            shelving
        protect_s = protect_b + protect_t
        # Shelve
        shelve_b = protect_s + safe
        shelve_s = shelve_b + shelve_t
        # Readout, the red and blue should be open at expose_s
        expose_b = shelve_s + 2*safe # Doppler cooling sholdn't start until the
            shelving is off
        expose_s = expose_b + expose_t
        # Deshelve
        deshelve_b = expose_s +safe
        deshelve_s = deshelve_b + deshelve_t
```

```
clist = {'b_doppler': ['/Dev5/Ctr6',trigchan5,True],
        'r_doppler': ['/Dev1/Ctr4',trigchan1,False],
        'NR_sigma': ['/Dev5/Ctr4',trigchan5,False],
        'NR_pi' : ['/Dev5/Ctr5',trigchan5,False],
        'b_shutter': ['/Dev5/Ctr2',trigchan5,False],
        'r_shutter': ['/Dev5/Ctr3',trigchan5,False],
        'FOR_sigma' : ['/Dev5/Ctr7',trigchan5,False],
        'FOR_pi' : ['/Dev1/Ctr0',trigchan1,False],
        'NR_sigma_protect': ['/Dev1/Ctr1',trigchan1,False],
        'shelve': ['/Dev1/Ctr5',trigchan1,False],
        'deshelve': ['/Dev1/Ctr6',trigchan1,False],
        'camera': ['/Dev5/Ctr1',trigchan5,False],
        'shelve_shutter': ['/Dev1/Ctr3',trigchan1,True],
        'raman_shutter': ['/Dev1/Ctr2',trigchan1,False], #False due to
           using OR + NOT
        'raman_shutter_protect': ['/Dev5/Ctr0',trigchan5,False], #False
            due to using OR + NOT
        'FOR_shutter' : ['/Dev6/Ctr0',trigchan6,True],
        'deshelve_shutter': ['/Dev6/Ctr1',trigchan6,True]}

ctimes = {'b_doppler': [0.0, shelve_s+safe],
        'r_doppler': [ramancool_NR_sigma_s+safe, shelve_s+safe],
        'NR_sigma': [ramancool_NR_sigma_b,ramancool_NR_sigma_s],
        'NR_pi' : [ramancool_NR_pi_b,ramancool_NR_pi_s],
        #'b_shutter': [0.0, shelve_s],
        #'r_shutter': [ramancool_NR_sigma_s+shutter_t, shelve_s+
           shutter_t],
        'FOR_sigma' : [pi_pulse_b,pi_pulse_s],
        'FOR_pi' : [pi_pulse_b,pi_pulse_s],
        'NR_sigma_protect': [protect_b,protect_s],
        'shelve': [shelve_b, shelve_s],
        'deshelve': [deshelve_b, deshelve_s],
        'camera': [expose_b, expose_s],
        #'shelve_shutter': [shelve_b, shelve_s],
        #'raman_shutter': [0.0, ramancool_NR_sigma_s],
        #'raman_shutter_protect': [pi_pulse_s, protect_s],
        #'FOR_shutter': [ramancool_NR_sigma_s+shutter_t,pi_pulse_s],
        #'deshelve_shutter': [deshelve_b,deshelve_s]
        }
        for k in ctimes.keys():
            oneshots[k] = OneShot(*(clist[k]+ctimes[k]))

        # Then start all the oneshots
        for k in oneshots.keys():
            oneshots[k].begin()
        if plot_me:
            hold = plotOneShots(ctimes)

#        'raman_shutter': [shutter_t,stirap_protect_b],
#        'stirap_trig': [stirap_a_b+safe,stirap_a_b+2*safe],
#        #'shelve': [stirap_protect_b+shutter_t*2,stirap_protect_b+shutter_t*2+
    shelve_t],
#        #'shelve_shutter': [stirap_protect_b+shutter_t,stirap_protect_b+
    shutter_t*2+shelve_t],
#        'shelve': [stirap_protect_b+shutter_t,stirap_protect_b+shutter_t+
    shelve_t],
#        'shelve_shutter': [stirap_protect_b,stirap_protect_b+shutter_t+
    shelve_t],
#        'deshelve': [deshelve_b,deshelve_b+deshelve_t],
#        'camera': [cooling_s+shutter_t,cooling_s+shutter_t+expose_t]

        data = np.zeros((for_frequencies.shape[0],reps+1))
        for (i,for_freq) in enumerate(for_frequencies):
            # Set the dds frequencies. I am making Raman A be the negative one.
            # Also, I am making them be symmetric around 80, because I can.
            dds.SelectChannel(0)
            #dds.SetFrequency(80.0-f/2.0)
            dds.SetFrequency(80.0-11/2.0)
            dds.SelectChannel(1)
            #dds.SetFrequency(80.0+f/2.0)
            dds.SetFrequency(80.0-11/2.0-for_freq/2)
            data[i,0] = for_freq
            print('% of Scan Completed:')
            print((i+1.0)/for_frequencies.shape[0]*100.0)
            for j in range(reps):
                AO_detect_control.start()
                pmt.startRead()
                trig.setHigh()
                time.sleep(deshelve_b+deshelve_t+line_t)
                res = pmt.getRead()
                trig.setLow()
                data[i,j+1] = res
                time.sleep(recover_t) # Make sure that the cycle has finished
                AO_detect_control.stop()

    # Build a suitable filename
    t = time.localtime()
    filename = "{}{:02g}{:02g}".format(t.tm_year,t.tm_mon,t.tm_mday)+ "
        _CWRamancool_FOR_Freq_NS_scanPMT"
    i = 0
    while os.path.exists(os.path.normpath(path + filename + "_{}.npy".format(i)
        )):
        i = i + 1
    filenamen = os.path.normpath(path + filename + "_{}.npy".format(i))

    # And save the data
    np.save(filenamen,data)

    for k in oneshots.keys():
        oneshots[k].stop()
    del oneshots
    del clist
    del ctimes
    oneshots = {}

    clist = {'b_shutter': ['/Dev5/Ctr2',trigchan5,False],
                'r_shutter': ['/Dev5/Ctr3',trigchan5,False],
                'shelve_shutter': ['/Dev1/Ctr3',trigchan1,True],
                'raman_shutter': ['/Dev1/Ctr2',trigchan1,False], #False due to
                    using OR + NOT
                'raman_shutter_protect': ['/Dev5/Ctr0',trigchan5,False], #False
                     due to using OR + NOT
                'FOR_shutter': ['/Dev6/Ctr0',trigchan6,True],
                'deshelve_shutter': ['/Dev6/Ctr1',trigchan6,True]}

    ctimes = {'b_shutter': [0.0, 2*shutter_t],
             'r_shutter': [0.0, 2*shutter_t],
             'shelve_shutter': [0.0, 2*shutter_t],
             'raman_shutter': [0.0, 2*shutter_t],
             'raman_shutter_protect': [0.0, 2*shutter_t],
             'FOR_shutter': [0.0, 2*shutter_t],
             'deshelve_shutter': [0.0,2*shutter_t]}
    for k in ctimes.keys():
        oneshots[k] = OneShot(*(clist[k]+ctimes[k]))
    for k in oneshots.keys():
        oneshots[k].begin()
    trig.setHigh()
    time.sleep(2*shutter_t)
    trig.setLow()
    time.sleep(recover_t)
    for k in oneshots.keys():
        oneshots[k].stop()
    del oneshots
    del clist

    #Clear the trigger and PMT tasks.
    trig.stop()
    del trig
    pmt.stop()
    del pmt
```

E.3.6.5. FOR Heating Rate Scan. The code below scans the delay after optical pumping and following the FOR Raman state readout steps in order to measure the trap heating rate.

```
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 16 15:09:02 2015

The main code for executing a frequency scan with STIRAP.

@author: CMS
@secondauthor: MGK
"""
# The path to save the data in
path = 'c:/data/Rabi/Heating/'

# Start PMT task.
pmt = PMT(pmtc)

if __name__ == '__main__':
    # Setup trigger out
    trig = TrigControl(trigsource)

    # And now all the oneshots
    oneshots = {}
    # First initialize all the oneshots

    time.sleep(1)
    # The first column is the frequencies, all others are the data collected
    # for that frequency

    # FOR beams for pi pulses

#       for (i,f) in enumerate(frequencies):
#           # Set the dds frequencies. I am making Raman A be the negative one.
#           # Also, I am making them be symmetric around 80, because I can.
#           dds.SelectChannel(0)
#           #dds.SetFrequency(80.0-f/2.0)
#           dds.SetFrequency(80.0-15/2.0)
#           dds.SelectChannel(1)
#           #dds.SetFrequency(80.0+f/2.0)
#           dds.SetFrequency(80.0-15/2.0+f)
#           data[i,0] = f

    # For a scan, this will be incorporated into a for loop
    # FOR_Sigma pulse length
#       FOR_sigma = OneShot('/Dev5/Ctr7',trigchan5,True,stirap_a_b-safe,
    protect_b+safe)
    # FOR_Pi pulse length
#       FOR_pi = OneShot('/Dev1/Ctr0',trigchan1,True,stirap_b_b-safe,protect_b+
    safe)

    #Manually set the default states of the shutters.
    try:
        del clist
    except NameError:
        print
    clist = {'b_shutter': ['/Dev5/Ctr2',trigchan5,False],
            'r_shutter': ['/Dev5/Ctr3',trigchan5,False],
            'shelve_shutter': ['/Dev1/Ctr3',trigchan1,False],
            'raman_shutter': ['/Dev1/Ctr2',trigchan1,True], #False due to using
                 OR + NOT
            'raman_shutter_protect': ['/Dev5/Ctr0',trigchan5,True], #False due
                to using OR + NOT
            'FOR_shutter': ['/Dev6/Ctr0',trigchan6,False],
            'deshelve_shutter': ['/Dev6/Ctr1',trigchan6,False]}

    ctimes = {'b_shutter': [0.0, 2*shutter_t],
             'r_shutter': [0.0, 2*shutter_t],
             'shelve_shutter': [0.0, 2*shutter_t],
             'raman_shutter': [0.0, 2*shutter_t],
             'raman_shutter_protect': [0.0, 2*shutter_t],
             'FOR_shutter': [0.0, 2*shutter_t],
             'deshelve_shutter': [0.0,2*shutter_t]}
    for k in ctimes.keys():
        oneshots[k] = OneShot(*(clist[k]+ctimes[k]))
    for k in oneshots.keys():
        oneshots[k].begin()
    trig.setHigh()
    time.sleep(2*shutter_t)
```

```
trig.setLow()
time.sleep(recover_t)
for k in oneshots.keys():
    oneshots[k].stop()
del oneshots
del clist
oneshots = {}
# b is begin, s is stop

data = np.zeros((heating_times_freqs.shape[0],reps+2))
for i in range(heating_times_freqs.shape[0]):
    heating_t = heating_times_freqs[i,0]
    for_freq = heating_times_freqs[i,1]

    with DDSControl(r'\\.\Ezssp-0',500.0) as dds:
        # Set the dds frequencies. I am making Raman A be the negative one.
        # Also, I am making them be symmetric around 80, because I can.
        dds.SelectChannel(2)
        #dds.SetFrequency(80.0-f/2.0)
        dds.SetFrequency(80.0-11/2.0)
        dds.SelectChannel(3)
        #dds.SetFrequency(80.0+f/2.0)
        dds.SetFrequency(80.0-11/2.0-cw_cool_freq/2)
        dds.SelectChannel(0)
        #dds.SetFrequency(80.0-f/2.0)
        dds.SetFrequency(80.0-11/2.0)
        dds.SelectChannel(1)
        #dds.SetFrequency(80.0+f/2.0)
        dds.SetFrequency(80.0-11/2.0-for_freq/2)

        # And now all the oneshots
        oneshots = {}
        # First initialize all the oneshots


        if np.logical_and(for_freq <= for_carrier+0.5, for_freq >=
            for_carrier-0.5):
            for_pi_pulse_t = for_carrier_pi_pulse_t
        else:
            for_pi_pulse_t = for_bsb_pi_pulse_t
```

```
# b is begin, s is stop
# Raman Cooling
ramancool_NR_sigma_b = safe
ramancool_NR_pi_b = safe
ramancool_NR_sigma_s = ramancool_NR_sigma_b + cw_cool_t + op_t #
    optical pumping
ramancool_NR_pi_s = ramancool_NR_sigma_b + cw_cool_t
# Pi Pulse, the stop time is defined down below, since it changes
    for a scan
pi_pulse_b = ramancool_NR_sigma_s+2*safe +heating_t# The NR lights
    must be off
pi_pulse_s = pi_pulse_b + for_pi_pulse_t
# Protect
protect_b = pi_pulse_s + safe # Optical pumping shouldnt be on
    during shelving
protect_s = protect_b + protect_t
# Shelve
shelve_b = protect_s + safe
shelve_s = shelve_b + shelve_t
# Readout, the red and blue should be open at expose_s
expose_b = shelve_s + 2*safe # Doppler cooling sholdn’t start until
    the shelving is off
expose_s = expose_b + expose_t
# Deshelve
deshelve_b = expose_s +safe
deshelve_s = deshelve_b + deshelve_t

clist = {’b_doppler’: [’/Dev5/Ctr6’,trigchan5,True],
        ’r_doppler’: [’/Dev1/Ctr4’,trigchan1,False],
        ’NR_sigma’: [’/Dev5/Ctr4’,trigchan5,False],
        ’NR_pi’ : [’/Dev5/Ctr5’,trigchan5,False],
        ’b_shutter’: [’/Dev5/Ctr2’,trigchan5,False],
        ’r_shutter’: [’/Dev5/Ctr3’,trigchan5,False],
        ’FOR_sigma’ : [’/Dev5/Ctr7’,trigchan5,False],
        ’FOR_pi’ : [’/Dev1/Ctr0’,trigchan1,False],
        ’NR_sigma_protect’: [’/Dev1/Ctr1’,trigchan1,False],
        ’shelve’: [’/Dev1/Ctr5’,trigchan1,False],
        ’deshelve’: [’/Dev1/Ctr6’,trigchan1,False],
        ’camera’: [’/Dev5/Ctr1’,trigchan5,False],
        ’shelve_shutter’: [’/Dev1/Ctr3’,trigchan1,True],
                'raman_shutter': ['/Dev1/Ctr2',trigchan1,False], #False due
                   to using OR + NOT
                'raman_shutter_protect': ['/Dev5/Ctr0',trigchan5,False], #
                   False due to using OR + NOT
                'FOR_shutter' : ['/Dev6/Ctr0',trigchan6,True],
                'deshelve_shutter': ['/Dev6/Ctr1',trigchan6,True]}

        ctimes = {'b_doppler': [0.0, shelve_s+safe],
                'r_doppler': [ramancool_NR_sigma_s+safe, shelve_s+safe],
                'NR_sigma': [ramancool_NR_sigma_b,ramancool_NR_sigma_s],
                'NR_pi' : [ramancool_NR_pi_b,ramancool_NR_pi_s],
                #'b_shutter': [0.0, shelve_s],
                #'r_shutter': [ramancool_NR_sigma_s+shutter_t, shelve_s+
                   shutter_t],
                'FOR_sigma' : [pi_pulse_b,pi_pulse_s],
                'FOR_pi' : [pi_pulse_b,pi_pulse_s],
                'NR_sigma_protect': [protect_b,protect_s],
                'shelve': [shelve_b, shelve_s],
                'deshelve': [deshelve_b, deshelve_s],
                'camera': [expose_b, expose_s],
                #'shelve_shutter': [shelve_b, shelve_s],
                #'raman_shutter': [0.0, ramancool_NR_sigma_s],
                #'raman_shutter_protect': [pi_pulse_s, protect_s],
                #'FOR_shutter': [ramancool_NR_sigma_s+shutter_t,pi_pulse_s],
                #'deshelve_shutter': [deshelve_b,deshelve_s]
                }

        for k in ctimes.keys():
            oneshots[k] = OneShot(*(clist[k]+ctimes[k]))

        # Then start all the oneshots
        for k in oneshots.keys():
            oneshots[k].begin()
        if np.logical_and(plot_me == True, i == 0):
            hold = plotOneShots(ctimes)

#        'raman_shutter': [shutter_t,stirap_protect_b],
#        'stirap_trig': [stirap_a_b+safe,stirap_a_b+2*safe],
#        #'shelve': [stirap_protect_b+shutter_t*2,stirap_protect_b+
   shutter_t*2+shelve_t],
#        #’shelve_shutter’: [stirap_protect_b+shutter_t,stirap_protect_b+
    shutter_t*2+shelve_t],
#        ’shelve’: [stirap_protect_b+shutter_t,stirap_protect_b+shutter_t+
    shelve_t],
#        ’shelve_shutter’: [stirap_protect_b,stirap_protect_b+shutter_t+
    shelve_t],
#        ’deshelve’: [deshelve_b,deshelve_b+deshelve_t],
#        ’camera’: [cooling_s+shutter_t,cooling_s+shutter_t+expose_t]




        # Set the dds frequencies. I am making Raman A be the negative one.
        # Also, I am making them be symmetric around 80, because I can.

        data[i,0] = heating_t
        data[i,1] = for_freq
        print(’% of Scan Completed:’)
        print((i+1.0)/(heating_times_freqs.shape[0])*100.0)
        for j in range(reps):
            AO_detect_control.start()
            pmt.startRead()
            trig.setHigh()
            time.sleep(deshelve_b+deshelve_t+line_t)
            res = pmt.getRead()
            trig.setLow()
            data[i,j+2] = res
            time.sleep(recover_t) # Make sure that the cycle has finished
            AO_detect_control.stop()

        #clear the oneshots to start again!
        for k in oneshots.keys():
            oneshots[k].stop()
        del oneshots

# Build a suitable filename
t = time.localtime()
filename = "{}{:02g}{:02g}".format(t.tm_year,t.tm_mon,t.tm_mday)+ "
    _CWRamancool_FOR_Heating_NS_scanPMT"
i = 0
```

```
while os.path.exists(os.path.normpath(path + filename + "_{}.npy".format(i)
    )):
    i = i + 1
filenamen = os.path.normpath(path + filename + "_{}.npy".format(i))

# And save the data
np.save(filenamen,data)

del clist
del ctimes
oneshots = {}

clist = {’b_shutter’: [’/Dev5/Ctr2’,trigchan5,False],
            ’r_shutter’: [’/Dev5/Ctr3’,trigchan5,False],
            ’shelve_shutter’: [’/Dev1/Ctr3’,trigchan1,True],
            ’raman_shutter’: [’/Dev1/Ctr2’,trigchan1,False], #False due to
               using OR + NOT
            ’raman_shutter_protect’: [’/Dev5/Ctr0’,trigchan5,False], #False
                due to using OR + NOT
            ’FOR_shutter’: [’/Dev6/Ctr0’,trigchan6,True],
            ’deshelve_shutter’: [’/Dev6/Ctr1’,trigchan6,True]}

ctimes = {’b_shutter’: [0.0, 2*shutter_t],
         ’r_shutter’: [0.0, 2*shutter_t],
         ’shelve_shutter’: [0.0, 2*shutter_t],
         ’raman_shutter’: [0.0, 2*shutter_t],
         ’raman_shutter_protect’: [0.0, 2*shutter_t],
         ’FOR_shutter’: [0.0, 2*shutter_t],
         ’deshelve_shutter’: [0.0,2*shutter_t]}
for k in ctimes.keys():
    oneshots[k] = OneShot(*(clist[k]+ctimes[k]))
for k in oneshots.keys():
    oneshots[k].begin()
trig.setHigh()
time.sleep(2*shutter_t)
trig.setLow()
time.sleep(recover_t)
for k in oneshots.keys():
    oneshots[k].stop()
del oneshots
del clist
```

```
trig.stop()
del trig
pmt.stop()
del pmt
```

E.3.6.6. FOR Optical Pumping Time. The code below scans the optical pumping time to determine the shortest duration of time this can be.

```
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 16 15:09:02 2015

The main code for executing a protect time scan at a single frequency with FOR.

@author: CMS
@secondauthor: MGK
"""
# The path to save the data in
path = 'c:/data/Rabi/OP/'

# Start PMT task.
pmt = PMT(pmtc)

if __name__ == '__main__':
    # Setup trigger out
    trig = TrigControl(trigsource)

    with DDSControl(r'\\.\Ezssp-0',500.0) as dds:
        # Set the dds frequencies. Ch0,Ch1 are FOR sigma,pi. Ch2,Ch3 are NR.
        #The offset is to center the AOM's at resonance.
        dds.SelectChannel(0)
        dds.SetFrequency(80.0-11/2.0)
        dds.SelectChannel(1)
        dds.SetFrequency(80.0-11/2.0-for_freq/2)
        dds.SelectChannel(2)
        dds.SetFrequency(80.0-11/2.0)
        dds.SelectChannel(3)
        dds.SetFrequency(80.0-11/2.0-cw_cool_freq/2)

    try:
        del clist
    except NameError:
        print
    oneshots = {}
    clist = {'b_shutter': ['/Dev5/Ctr2',trigchan5,False],
        'r_shutter': ['/Dev5/Ctr3',trigchan5,False],
        'shelve_shutter': ['/Dev1/Ctr3',trigchan1,False],
        'raman_shutter': ['/Dev1/Ctr2',trigchan1,True], #False due to using
             OR + NOT
        'raman_shutter_protect': ['/Dev5/Ctr0',trigchan5,True], #False due
            to using OR + NOT
        'FOR_shutter': ['/Dev6/Ctr0',trigchan6,False],
        'deshelve_shutter': ['/Dev6/Ctr1',trigchan6,False]}

ctimes = {'b_shutter': [0.0, 2*shutter_t],
         'r_shutter': [0.0, 2*shutter_t],
         'shelve_shutter': [0.0, 2*shutter_t],
         'raman_shutter': [0.0, 2*shutter_t],
         'raman_shutter_protect': [0.0, 2*shutter_t],
         'FOR_shutter': [0.0, 2*shutter_t],
         'deshelve_shutter': [0.0,2*shutter_t]}
for k in ctimes.keys():
    oneshots[k] = OneShot(*(clist[k]+ctimes[k]))
for k in oneshots.keys():
    oneshots[k].begin()
trig.setHigh()
time.sleep(2*shutter_t)
trig.setLow()
time.sleep(recover_t)
for k in oneshots.keys():
    oneshots[k].stop()
del oneshots
del clist

time.sleep(1)
data = np.zeros((op_times.shape[0],reps+1))
for (i,op_t) in enumerate(op_times):
    # And now all the oneshots
    oneshots = {}

    # Raman Cooling
    ramancool_NR_sigma_b = safe
    ramancool_NR_pi_b = safe
    ramancool_NR_sigma_s = ramancool_NR_sigma_b + cw_cool_t + op_t # optical
         pumping
    ramancool_NR_pi_s = ramancool_NR_sigma_b + cw_cool_t
```

```
# Pi Pulse, the stop time is defined down below, since it changes for a scan
pi_pulse_b = ramancool_NR_sigma_s+2*safe +heating_t# The NR lights must be off
pi_pulse_s = pi_pulse_b + for_pi_pulse_t
# Protect
protect_b = pi_pulse_s + safe # Optical pumping shouldnt be on during shelving
protect_s = protect_b + protect_t
# Shelve
shelve_b = protect_s + safe
shelve_s = shelve_b + shelve_t
# Readout, the red and blue should be open at expose_s
expose_b = shelve_s + 2*safe # Doppler cooling sholdn’t start until the shelving is off
expose_s = expose_b + expose_t
# Deshelve
deshelve_b = expose_s +safe
deshelve_s = deshelve_b + deshelve_t

clist = {’b_doppler’: [’/Dev5/Ctr6’,trigchan5,True],
        ’r_doppler’: [’/Dev1/Ctr4’,trigchan1,False],
        ’NR_sigma’: [’/Dev5/Ctr4’,trigchan5,False],
        ’NR_pi’ : [’/Dev5/Ctr5’,trigchan5,False],
        ’b_shutter’: [’/Dev5/Ctr2’,trigchan5,False],
        ’r_shutter’: [’/Dev5/Ctr3’,trigchan5,False],
        ’FOR_sigma’ : [’/Dev5/Ctr7’,trigchan5,False],
        ’FOR_pi’ : [’/Dev1/Ctr0’,trigchan1,False],
        ’NR_sigma_protect’: [’/Dev1/Ctr1’,trigchan1,False],
        ’shelve’: [’/Dev1/Ctr5’,trigchan1,False],
        ’deshelve’: [’/Dev1/Ctr6’,trigchan1,False],
        ’camera’: [’/Dev5/Ctr1’,trigchan5,False],
        ’shelve_shutter’: [’/Dev1/Ctr3’,trigchan1,True],
        ’raman_shutter’: [’/Dev1/Ctr2’,trigchan1,False], #False due to using OR + NOT
        ’raman_shutter_protect’: [’/Dev5/Ctr0’,trigchan5,False], #False due to using OR + NOT
        ’FOR_shutter’ : [’/Dev6/Ctr0’,trigchan6,True],
        ’deshelve_shutter’: [’/Dev6/Ctr1’,trigchan6,True]}

ctimes = {’b_doppler’: [0.0, shelve_s+safe],
        ’r_doppler’: [ramancool_NR_sigma_s+safe, shelve_s+safe],
        ’NR_sigma’: [ramancool_NR_sigma_b,ramancool_NR_sigma_s],
        ’NR_pi’ : [ramancool_NR_pi_b,ramancool_NR_pi_s],
        #’b_shutter’: [0.0, shelve_s],
        #’r_shutter’: [ramancool_NR_sigma_s+shutter_t, shelve_s+
            shutter_t],
        ’FOR_sigma’ : [pi_pulse_b,pi_pulse_s],
        ’FOR_pi’ : [pi_pulse_b,pi_pulse_s],
        ’NR_sigma_protect’: [protect_b,protect_s],
        ’shelve’: [shelve_b, shelve_s],
        ’deshelve’: [deshelve_b, deshelve_s],
        ’camera’: [expose_b, expose_s],
        #’shelve_shutter’: [shelve_b, shelve_s],
        #’raman_shutter’: [0.0, ramancool_NR_sigma_s],
        #’raman_shutter_protect’: [pi_pulse_s, protect_s],
        #’FOR_shutter’: [ramancool_NR_sigma_s+shutter_t,pi_pulse_s],
        #’deshelve_shutter’: [deshelve_b,deshelve_s]
        }

for k in ctimes.keys():
    oneshots[k] = OneShot(*(clist[k]+ctimes[k]))

# Then start all the oneshots
for k in oneshots.keys():
    oneshots[k].begin()
if np.logical_and(plot_me == True, i == 0):
    hold = plotOneShots(ctimes)

data[i,0] = op_t
print(’% of Scan Completed:’)
print((i+1.0)/op_times.shape[0]*100.0)
for j in range(reps):
    AO_detect_control.start()
    pmt.startRead()
    trig.setHigh()
    time.sleep(deshelve_s+shutter_t+line_t)
    res = pmt.getRead()
    trig.setLow()
    data[i,j+1] = res
    time.sleep(recover_t) # Make sure that the cycle has finished
    AO_detect_control.stop()
    #clear the oneshots to start again!
    for k in oneshots.keys():
        oneshots[k].stop()
    del oneshots

# Build a suitable filename
t = time.localtime()
filename = "{}{:02g}{:02g}".format(t.tm_year,t.tm_mon,t.tm_mday)+ "
    _CWRamancool_FOR_OP_NS_scanPMT"
i = 0
while os.path.exists(os.path.normpath(path + filename + "_{}.npy".format(i)
    )):
    i = i + 1
filenamen = os.path.normpath(path + filename + "_{}.npy".format(i))

# And save the data
np.save(filenamen,data)

del clist
del ctimes
oneshots = {}

clist = {'b_shutter': ['/Dev5/Ctr2',trigchan5,False],
            'r_shutter': ['/Dev5/Ctr3',trigchan5,False],
            'shelve_shutter': ['/Dev1/Ctr3',trigchan1,True],
            'raman_shutter': ['/Dev1/Ctr2',trigchan1,False], #False due to
                using OR + NOT
            'raman_shutter_protect': ['/Dev5/Ctr0',trigchan5,False], #False
                 due to using OR + NOT
            'FOR_shutter': ['/Dev6/Ctr0',trigchan6,True],
            'deshelve_shutter': ['/Dev6/Ctr1',trigchan6,True]}

ctimes = {'b_shutter': [0.0, 2*shutter_t],
         'r_shutter': [0.0, 2*shutter_t],
         'shelve_shutter': [0.0, 2*shutter_t],
         'raman_shutter': [0.0, 2*shutter_t],
         'raman_shutter_protect': [0.0, 2*shutter_t],
         'FOR_shutter': [0.0, 2*shutter_t],
         'deshelve_shutter': [0.0,2*shutter_t]}
for k in ctimes.keys():
    oneshots[k] = OneShot(*(clist[k]+ctimes[k]))
for k in oneshots.keys():
    oneshots[k].begin()
trig.setHigh()
time.sleep(2*shutter_t)
trig.setLow()
time.sleep(recover_t)
for k in oneshots.keys():
    oneshots[k].stop()
del oneshots
del clist

#Clear the trigger and PMT tasks.
trig.stop()
del trig
pmt.stop()
del pmt
```

E.3.6.7. FOR Protect Time. The code below scans the protect time to determine the optimal value of this time.

```
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 16 15:09:02 2015

The main code for executing a protect time scan at a single frequency with FOR.

@author: CMS
@secondauthor: MGK
"""
# The path to save the data in
path = 'c:/data/Rabi/Protect/'

# Start PMT task.
pmt = PMT(pmtc)

if __name__ == '__main__':
    # Setup trigger out
    trig = TrigControl(trigsource)

    with DDSControl(r'\\.\Ezssp-0',500.0) as dds:
        # Set the dds frequencies. Ch0,Ch1 are FOR sigma,pi. Ch2,Ch3 are NR.
        #The offset is to center the AOM's at resonance.
        dds.SelectChannel(0)
        dds.SetFrequency(80.0-11/2.0)
        dds.SelectChannel(1)
        dds.SetFrequency(80.0-11/2.0-for_freq/2)
        dds.SelectChannel(2)
        dds.SetFrequency(80.0-11/2.0)
        dds.SelectChannel(3)
        dds.SetFrequency(80.0-11/2.0-cw_cool_freq/2)

    try:
        del clist
    except NameError:
        print
    oneshots = {}
    clist = {'b_shutter': ['/Dev5/Ctr2',trigchan5,False],
         'r_shutter': ['/Dev5/Ctr3',trigchan5,False],
         'shelve_shutter': ['/Dev1/Ctr3',trigchan1,False],
         'raman_shutter': ['/Dev1/Ctr2',trigchan1,True], #False due to using
             OR + NOT
         'raman_shutter_protect': ['/Dev5/Ctr0',trigchan5,True], #False due
            to using OR + NOT
         'FOR_shutter': ['/Dev6/Ctr0',trigchan6,False],
         'deshelve_shutter': ['/Dev6/Ctr1',trigchan6,False]}

ctimes = {'b_shutter': [0.0, 2*shutter_t],
          'r_shutter': [0.0, 2*shutter_t],
          'shelve_shutter': [0.0, 2*shutter_t],
          'raman_shutter': [0.0, 2*shutter_t],
          'raman_shutter_protect': [0.0, 2*shutter_t],
          'FOR_shutter': [0.0, 2*shutter_t],
          'deshelve_shutter': [0.0,2*shutter_t]}
for k in ctimes.keys():
    oneshots[k] = OneShot(*(clist[k]+ctimes[k]))
for k in oneshots.keys():
    oneshots[k].begin()
trig.setHigh()
time.sleep(2*shutter_t)
trig.setLow()
time.sleep(recover_t)
for k in oneshots.keys():
    oneshots[k].stop()
del oneshots
del clist

time.sleep(1)
data = np.zeros((protect_times.shape[0],reps+1))
for (i,protect_t) in enumerate(protect_times):
    # And now all the oneshots
    oneshots = {}

    # Raman Cooling
    ramancool_NR_sigma_b = safe
    ramancool_NR_pi_b = safe
    ramancool_NR_sigma_s = ramancool_NR_sigma_b + cw_cool_t + op_t # optical
        pumping
    ramancool_NR_pi_s = ramancool_NR_sigma_b + cw_cool_t
```

```
# Pi Pulse, the stop time is defined down below, since it changes for a
    scan
pi_pulse_b = ramancool_NR_sigma_s+2*safe +heating_t# The NR lights must
    be off
pi_pulse_s = pi_pulse_b + for_pi_pulse_t
# Protect
protect_b = pi_pulse_s + safe # Optical pumping shouldnt be on during
    shelving
protect_s = protect_b + protect_t
# Shelve
shelve_b = protect_s + safe
shelve_s = shelve_b + shelve_t
# Readout, the red and blue should be open at expose_s
expose_b = shelve_s + 2*safe # Doppler cooling sholdn’t start until the
    shelving is off
expose_s = expose_b + expose_t
# Deshelve
deshelve_b = expose_s +safe
deshelve_s = deshelve_b + deshelve_t

clist = {’b_doppler’: [’/Dev5/Ctr6’,trigchan5,True],
        ’r_doppler’: [’/Dev1/Ctr4’,trigchan1,False],
        ’NR_sigma’: [’/Dev5/Ctr4’,trigchan5,False],
        ’NR_pi’ : [’/Dev5/Ctr5’,trigchan5,False],
        ’b_shutter’: [’/Dev5/Ctr2’,trigchan5,False],
        ’r_shutter’: [’/Dev5/Ctr3’,trigchan5,False],
        ’FOR_sigma’ : [’/Dev5/Ctr7’,trigchan5,False],
        ’FOR_pi’ : [’/Dev1/Ctr0’,trigchan1,False],
        ’NR_sigma_protect’: [’/Dev1/Ctr1’,trigchan1,False],
        ’shelve’: [’/Dev1/Ctr5’,trigchan1,False],
        ’deshelve’: [’/Dev1/Ctr6’,trigchan1,False],
        ’camera’: [’/Dev5/Ctr1’,trigchan5,False],
        ’shelve_shutter’: [’/Dev1/Ctr3’,trigchan1,True],
        ’raman_shutter’: [’/Dev1/Ctr2’,trigchan1,False], #False due to
            using OR + NOT
        ’raman_shutter_protect’: [’/Dev5/Ctr0’,trigchan5,False], #False
             due to using OR + NOT
        ’FOR_shutter’ : [’/Dev6/Ctr0’,trigchan6,True],
        ’deshelve_shutter’: [’/Dev6/Ctr1’,trigchan6,True]}

ctimes = {’b_doppler’: [0.0, shelve_s+safe],
        'r_doppler': [ramancool_NR_sigma_s+safe, shelve_s+safe],
        'NR_sigma': [ramancool_NR_sigma_b,ramancool_NR_sigma_s],
        'NR_pi' : [ramancool_NR_pi_b,ramancool_NR_pi_s],
        #'b_shutter': [0.0, shelve_s],
        #'r_shutter': [ramancool_NR_sigma_s+shutter_t, shelve_s+
            shutter_t],
        'FOR_sigma' : [pi_pulse_b,pi_pulse_s],
        'FOR_pi' : [pi_pulse_b,pi_pulse_s],
        'NR_sigma_protect': [protect_b,protect_s],
        'shelve': [shelve_b, shelve_s],
        'deshelve': [deshelve_b, deshelve_s],
        'camera': [expose_b, expose_s],
        #'shelve_shutter': [shelve_b, shelve_s],
        #'raman_shutter': [0.0, ramancool_NR_sigma_s],
        #'raman_shutter_protect': [pi_pulse_s, protect_s],
        #'FOR_shutter': [ramancool_NR_sigma_s+shutter_t,pi_pulse_s],
        #'deshelve_shutter': [deshelve_b,deshelve_s]
        }

for k in ctimes.keys():
    oneshots[k] = OneShot(*(clist[k]+ctimes[k]))

# Then start all the oneshots
for k in oneshots.keys():
    oneshots[k].begin()
if np.logical_and(plot_me == True, i == 0):
    hold = plotOneShots(ctimes)

data[i,0] = protect_t
print('% of Scan Completed:')
print((i+1.0)/protect_times.shape[0]*100.0)
for j in range(reps):
    AO_detect_control.start()
    pmt.startRead()
    trig.setHigh()
    time.sleep(deshelve_s+shutter_t+line_t)
    res = pmt.getRead()
    trig.setLow()
    data[i,j+1] = res
    time.sleep(recover_t) # Make sure that the cycle has finished
    AO_detect_control.stop()
    #clear the oneshots to start again!
    for k in oneshots.keys():
        oneshots[k].stop()
    del oneshots

# Build a suitable filename
t = time.localtime()
filename = "{}{:02g}{:02g}".format(t.tm_year,t.tm_mon,t.tm_mday)+ "
    _CWRamancool_FOR_Protect_NS_scanPMT"
i = 0
while os.path.exists(os.path.normpath(path + filename + "_{}.npy".format(i)
    )):
    i = i + 1
filenamen = os.path.normpath(path + filename + "_{}.npy".format(i))

# And save the data
np.save(filenamen,data)

del clist
del ctimes
oneshots = {}

clist = {'b_shutter': ['/Dev5/Ctr2',trigchan5,False],
            'r_shutter': ['/Dev5/Ctr3',trigchan5,False],
            'shelve_shutter': ['/Dev1/Ctr3',trigchan1,True],
            'raman_shutter': ['/Dev1/Ctr2',trigchan1,False], #False due to
                using OR + NOT
            'raman_shutter_protect': ['/Dev5/Ctr0',trigchan5,False], #False
                 due to using OR + NOT
            'FOR_shutter': ['/Dev6/Ctr0',trigchan6,True],
            'deshelve_shutter': ['/Dev6/Ctr1',trigchan6,True]}

ctimes = {'b_shutter': [0.0, 2*shutter_t],
         'r_shutter': [0.0, 2*shutter_t],
         'shelve_shutter': [0.0, 2*shutter_t],
         'raman_shutter': [0.0, 2*shutter_t],
         'raman_shutter_protect': [0.0, 2*shutter_t],
         'FOR_shutter': [0.0, 2*shutter_t],
         'deshelve_shutter': [0.0,2*shutter_t]}
for k in ctimes.keys():
    oneshots[k] = OneShot(*(clist[k]+ctimes[k]))
for k in oneshots.keys():
    oneshots[k].begin()
trig.setHigh()
time.sleep(2*shutter_t)
trig.setLow()
time.sleep(recover_t)
for k in oneshots.keys():
    oneshots[k].stop()
del oneshots
del clist

#Clear the trigger and PMT tasks.
trig.stop()
del trig
pmt.stop()
del pmt
```

E.3.6.8. FOR Temperature Scan. The code below scans the FOR Raman frequencies around the carrier, BSB, and RSB with corresponding π pulse time to measure the final $\bar{n}$ value.

```
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 16 15:09:02 2015

The main code for executing a frequency scan with STIRAP.

@author: CMS
@secondauthor: MGK
"""
# The path to save the data in
path = 'c:/data/Rabi/Temperature/'

# Start PMT task.
pmt = PMT(pmtc)

if __name__ == '__main__':
    # Setup trigger out
    trig = TrigControl(trigsource)

    # And now all the oneshots
    oneshots = {}
    # First initialize all the oneshots

    time.sleep(1)
    # The first column is the frequencies, all others are the data collected
    # for that frequency
    with DDSControl(r'\\.\Ezssp-0',500.0) as dds:
        # Set the dds frequencies. I am making Raman A be the negative one.
        # Also, I am making them be symmetric around 80, because I can.
        dds.SelectChannel(2)
        #dds.SetFrequency(80.0-f/2.0)
        dds.SetFrequency(80.0-11/2.0)
        dds.SelectChannel(3)
        #dds.SetFrequency(80.0+f/2.0)
        dds.SetFrequency(80.0-11/2.0-cw_cool_freq/2)
```

```
       # FOR beams for pi pulses

#       for (i,f) in enumerate(frequencies):
#           # Set the dds frequencies. I am making Raman A be the negative one.
#           # Also, I am making them be symmetric around 80, because I can.
#           dds.SelectChannel(0)
#           #dds.SetFrequency(80.0-f/2.0)
#           dds.SetFrequency(80.0-15/2.0)
#           dds.SelectChannel(1)
#           #dds.SetFrequency(80.0+f/2.0)
#           dds.SetFrequency(80.0-15/2.0+f)
#           data[i,0] = f

       # For a scan, this will be incorporated into a for loop
       # FOR_Sigma pulse length
#       FOR_sigma = OneShot(’/Dev5/Ctr7’,trigchan5,True,stirap_a_b-safe,
   protect_b+safe)
       # FOR_Pi pulse length
#       FOR_pi = OneShot(’/Dev1/Ctr0’,trigchan1,True,stirap_b_b-safe,protect_b+
   safe)

       #Manually set the default states of the shutters.
       try:
           del clist
       except NameError:
           print
       clist = {’b_shutter’: [’/Dev5/Ctr2’,trigchan5,False],
               ’r_shutter’: [’/Dev5/Ctr3’,trigchan5,False],
               ’shelve_shutter’: [’/Dev1/Ctr3’,trigchan1,False],
               ’raman_shutter’: [’/Dev1/Ctr2’,trigchan1,True], #False due to
                   using OR + NOT
               ’raman_shutter_protect’: [’/Dev5/Ctr0’,trigchan5,True], #False
                   due to using OR + NOT
               ’FOR_shutter’: [’/Dev6/Ctr0’,trigchan6,False],
               ’deshelve_shutter’: [’/Dev6/Ctr1’,trigchan6,False]}

       ctimes = {’b_shutter’: [0.0, 2*shutter_t],
                ’r_shutter’: [0.0, 2*shutter_t],
                ’shelve_shutter’: [0.0, 2*shutter_t],
                ’raman_shutter’: [0.0, 2*shutter_t],
                ’raman_shutter_protect’: [0.0, 2*shutter_t],
                'FOR_shutter': [0.0, 2*shutter_t],
                'deshelve_shutter': [0.0,2*shutter_t]}
       for k in ctimes.keys():
           oneshots[k] = OneShot(*(clist[k]+ctimes[k]))
       for k in oneshots.keys():
           oneshots[k].begin()
       trig.setHigh()
       time.sleep(2*shutter_t)
       trig.setLow()
       time.sleep(recover_t)
       for k in oneshots.keys():
           oneshots[k].stop()
       del oneshots
       del clist
       oneshots = {}
       # b is begin, s is stop

       data = np.zeros((for_frequencies_temp.shape[0],reps+1))
       for (i,for_freq) in enumerate(for_frequencies_temp):

           # And now all the oneshots
           oneshots = {}
           # First initialize all the oneshots


           if np.logical_and(for_freq <= for_carrier+0.5, for_freq >=
              for_carrier-0.5):
              for_pi_pulse_t = for_carrier_pi_pulse_t
           else:
              for_pi_pulse_t = for_bsb_pi_pulse_t

           # b is begin, s is stop
           # Raman Cooling
           ramancool_NR_sigma_b = safe
           ramancool_NR_pi_b = safe
           ramancool_NR_sigma_s = ramancool_NR_sigma_b + cw_cool_t + op_t #
              optical pumping
           ramancool_NR_pi_s = ramancool_NR_sigma_b + cw_cool_t
           # Pi Pulse, the stop time is defined down below, since it changes
              for a scan
```

```
pi_pulse_b = ramancool_NR_sigma_s+2*safe +heating_t# The NR lights
    must be off
pi_pulse_s = pi_pulse_b + for_pi_pulse_t
# Protect
protect_b = pi_pulse_s + safe # Optical pumping shouldnt be on
    during shelving
protect_s = protect_b + protect_t
# Shelve
shelve_b = protect_s + safe
shelve_s = shelve_b + shelve_t
# Readout, the red and blue should be open at expose_s
expose_b = shelve_s + 2*safe # Doppler cooling sholdn’t start until
    the shelving is off
expose_s = expose_b + expose_t
# Deshelve
deshelve_b = expose_s +safe
deshelve_s = deshelve_b + deshelve_t

clist = {’b_doppler’: [’/Dev5/Ctr6’,trigchan5,True],
        ’r_doppler’: [’/Dev1/Ctr4’,trigchan1,False],
        ’NR_sigma’: [’/Dev5/Ctr4’,trigchan5,False],
        ’NR_pi’ : [’/Dev5/Ctr5’,trigchan5,False],
        ’b_shutter’: [’/Dev5/Ctr2’,trigchan5,False],
        ’r_shutter’: [’/Dev5/Ctr3’,trigchan5,False],
        ’FOR_sigma’ : [’/Dev5/Ctr7’,trigchan5,False],
        ’FOR_pi’ : [’/Dev1/Ctr0’,trigchan1,False],
        ’NR_sigma_protect’: [’/Dev1/Ctr1’,trigchan1,False],
        ’shelve’: [’/Dev1/Ctr5’,trigchan1,False],
        ’deshelve’: [’/Dev1/Ctr6’,trigchan1,False],
        ’camera’: [’/Dev5/Ctr1’,trigchan5,False],
        ’shelve_shutter’: [’/Dev1/Ctr3’,trigchan1,True],
        ’raman_shutter’: [’/Dev1/Ctr2’,trigchan1,False], #False due
            to using OR + NOT
        ’raman_shutter_protect’: [’/Dev5/Ctr0’,trigchan5,False], #
            False due to using OR + NOT
        ’FOR_shutter’ : [’/Dev6/Ctr0’,trigchan6,True],
        ’deshelve_shutter’: [’/Dev6/Ctr1’,trigchan6,True]}

ctimes = {’b_doppler’: [0.0, shelve_s+safe],
        ’r_doppler’: [ramancool_NR_sigma_s+safe, shelve_s+safe],
        ’NR_sigma’: [ramancool_NR_sigma_b,ramancool_NR_sigma_s],
                ’NR_pi’ : [ramancool_NR_pi_b,ramancool_NR_pi_s],
                #’b_shutter’: [0.0, shelve_s],
                #’r_shutter’: [ramancool_NR_sigma_s+shutter_t, shelve_s+
                    shutter_t],
                ’FOR_sigma’ : [pi_pulse_b,pi_pulse_s],
                ’FOR_pi’ : [pi_pulse_b,pi_pulse_s],
                ’NR_sigma_protect’: [protect_b,protect_s],
                ’shelve’: [shelve_b, shelve_s],
                ’deshelve’: [deshelve_b, deshelve_s],
                ’camera’: [expose_b, expose_s],
                #’shelve_shutter’: [shelve_b, shelve_s],
                #’raman_shutter’: [0.0, ramancool_NR_sigma_s],
                #’raman_shutter_protect’: [pi_pulse_s, protect_s],
                #’FOR_shutter’: [ramancool_NR_sigma_s+shutter_t,pi_pulse_s],
                #’deshelve_shutter’: [deshelve_b,deshelve_s]
                }

        for k in ctimes.keys():
            oneshots[k] = OneShot(*(clist[k]+ctimes[k]))

        # Then start all the oneshots
        for k in oneshots.keys():
            oneshots[k].begin()
        if np.logical_and(plot_me == True, i == 0):
            hold = plotOneShots(ctimes)

#        ’raman_shutter’: [shutter_t,stirap_protect_b],
#        ’stirap_trig’: [stirap_a_b+safe,stirap_a_b+2*safe],
#        #’shelve’: [stirap_protect_b+shutter_t*2,stirap_protect_b+
   shutter_t*2+shelve_t],
#        #’shelve_shutter’: [stirap_protect_b+shutter_t,stirap_protect_b+
   shutter_t*2+shelve_t],
#        ’shelve’: [stirap_protect_b+shutter_t,stirap_protect_b+shutter_t+
   shelve_t],
#        ’shelve_shutter’: [stirap_protect_b,stirap_protect_b+shutter_t+
   shelve_t],
#        ’deshelve’: [deshelve_b,deshelve_b+deshelve_t],
#        ’camera’: [cooling_s+shutter_t,cooling_s+shutter_t+expose_t]
```

```
        # Set the dds frequencies. I am making Raman A be the negative one.
        # Also, I am making them be symmetric around 80, because I can.
        dds.SelectChannel(0)
        #dds.SetFrequency(80.0-f/2.0)
        dds.SetFrequency(80.0-11/2.0)
        dds.SelectChannel(1)
        #dds.SetFrequency(80.0+f/2.0)
        dds.SetFrequency(80.0-11/2.0-for_freq/2)
        data[i,0] = for_freq
        print(’% of Scan Completed:’)
        print((i+1.0)/for_frequencies_temp.shape[0]*100.0)
        for j in range(reps):
            AO_detect_control.start()
            pmt.startRead()
            trig.setHigh()
            time.sleep(deshelve_b+deshelve_t+line_t)
            res = pmt.getRead()
            trig.setLow()
            data[i,j+1] = res
            time.sleep(recover_t) # Make sure that the cycle has finished
            AO_detect_control.stop()

        #clear the oneshots to start again!
        for k in oneshots.keys():
            oneshots[k].stop()
        del oneshots

# Build a suitable filename
t = time.localtime()
filename = "{}{:02g}{:02g}".format(t.tm_year,t.tm_mon,t.tm_mday)+ "
    _CWRamancool_FOR_Temperature_NS_scanPMT"
i = 0
while os.path.exists(os.path.normpath(path + filename + "_{}.npy".format(i)
    )):
    i = i + 1
filenamen = os.path.normpath(path + filename + "_{}.npy".format(i))

# And save the data
np.save(filenamen,data)
```

```
del clist
del ctimes
oneshots = {}

clist = {'b_shutter': ['/Dev5/Ctr2',trigchan5,False],
            'r_shutter': ['/Dev5/Ctr3',trigchan5,False],
            'shelve_shutter': ['/Dev1/Ctr3',trigchan1,True],
            'raman_shutter': ['/Dev1/Ctr2',trigchan1,False], #False due to
               using OR + NOT
            'raman_shutter_protect': ['/Dev5/Ctr0',trigchan5,False], #False
                due to using OR + NOT
            'FOR_shutter': ['/Dev6/Ctr0',trigchan6,True],
            'deshelve_shutter': ['/Dev6/Ctr1',trigchan6,True]}

ctimes = {'b_shutter': [0.0, 2*shutter_t],
         'r_shutter': [0.0, 2*shutter_t],
         'shelve_shutter': [0.0, 2*shutter_t],
         'raman_shutter': [0.0, 2*shutter_t],
         'raman_shutter_protect': [0.0, 2*shutter_t],
         'FOR_shutter': [0.0, 2*shutter_t],
         'deshelve_shutter': [0.0,2*shutter_t]}
for k in ctimes.keys():
    oneshots[k] = OneShot(*(clist[k]+ctimes[k]))
for k in oneshots.keys():
    oneshots[k].begin()
trig.setHigh()
time.sleep(2*shutter_t)
trig.setLow()
time.sleep(recover_t)
for k in oneshots.keys():
    oneshots[k].stop()
del oneshots
del clist

trig.stop()
del trig
pmt.stop()
del pmt
```

E.3.6.9. FOR Time Scan. The code below scans the FOR Raman exposure time to plot and fit Rabi oscillations.

```
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 16 15:09:02 2015

The main code for executing a pi-pulse time scan at a single frequency with FOR
    .

@author: CMS
@secondauthor: MGK
"""
# The path to save the data in
path = 'c:/data/Rabi/Time/'

# Start PMT Task
pmt = PMT(pmtc)

if __name__ == '__main__':
    # Setup trigger out
    trig = TrigControl(trigsource)

    with DDSControl(r'\\.\Ezssp-0',500.0) as dds:
        # Set the dds frequencies. Ch0,Ch1 are FOR sigma,pi. Ch2,Ch3 are NR.
        #The offset is to center the AOM's at resonance.
        dds.SelectChannel(0)
        dds.SetFrequency(80.0-11/2.0)
        dds.SelectChannel(1)
        dds.SetFrequency(80.0-11/2.0-for_freq/2)
        dds.SelectChannel(2)
        dds.SetFrequency(80.0-11/2.0)
        dds.SelectChannel(3)
        dds.SetFrequency(80.0-11/2.0-cw_cool_freq/2)

    try:
        del clist
    except NameError:
        print
    oneshots = {}
```

```
clist = {'b_shutter': ['/Dev5/Ctr2',trigchan5,False],
        'r_shutter': ['/Dev5/Ctr3',trigchan5,False],
        'shelve_shutter': ['/Dev1/Ctr3',trigchan1,False],
        'raman_shutter': ['/Dev1/Ctr2',trigchan1,True], #False due to using
            OR + NOT
        'raman_shutter_protect': ['/Dev5/Ctr0',trigchan5,True], #False due
           to using OR + NOT
        'FOR_shutter': ['/Dev6/Ctr0',trigchan6,False],
        'deshelve_shutter': ['/Dev6/Ctr1',trigchan6,False]}

ctimes = {'b_shutter': [0.0, 2*shutter_t],
         'r_shutter': [0.0, 2*shutter_t],
         'shelve_shutter': [0.0, 2*shutter_t],
         'raman_shutter': [0.0, 2*shutter_t],
         'raman_shutter_protect': [0.0, 2*shutter_t],
         'FOR_shutter': [0.0, 2*shutter_t],
         'deshelve_shutter': [0.0,2*shutter_t]}
for k in ctimes.keys():
    oneshots[k] = OneShot(*(clist[k]+ctimes[k]))
for k in oneshots.keys():
    oneshots[k].begin()
trig.setHigh()
time.sleep(2*shutter_t)
trig.setLow()
time.sleep(recover_t)
for k in oneshots.keys():
    oneshots[k].stop()
del oneshots
del clist

time.sleep(1)
data = np.zeros((for_pulse_times.shape[0],reps+1))
for (i,for_pulse_t) in enumerate(for_pulse_times):
    # And now all the oneshots
    oneshots = {}

    # Raman Cooling
    ramancool_NR_sigma_b = safe
    ramancool_NR_pi_b = safe
    ramancool_NR_sigma_s = ramancool_NR_sigma_b + cw_cool_t + op_t # optical
        pumping
```

```
ramancool_NR_pi_s = ramancool_NR_sigma_b + cw_cool_t
# Pi Pulse, the stop time is defined down below, since it changes for a
    scan
pi_pulse_b = ramancool_NR_sigma_s+2*safe +heating_t# The NR lights must
    be off
pi_pulse_s = pi_pulse_b + for_pulse_t
# Protect
protect_b = pi_pulse_s + safe # Optical pumping shouldnt be on during
    shelving
protect_s = protect_b + protect_t
# Shelve
shelve_b = protect_s + safe
shelve_s = shelve_b + shelve_t
# Readout, the red and blue should be open at expose_s
expose_b = shelve_s + 2*safe # Doppler cooling sholdn’t start until the
    shelving is off
expose_s = expose_b + expose_t
# Deshelve
deshelve_b = expose_s +safe
deshelve_s = deshelve_b + deshelve_t

clist = {’b_doppler’: [’/Dev5/Ctr6’,trigchan5,True],
        ’r_doppler’: [’/Dev1/Ctr4’,trigchan1,False],
        ’NR_sigma’: [’/Dev5/Ctr4’,trigchan5,False],
        ’NR_pi’ : [’/Dev5/Ctr5’,trigchan5,False],
        ’b_shutter’: [’/Dev5/Ctr2’,trigchan5,False],
        ’r_shutter’: [’/Dev5/Ctr3’,trigchan5,False],
        ’FOR_sigma’ : [’/Dev5/Ctr7’,trigchan5,False],
        ’FOR_pi’ : [’/Dev1/Ctr0’,trigchan1,False],
        ’NR_sigma_protect’: [’/Dev1/Ctr1’,trigchan1,False],
        ’shelve’: [’/Dev1/Ctr5’,trigchan1,False],
        ’deshelve’: [’/Dev1/Ctr6’,trigchan1,False],
        ’camera’: [’/Dev5/Ctr1’,trigchan5,False],
        ’shelve_shutter’: [’/Dev1/Ctr3’,trigchan1,True],
        ’raman_shutter’: [’/Dev1/Ctr2’,trigchan1,False], #False due to
            using OR + NOT
        ’raman_shutter_protect’: [’/Dev5/Ctr0’,trigchan5,False], #False
             due to using OR + NOT
        ’FOR_shutter’ : [’/Dev6/Ctr0’,trigchan6,True],
        ’deshelve_shutter’: [’/Dev6/Ctr1’,trigchan6,True]}
```

```
ctimes = {’b_doppler’: [0.0, shelve_s+safe],
        ’r_doppler’: [ramancool_NR_sigma_s+safe, shelve_s+safe],
        ’NR_sigma’: [ramancool_NR_sigma_b,ramancool_NR_sigma_s],
        ’NR_pi’ : [ramancool_NR_pi_b,ramancool_NR_pi_s],
        #’b_shutter’: [0.0, shelve_s],
        #’r_shutter’: [ramancool_NR_sigma_s+shutter_t, shelve_s+
            shutter_t],
        ’FOR_sigma’ : [pi_pulse_b,pi_pulse_s],
        ’FOR_pi’ : [pi_pulse_b,pi_pulse_s],
        ’NR_sigma_protect’: [protect_b,protect_s],
        ’shelve’: [shelve_b, shelve_s],
        ’deshelve’: [deshelve_b, deshelve_s],
        ’camera’: [expose_b, expose_s],
        #’shelve_shutter’: [shelve_b, shelve_s],
        #’raman_shutter’: [0.0, ramancool_NR_sigma_s],
        #’raman_shutter_protect’: [pi_pulse_s, protect_s],
        #’FOR_shutter’: [ramancool_NR_sigma_s+shutter_t,pi_pulse_s],
        #’deshelve_shutter’: [deshelve_b,deshelve_s]
        }

for k in ctimes.keys():
    oneshots[k] = OneShot(*(clist[k]+ctimes[k]))

# Then start all the oneshots
for k in oneshots.keys():
    oneshots[k].begin()
if np.logical_and(plot_me == True, i == 0):
    hold = plotOneShots(ctimes)

data[i,0] = for_pulse_t
print(’% of Scan Completed:’)
print((i+1.0)/for_pulse_times.shape[0]*100.0)
for j in range(reps):
    AO_detect_control.start()
    pmt.startRead()
    trig.setHigh()
    time.sleep(deshelve_s+shutter_t+line_t)
    res = pmt.getRead()
    trig.setLow()
    data[i,j+1] = res
    time.sleep(recover_t) # Make sure that the cycle has finished
```

```
        AO_detect_control.stop()

    #clear the oneshots to start again!
    for k in oneshots.keys():
        oneshots[k].stop()
    del oneshots

# Build a suitable filename
t = time.localtime()
filename = "{}{:02g}{:02g}".format(t.tm_year,t.tm_mon,t.tm_mday)+ "
    _CWRamancool_FOR_Time_NS_scanPMT"
i = 0
while os.path.exists(os.path.normpath(path + filename + "_{}.npy".format(i)
    )):
    i = i + 1
filenamen = os.path.normpath(path + filename + "_{}.npy".format(i))

# And save the data
np.save(filenamen,data)

del clist
del ctimes
oneshots = {}

clist = {’b_shutter’: [’/Dev5/Ctr2’,trigchan5,False],
            ’r_shutter’: [’/Dev5/Ctr3’,trigchan5,False],
            ’shelve_shutter’: [’/Dev1/Ctr3’,trigchan1,True],
            ’raman_shutter’: [’/Dev1/Ctr2’,trigchan1,False], #False due to
                using OR + NOT
            ’raman_shutter_protect’: [’/Dev5/Ctr0’,trigchan5,False], #False
                 due to using OR + NOT
            ’FOR_shutter’: [’/Dev6/Ctr0’,trigchan6,True],
            ’deshelve_shutter’: [’/Dev6/Ctr1’,trigchan6,True]}

ctimes = {’b_shutter’: [0.0, 2*shutter_t],
         ’r_shutter’: [0.0, 2*shutter_t],
         ’shelve_shutter’: [0.0, 2*shutter_t],
         ’raman_shutter’: [0.0, 2*shutter_t],
         ’raman_shutter_protect’: [0.0, 2*shutter_t],
         ’FOR_shutter’: [0.0, 2*shutter_t],
         ’deshelve_shutter’: [0.0,2*shutter_t]}
```

```
for k in ctimes.keys():
    oneshots[k] = OneShot(*(clist[k]+ctimes[k]))
for k in oneshots.keys():
    oneshots[k].begin()
trig.setHigh()
time.sleep(2*shutter_t)
trig.setLow()
time.sleep(recover_t)
for k in oneshots.keys():
    oneshots[k].stop()
del oneshots
del clist

#Clear the trigger and PMT tasks.
trig.stop()
del trig
pmt.stop()
del pmt
```

E.3.7. Magnetic Field Trimming

The code below scans the trimming magnetic fields (x- and z-axis) while recording background-subtracted fluorescence from the σ^+-polarized NR Raman pump beam. The background points are collected by shutting off the 650 nm Doppler repump beam, so the ion is dark to the 493 nm light. Data analysis is performed in the iPython notebook in App. E.1.6. Created by MRD and modified by CMS.

```
# -*- coding: utf-8 -*-
"""
Created on Wed Mar 18 18:16:42 2015

A script intended to scan X and Z coil currents, and measure the fluorescence
with sigma light on in order to optimize the trimming.

This program talks with the camera (EMCCD5.vi as of right now) to collect data.
Thus, it will wait 20 seconds for you to start this program in External Trigger
mode, and with the "Send Data on Network" button depressed. When done, you
should make sure that all python variables are cleared (by stopping the
interpreter or using %reset), and then use the "Force Shutdown" button on
the camera program.

Also, the red control TTL has to be moved to a digital out. It is probably
on a counter most of the time. The correct new location is /dev5/P0.4
(usually Dev1/Ctr4).

@author: lynx
"""

from OneShot import OneShot
from TrigControl import TrigControl, DOControl
import time
import numpy as np
import os
import pdb
from AIWrite import AOWrite
from plotoneshots import plotOneShots
from pmt import PMT


# Currents to scan
Xcoilvals = np.linspace(0,6,7)
Zcoilvals = np.linspace(0,0.6,7)
# How much data to take at each setting
reps = 10

# A 'safe interval', which is a period added in to various places to ensure
# the light has actually turned off, or on. Also used as the duration of a
# trig pulse.
safe_t = 1e-4
shutter_t = 10e-3
op_t = 1e-3
expose_t = 500e-3
doppler_t = 100e-3
recover_t = 100e-3

# The path to save the data in
path = 'c:/data/BTrim/'

# If this is true, then plot the ctimes sequence
plot_me = True

Xcoilchan = '/Dev3/ao8'
Ycoilchan = '/Dev3/ao12'
Zcoilchan = '/Dev3/ao10'
trigchan1 = '/dev1/PFI14'
trigchan5 = '/dev5/PFI3'
trigsource = '/dev5/port0/line0'

#Explicit control of the red light
#This is Dev5/P0.3
r_doppler_c = '/dev5/port0/line4'

#PMT info
pmtc = '/dev1/ctr7'
pmt = PMT(pmtc)

# A dictionary of all the relevant channels used below.
# Doppler blue and red should be set to be ordinarily on, all others off
clist = {'b_doppler': ['/Dev5/Ctr6',trigchan5,True],
        'r_doppler': ['/Dev1/Ctr4',trigchan1,False],
        'raman_a': ['/Dev5/Ctr4',trigchan5,False],
        'raman_b': ['/Dev5/Ctr5',trigchan5,False],
        'camera': ['/Dev5/Ctr1',trigchan5,False],
        'b_shutter': ['/Dev5/Ctr2',trigchan5,False],
        'r_shutter': ['/Dev5/Ctr3',trigchan5,False],
        'raman_shutter': ['/Dev1/Ctr2',trigchan1,True]}

doppler_b = doppler_t
raman_shutter_b = doppler_b + shutter_t
raman_a_b = raman_shutter_b + shutter_t
camera_e = raman_a_b + op_t + safe_t
raman_a_e = camera_e + expose_t + safe_t
doppler_e = raman_a_e + shutter_t

ctimes = {'b_doppler': [doppler_b,doppler_e],
         'b_shutter': [doppler_b,doppler_e],
         'raman_shutter': [raman_shutter_b,raman_a_e],
         'raman_a': [raman_a_b,raman_a_e],
         'camera': [camera_e,camera_e + expose_t]}

if __name__ == '__main__':
    # Setup trigger out
    trig = TrigControl(trigsource)

    # And control of the red laser
    red = DOControl(r_doppler_c)


    # And the Analog out channels
#    Xcoil = AOWrite(Xcoilchan,0)
#    ZCoil = AOWrite(Zcoilchan,0)




    Xcoil = AOWrite(Xcoilchan)
    ZCoil = AOWrite(Zcoilchan)
```

```
# And now all the oneshots
oneshots = {}
# First initialize all the oneshots
for k in ctimes.keys():
    oneshots[k] = OneShot(*(clist[k]+ctimes[k]))

# Then start all the oneshots
for k in oneshots.keys():
    try:
        oneshots[k].begin()
    except:
        pdb.post_mortem()
if plot_me:
    hold = plotOneShots(ctimes)

time.sleep(5) # Give the user some time to start the camera program

data = np.zeros((Xcoilvals.shape[0],Zcoilvals.shape[0]))
err = np.zeros(data.shape)
tmp = np.zeros(reps)
res_sig = np.zeros(reps)
res_bg = np.zeros(reps)
for a,xv in enumerate(Xcoilvals):
    Xcoil.setVoltage(xv)
    time.sleep(5)
    for b,zv in enumerate(Zcoilvals):
        ZCoil.setVoltage(zv)
        time.sleep(5)
        for i in range(reps):
            pmt.startRead()
            red.setHigh()
            trig.setHigh()
            time.sleep(doppler_e)
            res_bg[i] = pmt.getRead()
            trig.setLow()
            time.sleep(recover_t)

            pmt.startRead()
            red.setLow()
            trig.setHigh()
            time.sleep(doppler_e)
            res_sig[i] = pmt.getRead()
            trig.setLow()
            time.sleep(recover_t)

            #Data processing here to exclude missed camera exposures.
        n1 = 0.0
        for j in range(res_sig.shape[0]):
            if res_sig[j] == -1:
                res_sig[j] = np.NaN
            else:
                n1 = n1 + 1
        n2 = 0.0
        for j in range(res_bg.shape[0]):
            if res_bg[j] == -1:
                res_bg[j] = np.NaN
            else:
                n2 = n2 + 1
        tmp = res_sig - res_bg
        data[a,b] = np.nanmean(tmp)
        err[a,b] = np.nanstd(tmp)/np.sqrt(np.min([n1,n2]))
        print "Fluorescence at {:.2f}, {:.2f}: {} pm {}".format(xv, zv, data
            [a,b], err[a,b])

# Build a suitable filename
t = time.localtime()
filename = "{}{:02g}{:02g}".format(t.tm_year,t.tm_mon,t.tm_mday)+ "_BTrim"
i = 0
while os.path.exists(os.path.normpath(path + filename + "_{}.npy".format(i)
    )):
    i = i + 1
filenamen = os.path.normpath(path + filename + "_{}.npy".format(i))
errfilename = os.path.normpath(path + filename + "_{}_err.npy".format(i))

# And save the data
np.save(filenamen,data)
np.save(errfilename,err)
```

E.3.8. Shelving Test

The code below tests the 455 nm exposure time for maximum shelving efficiency.

```
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 16 15:09:02 2015

The main code for executing a frequency scan with STIRAP.

@author: matt
"""

from OneShot import OneShot
from TrigControl import TrigControl
import time
import numpy as np
import os
import pdb
from plotoneshots import plotOneShots
from pmt import PMT
from AIWrite import AOWrite

# How much data to take at each setting
reps = 200

# The path to save the data in
path = 'c:/data/Shelving/'

# If this is true, then plot the ctimes sequence
plot_me = False

trigchan1 = '/dev1/PFI14'
trigchan5 = '/dev5/PFI3'
trigchan6 = '/dev6/PFI14'
trigsource = '/dev5/port0/line0'

# Expose time of camera
expose_t = 5.0e-3
```

```
# Change the Doppler cooling settings for ion detection after shelving. This
# sets up the AO task. It has to be defined as a single task to share the
# on-board clock. AO1 = b_FM, AO4 = b_AM, AO5 = r_FM.
trigchan2_camera = '/dev2/PFI7'
b_detect_FM_v = 0.25
b_cool_FM_v = 0.10
b_FM_voltages = np.array([b_detect_FM_v,b_cool_FM_v],dtype=np.float64)
b_detect_AM_v = -1.5
b_cool_AM_v = -0.15
b_AM_voltages = np.array([b_detect_AM_v,b_cool_AM_v],dtype=np.float64)
r_detect_AM_v = -0.10
r_cool_AM_v = -0.02
r_AM_voltages = np.array([r_detect_AM_v,r_cool_AM_v],dtype=np.float64)
AO_detect_voltages = np.concatenate((b_FM_voltages,b_AM_voltages,r_AM_voltages)
    )
AO_detect_control = AOWrite("/dev2/ao1,/dev2/ao4:5")
AO_detect_control.setTrig(1/expose_t,2,trigchan2_camera)
AO_detect_control.voltSequence(3,AO_detect_voltages)

#PMT info
pmtc = '/dev1/ctr7'
pmt = PMT(pmtc)

# A dictionary of all the relevant channels used below.
# Doppler blue and red should be set to be ordinarily on, all others off
clist = {'b_doppler': ['/Dev5/Ctr6',trigchan5,True],
        'r_doppler': ['/Dev1/Ctr4',trigchan1,False],
        'shelve': ['/Dev1/Ctr5',trigchan1,False],
        'deshelve': ['/Dev1/Ctr6',trigchan1,False],
        'camera': ['/Dev5/Ctr1',trigchan5,False],
        'b_shutter': ['/Dev5/Ctr2',trigchan5,False],
        'r_shutter': ['/Dev5/Ctr3',trigchan5,False],
        'shelve_shutter': ['/Dev1/Ctr3',trigchan1,True],
        'deshelve_shutter': ['/Dev6/Ctr1',trigchan6,True]}
#        'raman_shutter': ['/Dev1/Ctr2',trigchan1,False]}

# A 'safe interval', which is a period added in to various places to ensure
# the light has actually turned off, or on. Also used as the duration of a
# trig pulse.
safe = 1e-5
# Time to allow for the mechanical shutter to close
```

```
shutter_t = 10e-3

# Time to expose the Shelving laser for
shelve_t = 10e-3

# Deshelving time
deshelve_t = 100e-3
# Recovery time between cycles to allow ion to cool
recover_t = 20.0e-3

shutter_open_b = shelve_t+shutter_t*2
shutter_open_r = shutter_open_b-shutter_t*2
camera_b = shutter_open_b+shutter_t*2
deshelve_b = camera_b+3*safe+expose_t

ctimes = {'b_doppler': [0.0,camera_b],
         'b_shutter': [0.0,shutter_open_b],
         'r_doppler': [shutter_t,camera_b],
         'r_shutter': [shutter_t,shutter_open_b],
         'shelve': [2*shutter_t,shutter_open_b],
         'shelve_shutter': [shutter_t,shutter_open_b],
         'deshelve': [deshelve_b+shutter_t,deshelve_b+deshelve_t],
         'deshelve_shutter': [deshelve_b,deshelve_b+deshelve_t],
         'camera': [camera_b+safe,camera_b+safe+expose_t]}


if __name__ == '__main__':
    # Setup trigger out
    trig = TrigControl(trigsource)

    # And now all the oneshots
    oneshots = {}
    # First initialize all the oneshots
    for k in ctimes.keys():
        oneshots[k] = OneShot(*(clist[k]+ctimes[k]))

    # Then start all the oneshots
    for k in oneshots.keys():
        try:
            oneshots[k].begin()
        except:
```

```
        pdb.post_mortem()
if plot_me:
    tmp = plotOneShots(ctimes)

time.sleep(1)
# The first column is the frequencies, all others are the data collected
# for that frequency
data = np.zeros((reps,))
for j in range(reps):
    AO_detect_control.start()
    pmt.startRead()
    trig.setHigh()
    time.sleep(deshelve_b+deshelve_t)
    res = pmt.getRead()
    trig.setLow()
    data[j] = res
    time.sleep(recover_t) # Make sure that the cycle has finished
    AO_detect_control.stop()
    print(’% of Scan Completed:’)
    print((j+1.0)/reps*100.0)



 # Build a suitable filename
t = time.localtime()
filename = "{}{:02g}{:02g}".format(t.tm_year,t.tm_mon,t.tm_mday)+ "
    _shelvingPMT"
i = 0
while os.path.exists(os.path.normpath(path + filename + "_{}.npy".format(i)
    )):
    i = i + 1
filenamen = os.path.normpath(path + filename + "_{}.npy".format(i))

# And save the data
np.save(filenamen,data)

for k in oneshots.keys():
    oneshots[k].stop()
del oneshots
trig.stop()
del trig
```

```
pmt.stop()
del pmt
AO_detect_control.clear()
del AO_detect_control
print('Im Done')
```

E.3.9. STIRAP Frequency Scan

The code below executes a frequency scan with STIRAP after performing NR RSC.

```
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 16 15:09:02 2015

The main code for executing a frequency scan with STIRAP.

@author: matt
@secondauthor: CMS
"""

from OneShot import OneShot
from DDSControl import DDSControl
from TrigControl import TrigControl
import time
import numpy as np
import os
#import pdb
from plotoneshots import plotOneShots
from pmt import PMT

# The Zeeman splitting should be about 12.3 MHz at 1 A on the horizontal coils
# Divided by 2 due to the double pass.
# But I don’t know if the right frequency is positive or negative
cool_freq = 6.4
#frequencies = np.linspace(4,8,5)
#frequencies = np.linspace(4.25,7,110)
#frequencies = np.linspace(5.25-2,5.25+2,80)
frequencies = np.linspace(4.5,6.5,80)
#frequencies = np.concatenate((frequencies,-frequencies))
#frequencies = np.linspace(1,-1,10)
#frequencies = 13
#frequencies = np.zeros(1)
#frequencies = np.random.permutation(frequencies)
# How much data to take at each setting
reps = 500
```

```
# The path to save the data in
path = 'c:/data/STIRAP/'

# If this is true, then plot the ctimes sequence
plot_me = True

#PMT info
pmtc = '/dev1/ctr7'
pmt = PMT(pmtc)

trigchan1 = '/dev1/PFI14'
trigchan5 = '/dev5/PFI3'
trigsource = '/dev5/port0/line0'
r_doppler_c = '/dev5/port0/line4'

# A dictionary of all the relevant channels used below.
# Doppler blue and red should be set to be ordinarily on, all others off
clist = {'b_doppler': ['/Dev5/Ctr6',trigchan5,True],
        'r_doppler': ['/Dev1/Ctr4',trigchan1,False],
        'raman_a': ['/Dev5/Ctr4',trigchan5,False],
        'raman_b': ['/Dev5/Ctr5',trigchan5,False],
        'stirap_a_on': ['/Dev5/Ctr7',trigchan5,True],
        'stirap_b_on': ['/Dev1/Ctr0',trigchan1,True],
        'stirap_trig': ['/Dev5/Ctr0',trigchan5,False],
        'shelve': ['/Dev1/Ctr5',trigchan1,False],
        'deshelve': ['/Dev1/Ctr6',trigchan1,False],
        'camera': ['/Dev5/Ctr1',trigchan5,False],
        'b_shutter': ['/Dev5/Ctr2',trigchan5,False],
        'r_shutter': ['/Dev5/Ctr3',trigchan5,False],
        'shelve_shutter': ['/Dev1/Ctr3',trigchan1,True],
        'raman_shutter': ['/Dev1/Ctr2',trigchan1,True]}

# A 'safe interval', which is a period added in to various places to ensure
# the light has actually turned off, or on. Also used as the duration of a
# trig pulse.
safe = 1e-5
# The amount of time to allow for the mechanical shutter to close or open
shutter_t = 10e-3
# The amount of time to optically pump for. Can safely be long.
op_t = 1.0e-3
#op_t = 0.0
```

```
# An interval set aside for the STIRAP pulses. Should be significantly longer
# than the STIRAP pulses
stirap_t = 4000.0e-6
# A period after STIRAP where Raman A is left on, with no red, to pump
# any transfered atoms to the D(3/2) state. This interval is potentially
# sensitive, since if its too short, the ion won’t be pumped, and if too long
# all atoms may be pumped by the imperfect B field trimming
protect_t = 1e-4
#stirap_protect_t = 5e-6
stirap_protect_t = 0.0
#stirap_protect_t = 0.0
# The amount of time to use 455 nm light to shelve any remaining ions into
# D(5/2)
shelve_t = 10e-3
# Expose time of camera
expose_t = 75e-3
# Deshelving time
deshelve_t = 150e-3
# Recovery time between cycles to allow ion to cool
recover_t = 100e-3
# Time to doppler cool. In practice, the ion cools for twice this time,
# because the ion is exposed at both the beginning and ending of a cycle
cool_t = 10e-3
# Time to expose the Raman cooling lasers for
raman_t = 10e-3
#raman_t = 0.0e-6

ramancool_b = 2*shutter_t
ramancool_s = ramancool_b + raman_t
stirap_b_b = ramancool_s
stirap_a_b = ramancool_s + op_t
protect_b = stirap_a_b + safe + stirap_t
stirap_protect_b = protect_b+safe+stirap_protect_t
cooling_b = stirap_protect_b + safe + protect_t + shelve_t +shutter_t*3
cooling_s = cooling_b+shutter_t
deshelve_b = cooling_s + shutter_t + expose_t

if stirap_protect_t == 0.0:
    stirap_protect_b = protect_b
    print(’ ’)
    print(’NO PROTECT STAGE!’)
    print(' ')

ctimes = {'b_doppler': [0.0,cooling_s],
          'b_shutter': [0.0,cooling_b],
          'r_doppler': [stirap_a_b,cooling_s],
          'r_shutter': [stirap_a_b,cooling_b],
          'raman_a': [ramancool_b,stirap_protect_b],
          'raman_shutter': [shutter_t,stirap_protect_b],
          'raman_b': [ramancool_b,protect_b],
          'stirap_a_on': [stirap_a_b-safe,protect_b+safe],
          'stirap_b_on': [stirap_b_b-safe,protect_b+safe],
          'stirap_trig': [stirap_a_b+safe,stirap_a_b+2*safe],
          #'shelve': [stirap_protect_b+shutter_t*2,stirap_protect_b+shutter_t*2+
              shelve_t],
          #'shelve_shutter': [stirap_protect_b+shutter_t,stirap_protect_b+
              shutter_t*2+shelve_t],
          'shelve': [stirap_protect_b+shutter_t,stirap_protect_b+shutter_t+
              shelve_t],
          'shelve_shutter': [stirap_protect_b,stirap_protect_b+shutter_t+
              shelve_t],
          'deshelve': [deshelve_b,deshelve_b+deshelve_t],
          'camera': [cooling_s+shutter_t,cooling_s+shutter_t+expose_t]}

if __name__ == '__main__':
    # Setup trigger out
    trig = TrigControl(trigsource)

    # And now all the oneshots
    oneshots = {}
    # First initialize all the oneshots
    for k in ctimes.keys():
        oneshots[k] = OneShot(*(clist[k]+ctimes[k]))

    # Then start all the oneshots
    for k in oneshots.keys():
        oneshots[k].begin()
    if plot_me:
        hold = plotOneShots(ctimes)

    time.sleep(5)
    # The first column is the frequencies, all others are the data collected
    # for that frequency
    data = np.zeros((frequencies.shape[0],reps+1))
    with DDSControl(r'\\.\Ezssp-0',500.0) as dds:
        # Set the dds frequencies. I am making Raman A be the negative one.
        # Also, I am making them be symmetric around 80, because I can.
        dds.SelectChannel(2)
        #dds.SetFrequency(80.0-f/2.0)
        dds.SetFrequency(80.0-15/2.0)
        dds.SelectChannel(3)
        #dds.SetFrequency(80.0+f/2.0)
        dds.SetFrequency(80.0-15/2.0+cool_freq)
        for (i,f) in enumerate(frequencies):
            # Set the dds frequencies. I am making Raman A be the negative one.
            # Also, I am making them be symmetric around 80, because I can.
            dds.SelectChannel(0)
            #dds.SetFrequency(80.0-f/2.0)
            dds.SetFrequency(80.0-15/2.0)
            dds.SelectChannel(1)
            #dds.SetFrequency(80.0+f/2.0)
            dds.SetFrequency(80.0-15/2.0+f)
            data[i,0] = f
            print('% of Scan Completed:')
            print((i+1.0)/frequencies.shape[0]*100.0)
            for j in range(reps):
                pmt.startRead()
                trig.setHigh()
                time.sleep(deshelve_b+deshelve_t)
                res = pmt.getRead()
                trig.setLow()
                data[i,j+1] = res
                time.sleep(recover_t) # Make sure that the cycle has finished

    # Build a suitable filename
    del oneshots['shelve']
    trig.setHigh()
    t = time.localtime()
    filename = "{}{:02g}{:02g}".format(t.tm_year,t.tm_mon,t.tm_mday)+ "
        _CWRamancool_STIRAPscanPMT"
    i = 0
    while os.path.exists(os.path.normpath(path + filename + "_{}.npy".format(i)
        )):
```

```
    i = i + 1
filenamen = os.path.normpath(path + filename + "_{}.npy".format(i))

# And save the data
np.save(filenamen,data)

trig.setLow()
```

E.3.10. Voltage Monitor

The code below monitors and records multiple AI channel voltages for stability monitoring.

```
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 16 15:09:02 2015

The main code for executing a frequency scan with STIRAP.

@author: matt
"""

import time
import numpy as np
import os
from AIRead import AIRead

# How much data to take at each setting
reps = 10

#How many analog input points to take to average per rep
AIAvg = 500

# Wait time between points.
wait_t = 10.0

# The path to save the data in
path = 'c:/data/Voltages Monitor/'

#Analog Input Channels
ai_flxgate = '/dev4/ai2'
ai_cpl_fwd = '/dev4/ai6'
ai_cpl_rev = '/dev4/ai7'
ai_endcap1 = '/dev4/ai3'
ai_endcap2 = '/dev4/ai4'

#Define AI Channels and set up triggers.
flxgate = AIRead(ai_flxgate)
cpl_fwd = AIRead(ai_cpl_fwd)
```

```
cpl_rev = AIRead(ai_cpl_rev)
endcap1 = AIRead(ai_endcap1)
endcap2 = AIRead(ai_endcap2)

data = np.zeros((1,6),dtype='complex')

# Build a suitable filename
t = time.localtime()
filename = "{}{:02g}{:02g}".format(t.tm_year,t.tm_mon,t.tm_mday)+ "
    _Voltages_Monitor"
i = 0
while os.path.exists(os.path.normpath(path + filename + "_{}.npy".format(i))):
    i = i + 1
filenamen = os.path.normpath(path + filename + "_{}.npy".format(i))

if __name__ == '__main__':
    j = 0
    while j >= 0:
        data = np.vstack((data,np.zeros((data.shape[1]))))
        temp_flxgate = np.array([flxgate.getVoltage() for x in range(AIAvg)])
        temp_cpl_fwd = np.array([cpl_fwd.getVoltage() for x in range(AIAvg)])
        temp_cpl_rev = np.array([cpl_rev.getVoltage() for x in range(AIAvg)])
        temp_endcap1 = np.array([endcap1.getVoltage() for x in range(AIAvg)])
        temp_endcap2 = np.array([endcap2.getVoltage() for x in range(AIAvg)])

        data[j,0] = time.mktime(time.localtime())
        data[j,1] = np.mean(temp_flxgate)+1j*np.std(temp_flxgate)/np.sqrt(
            temp_flxgate.size)
        data[j,2] = np.mean(temp_cpl_fwd)+1j*np.std(temp_cpl_fwd)/np.sqrt(
            temp_cpl_fwd.size)
        data[j,3] = np.mean(temp_cpl_rev)+1j*np.std(temp_cpl_rev)/np.sqrt(
            temp_cpl_rev.size)
        data[j,4] = np.mean(temp_endcap1)+1j*np.std(temp_endcap1)/np.sqrt(
            temp_endcap1.size)
        data[j,5] = np.mean(temp_endcap2)+1j*np.std(temp_endcap2)/np.sqrt(
            temp_endcap2.size)

        j = j + 1

        np.save(filenamen,data)
```

```
time.sleep(wait_t)
```

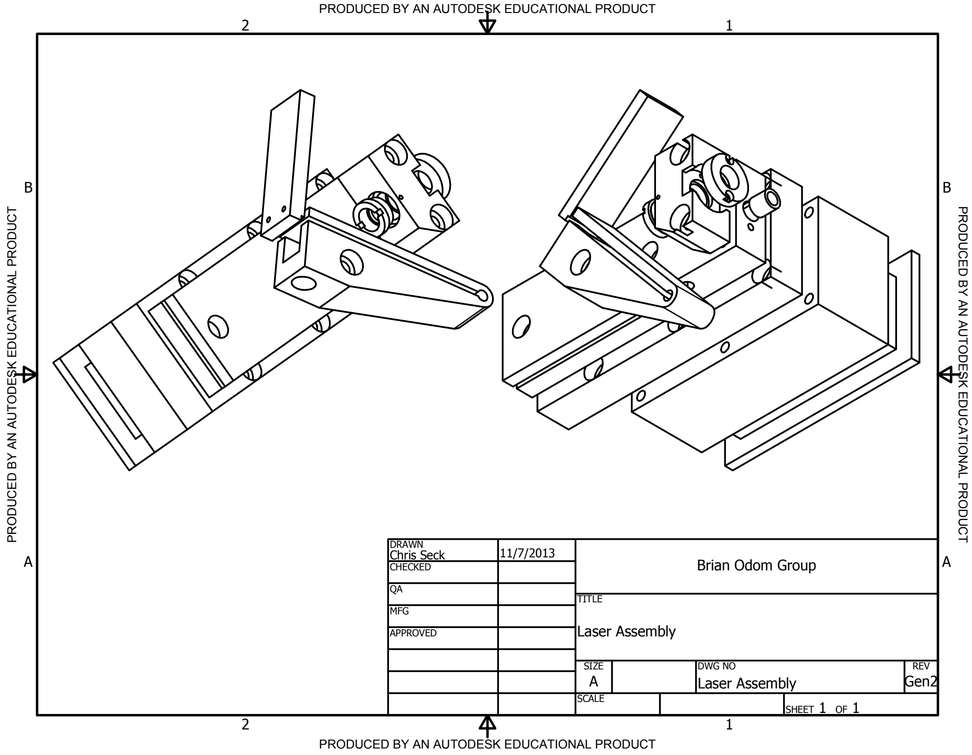
APPENDIX F

Mechanical Drawings

F.1. ECDL Project

F.1.1. ECDL Body

The lens casing spacer/adapters are omitted. The pieces simply fit inside the lens casings with a length determined by laser diode emitter location and working-distance of the collimation lens. The waveplate, if needed, is cemented between laser head and grating.

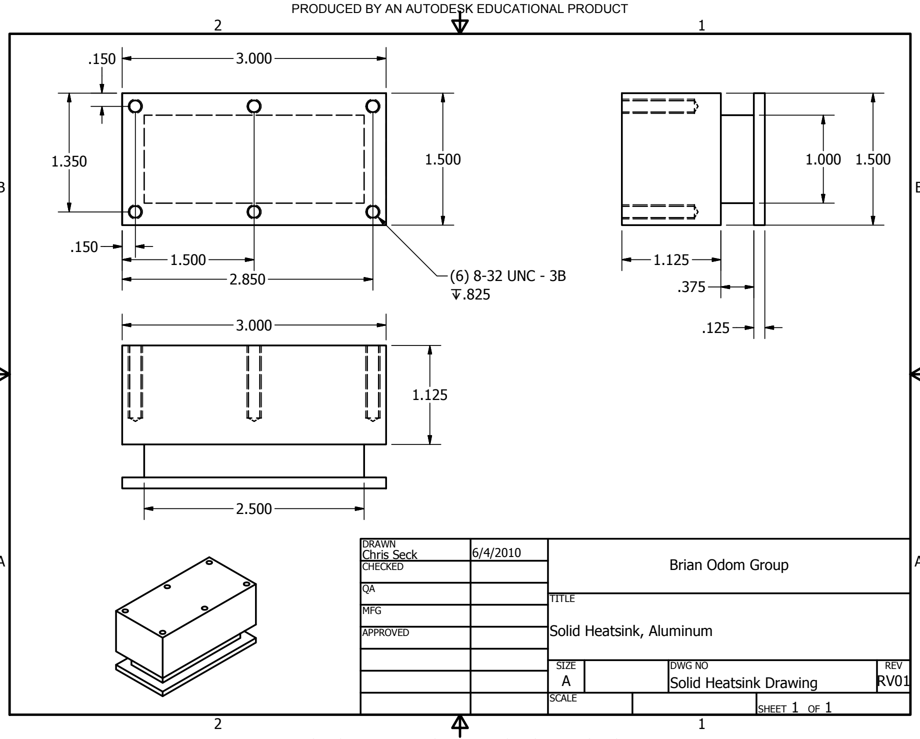

PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT
2
1
B
B
A
A
PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT
PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT
2
1
PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT

DRAWN
Chris Seck 11/7/2013
CHECKED
QA
MFG
APPROVED

Brian Odom Group

TITLE
Laser Assembly

SIZE A
DWG NO Laser Assembly
REV Gen2
SCALE
SHEET 1 OF 1



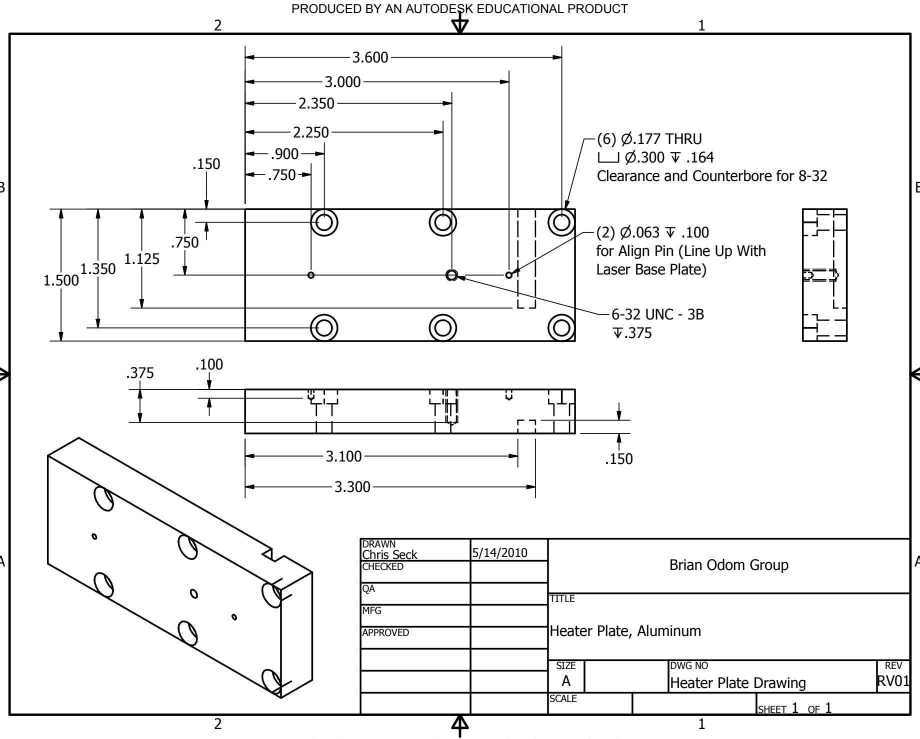
DRAWN
Chris Seck 6/4/2010
CHECKED
QA
MFG
APPROVED

Brian Odom Group

TITLE
Solid Heatsink, Aluminum

SIZE A DWG NO Solid Heatsink Drawing REV RV01

SCALE SHEET 1 OF 1



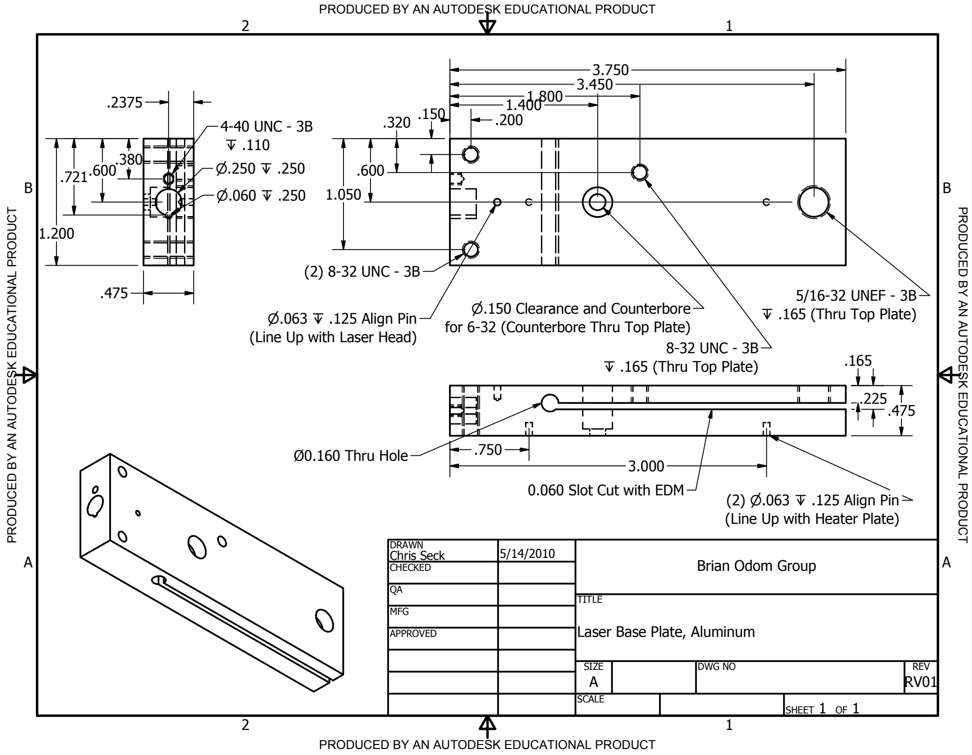
DRAWN
Chris Seck 5/14/2010
CHECKED
QA
MFG
APPROVED

Brian Odom Group

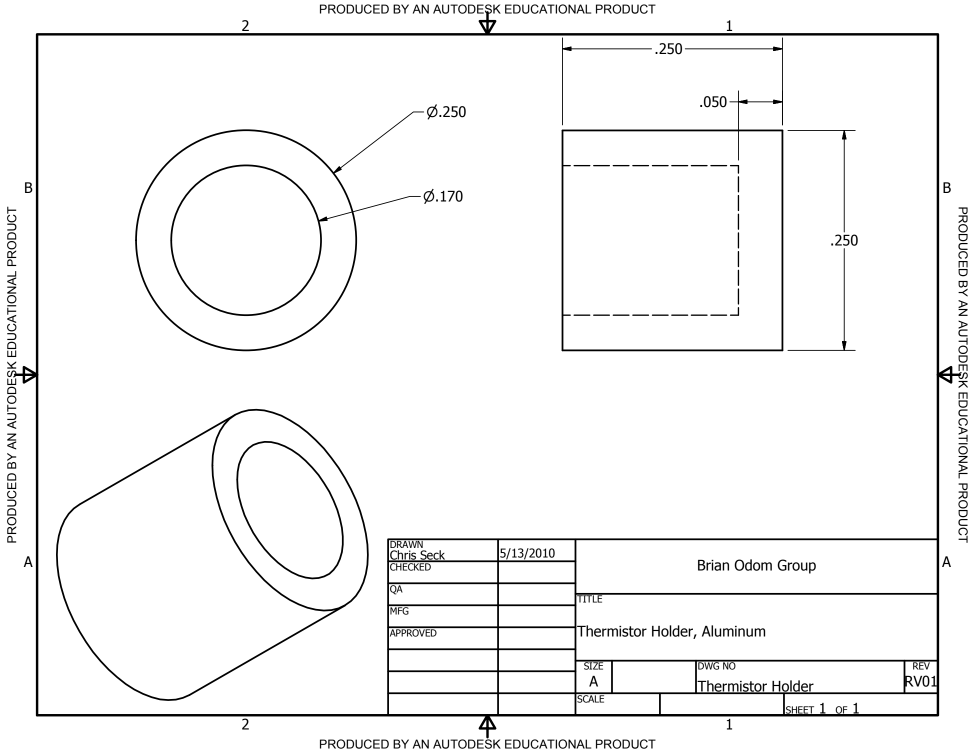
TITLE
Heater Plate, Aluminum

SIZE A DWG NO Heater Plate Drawing REV RV01

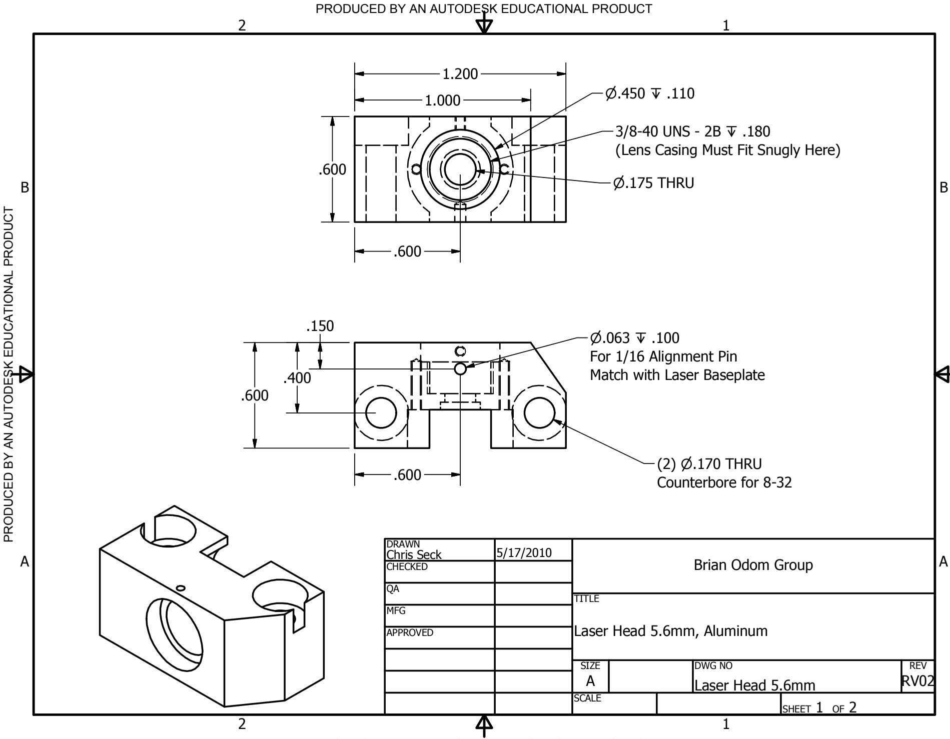
SCALE SHEET 1 OF 1


PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT
2
1
.2375
4-40 UNC - 3B
↧ .110
.380
.600
.721
Ø.250 ↧ .250
Ø.060 ↧ .250
1.200
.475
3.750
3.450
1.800
1.400
.150
.320
.200
.600
1.050
(2) 8-32 UNC - 3B
Ø.063 ↧ .125 Align Pin
(Line Up with Laser Head)
Ø.150 Clearance and Counterbore
for 6-32 (Counterbore Thru Top Plate)
5/16-32 UNEF - 3B
↧ .165 (Thru Top Plate)
8-32 UNC - 3B
↧ .165 (Thru Top Plate)
.165
.225
.475
Ø0.160 Thru Hole
.750
3.000
0.060 Slot Cut with EDM
(2) Ø.063 ↧ .125 Align Pin
(Line Up with Heater Plate)
B
A
PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT
PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT
PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT

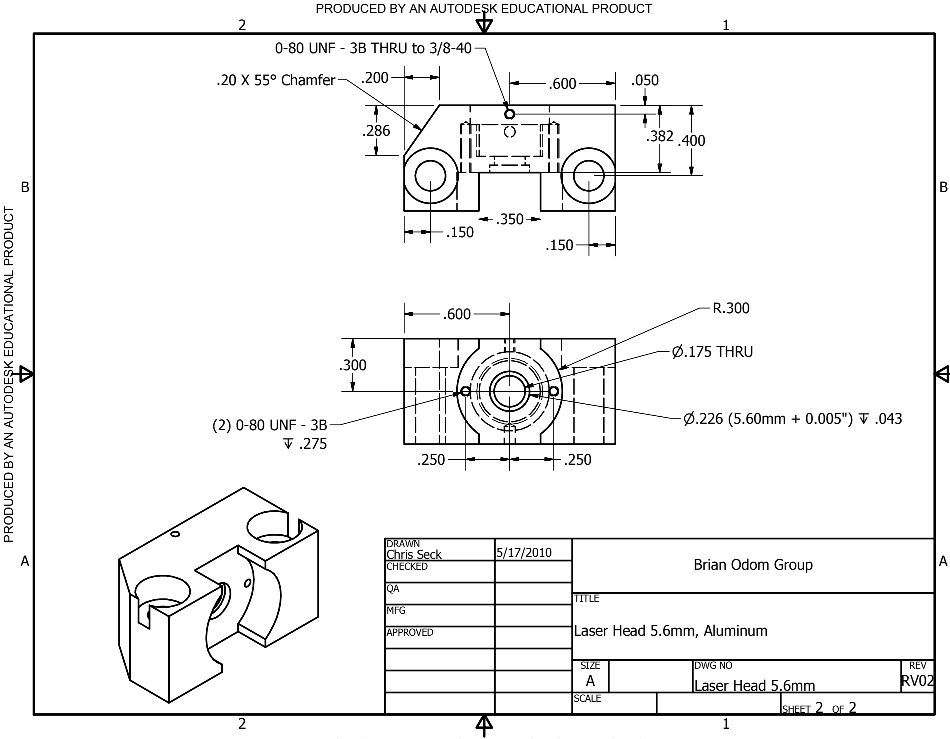
DRAWN Chris Seck	5/14/2010	Brian Odom Group
CHECKED		
QA		TITLE
MFG		Laser Base Plate, Aluminum
APPROVED		
		SIZE A · DWG NO · REV RV01
		SCALE · SHEET 1 OF 1


PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT
2
1
Ø.250
Ø.170
.250
.050
.250
B
A
PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT
PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT
PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT

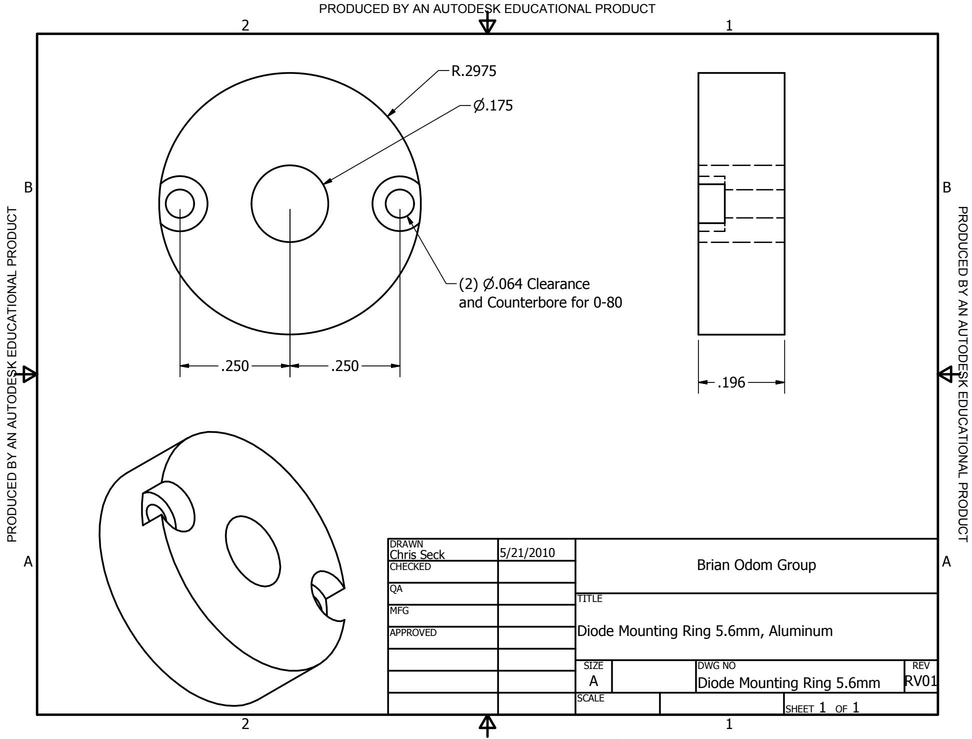
DRAWN Chris Seck	5/13/2010	Brian Odom Group
CHECKED		
QA		TITLE
MFG		Thermistor Holder, Aluminum
APPROVED		
		SIZE A · DWG NO Thermistor Holder · REV RV01
		SCALE · SHEET 1 OF 1



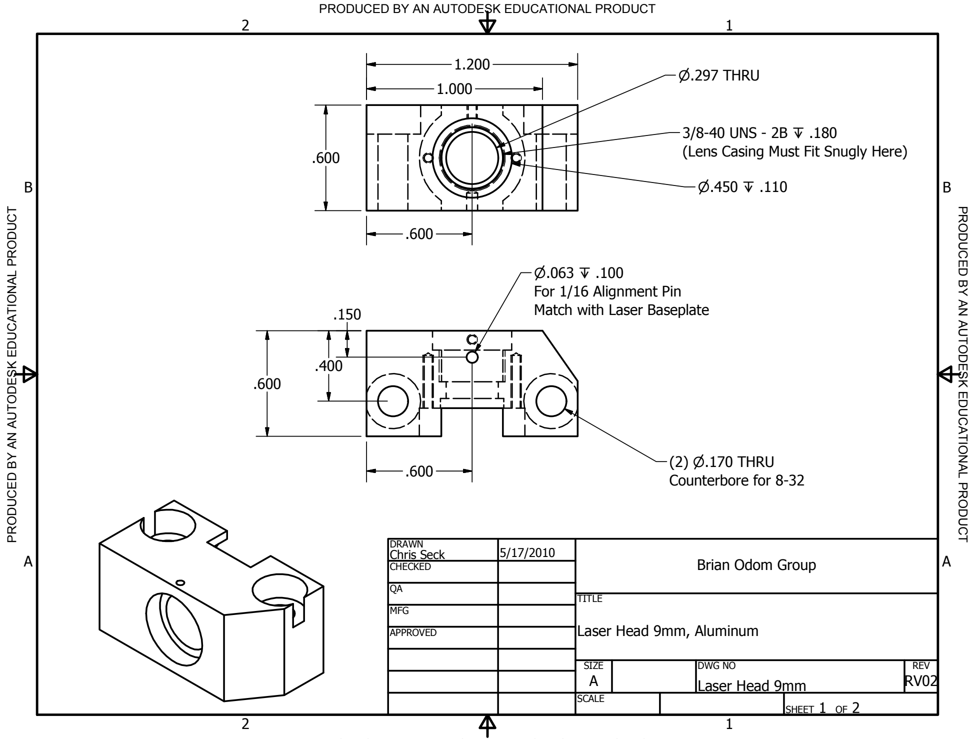
DRAWN Chris Seck	5/17/2010	Brian Odom Group
CHECKED		
QA		TITLE
MFG		
APPROVED		Laser Head 5.6mm, Aluminum
		SIZE A · DWG NO Laser Head 5.6mm · REV RV02
		SCALE · SHEET 1 OF 2



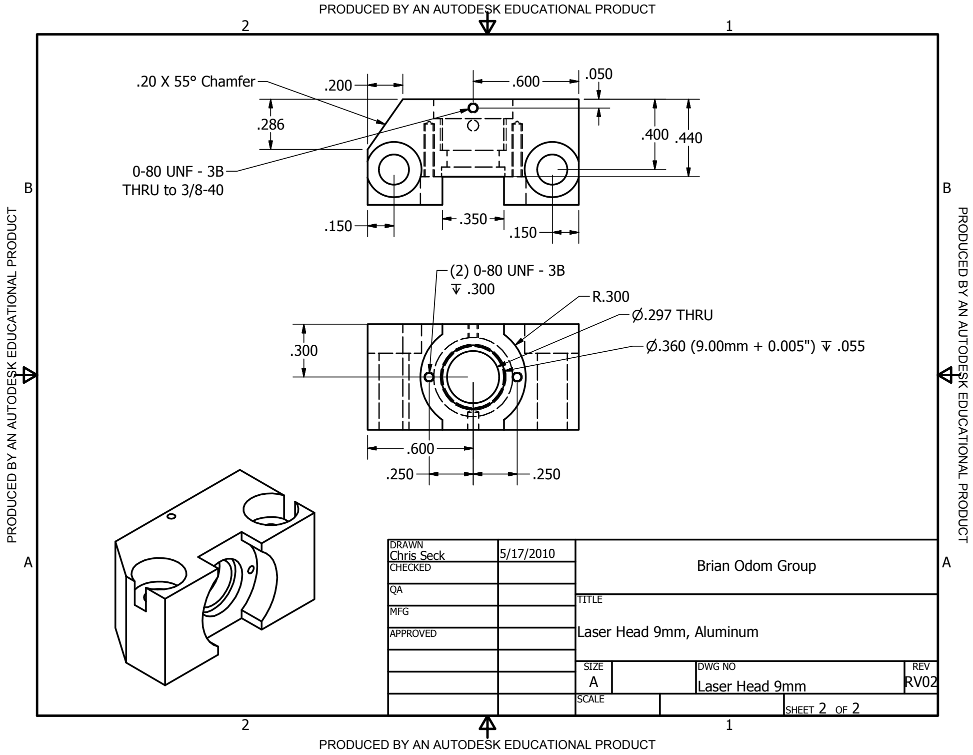
DRAWN Chris Seck	5/17/2010	Brian Odom Group
CHECKED		
QA		TITLE
MFG		
APPROVED		Laser Head 5.6mm, Aluminum
		SIZE A · DWG NO Laser Head 5.6mm · REV RV02
		SCALE · SHEET 2 OF 2



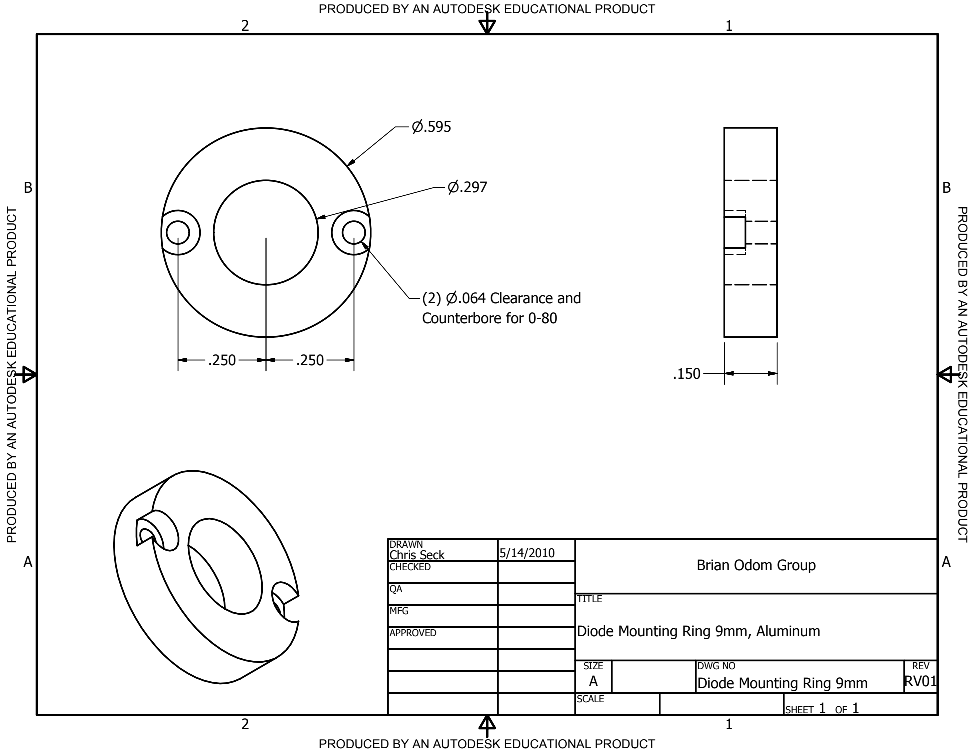
DRAWN Chris Seck	5/21/2010	Brian Odom Group
CHECKED		
QA		TITLE
MFG		Diode Mounting Ring 5.6mm, Aluminum
APPROVED		
		SIZE A · DWG NO Diode Mounting Ring 5.6mm · REV RV01
		SCALE · SHEET 1 OF 1



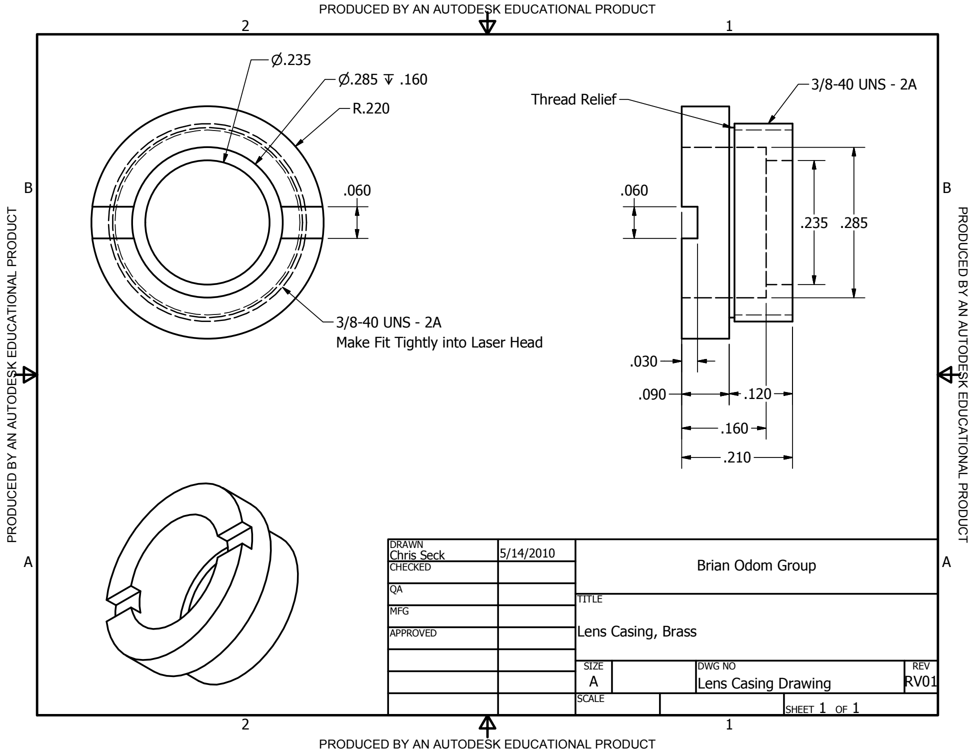
DRAWN Chris Seck	5/17/2010	Brian Odom Group
CHECKED		
QA		TITLE
MFG		Laser Head 9mm, Aluminum
APPROVED		
		SIZE A · DWG NO Laser Head 9mm · REV RV02
		SCALE · SHEET 1 OF 2



DRAWN Chris Seck	5/17/2010	Brian Odom Group
CHECKED		
QA		TITLE
MFG		Laser Head 9mm, Aluminum
APPROVED		
		SIZE A · DWG NO Laser Head 9mm · REV RV02
		SCALE · SHEET 2 OF 2



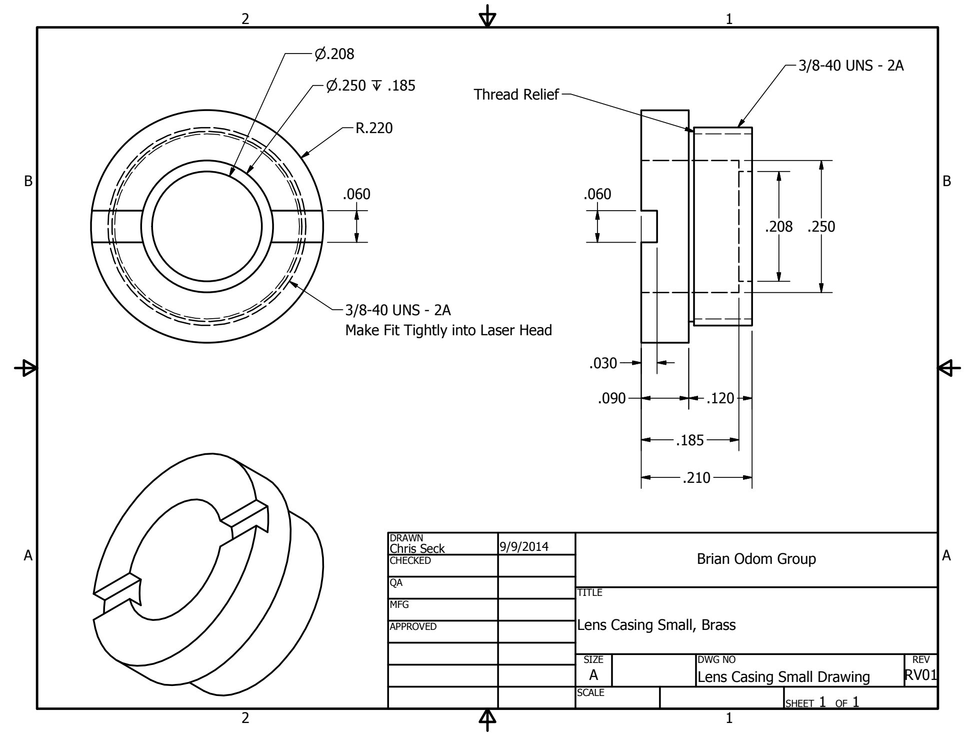
DRAWN Chris Seck	5/14/2010	Brian Odom Group
CHECKED		
QA		TITLE
MFG		Diode Mounting Ring 9mm, Aluminum
APPROVED		
		SIZE A · DWG NO Diode Mounting Ring 9mm · REV RV01
		SCALE · SHEET 1 OF 1



DRAWN Chris Seck	5/14/2010	Brian Odom Group
CHECKED		
QA		TITLE
MFG		
APPROVED		Lens Casing, Brass

SIZE A | DWG NO Lens Casing Drawing | REV RV01

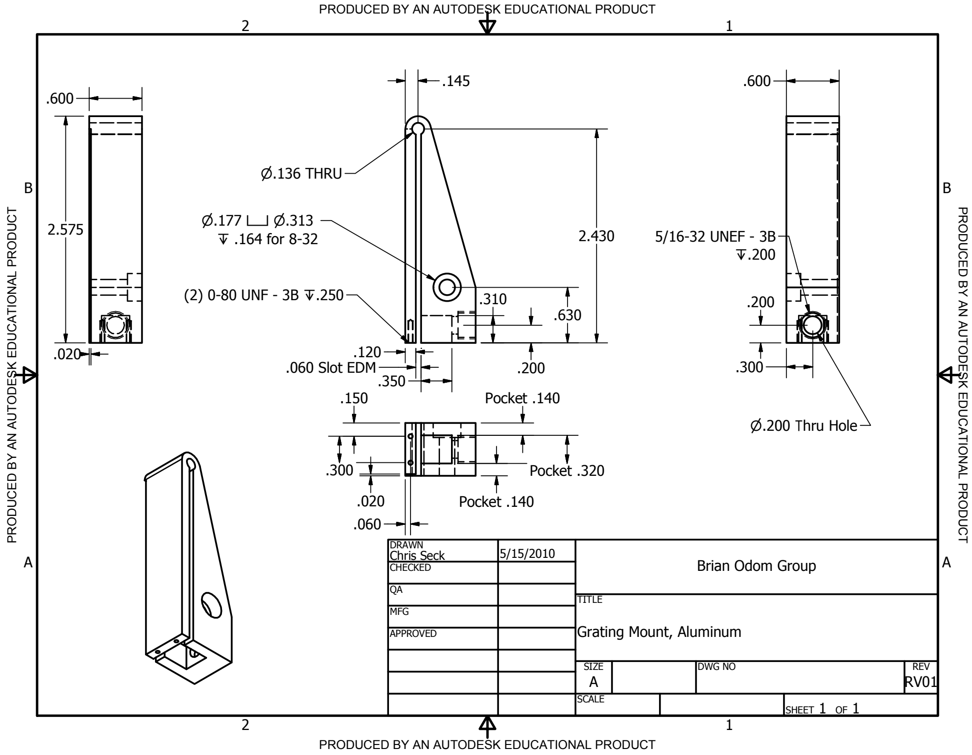
SCALE | SHEET 1 OF 1



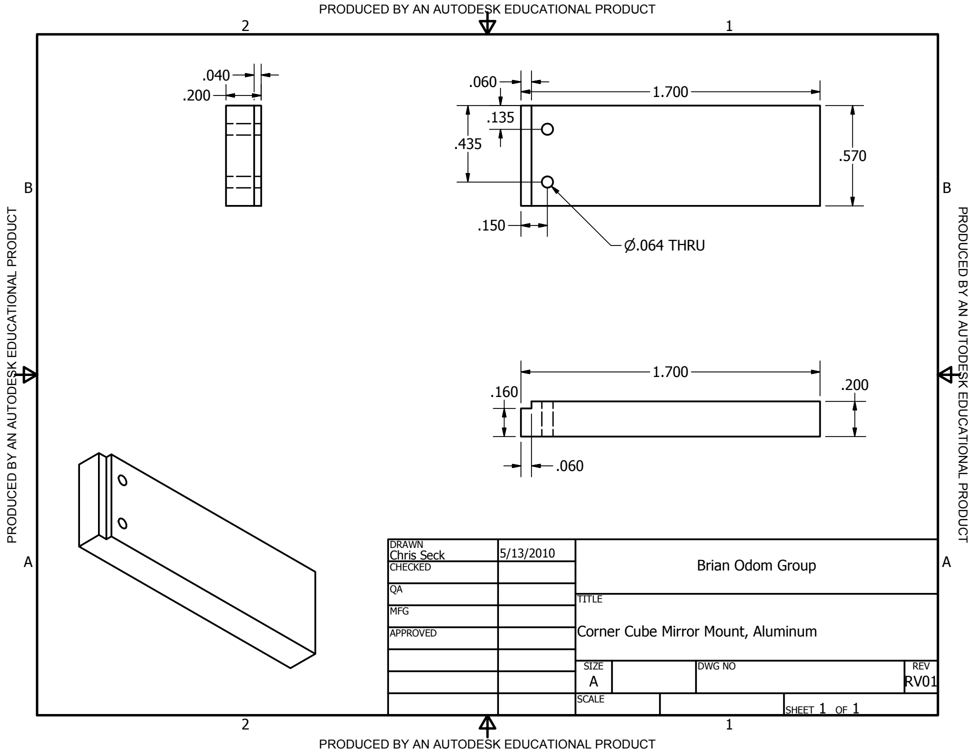
DRAWN Chris Seck	9/9/2014	Brian Odom Group
CHECKED		
QA		TITLE
MFG		
APPROVED		Lens Casing Small, Brass

SIZE A | DWG NO Lens Casing Small Drawing | REV RV01

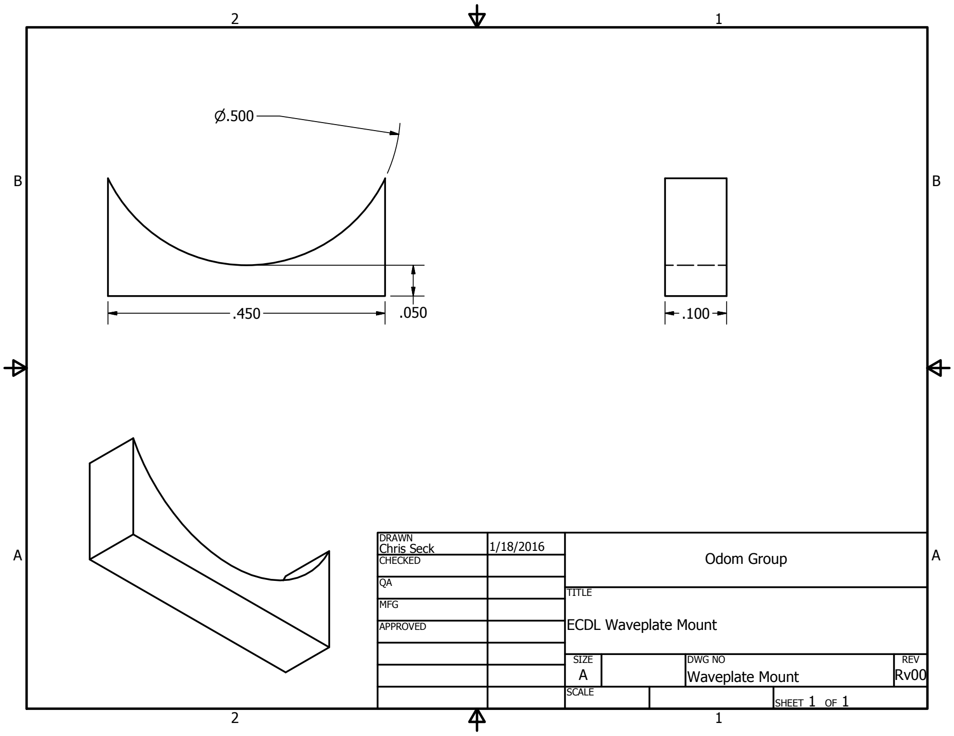
SCALE | SHEET 1 OF 1



DRAWN Chris Seck	5/15/2010	Brian Odom Group	
CHECKED			
QA		TITLE	
MFG		Grating Mount, Aluminum	
APPROVED			
		SIZE A	DWG NO	REV RV01
		SCALE	SHEET 1 OF 1



DRAWN Chris Seck	5/13/2010	Brian Odom Group	
CHECKED			
QA		TITLE	
MFG		Corner Cube Mirror Mount, Aluminum	
APPROVED			
		SIZE A	DWG NO	REV RV01
		SCALE	SHEET 1 OF 1



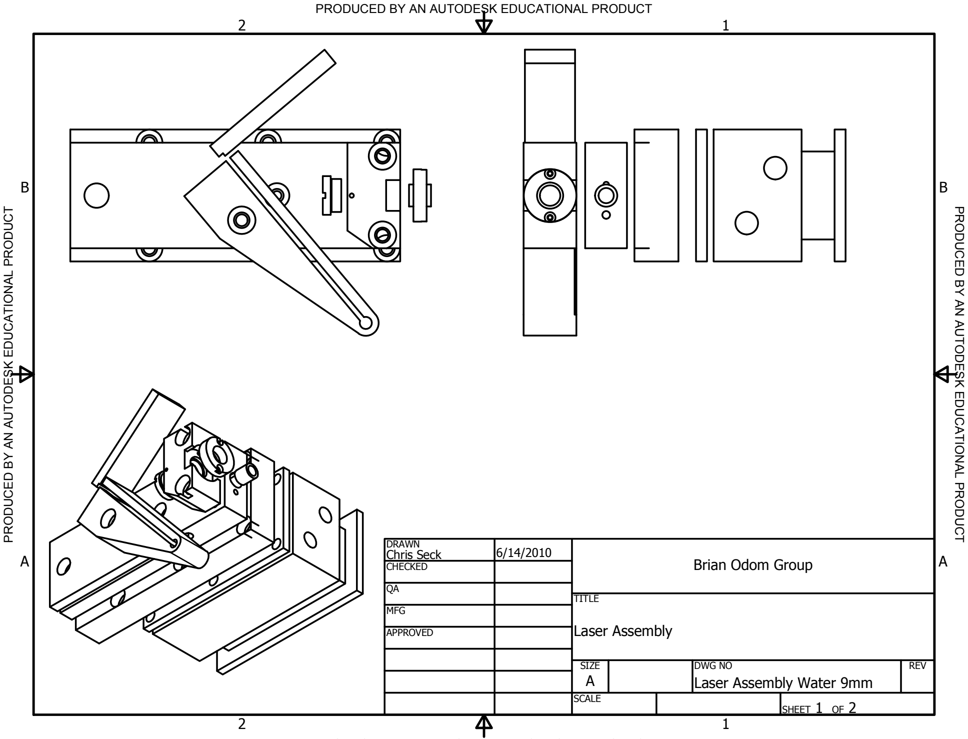
DRAWN Chris Seck 1/18/2016
CHECKED
QA
MFG
APPROVED

Odom Group

TITLE ECDL Waveplate Mount

SIZE A DWG NO Waveplate Mount REV Rv00

SCALE SHEET 1 OF 1



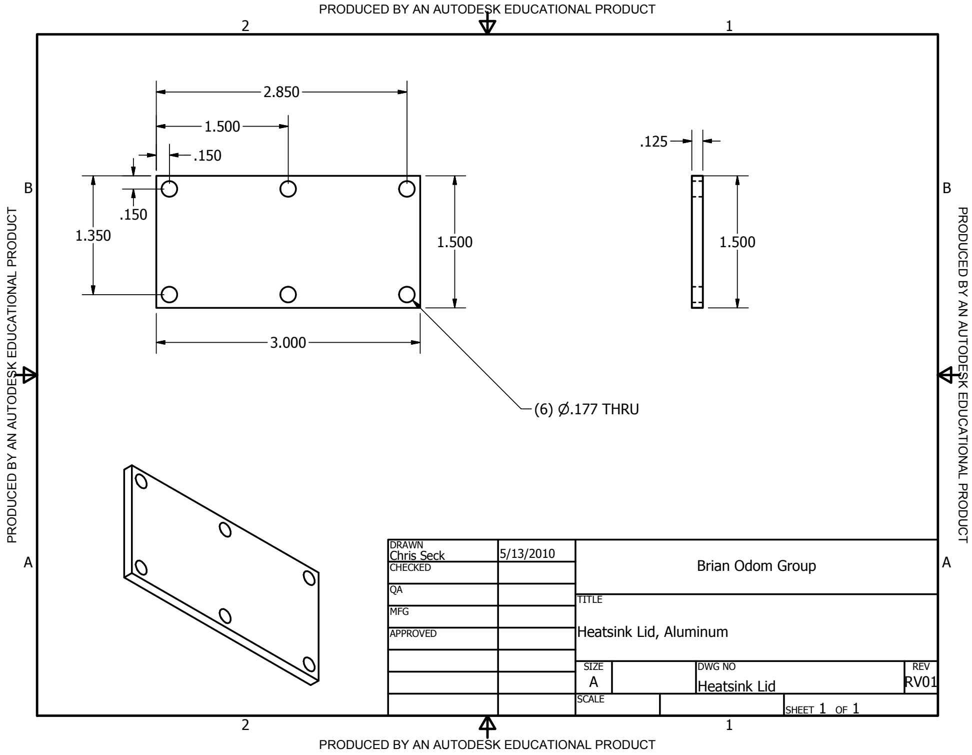
DRAWN Chris Seck 6/14/2010
CHECKED
QA
MFG
APPROVED

Brian Odom Group

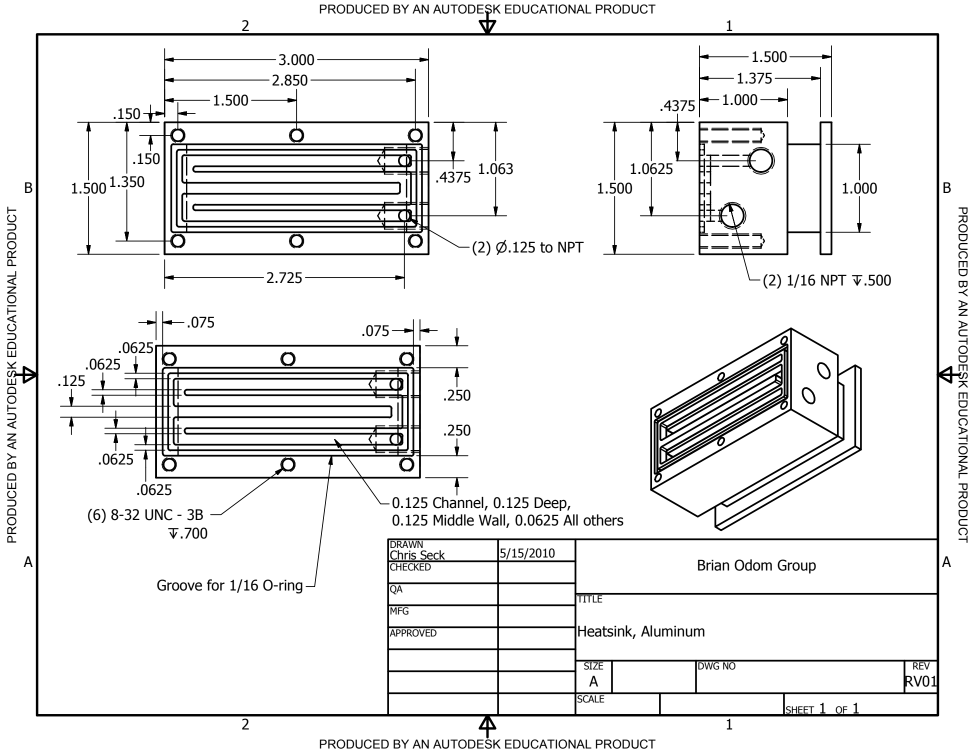
TITLE Laser Assembly

SIZE A DWG NO Laser Assembly Water 9mm REV

SCALE SHEET 1 OF 2

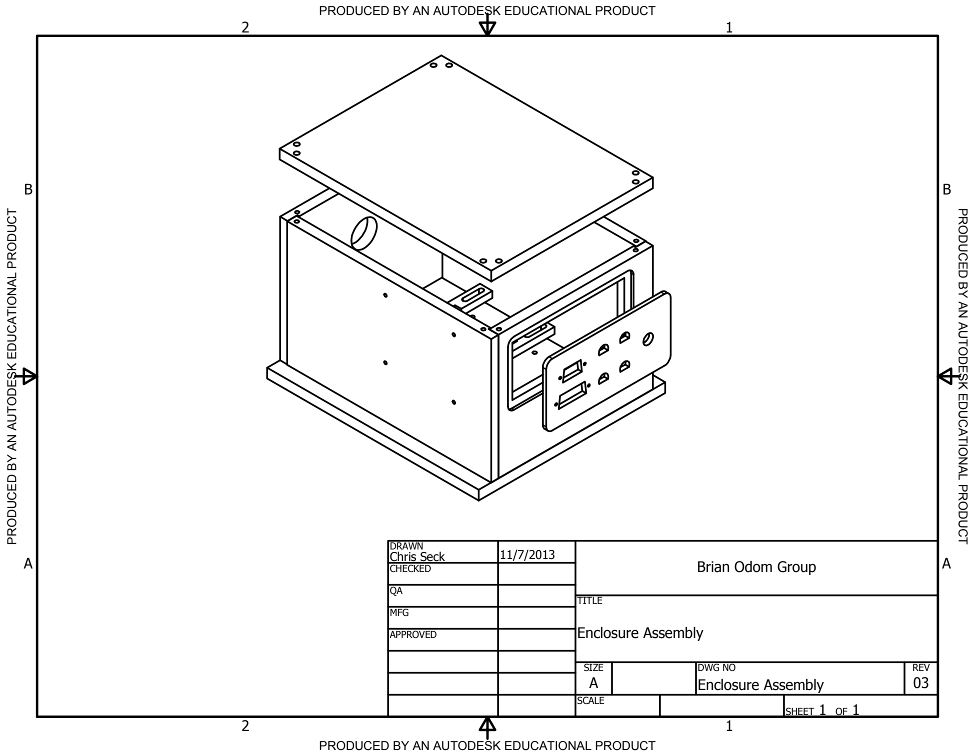


DRAWN Chris Seck	5/13/2010	Brian Odom Group
CHECKED		TITLE Heatsink Lid, Aluminum
QA		
MFG		
APPROVED		SIZE A · DWG NO Heatsink Lid · REV RV01
		SCALE · SHEET 1 OF 1

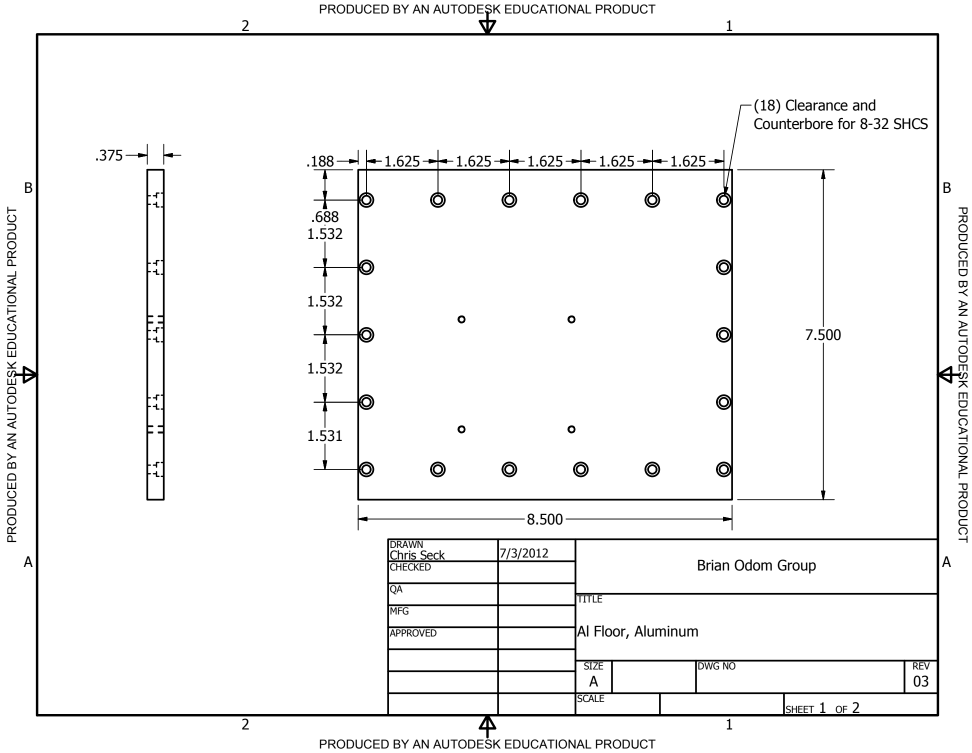


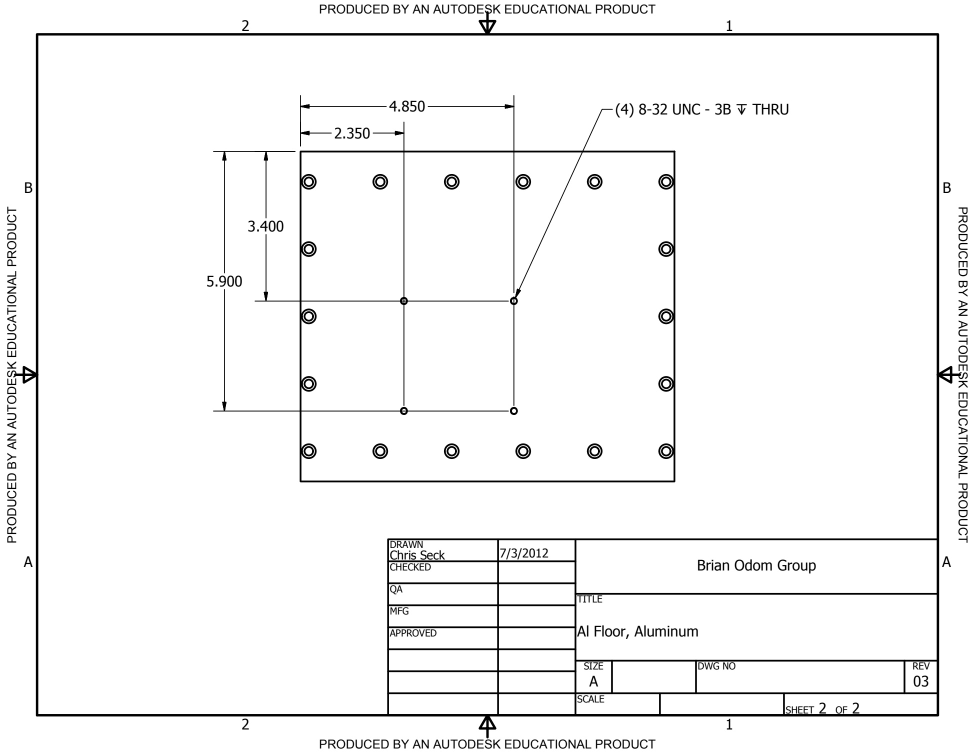
DRAWN Chris Seck	5/15/2010	Brian Odom Group
CHECKED		TITLE Heatsink, Aluminum
QA		
MFG		
APPROVED		SIZE A · DWG NO · REV RV01
		SCALE · SHEET 1 OF 1

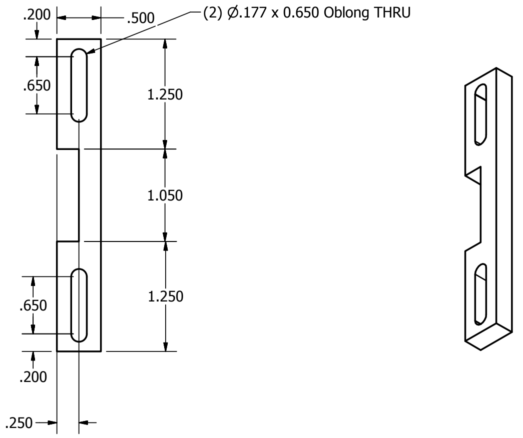
F.1.2. ECDL Enclosure


PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT
2
1
B
A
PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT
PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT
2
1
PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT

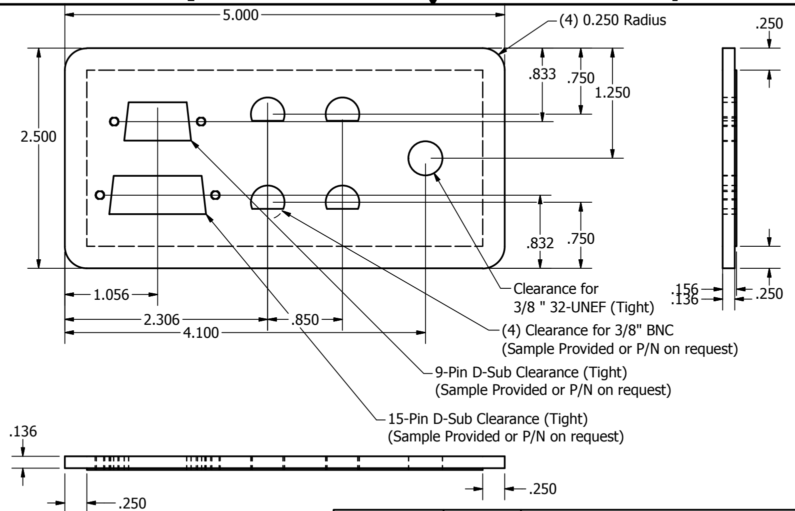
DRAWN Chris Seck	11/7/2013	Brian Odom Group		
CHECKED				
QA		TITLE		
MFG		Enclosure Assembly		
APPROVED				
		SIZE A	DWG NO Enclosure Assembly	REV 03
		SCALE	SHEET 1 OF 1	


PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT
2
1
B
A
.375
.188
1.625
1.625
1.625
1.625
1.625
(18) Clearance and Counterbore for 8-32 SHCS
.688
1.532
1.532
1.532
1.531
7.500
8.500
DRAWN
Chris Seck
7/3/2012
CHECKED
QA
MFG
APPROVED
Brian Odom Group
TITLE
Al Floor, Aluminum
SIZE
A
DWG NO
REV
03
SCALE
SHEET 1 OF 2

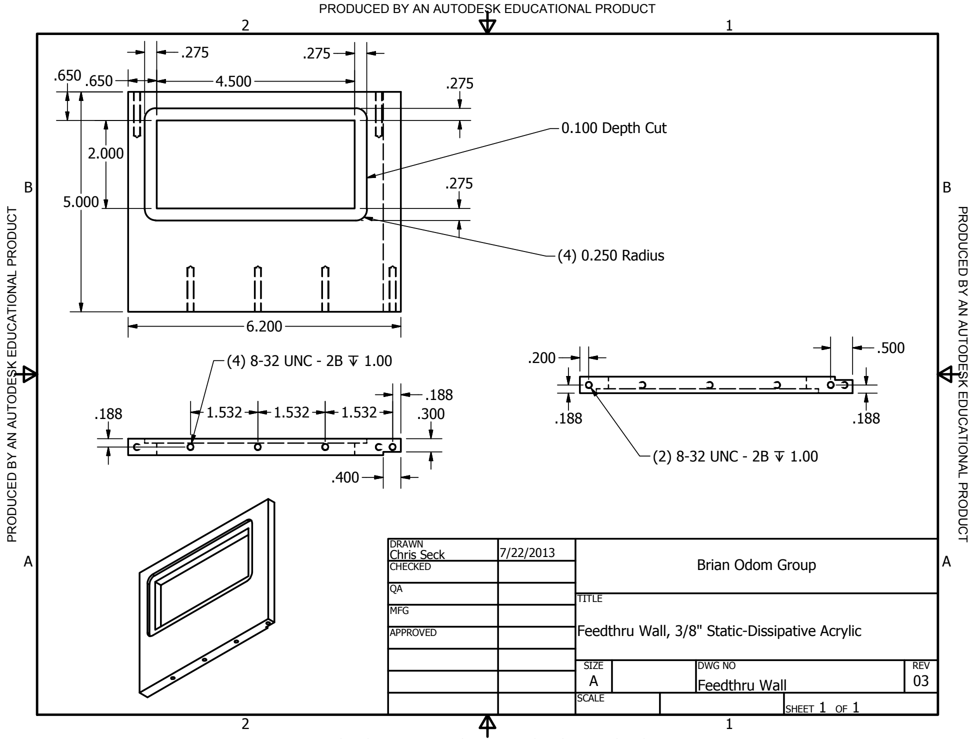

PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT
2
1
B
A
4.850
2.350
(4) 8-32 UNC - 3B ⊽ THRU
3.400
5.900
DRAWN
Chris Seck
7/3/2012
CHECKED
QA
MFG
APPROVED
Brian Odom Group
TITLE
Al Floor, Aluminum
SIZE
A
DWG NO
REV
03
SCALE
SHEET 2 OF 2



DRAWN Chris Seck	2/18/2011	Brian Odom Group		
CHECKED				
QA		TITLE		
MFG		Heatsink Bracket, 0.250" Aluminum (6061-T6)		
APPROVED				
		SIZE A	DWG NO Heatsink Bracket	REV 01
		SCALE		SHEET 1 OF 1


5.000
(4) 0.250 Radius
.250
.833
.750
1.250
2.500
.832
.750
Clearance for
3/8 " 32-UNEF (Tight)
.156
.136
.250
1.056
2.306
.850
4.100
(4) Clearance for 3/8" BNC
(Sample Provided or P/N on request)
9-Pin D-Sub Clearance (Tight)
(Sample Provided or P/N on request)
15-Pin D-Sub Clearance (Tight)
(Sample Provided or P/N on request)
.136
.250
.250

DRAWN Chris Seck	2/18/2011	Brian Odom Group		
CHECKED				
QA		TITLE		
MFG		Electronics Plate, 3/16" Static-Dissipative Acrylic		
APPROVED				
		SIZE A	DWG NO Electronics Plate	REV 02
		SCALE		SHEET 1 OF 1



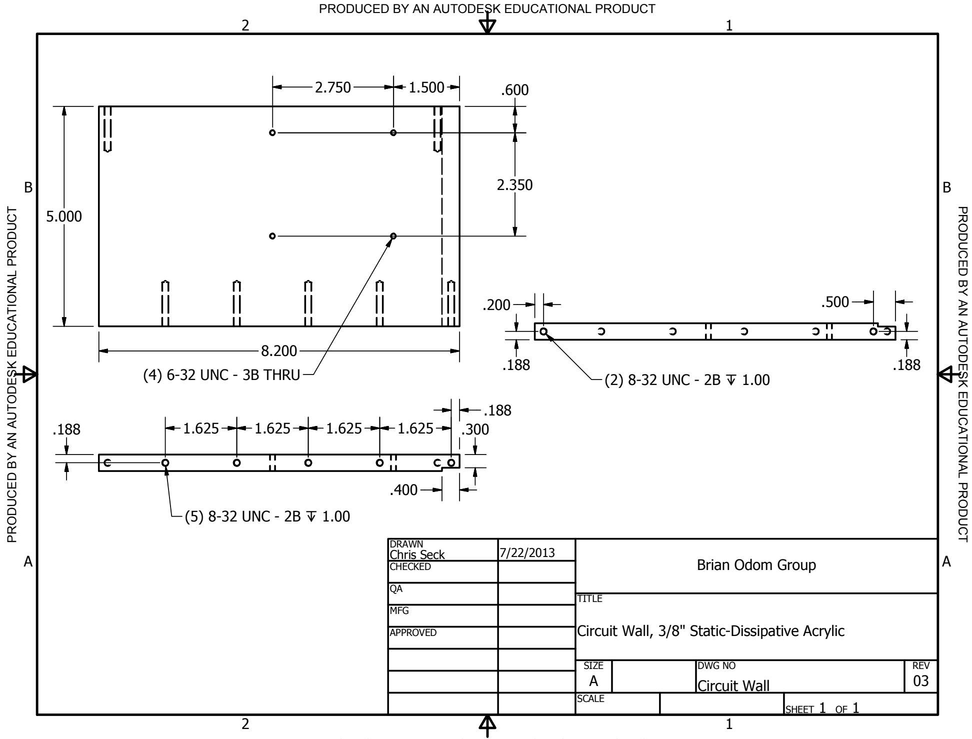
DRAWN Chris Seck	7/22/2013
CHECKED	
QA	
MFG	
APPROVED	

Brian Odom Group

TITLE
Feedthru Wall, 3/8" Static-Dissipative Acrylic

SIZE		DWG NO	REV
A		Feedthru Wall	03

SCALE SHEET 1 OF 1



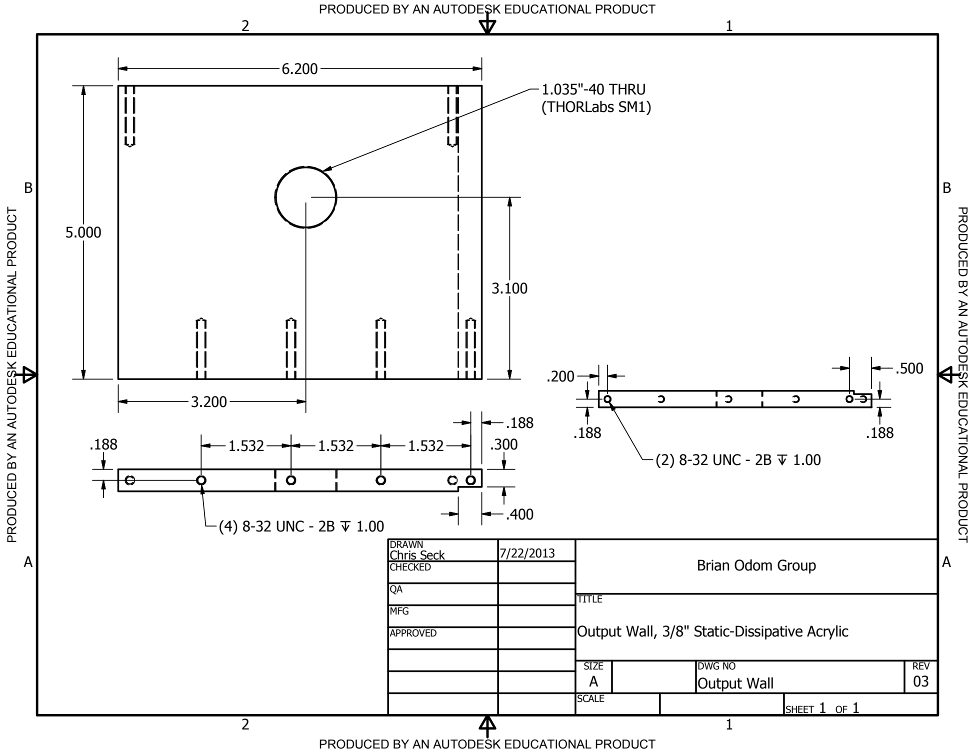
DRAWN Chris Seck	7/22/2013
CHECKED	
QA	
MFG	
APPROVED	

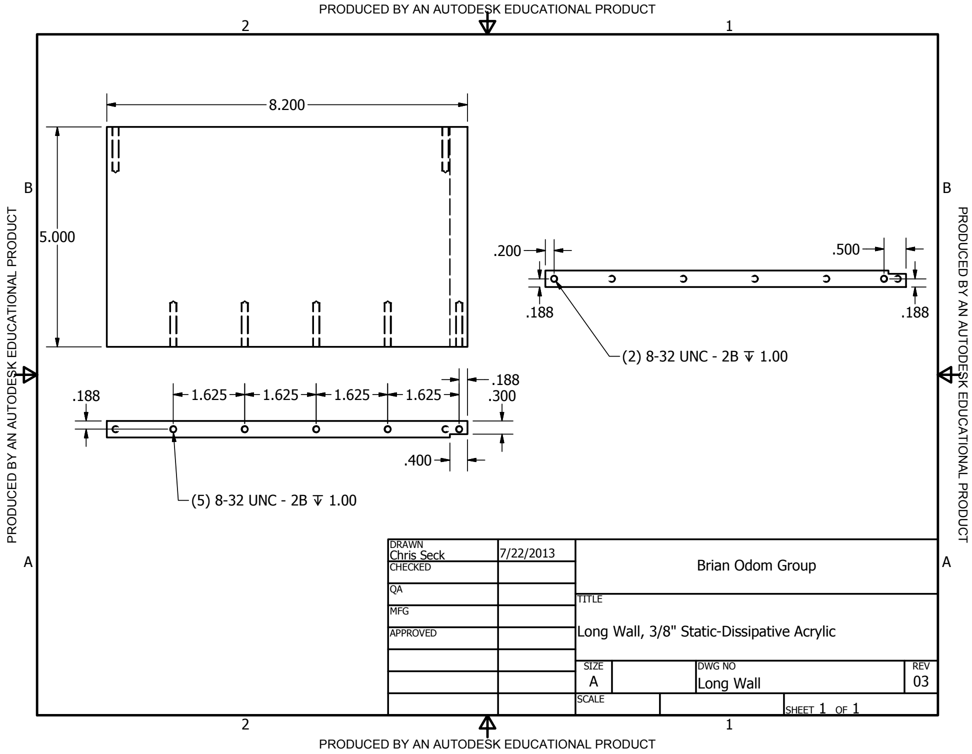
Brian Odom Group

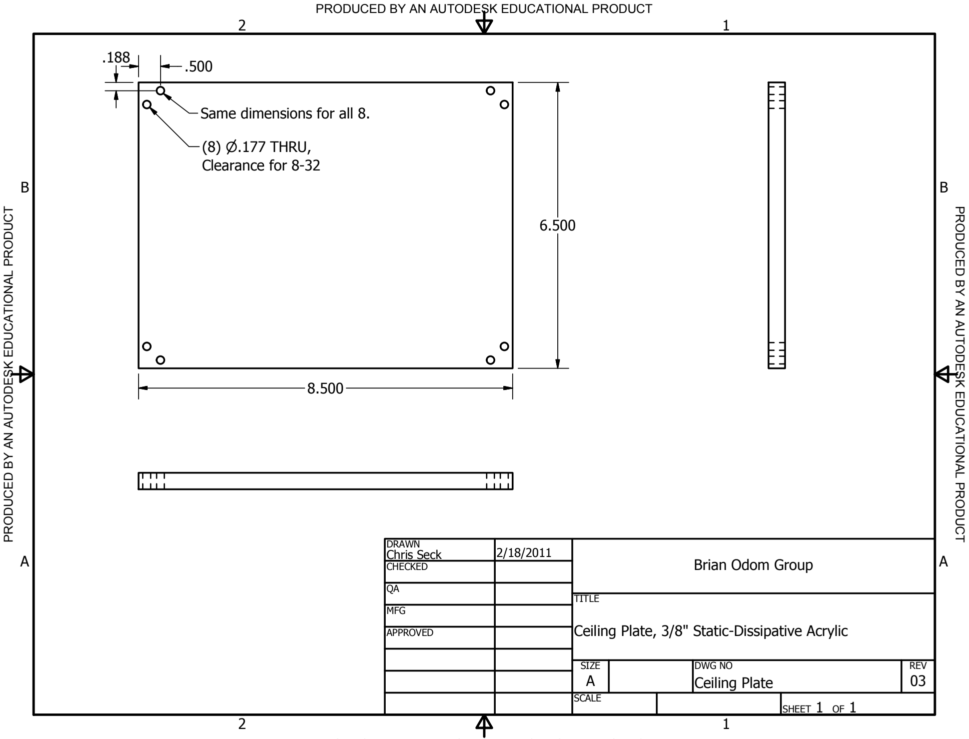
TITLE
Circuit Wall, 3/8" Static-Dissipative Acrylic

SIZE		DWG NO	REV
A		Circuit Wall	03

SCALE SHEET 1 OF 1


PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT
2
1
B
A
6.200
1.035"-40 THRU
(THORLabs SM1)
5.000
3.100
3.200
.200
.500
.188
.188
(2) 8-32 UNC - 2B ↧ 1.00
.188
.188
1.532
1.532
1.532
.300
.400
(4) 8-32 UNC - 2B ↧ 1.00
DRAWN
Chris Seck
7/22/2013
CHECKED
QA
MFG
APPROVED
Brian Odom Group
TITLE
Output Wall, 3/8" Static-Dissipative Acrylic
SIZE
A
DWG NO
Output Wall
REV
03
SCALE
SHEET 1 OF 1

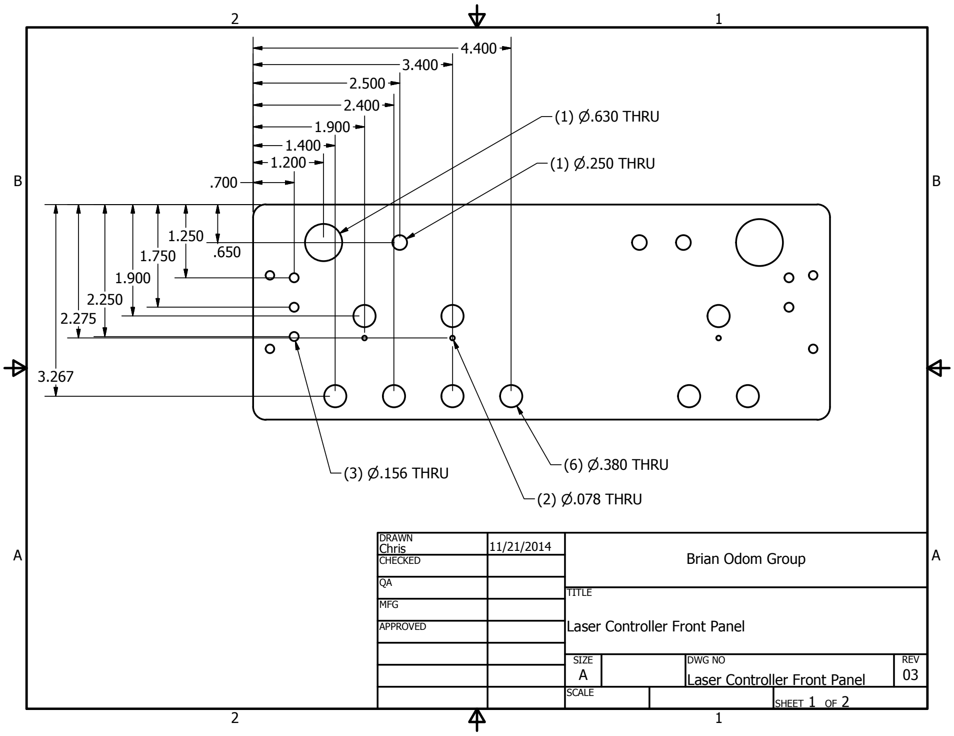

PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT
2
1
B
A
8.200
5.000
.200
.500
.188
.188
(2) 8-32 UNC - 2B ↧ 1.00
.188
1.625
1.625
1.625
1.625
.188
.300
.400
(5) 8-32 UNC - 2B ↧ 1.00
DRAWN
Chris Seck
7/22/2013
CHECKED
QA
MFG
APPROVED
Brian Odom Group
TITLE
Long Wall, 3/8" Static-Dissipative Acrylic
SIZE
A
DWG NO
Long Wall
REV
03
SCALE
SHEET 1 OF 1

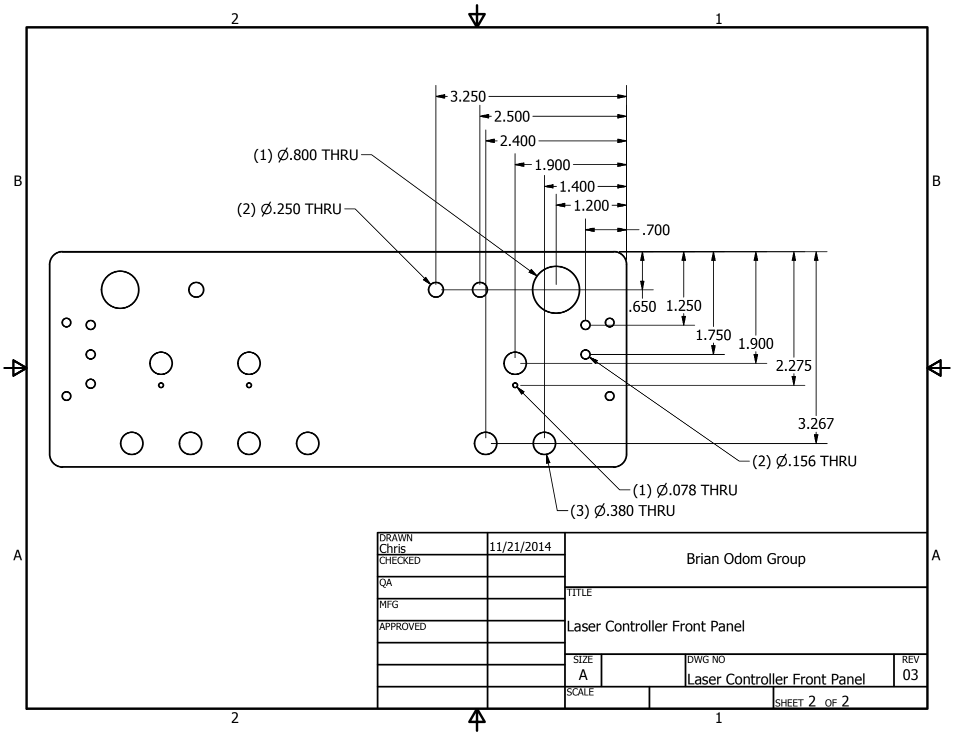


DRAWN Chris Seck	2/18/2011	Brian Odom Group
CHECKED		
QA		TITLE
MFG		Ceiling Plate, 3/8" Static-Dissipative Acrylic
APPROVED		
		SIZE A · DWG NO Ceiling Plate · REV 03
		SCALE · SHEET 1 OF 1

F.1.3. ECDL System Electronics

The rear-panel-mounted potentiometer plate uses all-metal standoffs. The base enclosure is a Hammond Manufacturing 1402KV. The paint is removed on the bottom and top panels such that the mounting screws ground them to the case.



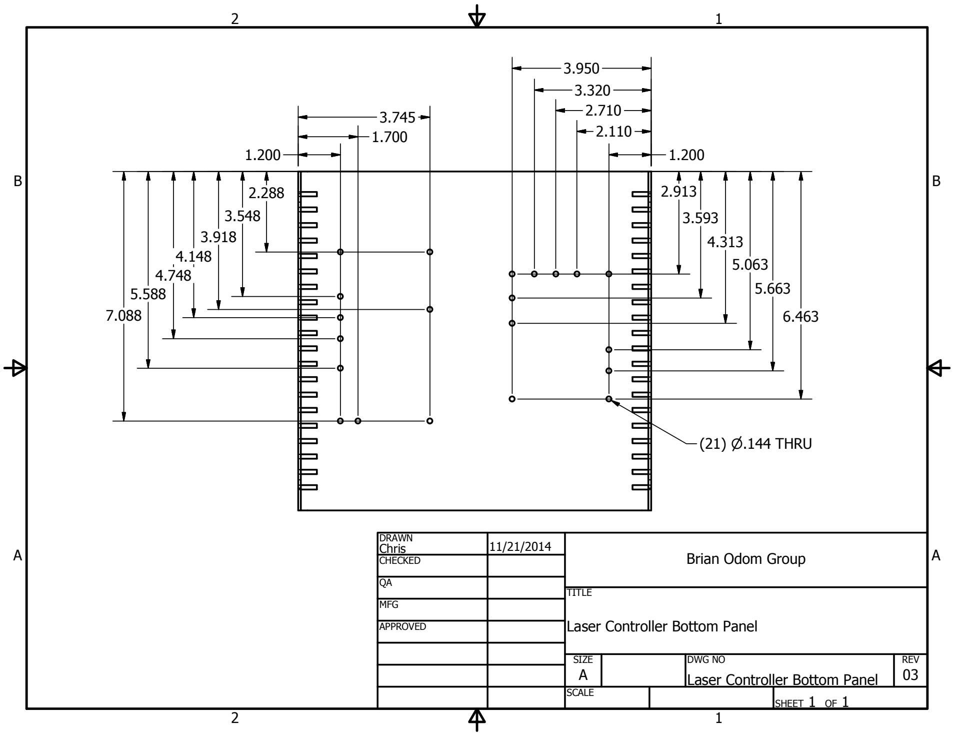


DRAWN Chris 11/21/2014
CHECKED
QA
MFG
APPROVED

Brian Odom Group

TITLE Laser Controller Front Panel

SIZE A
DWG NO Laser Controller Front Panel
REV 03
SCALE
SHEET 2 OF 2

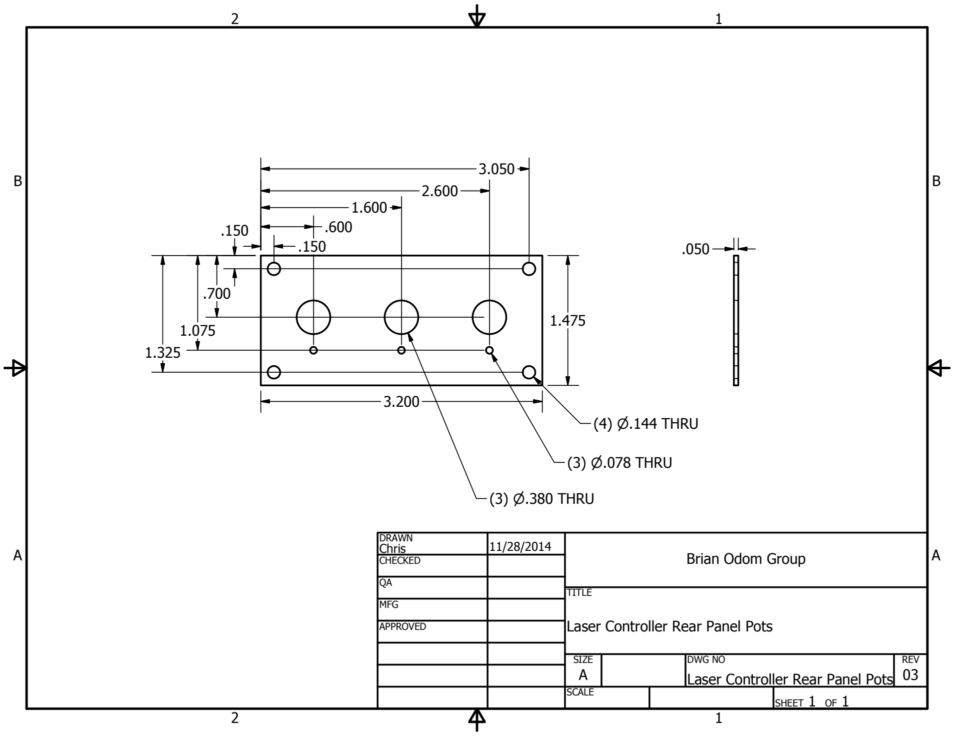


DRAWN Chris 11/21/2014
CHECKED
QA
MFG
APPROVED

Brian Odom Group

TITLE Laser Controller Bottom Panel

SIZE A
DWG NO Laser Controller Bottom Panel
REV 03
SCALE
SHEET 1 OF 1

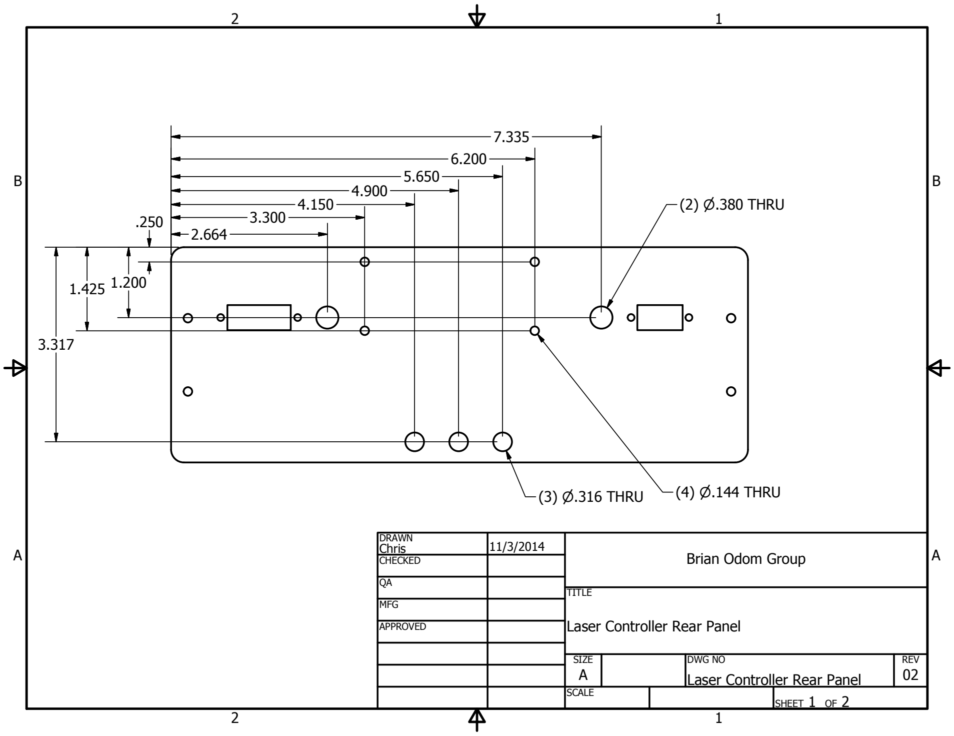


DRAWN Chris 11/28/2014
CHECKED
QA
MFG
APPROVED

Brian Odom Group

TITLE
Laser Controller Rear Panel Pots

SIZE A
DWG NO Laser Controller Rear Panel Pots
REV 03
SCALE
SHEET 1 OF 1

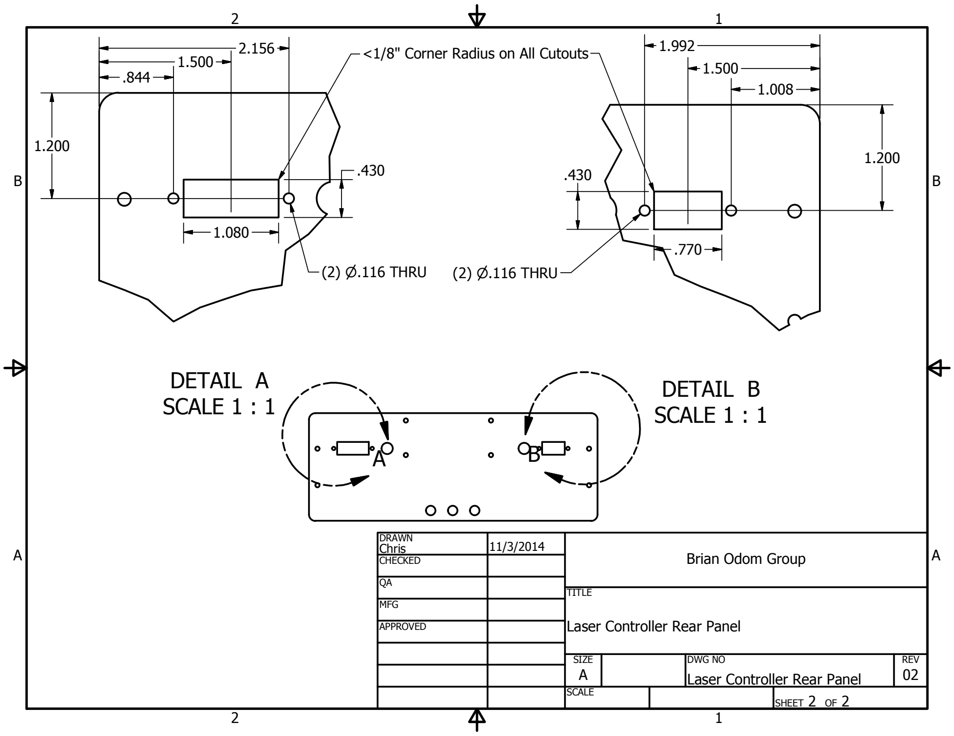


DRAWN Chris 11/3/2014
CHECKED
QA
MFG
APPROVED

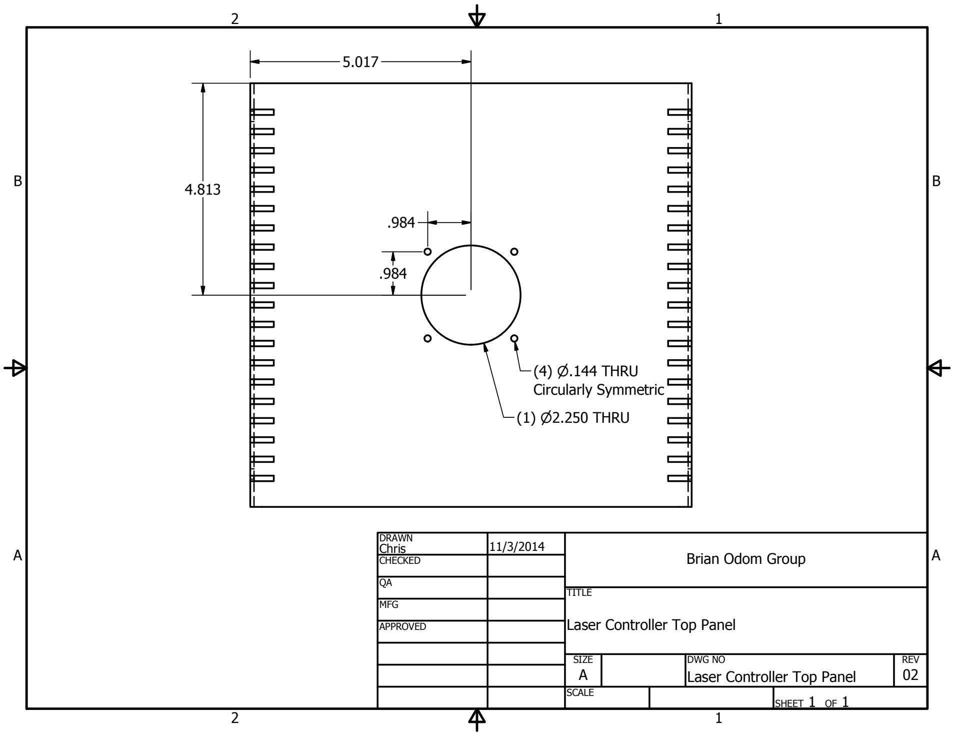
Brian Odom Group

TITLE
Laser Controller Rear Panel

SIZE A
DWG NO Laser Controller Rear Panel
REV 02
SCALE
SHEET 1 OF 2



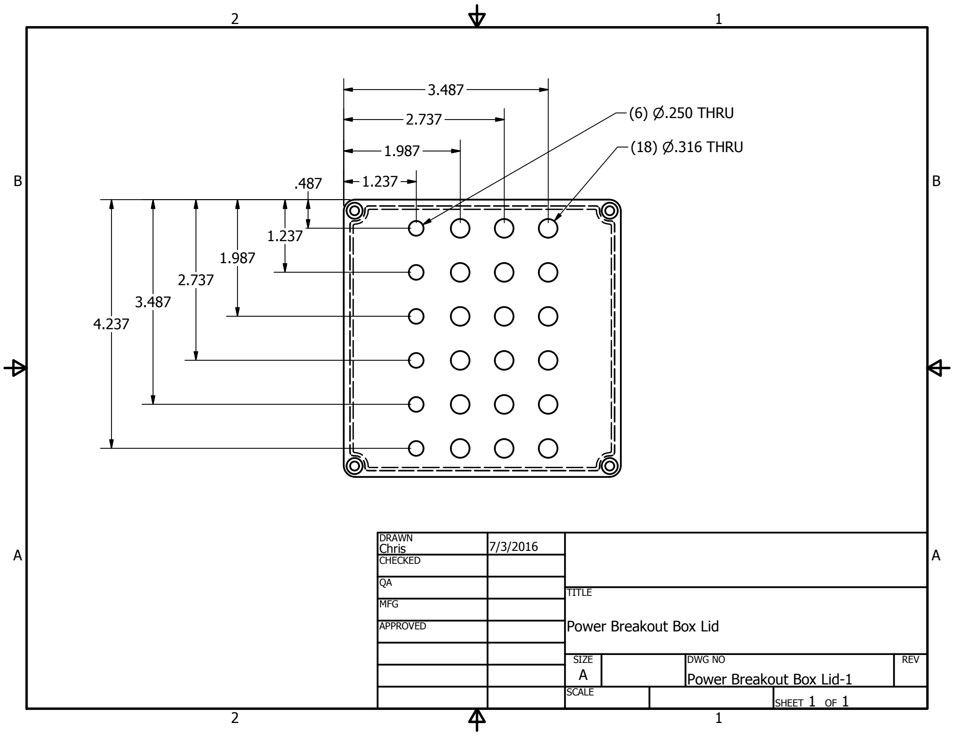
DRAWN Chris	11/3/2014	Brian Odom Group			
CHECKED		TITLE Laser Controller Rear Panel			
QA					
MFG					
APPROVED					
		SIZE A		DWG NO Laser Controller Rear Panel	REV 02
		SCALE		SHEET 2 OF 2	



DRAWN Chris	11/3/2014	Brian Odom Group			
CHECKED		TITLE Laser Controller Top Panel			
QA					
MFG					
APPROVED					
		SIZE A		DWG NO Laser Controller Top Panel	REV 02
		SCALE		SHEET 1 OF 1	

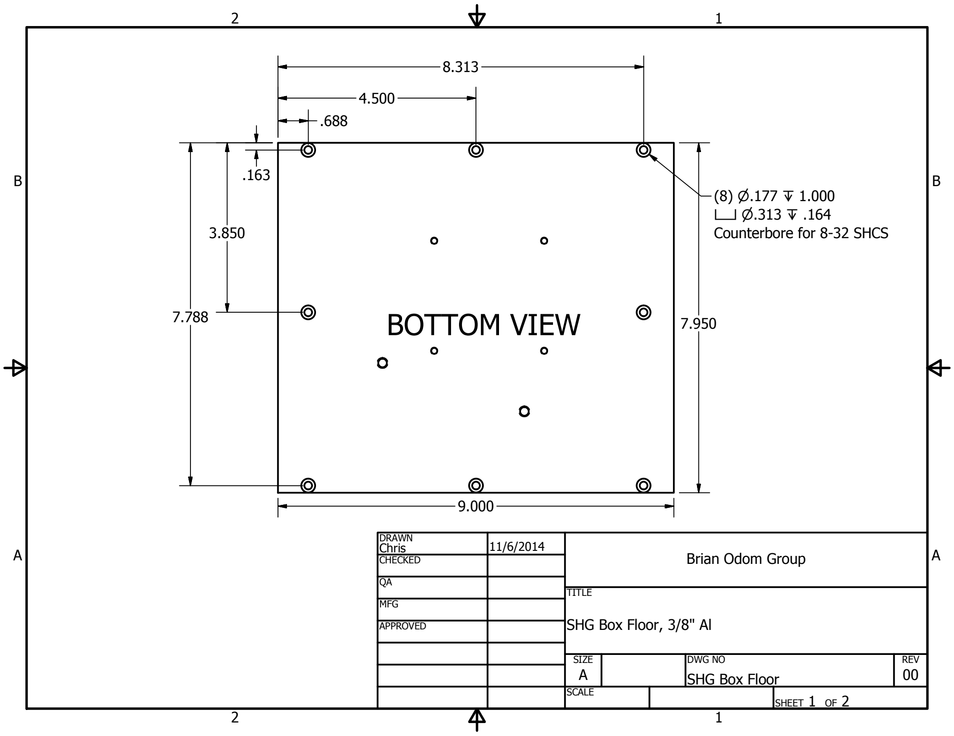
F.1.4. ECDL System Power Breakout Box Lid

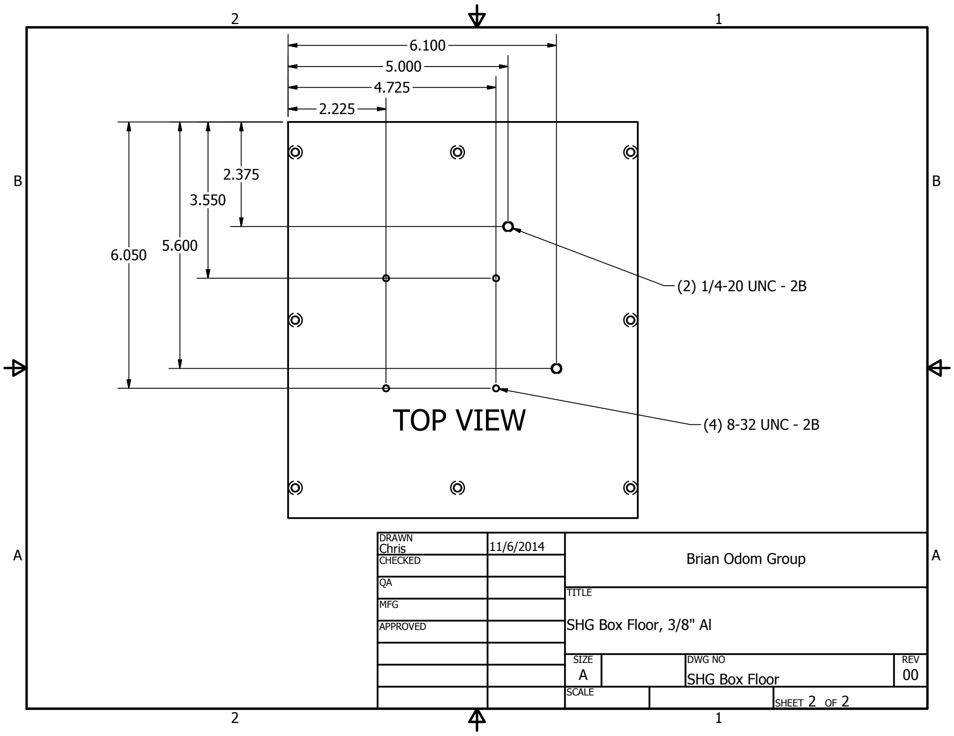
This lid provides a breakout box for a power supply to power multiple ECDL controllers and/or other electronics. Five of the six small holes are for DPDT switches for individual power switching. On the input-power row, a SHCS is bolted into this hole with a lead to the power supply ground banana jack to ground the distribution box. The base enclosure is a Hammond Manufacturing 1590U.

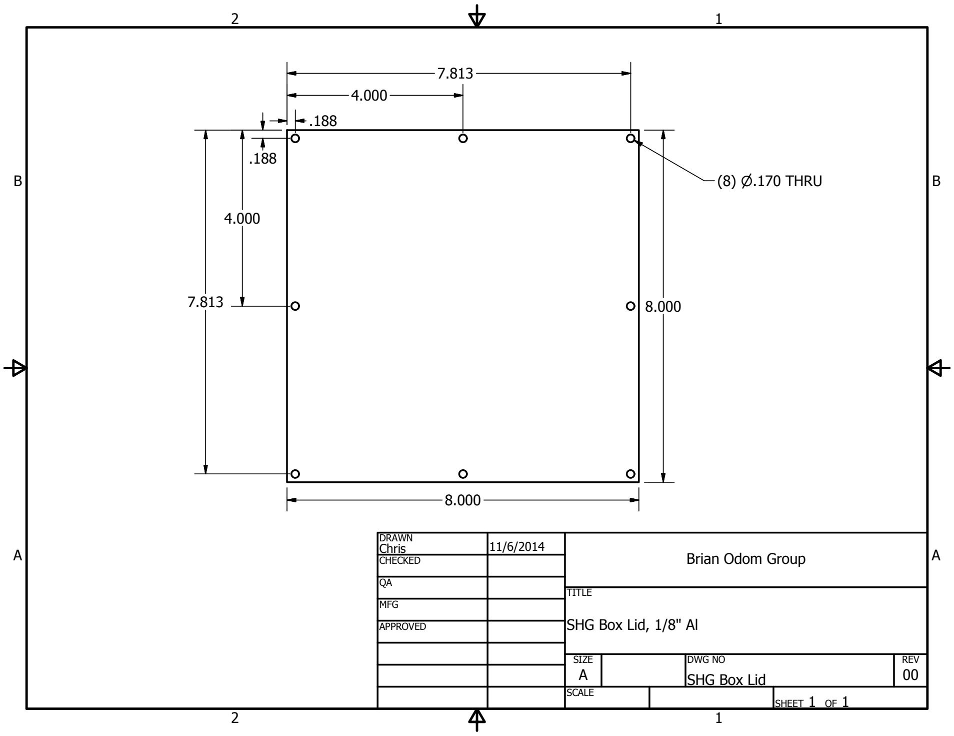


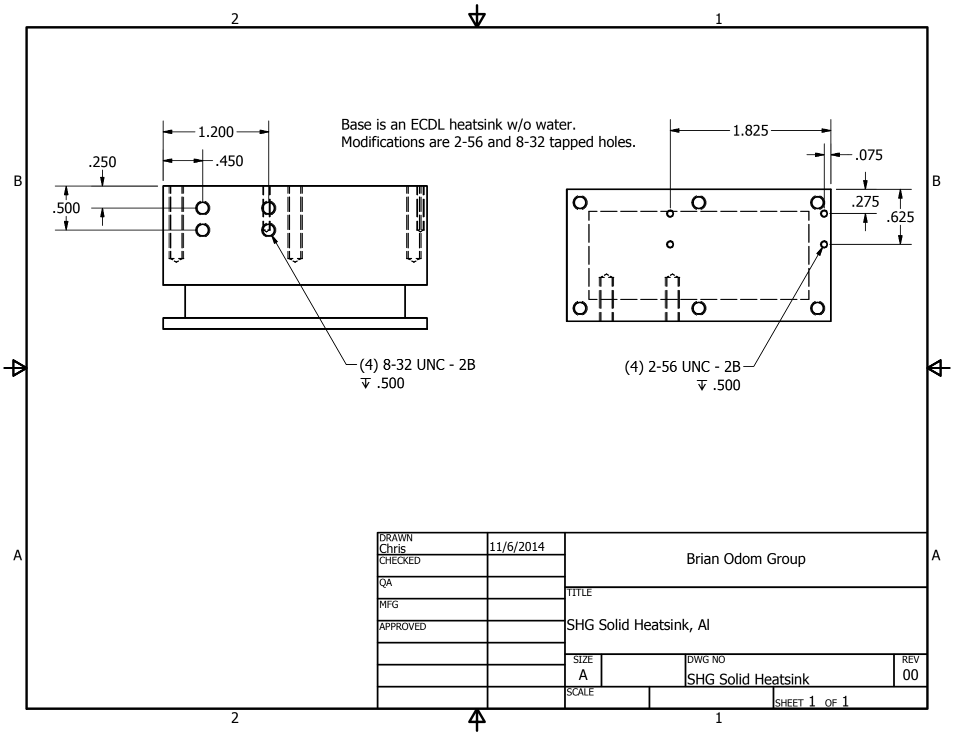
F.1.5. SHG Module Enclosure

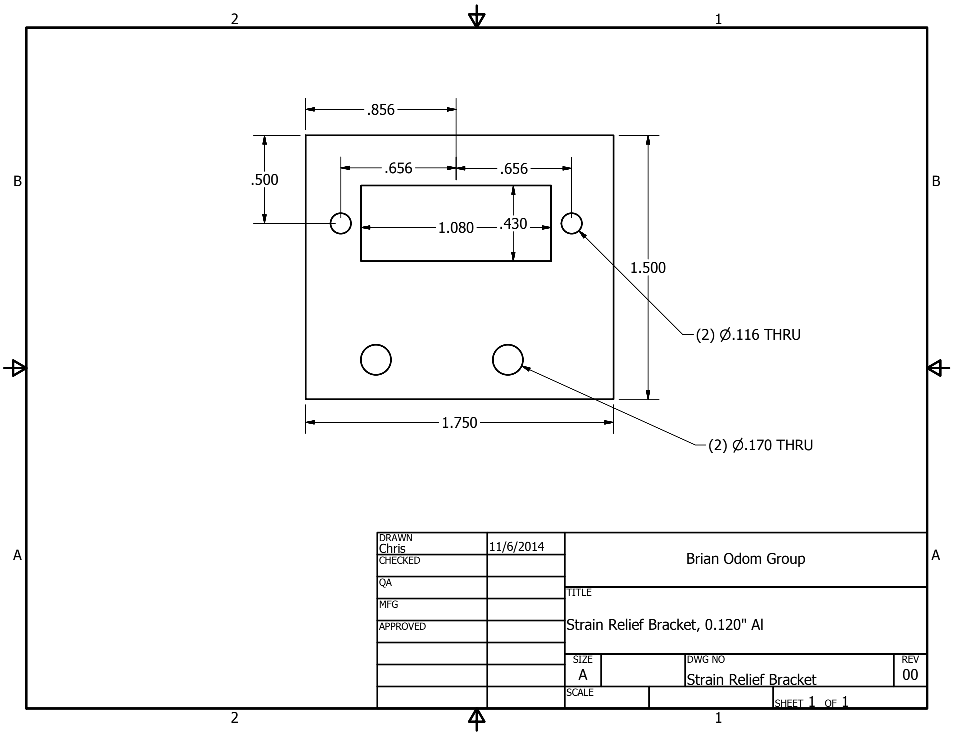
This encloses the AdvR SHG module. The module itself sits on a modified ECDL heatsink, clamped in place with heatsink brackets, to place output of the unit at the proper height. The two 1/4-20 taps on the floor mount a harmonic-separating dichroic with the fundamental being transmitted. The SHG box sides contain SM1-threaded holes to place AR-coated windows with zero-aperture irises as well as the input fiber mount and coupling lens.

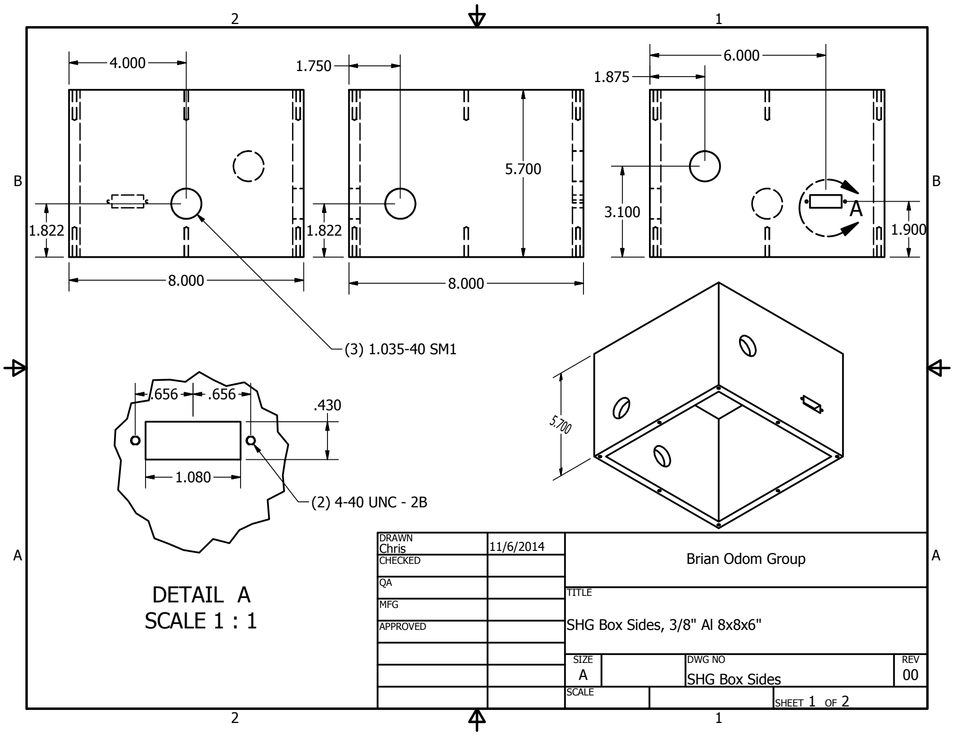



2
1
6.100
5.000
4.725
2.225
2.375
3.550
5.600
6.050
B
(2) 1/4-20 UNC - 2B
(4) 8-32 UNC - 2B
TOP VIEW
A
DRAWN
Chris
11/6/2014
CHECKED
QA
MFG
APPROVED
Brian Odom Group
TITLE
SHG Box Floor, 3/8" Al
SIZE
A
DWG NO
SHG Box Floor
REV
00
SCALE
SHEET 2 OF 2


2
1
7.813
4.000
.188
.188
4.000
7.813
B
(8) ∅.170 THRU
8.000
8.000
A
DRAWN
Chris
11/6/2014
CHECKED
QA
MFG
APPROVED
Brian Odom Group
TITLE
SHG Box Lid, 1/8" Al
SIZE
A
DWG NO
SHG Box Lid
REV
00
SCALE
SHEET 1 OF 1


Base is an ECDL heatsink w/o water.
Modifications are 2-56 and 8-32 tapped holes.
1.200
.450
.250
.500
(4) 8-32 UNC - 2B
↧ .500
1.825
.075
.275
.625
(4) 2-56 UNC - 2B
↧ .500
2
1
B
A
DRAWN
Chris
11/6/2014
CHECKED
QA
MFG
APPROVED
Brian Odom Group
TITLE
SHG Solid Heatsink, Al
SIZE
A
DWG NO
SHG Solid Heatsink
REV
00
SCALE
SHEET 1 OF 1


.856
.656
.656
.500
1.080
.430
1.500
(2) Ø.116 THRU
1.750
(2) Ø.170 THRU
2
1
B
A
DRAWN
Chris
11/6/2014
CHECKED
QA
MFG
APPROVED
Brian Odom Group
TITLE
Strain Relief Bracket, 0.120" Al
SIZE
A
DWG NO
Strain Relief Bracket
REV
00
SCALE
SHEET 1 OF 1



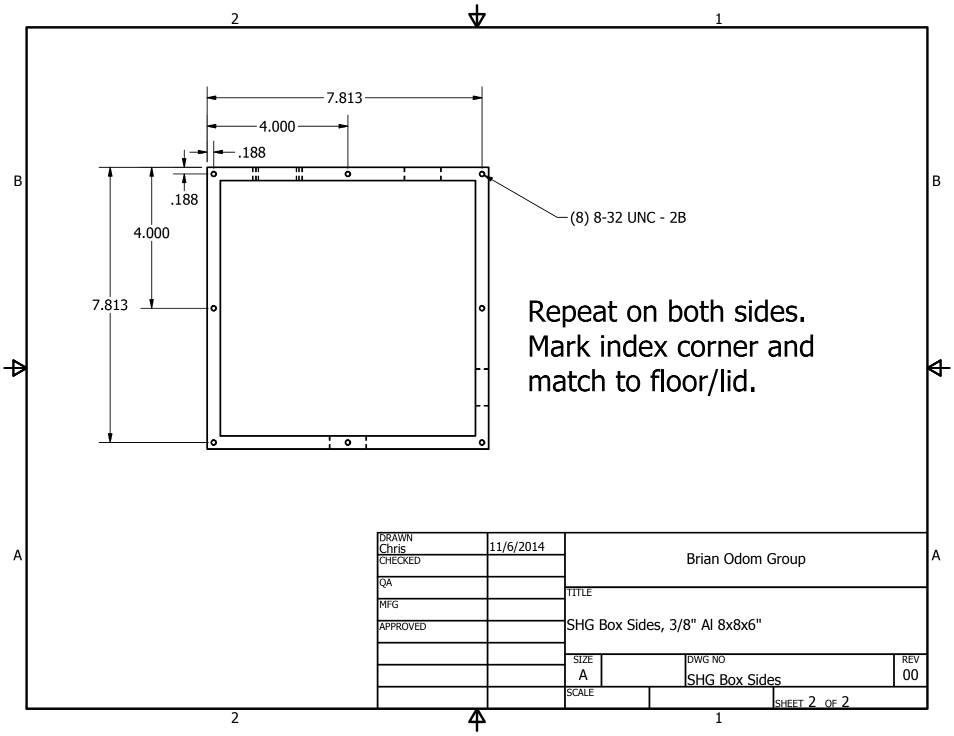
DRAWN Chris	11/6/2014
CHECKED	
QA	
MFG	
APPROVED	

Brian Odom Group

TITLE
SHG Box Sides, 3/8" Al 8x8x6"

SIZE A | DWG NO SHG Box Sides | REV 00

SCALE | SHEET 1 OF 2



Repeat on both sides.
Mark index corner and
match to floor/lid.

DRAWN Chris	11/6/2014
CHECKED	
QA	
MFG	
APPROVED	

Brian Odom Group

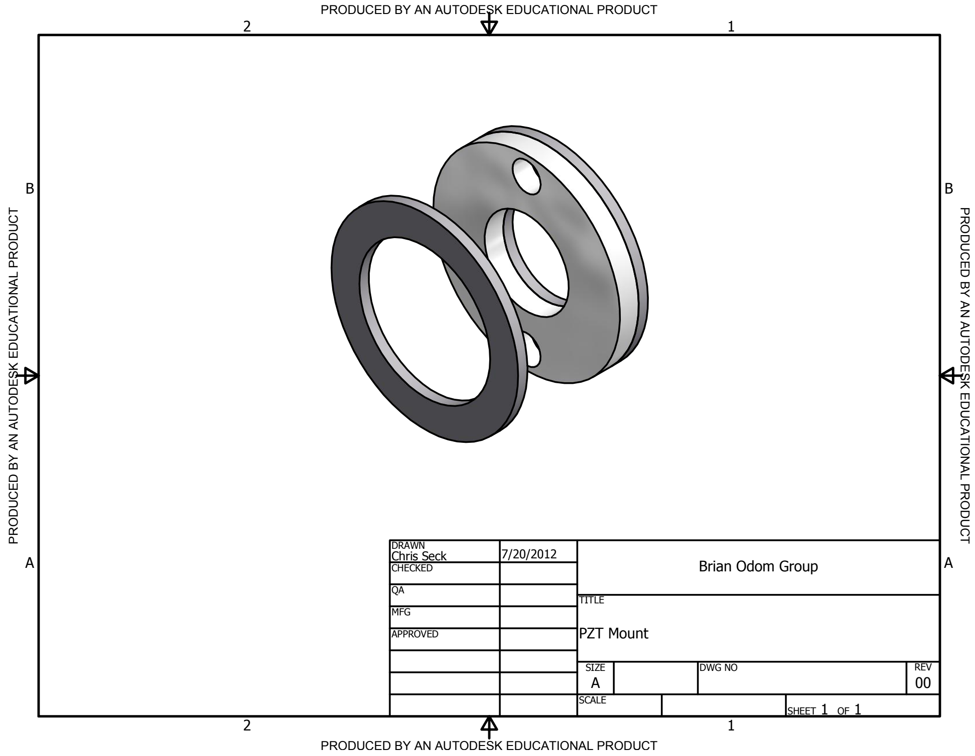
TITLE
SHG Box Sides, 3/8" Al 8x8x6"

SIZE A | DWG NO SHG Box Sides | REV 00

SCALE | SHEET 2 OF 2

F.1.6. Transfer Cavity Mounts

An o-ring is placed behind the first mirror support, then the mirror, PZT element, PZT endcap, and endcap support. The holes in the latter two are for the PZT wire leads.

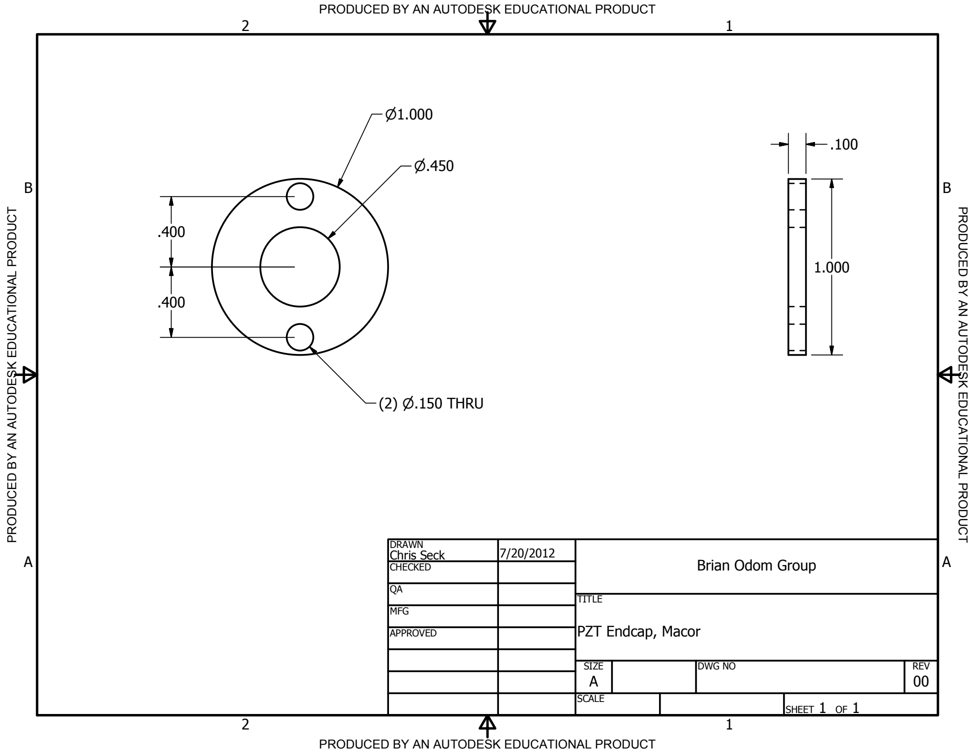

PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT
2
1
B
B
A
A
2
1
PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT
PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT
PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT

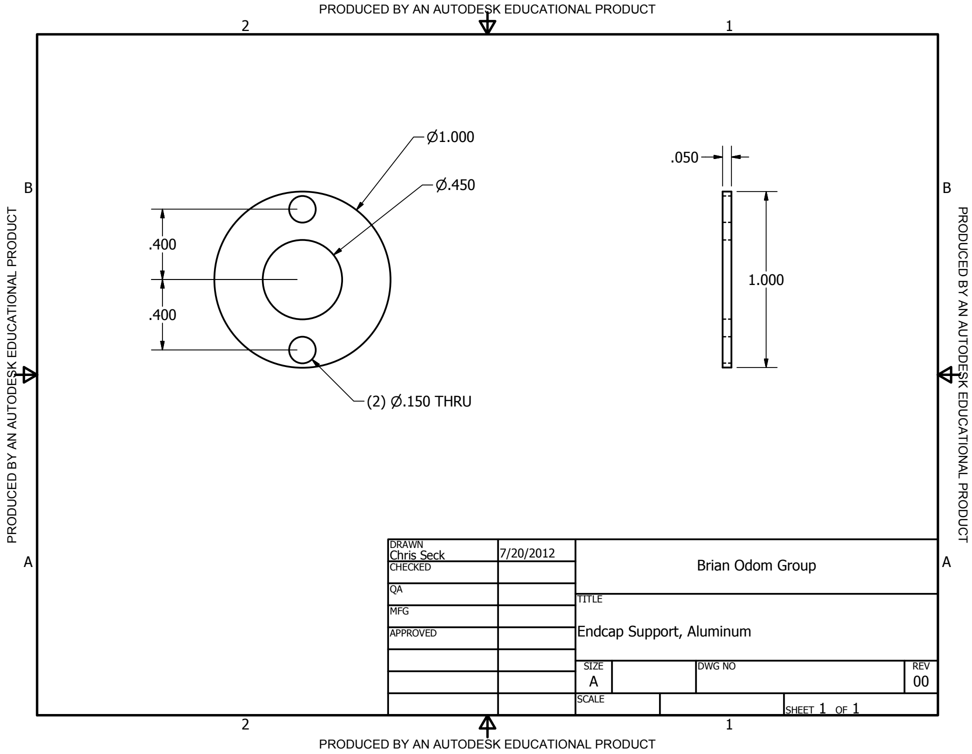
DRAWN Chris Seck 7/20/2012
CHECKED
QA
MFG
APPROVED

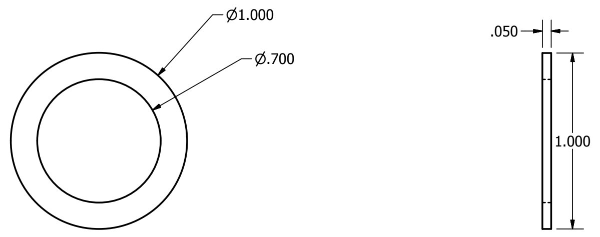
Brian Odom Group

TITLE PZT Mount

SIZE A
DWG NO
REV 00
SCALE
SHEET 1 OF 1


PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT
2
1
Ø1.000
Ø.450
.100
.400
.400
1.000
(2) Ø.150 THRU
B
A
DRAWN
Chris Seck
7/20/2012
CHECKED
QA
MFG
APPROVED
Brian Odom Group
TITLE
PZT Endcap, Macor
SIZE
A
DWG NO
REV
00
SCALE
SHEET 1 OF 1


PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT
2
1
Ø1.000
Ø.450
.050
.400
.400
1.000
(2) Ø.150 THRU
B
A
DRAWN
Chris Seck
7/20/2012
CHECKED
QA
MFG
APPROVED
Brian Odom Group
TITLE
Endcap Support, Aluminum
SIZE
A
DWG NO
REV
00
SCALE
SHEET 1 OF 1



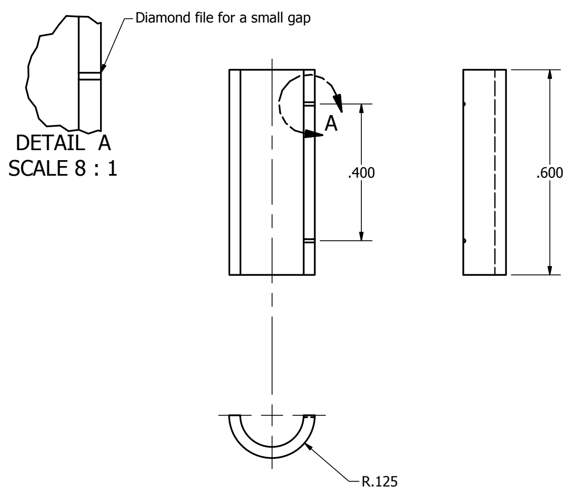
DRAWN Chris Seck	7/20/2012	Brian Odom Group			
CHECKED					
QA		TITLE Mirror Support, Aluminum			
MFG					
APPROVED					
		SIZE A		DWG NO	REV 00
		SCALE		SHEET 1 OF 1	

F.2. AlH$^+$ Rotational Cooling Project

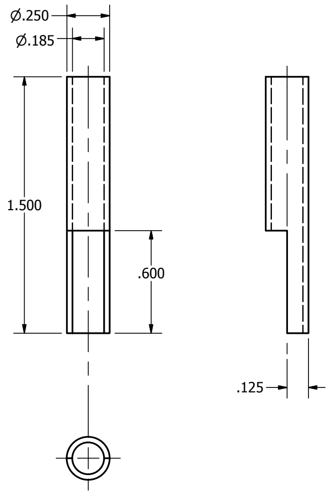
F.2.1. Alumina Nozzle

Original AlH neutral source design with tungsten-wire feedthrus. Design by CYL.

Al Oven Cap		Filename: Al Oven Cap.ipt
Units: inches/degree	Tolerances: +/-0.001	Chien-Yu Lien Chien-Yu.Lien@u.northwestern.edu Tech FB31 / 7-1959 Date: 4/5/2011
Quantity: 2 Material: Alumina Page: 1 / 1		

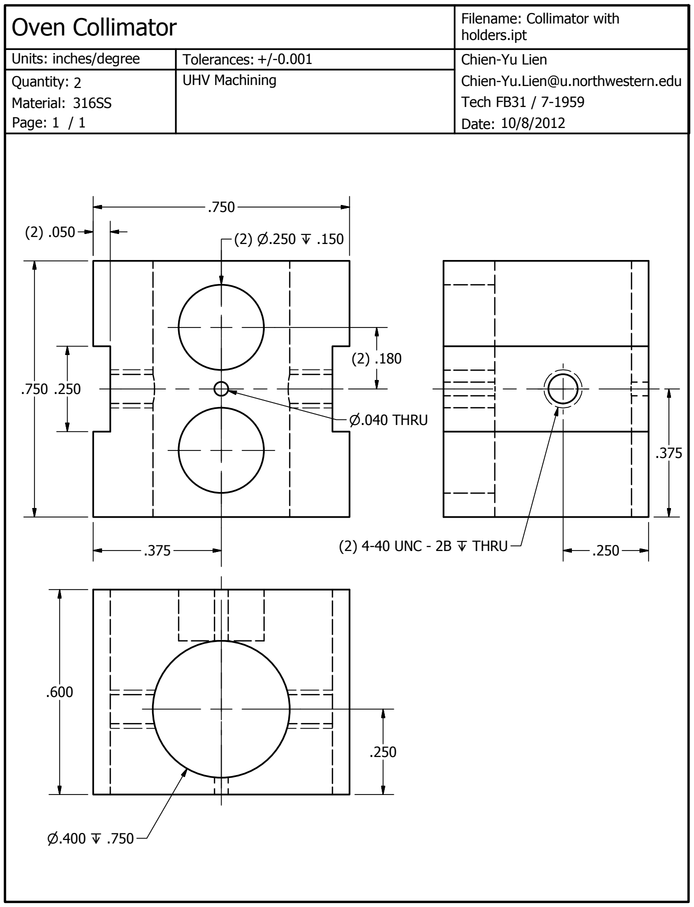


Al Oven		Filename: Al Oven.ipt
Units: inches/degree	Tolerances: +/-0.001	Chien-Yu Lien
Quantity: 2 Material: Alumina Page: 1 / 1		Chien-Yu.Lien@u.northwestern.edu Tech FB31 / 7-1959 Date: 4/5/2011



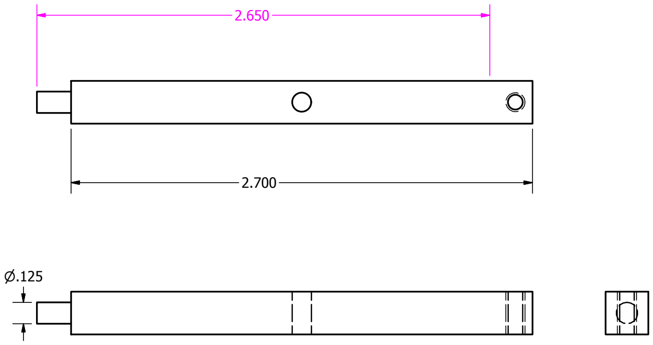
F.2.2. Barium Oven Skimmer

This is attached to the vacuum chamber using two groove-grabbers on the bottom octagon port. Design by CYL.



Oven Collimator		Filename: Collimator with holders.ipt
Units: inches/degree	Tolerances: +/-0.001	Chien-Yu Lien
Quantity: 2 Material: 316SS Page: 1 / 1	UHV Machining	Chien-Yu.Lien@u.northwestern.edu Tech FB31 / 7-1959 Date: 10/8/2012

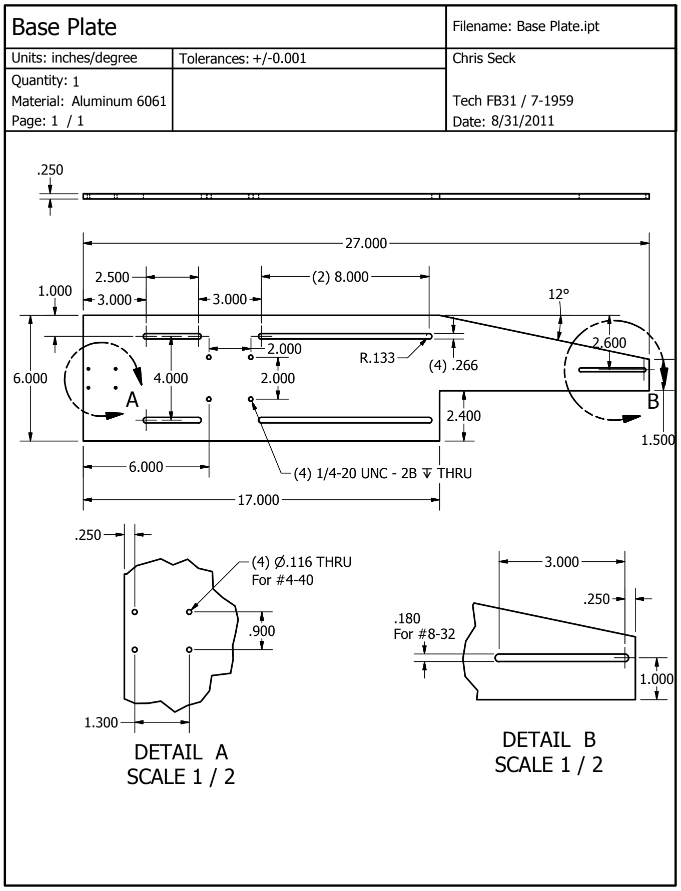
Trap Support		Filename: Trap Support v2.ipt
Units: inches/degree	Tolerances: +/- 0.001	Chien-Yu Lien
Quantity: 2 Material: 316ss Page: 1 / 1	UHV machining	chien-Yu.Lien@u.northwestern.edu Tech FB31 / 7-1959 Date: 6/7/2011



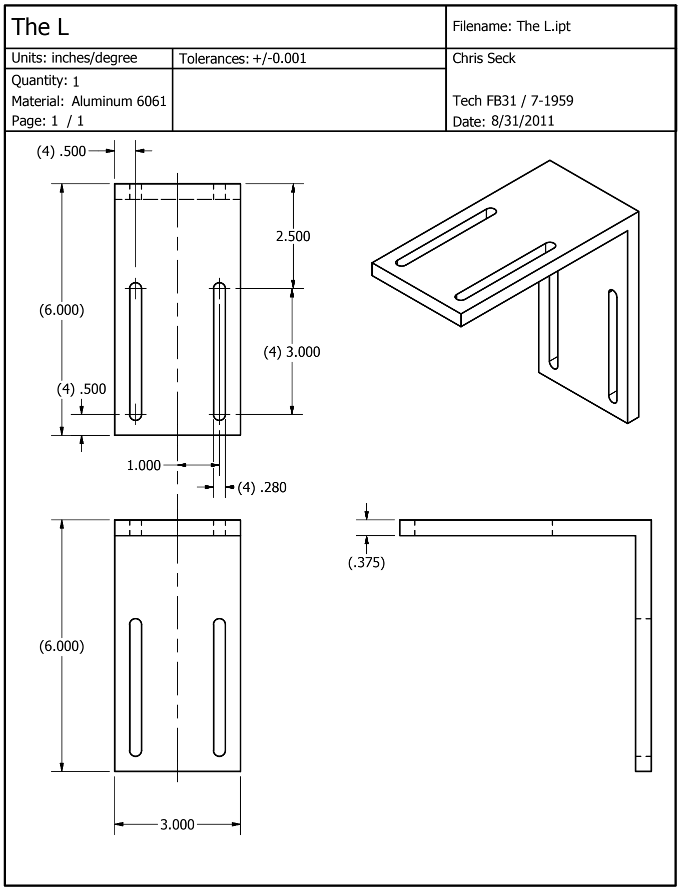
F.2.3. Camera Mount

The pieces below provided support and mounting of the camera optical system used in the AlH$^+$ experiment.

F.2.3.1. Optics Support Plate. This plate supports the entire optical system including camera.



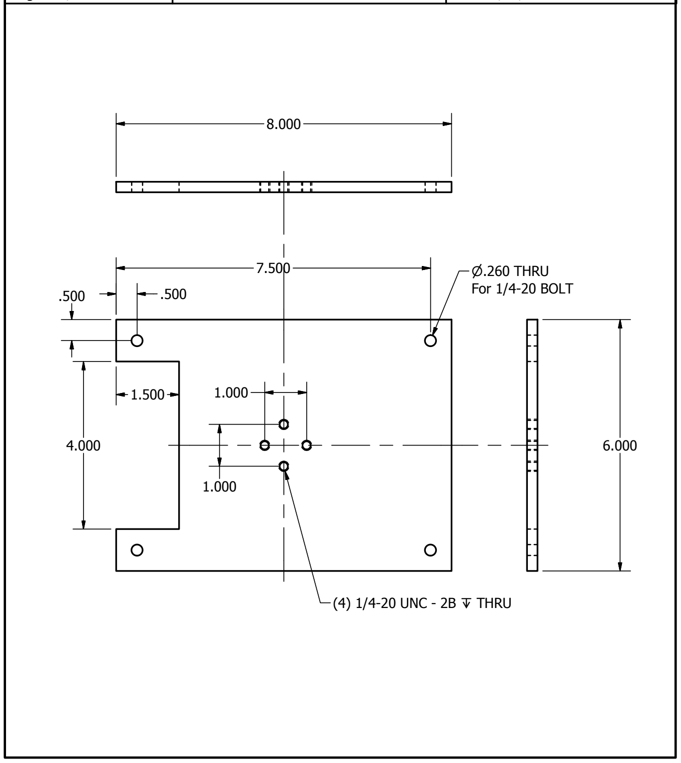
F.2.3.2. Angle Bracket. This bracket mounts the optical support plate to the translation stage platform.



The L		Filename: The L.ipt
Units: inches/degree	Tolerances: +/-0.001	Chris Seck
Quantity: 1 Material: Aluminum 6061 Page: 1 / 1		Tech FB31 / 7-1959 Date: 8/31/2011

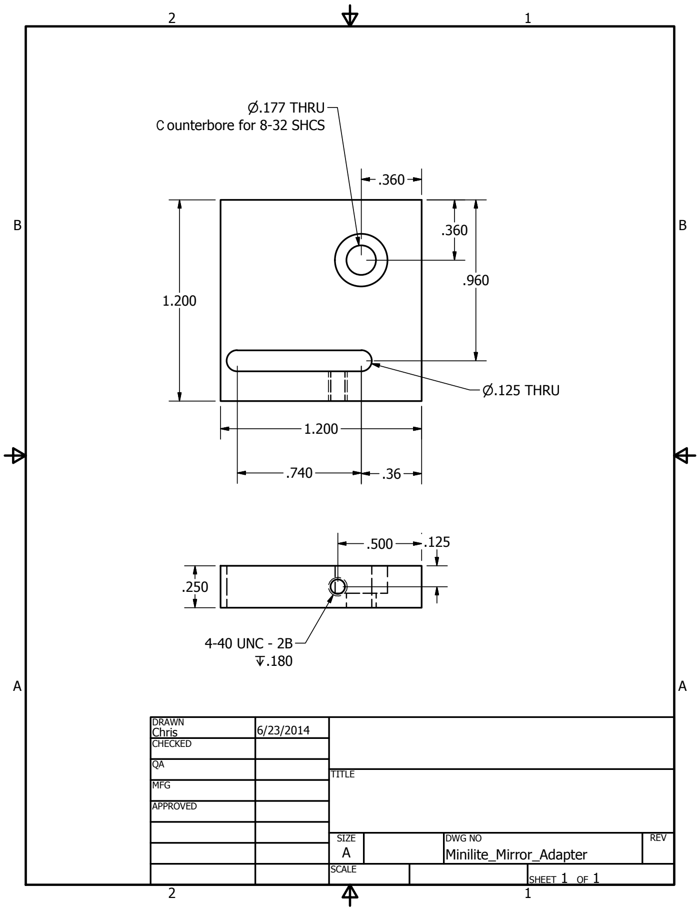
F.2.3.3. Translation Stage Platform. The XYZ translation stage mounts to this piece, which is supported by four 1'-inch-diameter aluminum posts to position the camera at the correct height above the optical table.

3D Base		Filename: 3D Base.ipt
Units: inches/degree	Tolerances: +/-0.001	Chris Seck
Quantity: 1 Material: Aluminum 6061 Page: 1 / 1		Tech FB31 / 7-1959 Date: 8/31/2011

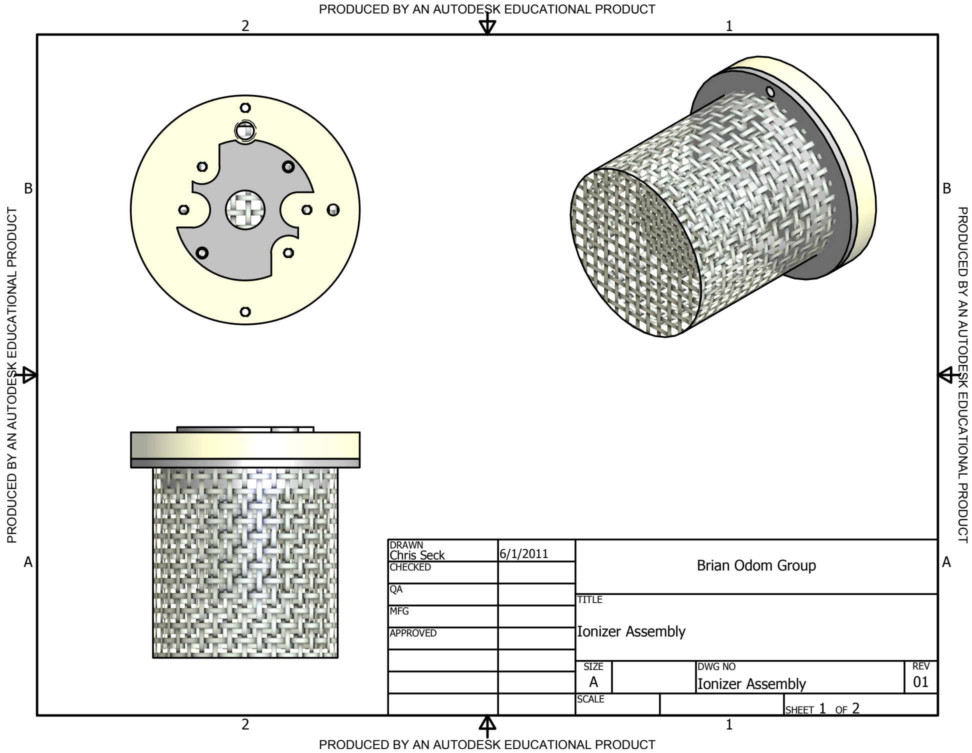


F.2.4. Continuum Minilite Mirror Mount Adapter

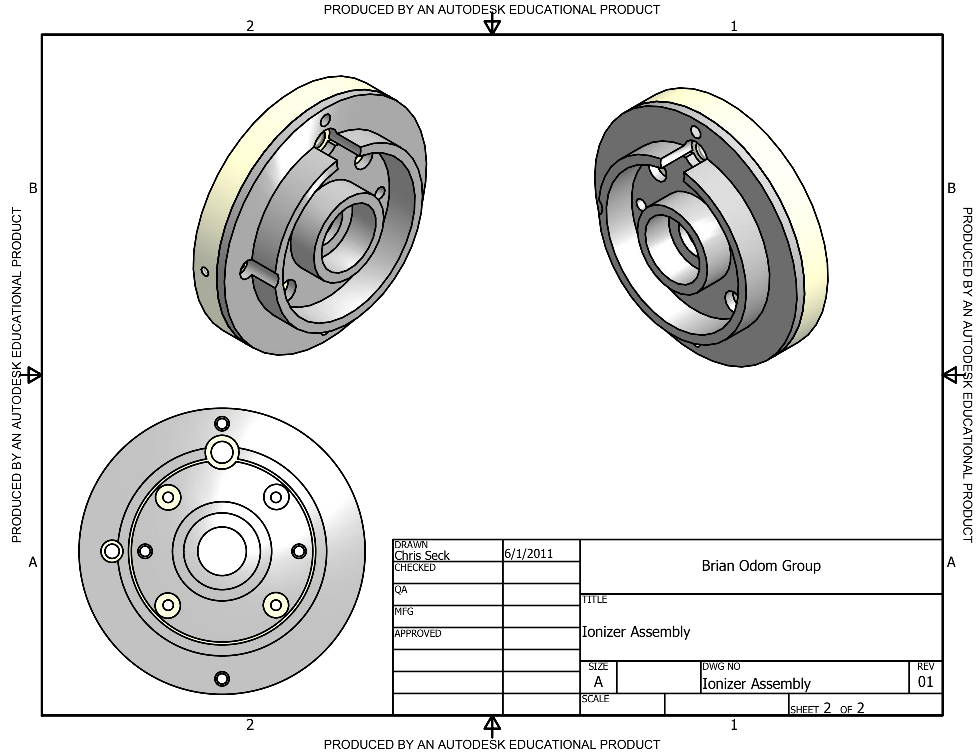
The part below permits a ThorLabs KM05 kinematic mirror mount to bolt to the Continuum Minilite laser head to permit dual-harmonic operation of the laser system.

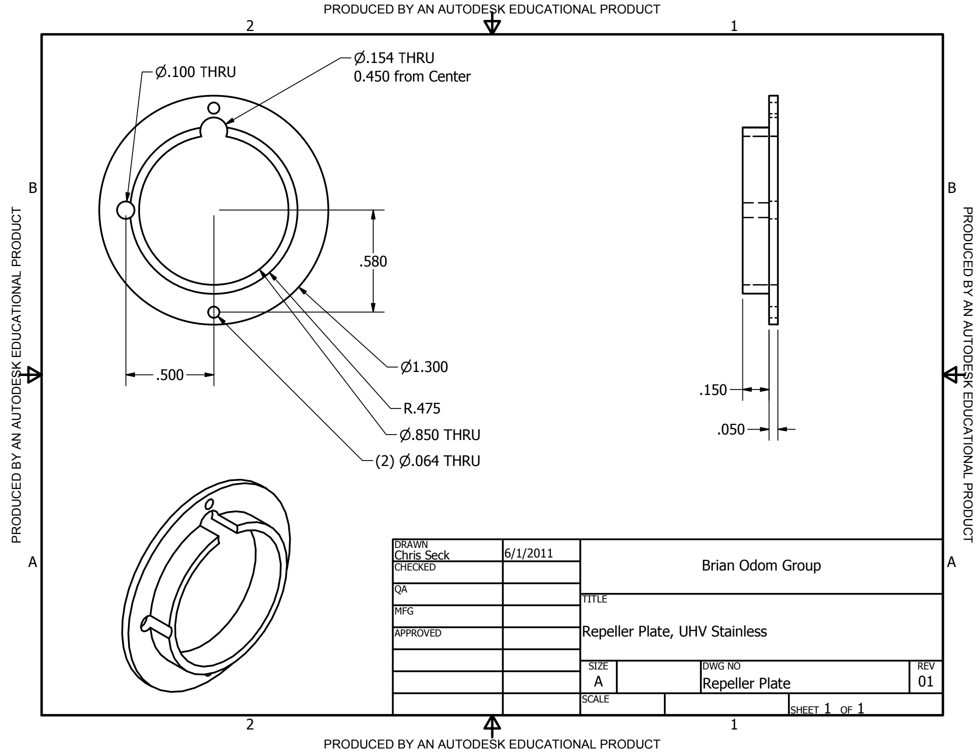


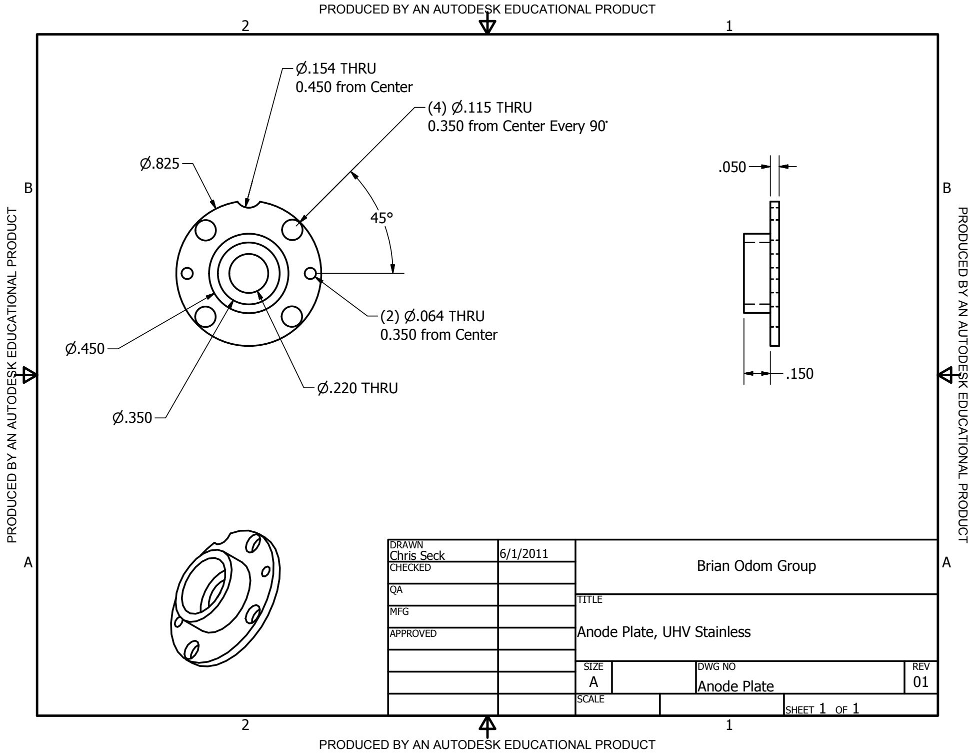
F.2.5. Electron Impact Ionizer


PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT
2
1
B
B
A
A
2
1
PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT
PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT
PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT

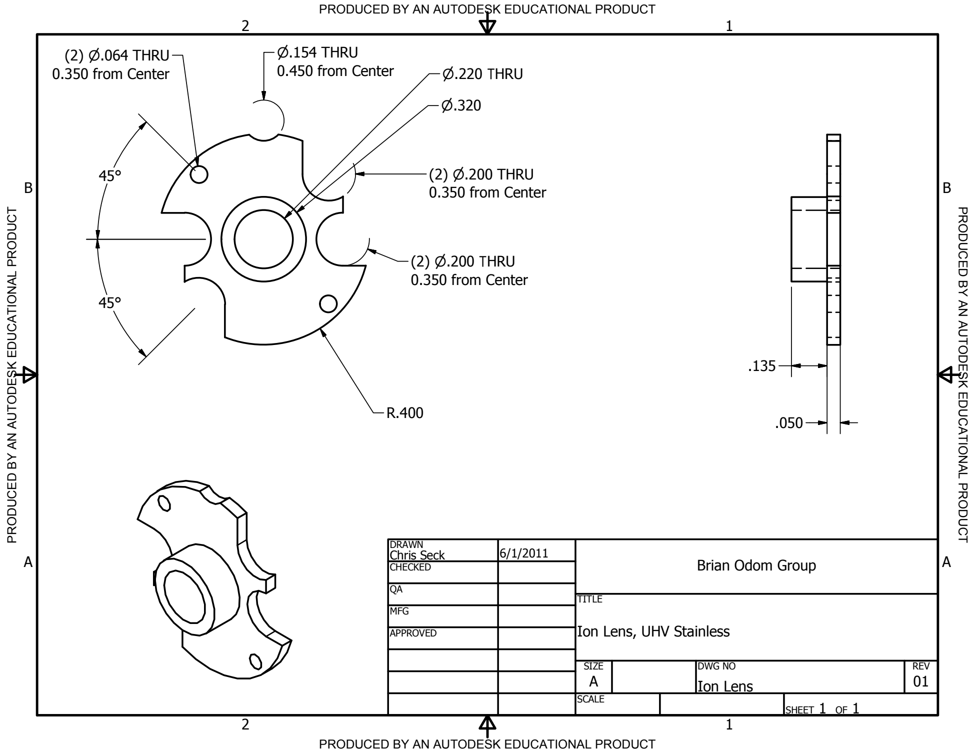
DRAWN Chris Seck	6/1/2011	Brian Odom Group
CHECKED		
QA		TITLE
MFG		Ionizer Assembly
APPROVED		
		SIZE A · DWG NO Ionizer Assembly · REV 01
		SCALE · SHEET 1 OF 2


PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT
2
1
B
A
DRAWN
Chris Seck
6/1/2011
CHECKED
QA
MFG
APPROVED
Brian Odom Group
TITLE
Ionizer Assembly
SIZE
A
DWG NO
Ionizer Assembly
REV
01
SCALE
SHEET 2 OF 2

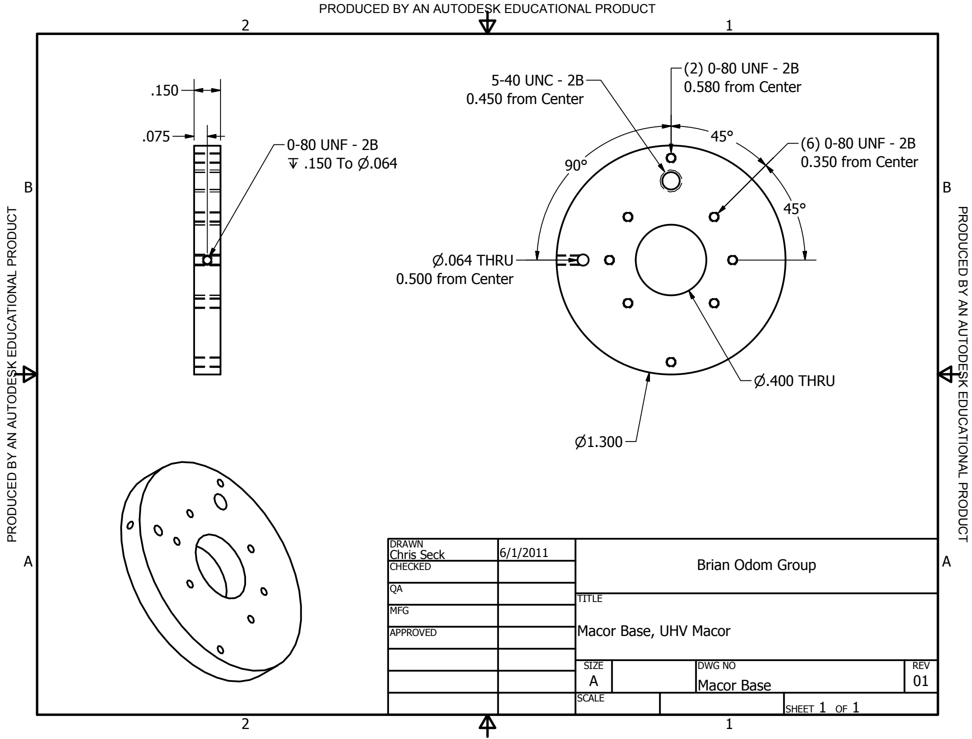

PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT
2
1
B
A
Ø.100 THRU
Ø.154 THRU
0.450 from Center
.580
.500
Ø1.300
R.475
Ø.850 THRU
(2) Ø.064 THRU
.150
.050
DRAWN
Chris Seck
6/1/2011
CHECKED
QA
MFG
APPROVED
Brian Odom Group
TITLE
Repeller Plate, UHV Stainless
SIZE
A
DWG NO
Repeller Plate
REV
01
SCALE
SHEET 1 OF 1

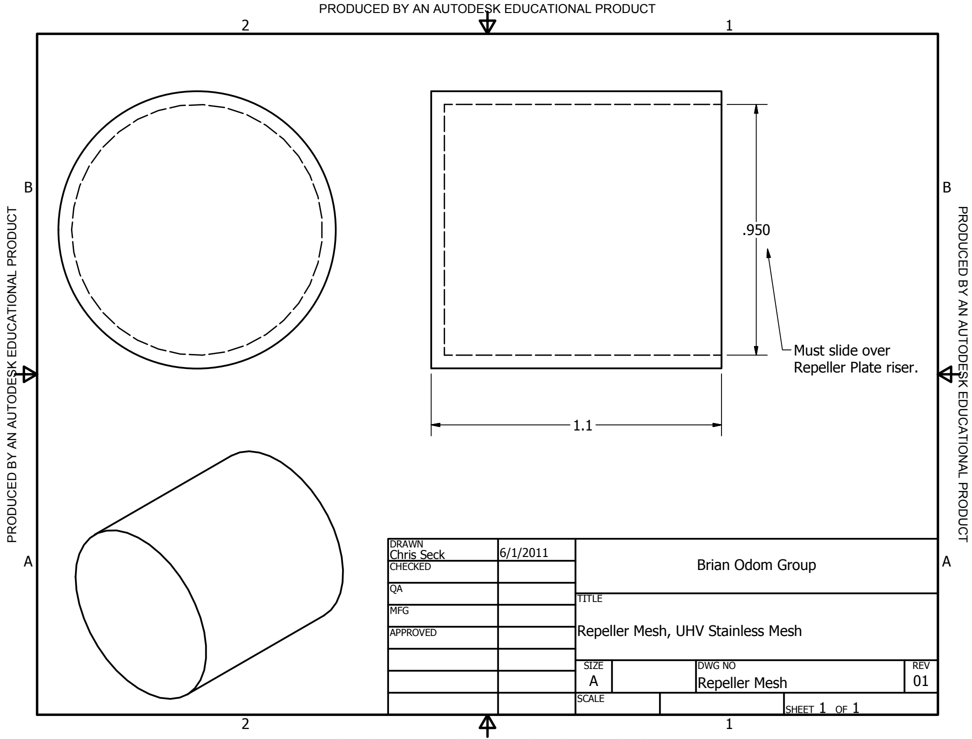


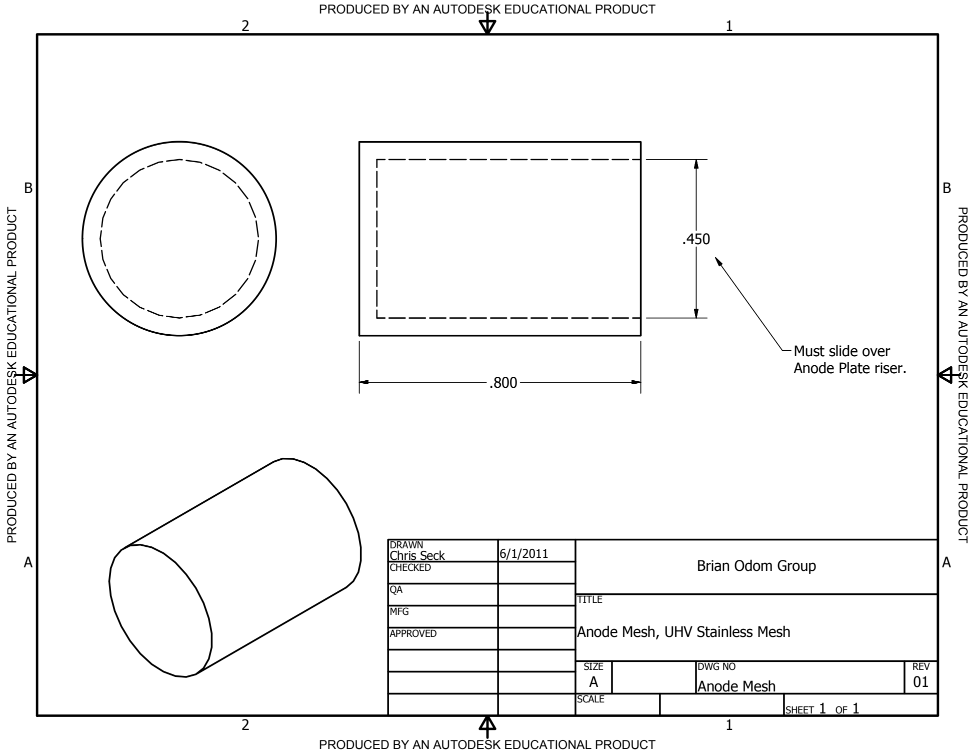
DRAWN Chris Seck	6/1/2011	Brian Odom Group
CHECKED		
QA		TITLE
MFG		Anode Plate, UHV Stainless
APPROVED		
SIZE A	DWG NO Anode Plate	REV 01
SCALE		SHEET 1 OF 1



DRAWN Chris Seck	6/1/2011	Brian Odom Group
CHECKED		
QA		TITLE
MFG		Ion Lens, UHV Stainless
APPROVED		
SIZE A	DWG NO Ion Lens	REV 01
SCALE		SHEET 1 OF 1


2
1
B
A
.150
.075
0-80 UNF - 2B
↧ .150 To ∅.064
5-40 UNC - 2B
0.450 from Center
(2) 0-80 UNF - 2B
0.580 from Center
45°
(6) 0-80 UNF - 2B
0.350 from Center
90°
45°
∅.064 THRU
0.500 from Center
∅.400 THRU
∅1.300
DRAWN
Chris Seck
6/1/2011
CHECKED
QA
MFG
APPROVED
Brian Odom Group
TITLE
Macor Base, UHV Macor
SIZE
A
DWG NO
Macor Base
REV
01
SCALE
SHEET 1 OF 1


2
1
B
A
.950
Must slide over
Repeller Plate riser.
1.1
DRAWN
Chris Seck
6/1/2011
CHECKED
QA
MFG
APPROVED
Brian Odom Group
TITLE
Repeller Mesh, UHV Stainless Mesh
SIZE
A
DWG NO
Repeller Mesh
REV
01
SCALE
SHEET 1 OF 1



DRAWN Chris Seck 6/1/2011
CHECKED
QA
MFG
APPROVED

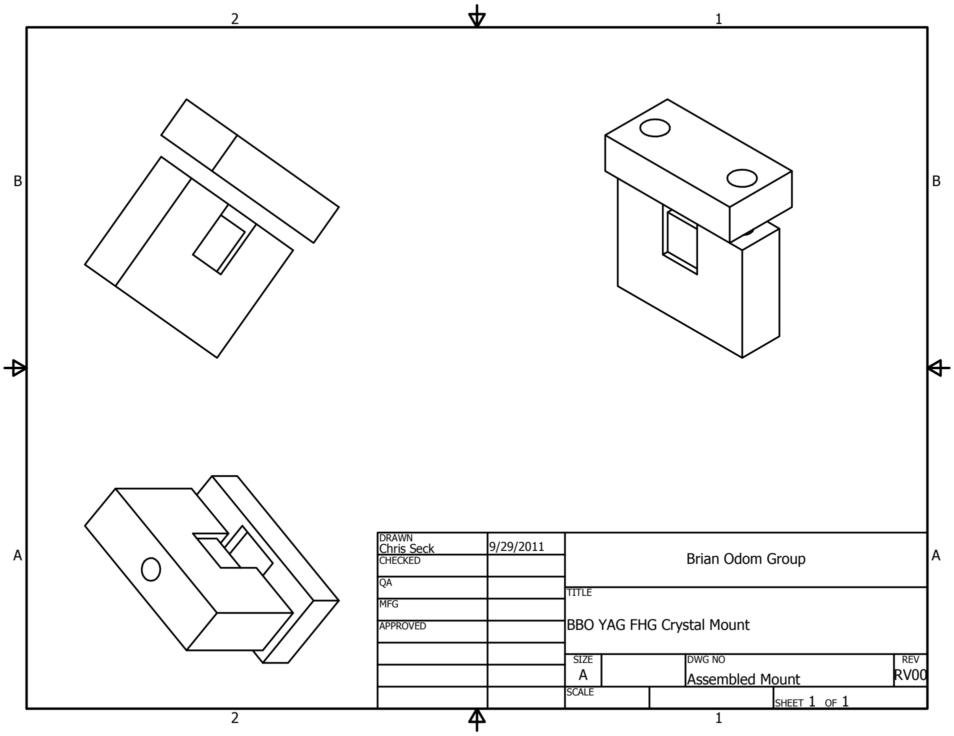
Brian Odom Group

TITLE
Anode Mesh, UHV Stainless Mesh

SIZE A
DWG NO Anode Mesh
REV 01
SCALE
SHEET 1 OF 1

F.2.6. FHG Crystal Mount

This is used for 266 nm generation before the Continuum Minilite modifications are performed by splitting some of the 532 nm power from the dye laser pump. A piece of springed-steel (or similar) is placed in the lid pocket such that the BBO Nd:YAG FHG crystal is not overconstrained. The mount should be placed on at least a single-axis rotation stage.



DRAWN Chris Seck 9/29/2011

CHECKED

QA

MFG

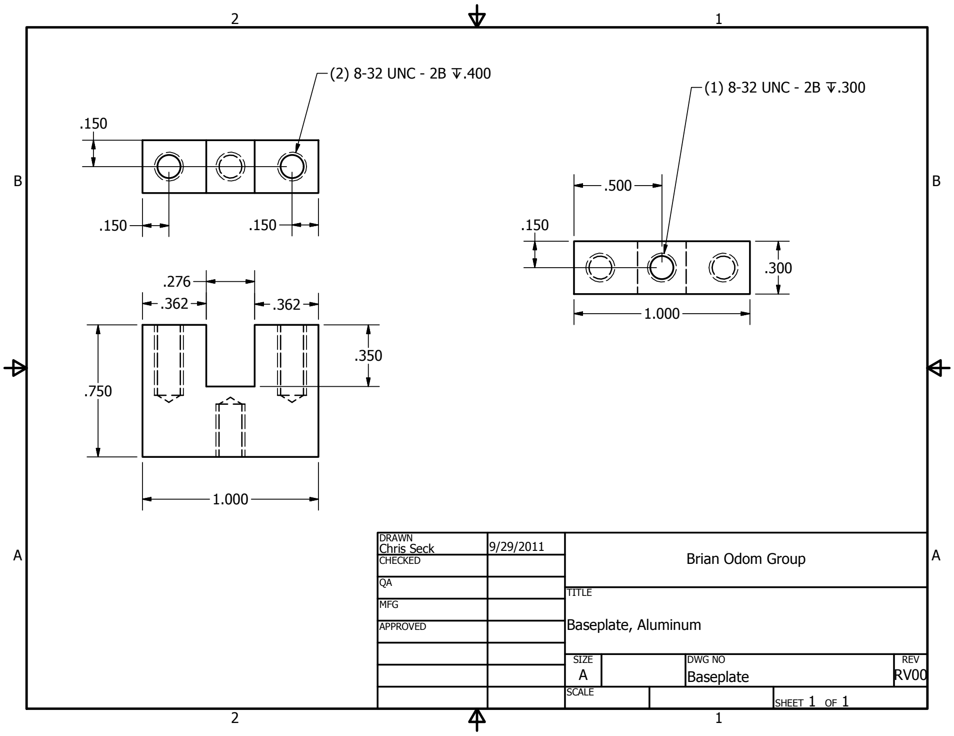
APPROVED

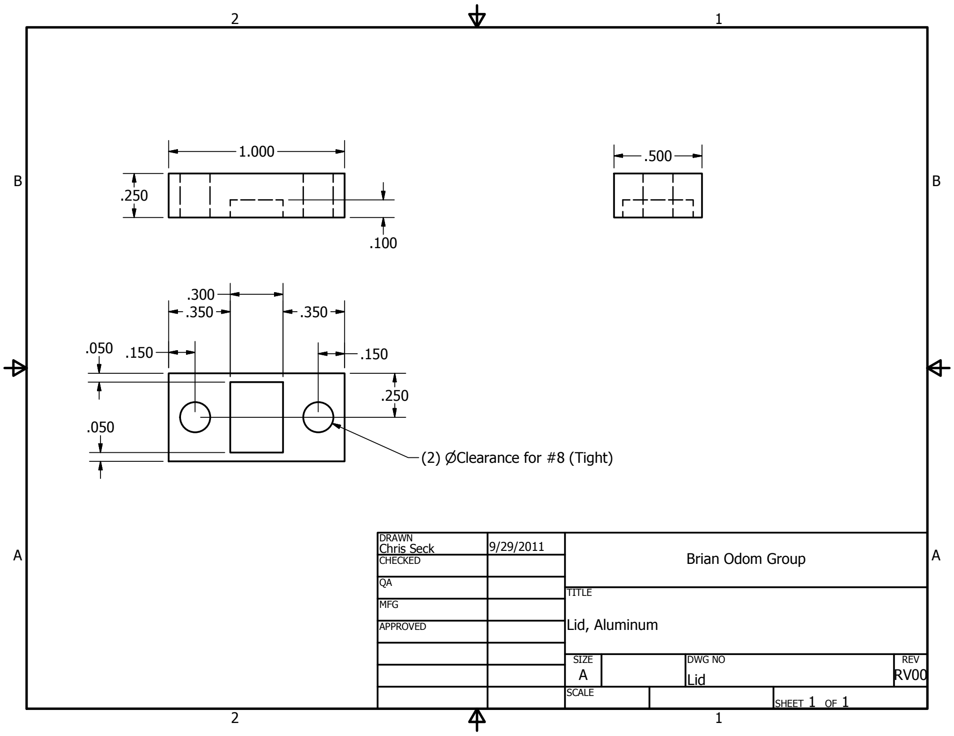
Brian Odom Group

TITLE BBO YAG FHG Crystal Mount

SIZE A DWG NO Assembled Mount REV RV00

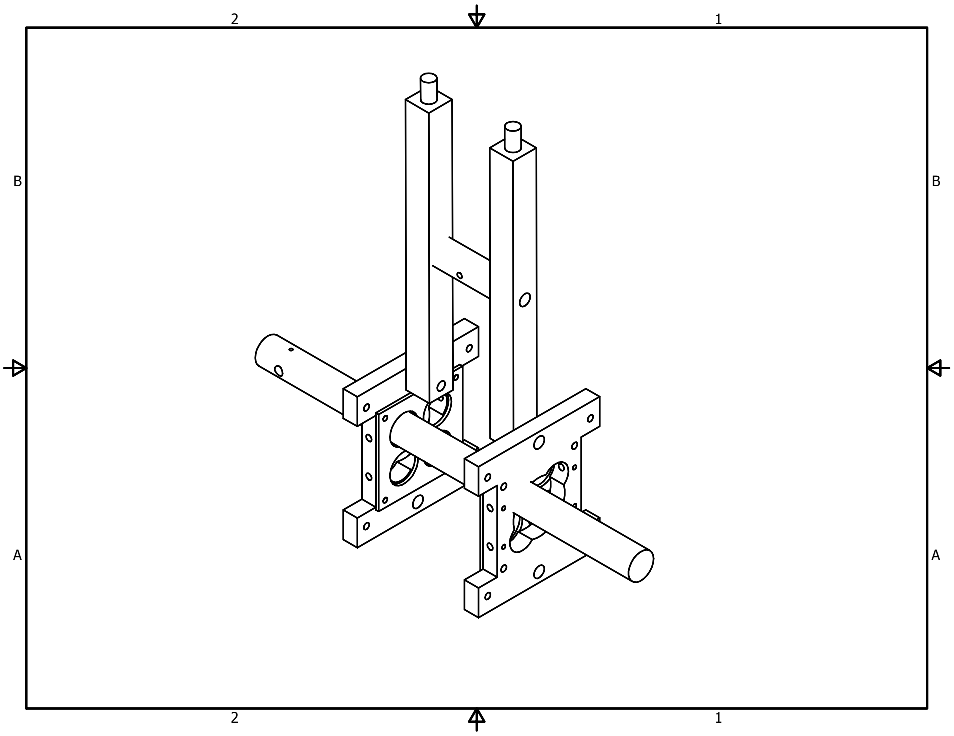
SCALE SHEET 1 OF 1


(2) 8-32 UNC - 2B ↧.400
(1) 8-32 UNC - 2B ↧.300
.150
.150
.150
.500
.150
.300
1.000
.276
.362
.362
.350
.750
1.000
2
1
B
A
DRAWN
Chris Seck
9/29/2011
CHECKED
QA
MFG
APPROVED
Brian Odom Group
TITLE
Baseplate, Aluminum
SIZE
A
DWG NO
Baseplate
REV
RV00
SCALE
SHEET 1 OF 1

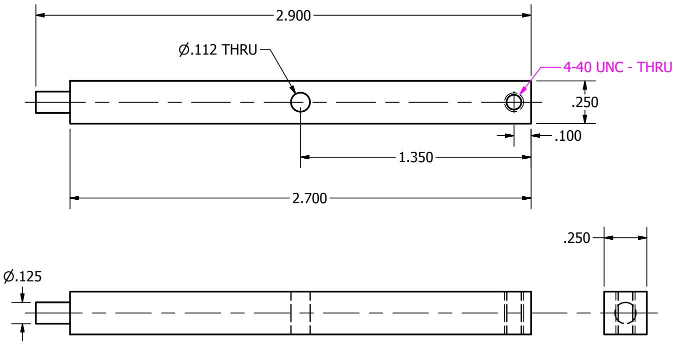

1.000
.500
.250
.100
.300
.350
.350
.050
.150
.150
.250
.050
(2) ∅Clearance for #8 (Tight)
2
1
B
A
DRAWN
Chris Seck
9/29/2011
CHECKED
QA
MFG
APPROVED
Brian Odom Group
TITLE
Lid, Aluminum
SIZE
A
DWG NO
Lid
REV
RV00
SCALE
SHEET 1 OF 1

F.2.7. Ion Trap and Time-of-flight Mass Spectrometry

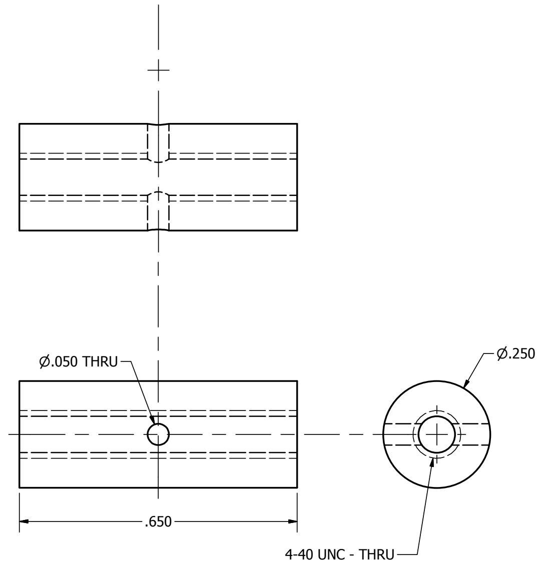
The ion trap is attached to the vacuum chamber using two groove-grabbers on the top octagon port. The time-of-flight components are attached to the vacuum chamber using two groove-grabbers on the time-of-flight-mass-spectrometer-side octagon port. Design by CYL.



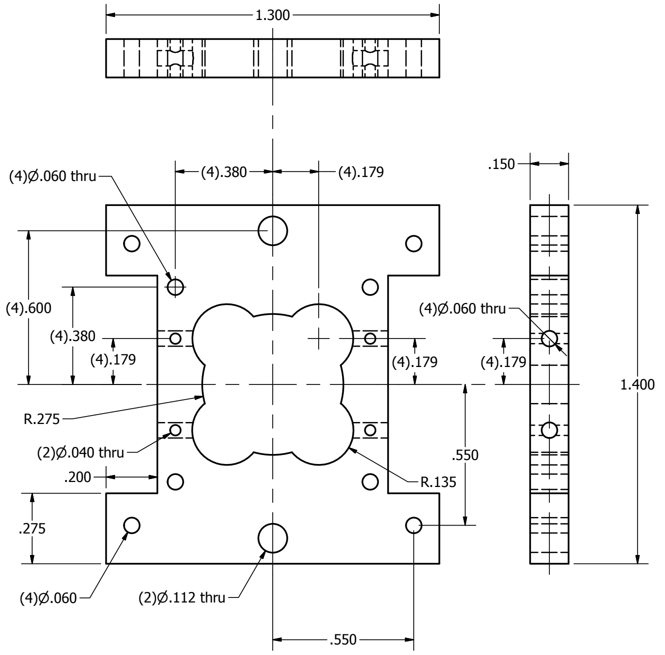
Trap Support		Filename: Trap Support v2.ipt
Units: inches/degree	Tolerances: +/- 0.001	Chien-Yu Lien
Quantity: 9 Material: 316ss Page: 1 / 1	UHV machining	chien-Yu.Lien@u.northwestern.edu Tech FB31 / 7-1959 Date: 2/5/2011



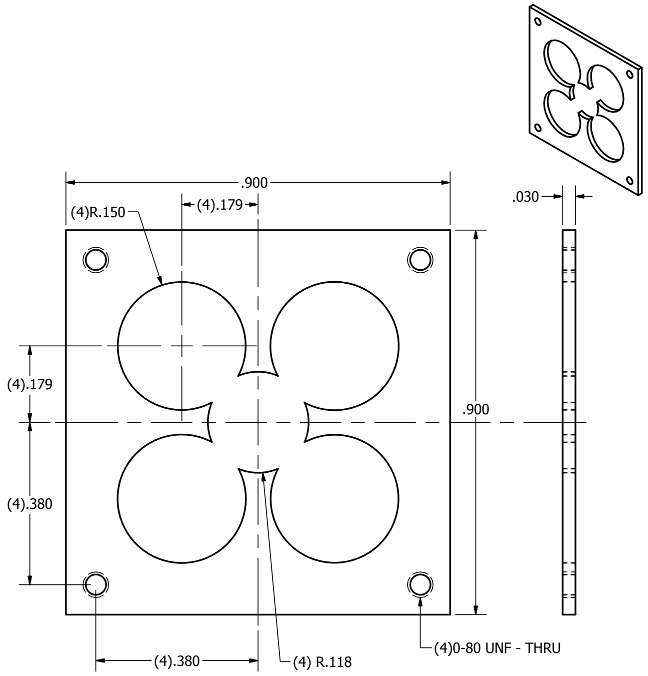
Support Link		Filename: Support link.ipt
Units: inches/degree	Tolerances: +/-0.001	Chien-Yu Lien
Quantity: 5 Material: 316ss Page: 1 / 1	UHV Machining	Chien-Yu.Lien@u.northwestern.edu Tech FB31 / 7-1959 Date: 2/5/2011



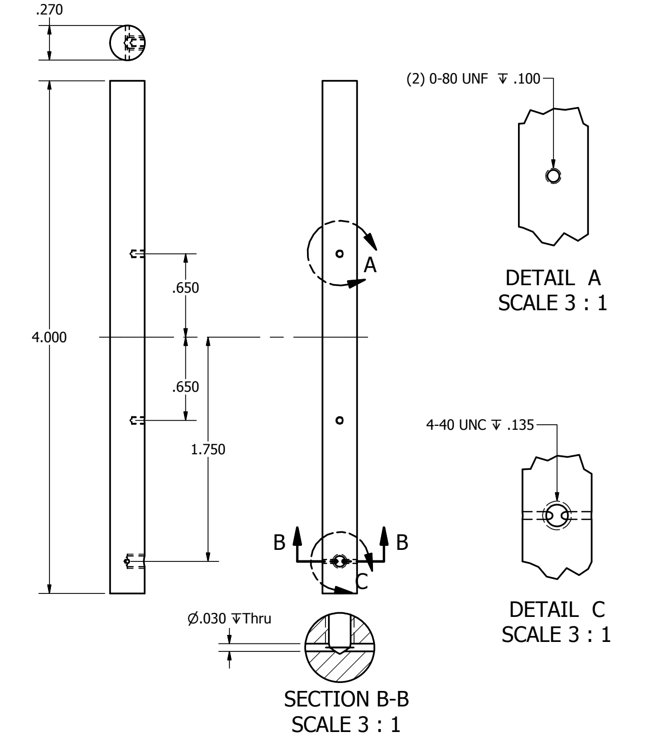
RFRod Mount		Filename: RFRod Mount.ipt
Units: inches/degree	Tolerances: +/-0.001	Chien-Yu Lien
Quantity: 5 Material: Macor Page: 1 / 1	UHV Machining	Chien-Yu.Lien@u.northwestern.edu Tech FB31 / 7-1959 Date: 2/5/2011

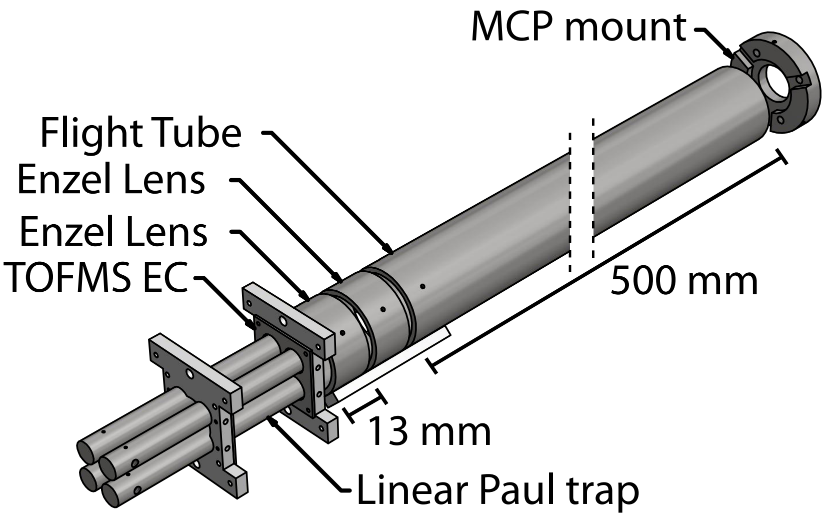


EndCap plate		Filename: Endcap v2.ipt
Units: inches/degree	Tolerances: +/-0.001	Chien-Yu Lien
Quantity: 5 Material: 316ss Page: 1 / 1	UHV Machining	Chien-Yu.Lien@u.northwestern.edu Tech FB31 / 7-1959 Date: 2/5/2011

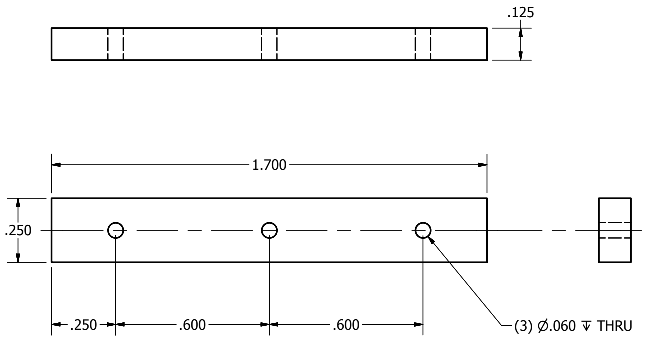


RF Rod		Filename: RFRod v2.ipt
Units: inches/degree	Tolerances: +/- 0.001	Chien-Yu Lien
Quantity: 9 Material: 316ss Page: 1 / 1	UHV Maching	Chien-Yu.Lien@u.northwestern.edu Tech FB31 / 7-1959 Date: 2/7/2011

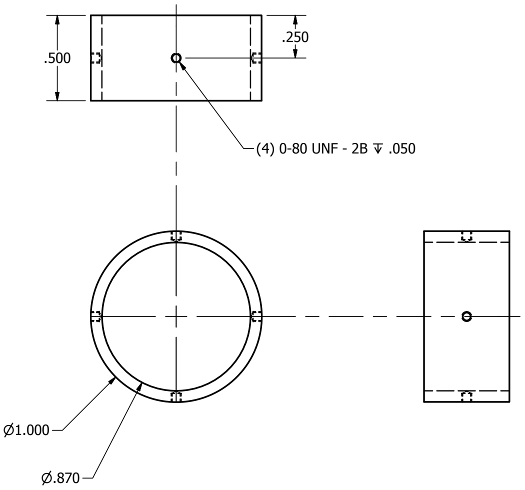



MCP mount
Flight Tube
Enzel Lens
Enzel Lens
TOFMS EC
500 mm
13 mm
Linear Paul trap

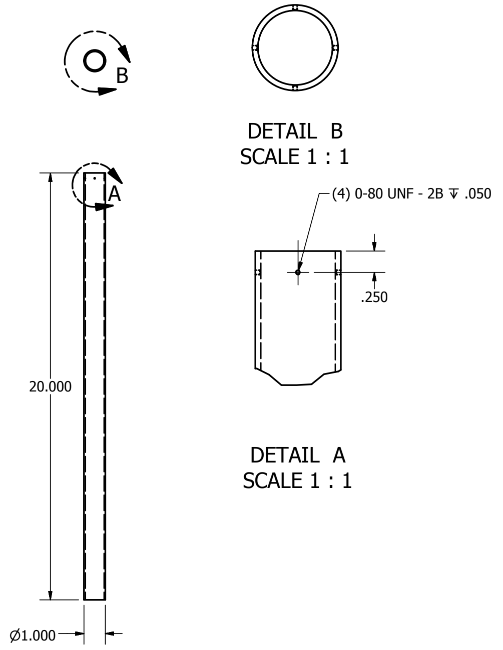
Lens Support		Filename: Lens support.ipt
Units: inches/degree	Tolerances: +/-0.001	Chien-Yu Lien
Quantity: 4 Material: Macor Page: 1 / 1	UHV Machining	Chien-Yu.Lien@u.northwestern.edu Tech FB31 / 7-1959 Date: 11/29/2012



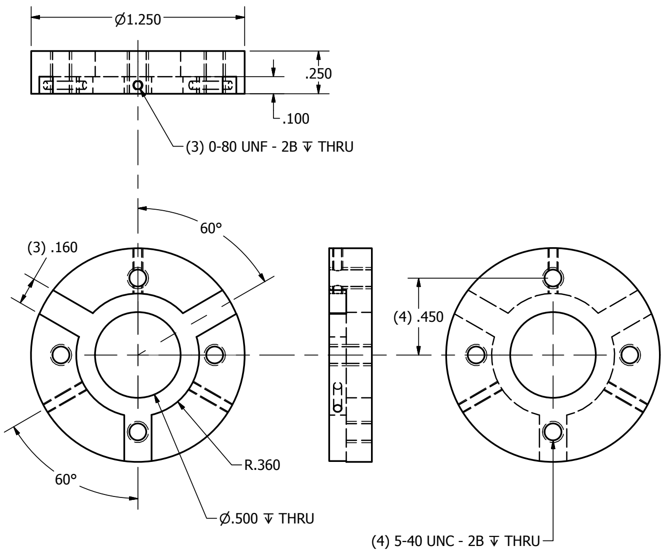
Enzel Lens		Filename: Enzel Lens.ipt
Units: inches/degree	Tolerances: +/-0.001	Chien-Yu Lien
Quantity: 2 Material: 316ss Page: 1 / 1	UHV Machining	Chien-Yu.Lien@u.northwestern.edu Tech FB31 / 7-1959 Date: 11/29/2012



Fllight tube 500		Filename: Flight tube 500.ipt
Units: inches/degree	Tolerances: +/-0.001	Chien-Yu Lien
Quantity: 1 Material: 316ss Page: 1 / 1	UHV Machining	Chien-Yu.Lien@u.northwestern.edu Tech FB31 / 7-1959 Date: 11/29/2012

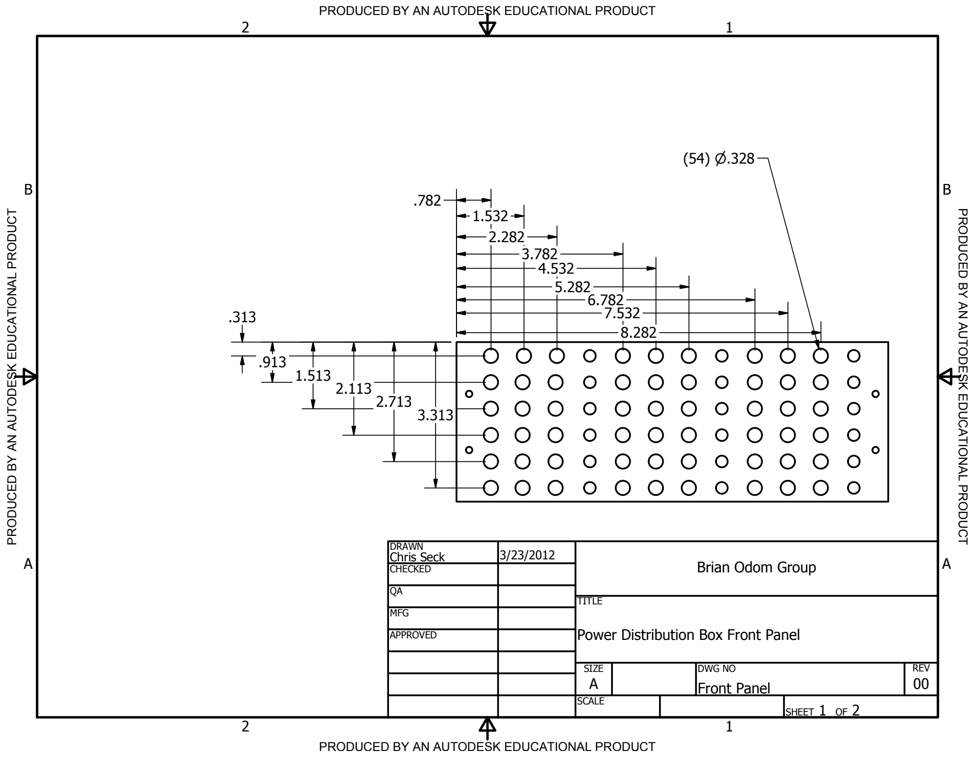


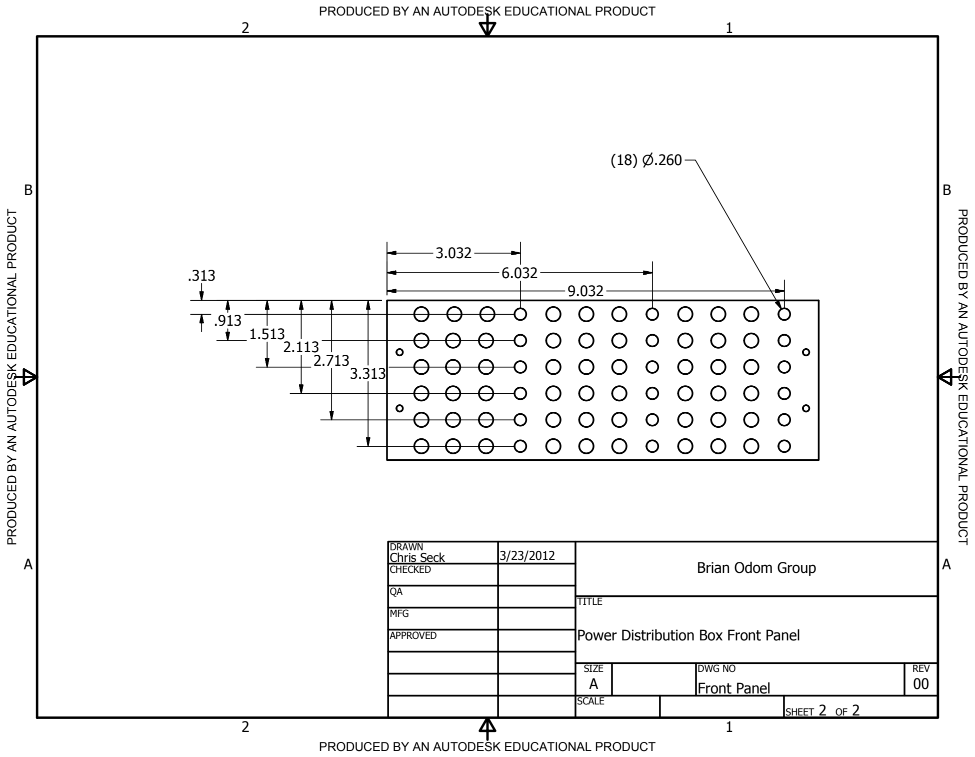
MCP mount		Filename: MCP mount.ipt
Units: inches/degree	Tolerances: +/-0.001	Chien-Yu Lien
Quantity: 2 Material: 316ss Page: 1 / 1	UHV Machining	Chien-Yu.Lien@u.northwestern.edu Tech FB31 / 7-1959 Date: 11/7/2012

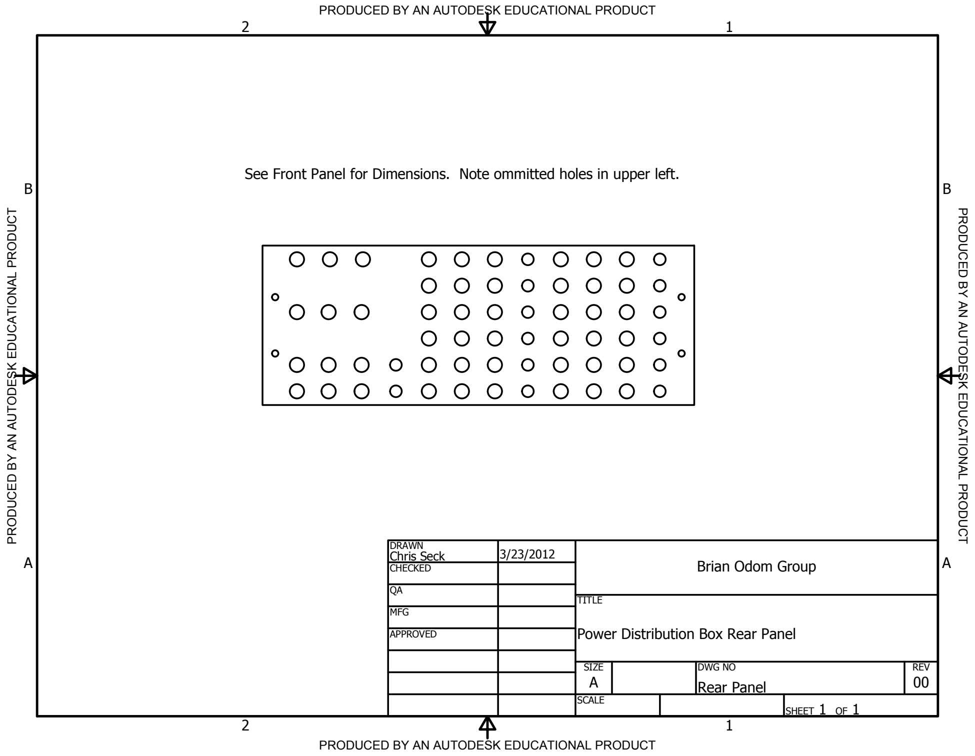


F.2.8. Power Distribution Box

The base enclosure is a Hammond Manufacturing 1402HV. The paint is removed on the bottom and top panels such that the mounting screws ground them to the case.



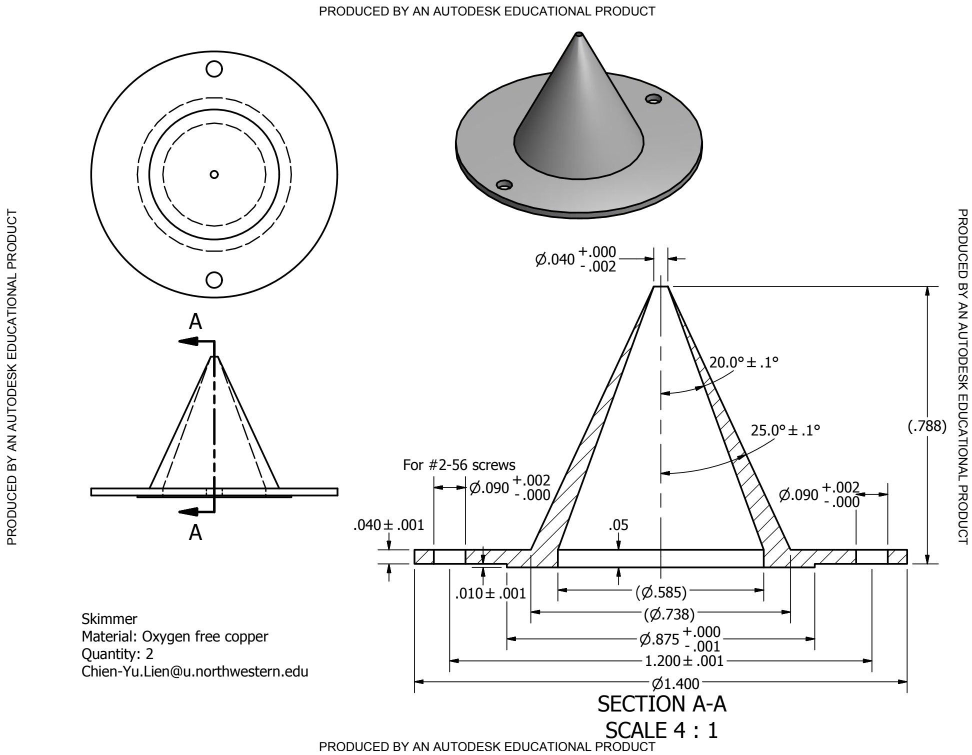




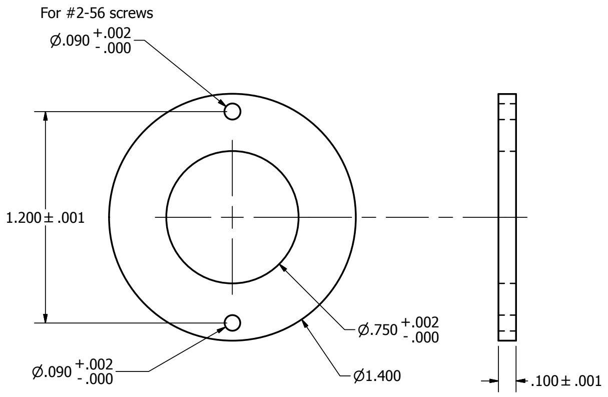
See Front Panel for Dimensions. Note ommitted holes in upper left.

F.2.9. Pulse Valve Skimmer

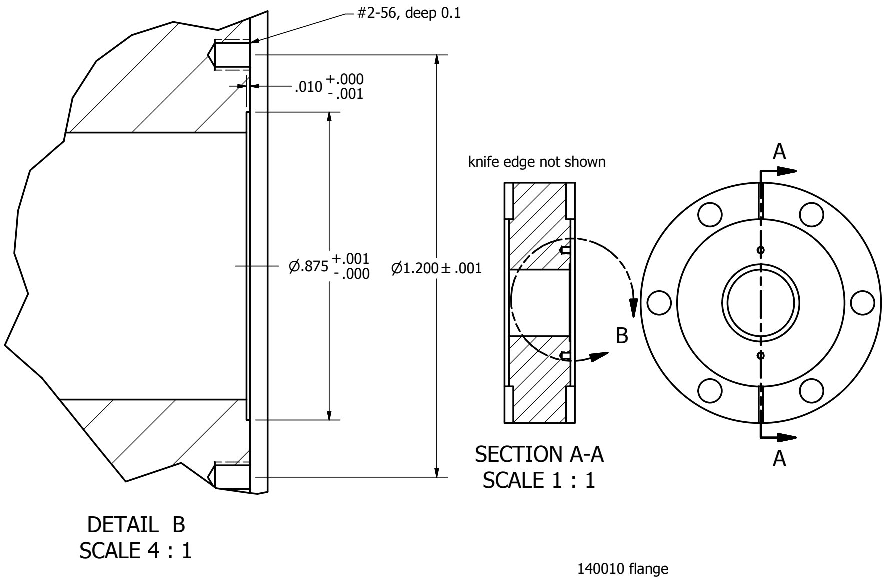
The flange is a MDC 140010 double-sided flange. Design by CYL.



Skimmer
Material: Oxygen free copper
Quantity: 2
Chien-Yu.Lien@u.northwestern.edu



Skimmer mount
Material: SS316
Quantity: 2
Chien-Yu.Lien@u.northwestern.edu

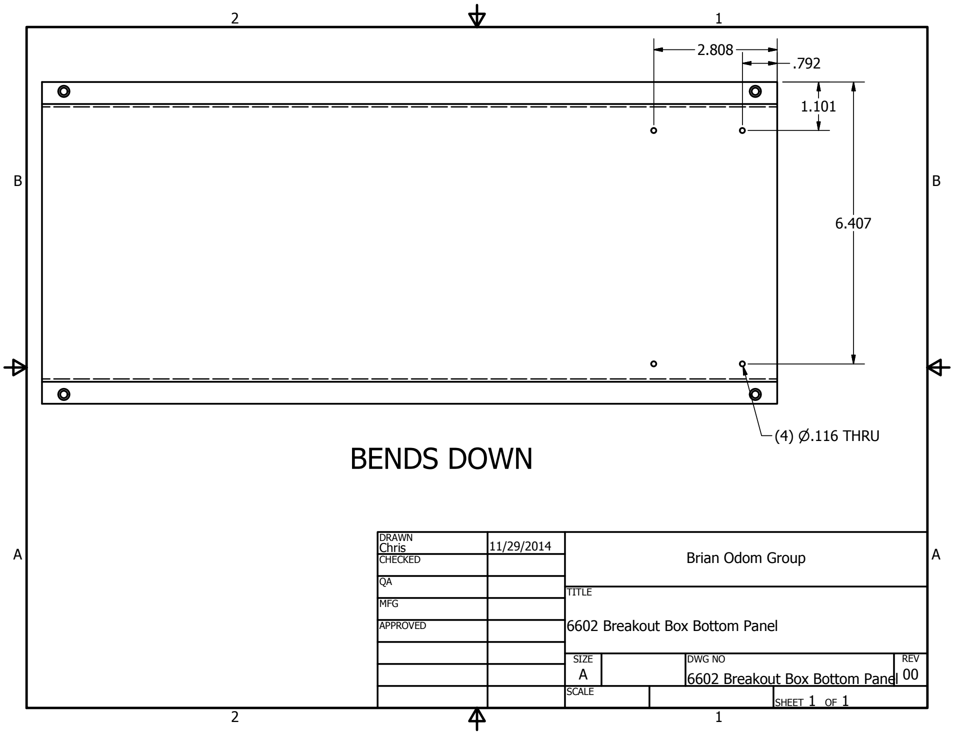


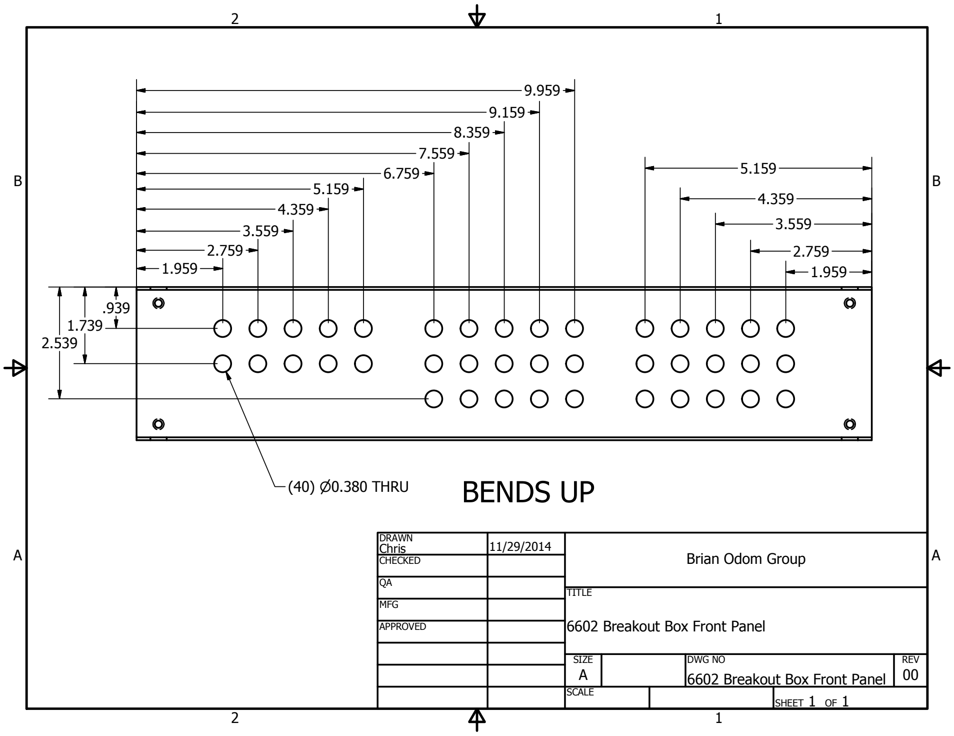
140010 flange
Chien-Yu.Lien@u.northwestern.edu

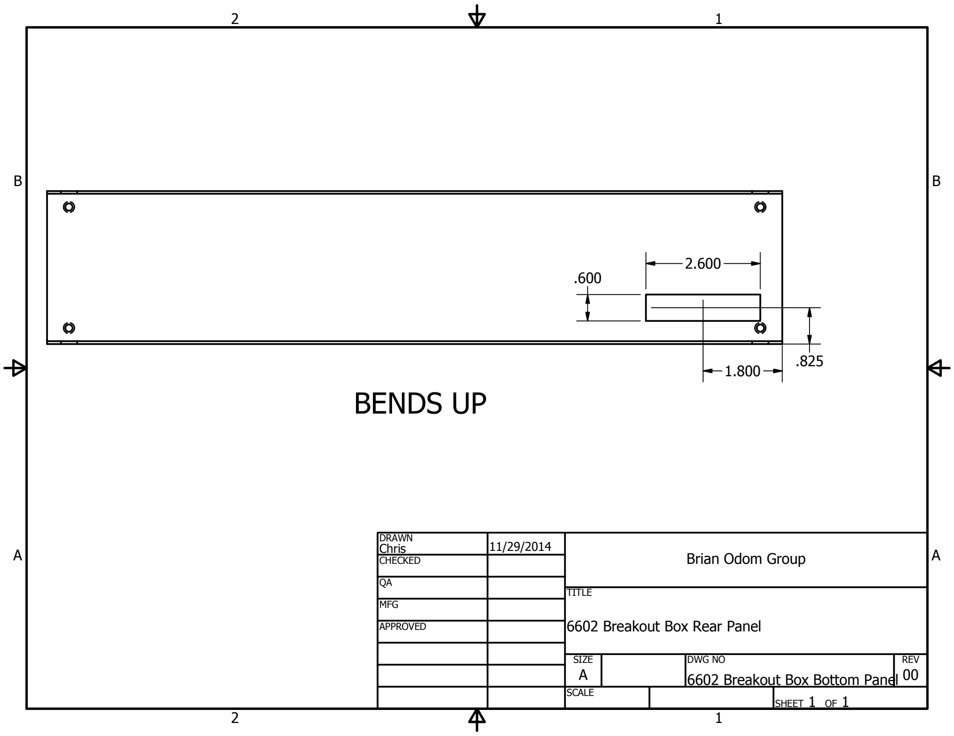
F.3. Motional Ground State Cooling Project

F.3.1. 6602 Breakout Box

The base enclosure is a Hammond Manufacturing RMCS19038.

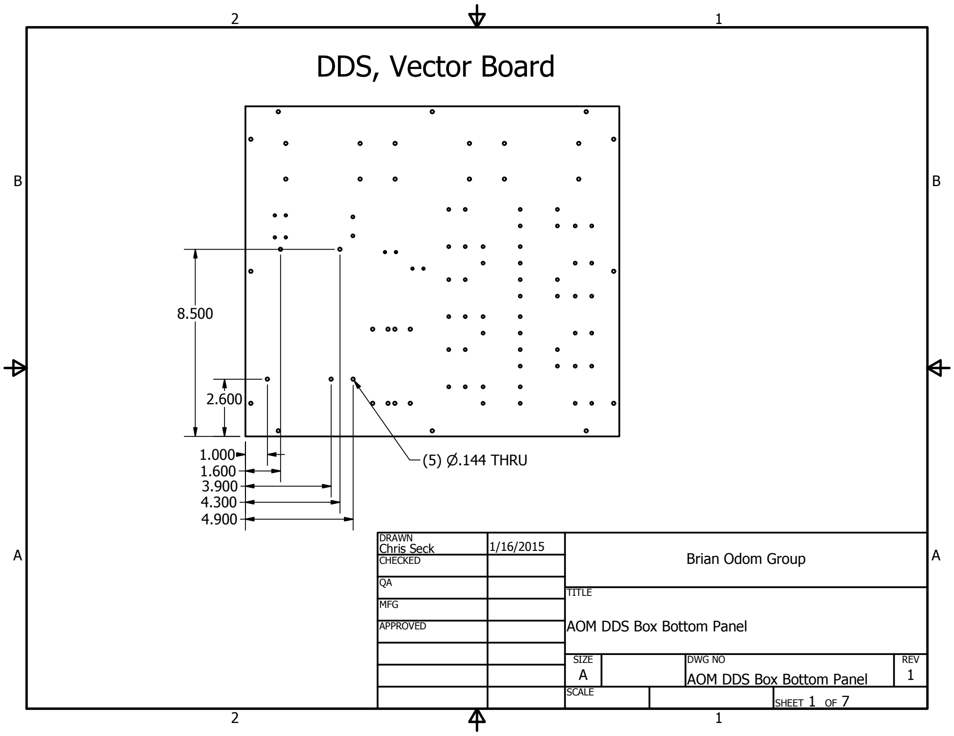


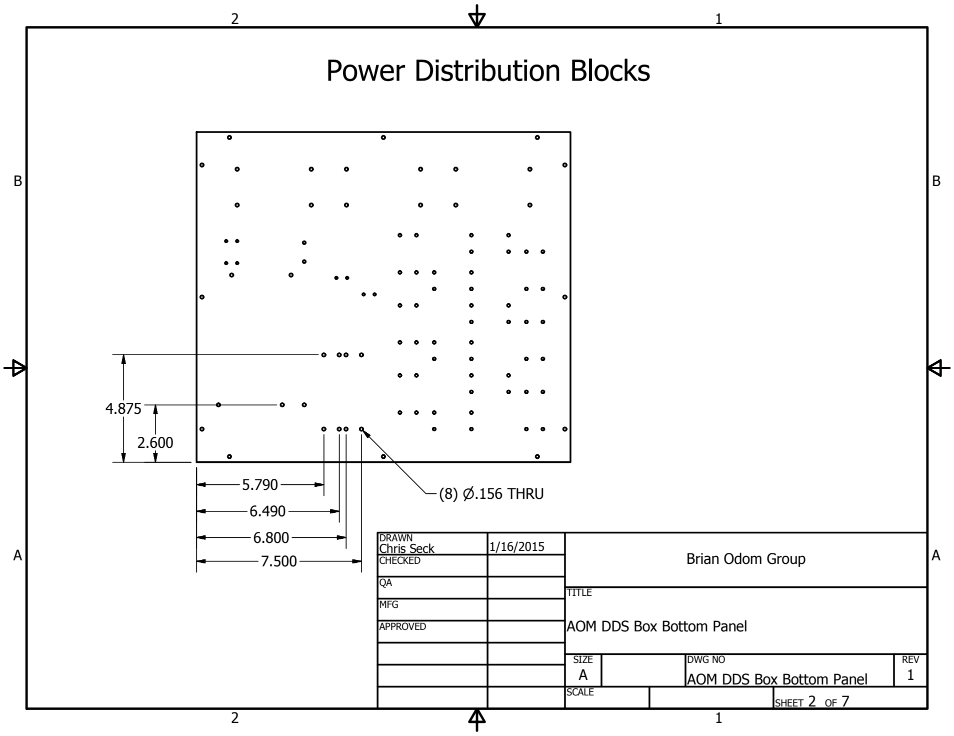

2
1
B
A
9.959
9.159
8.359
7.559
6.759
5.159
4.359
3.559
2.759
1.959
5.159
4.359
3.559
2.759
1.959
.939
1.739
2.539
(40) ∅0.380 THRU
BENDS UP
DRAWN
Chris
11/29/2014
CHECKED
QA
MFG
APPROVED
Brian Odom Group
TITLE
6602 Breakout Box Front Panel
SIZE
A
DWG NO
6602 Breakout Box Front Panel
REV
00
SCALE
SHEET 1 OF 1


2
1
B
A
2.600
.600
.825
1.800
BENDS UP
DRAWN
Chris
11/29/2014
CHECKED
QA
MFG
APPROVED
Brian Odom Group
TITLE
6602 Breakout Box Rear Panel
SIZE
A
DWG NO
6602 Breakout Box Bottom Panel
REV
00
SCALE
SHEET 1 OF 1

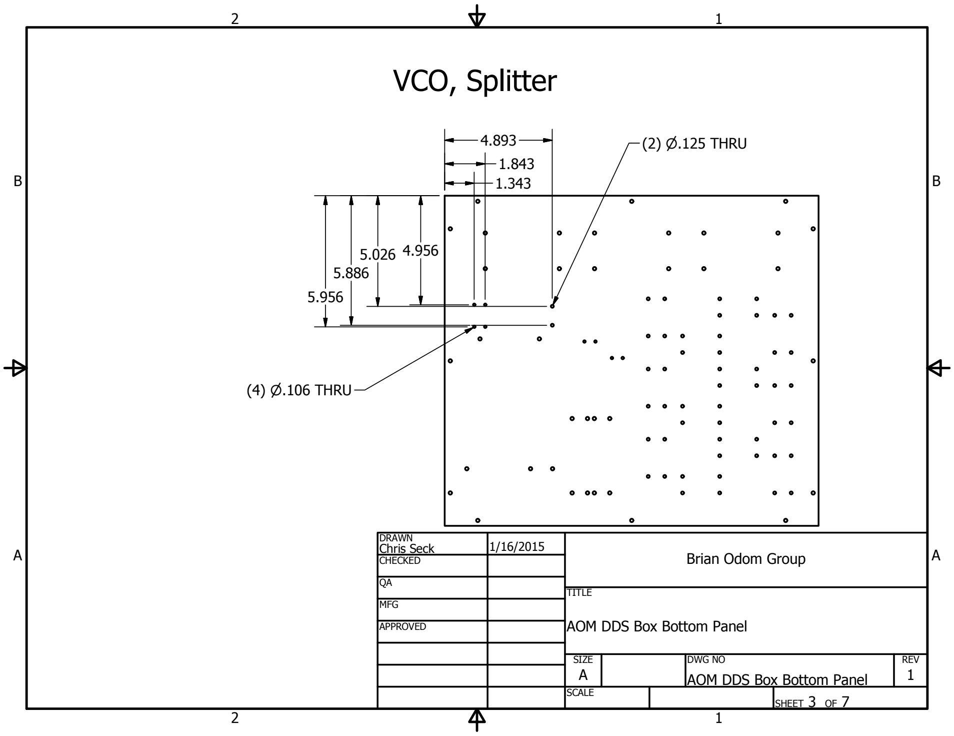
F.3.2. AOM DDS Box

The base enclosure is a Par-Metal 12-19155x.



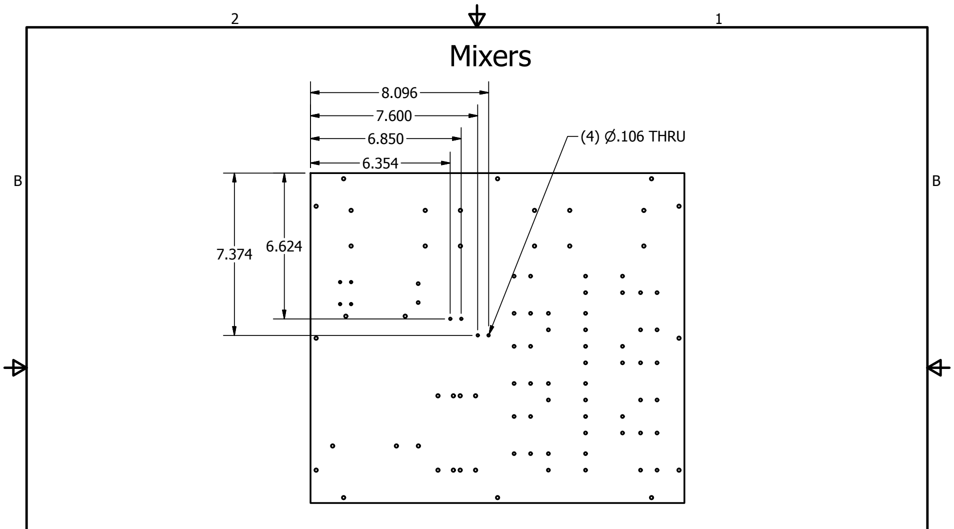


DRAWN Chris Seck	1/16/2015	Brian Odom Group		
CHECKED		TITLE AOM DDS Box Bottom Panel		
QA				
MFG				
APPROVED				
		SIZE A	DWG NO AOM DDS Box Bottom Panel	REV 1
		SCALE	SHEET 2 OF 7	



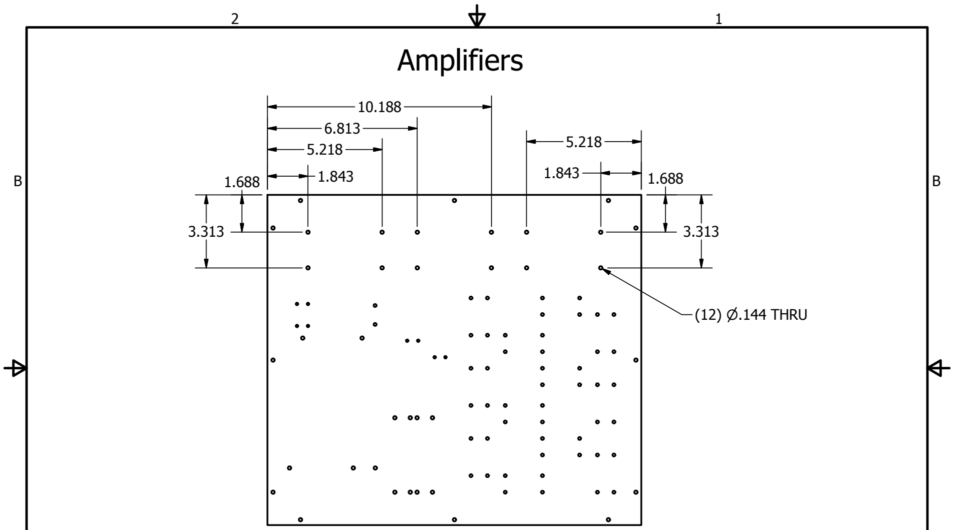
DRAWN Chris Seck	1/16/2015	Brian Odom Group		
CHECKED		TITLE AOM DDS Box Bottom Panel		
QA				
MFG				
APPROVED				
		SIZE A	DWG NO AOM DDS Box Bottom Panel	REV 1
		SCALE	SHEET 3 OF 7	

Mixers

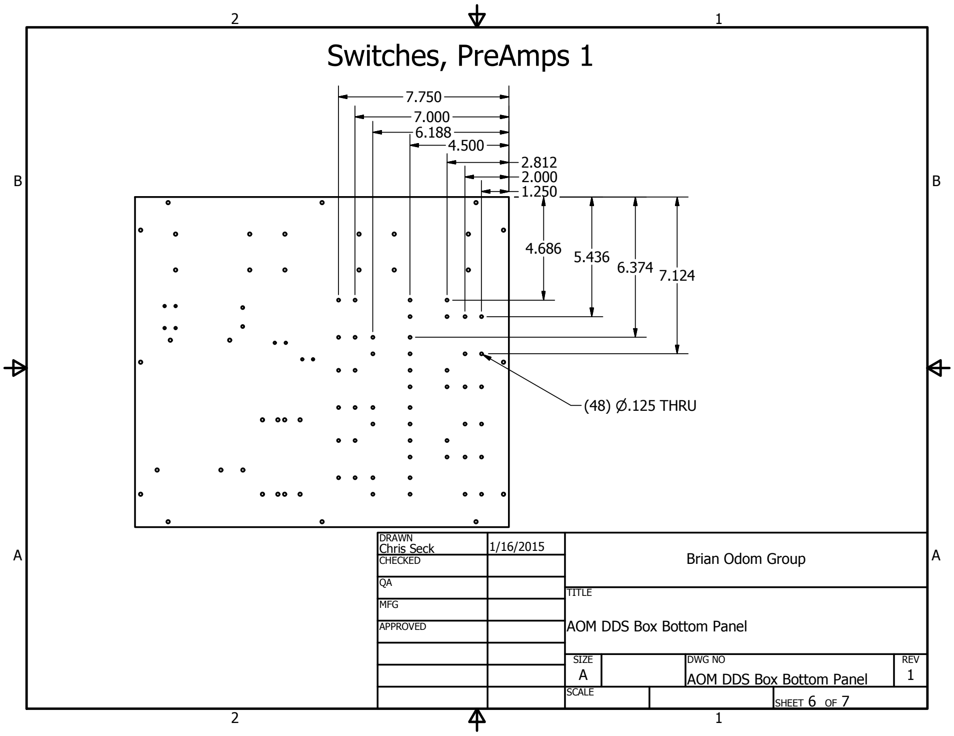


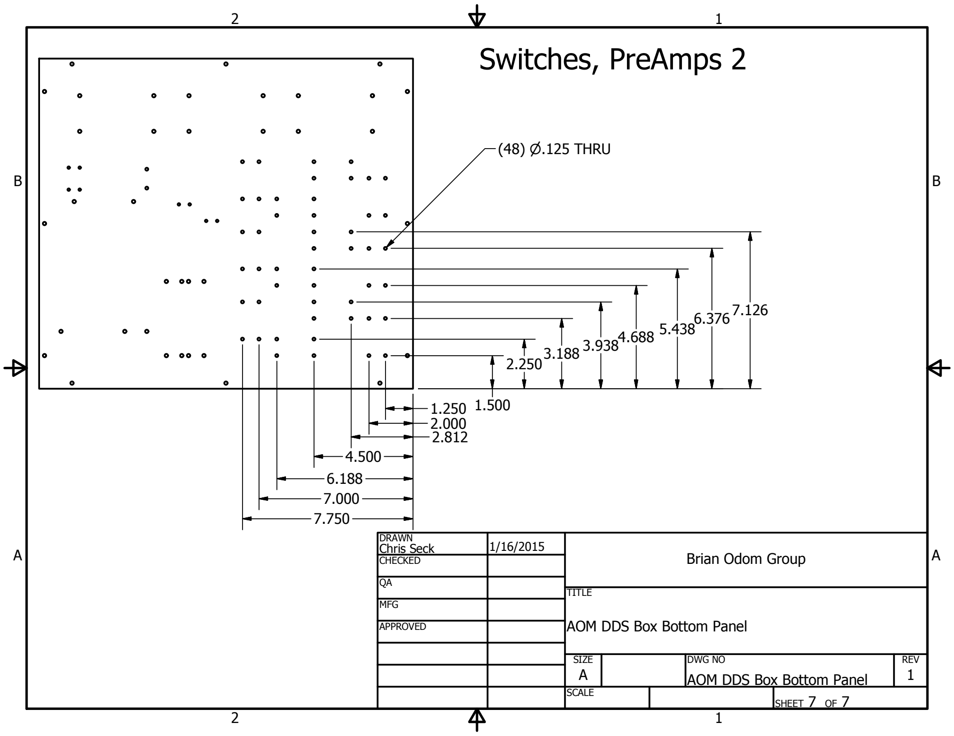
DRAWN Chris Seck	1/16/2015	Brian Odom Group
CHECKED		
QA		TITLE
MFG		
APPROVED		AOM DDS Box Bottom Panel
		SIZE A · DWG NO AOM DDS Box Bottom Panel · REV 1
		SCALE · SHEET 4 OF 7

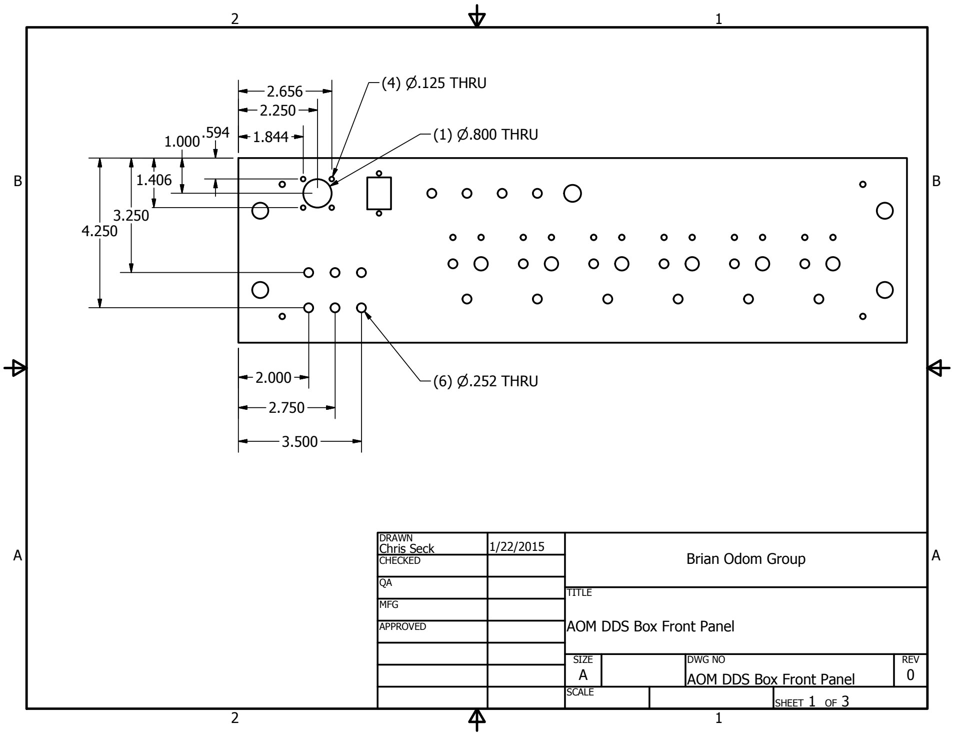
Amplifiers

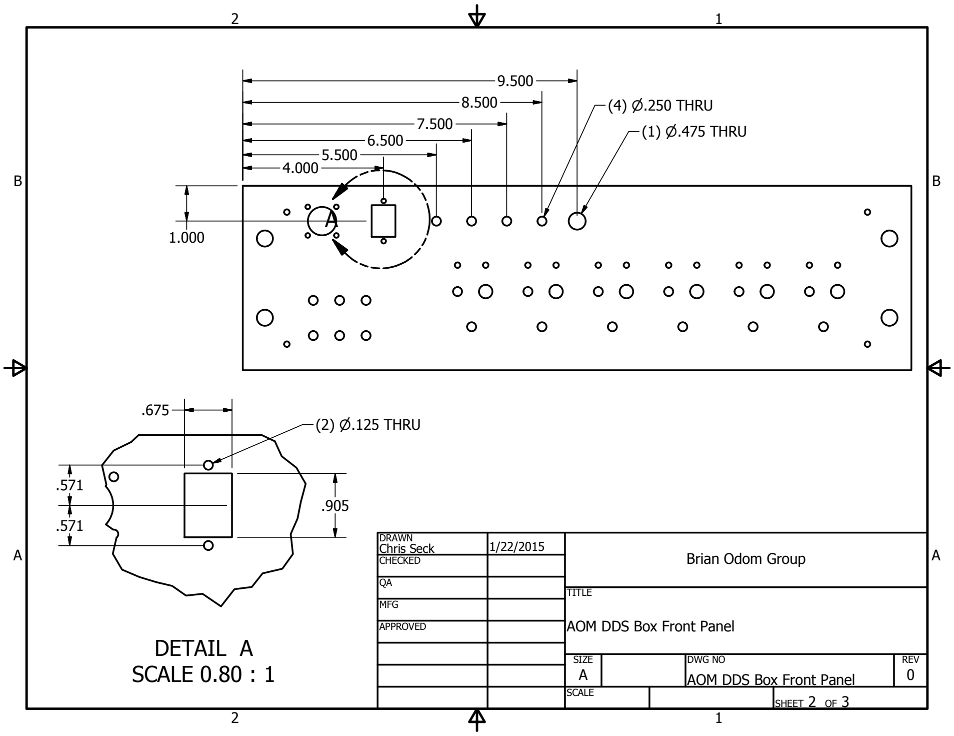


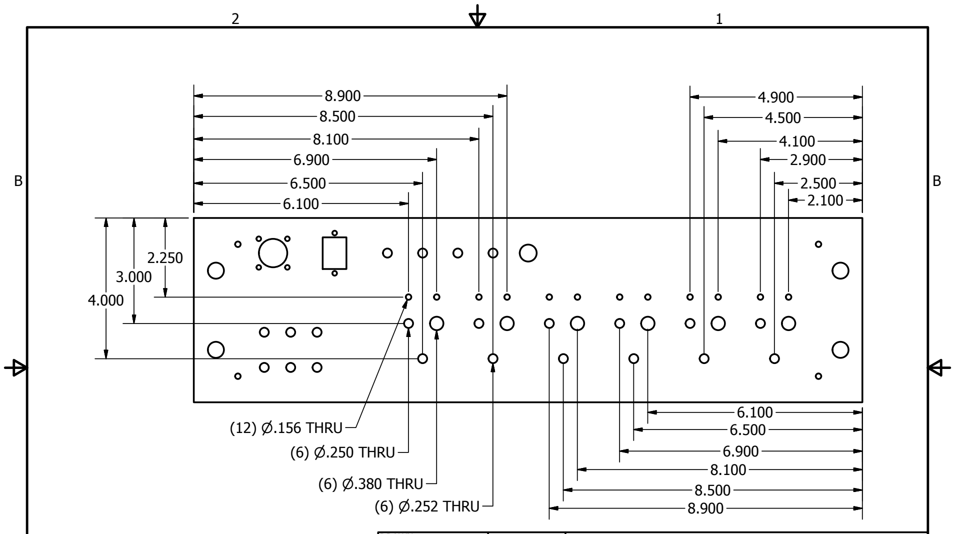
DRAWN Chris Seck	1/16/2015	Brian Odom Group
CHECKED		
QA		TITLE
MFG		
APPROVED		AOM DDS Box Bottom Panel
		SIZE A · DWG NO AOM DDS Box Bottom Panel · REV 1
		SCALE · SHEET 5 OF 7


2
1
Switches, PreAmps 1
7.750
7.000
6.188
4.500
2.812
2.000
1.250
4.686
5.436
6.374
7.124
(48) Ø.125 THRU
B
A
DRAWN
Chris Seck
1/16/2015
CHECKED
QA
MFG
APPROVED
Brian Odom Group
TITLE
AOM DDS Box Bottom Panel
SIZE
A
DWG NO
AOM DDS Box Bottom Panel
REV
1
SCALE
SHEET 6 OF 7

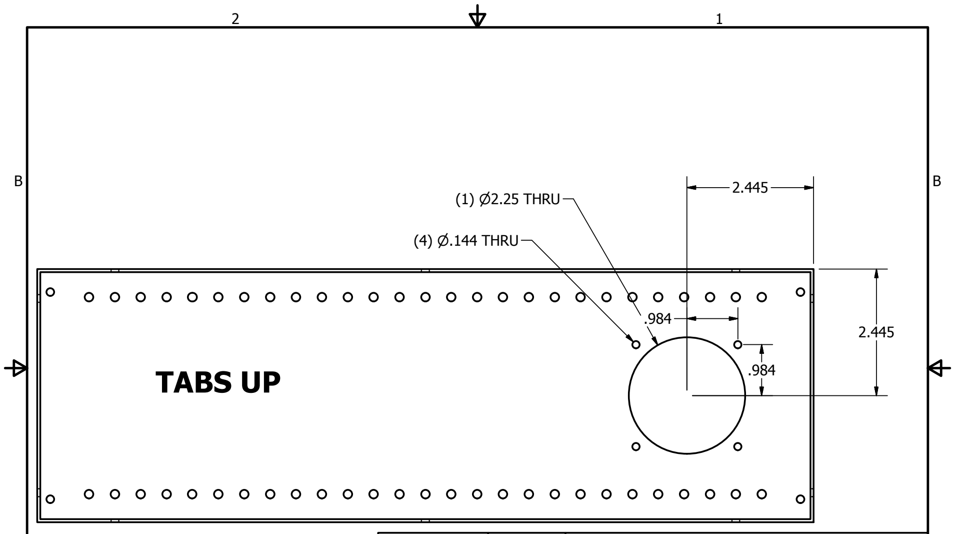

2
1
Switches, PreAmps 2
(48) Ø.125 THRU
7.126
6.376
5.438
4.688
3.938
3.188
2.250
1.500
1.250
2.000
2.812
4.500
6.188
7.000
7.750
B
A
DRAWN
Chris Seck
1/16/2015
CHECKED
QA
MFG
APPROVED
Brian Odom Group
TITLE
AOM DDS Box Bottom Panel
SIZE
A
DWG NO
AOM DDS Box Bottom Panel
REV
1
SCALE
SHEET 7 OF 7


2
1
B
A
(4) Ø.125 THRU
(1) Ø.800 THRU
(6) Ø.252 THRU
2.656
2.250
1.844
.594
1.000
1.406
3.250
4.250
2.000
2.750
3.500
DRAWN
Chris Seck
1/22/2015
CHECKED
QA
MFG
APPROVED
Brian Odom Group
TITLE
AOM DDS Box Front Panel
SIZE
A
DWG NO
AOM DDS Box Front Panel
REV
0
SCALE
SHEET 1 OF 3

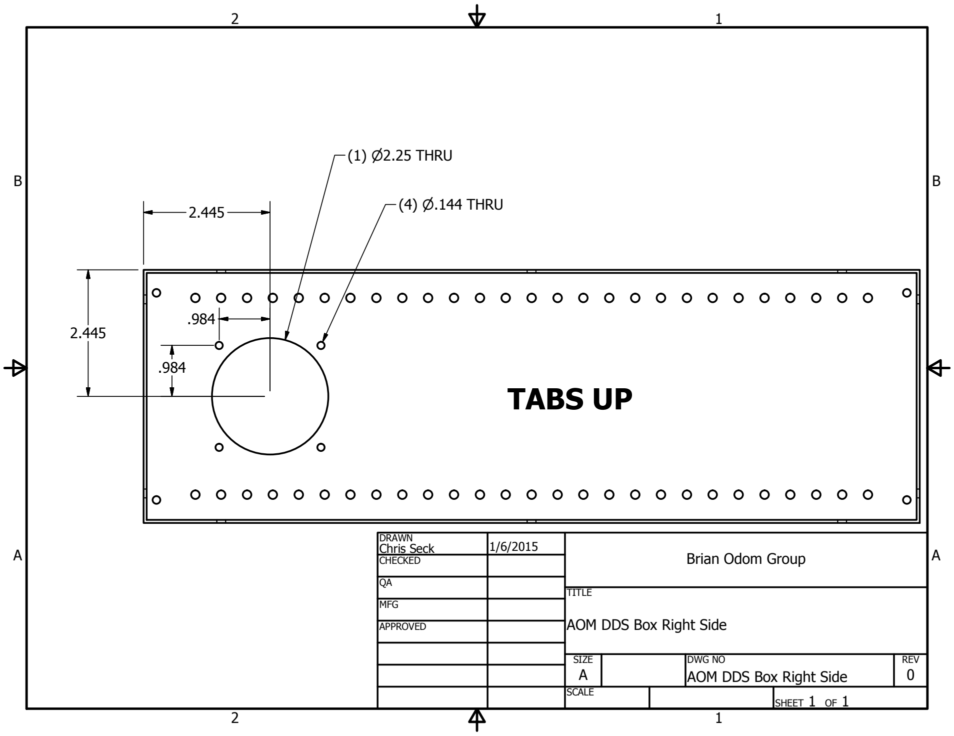

2
1
B
A
9.500
8.500
7.500
6.500
5.500
4.000
(4) Ø.250 THRU
(1) Ø.475 THRU
1.000
A
.675
(2) Ø.125 THRU
.571
.571
.905
DETAIL A
SCALE 0.80 : 1
DRAWN
Chris Seck
1/22/2015
CHECKED
QA
MFG
APPROVED
Brian Odom Group
TITLE
AOM DDS Box Front Panel
SIZE
A
DWG NO
AOM DDS Box Front Panel
REV
0
SCALE
SHEET 2 OF 3



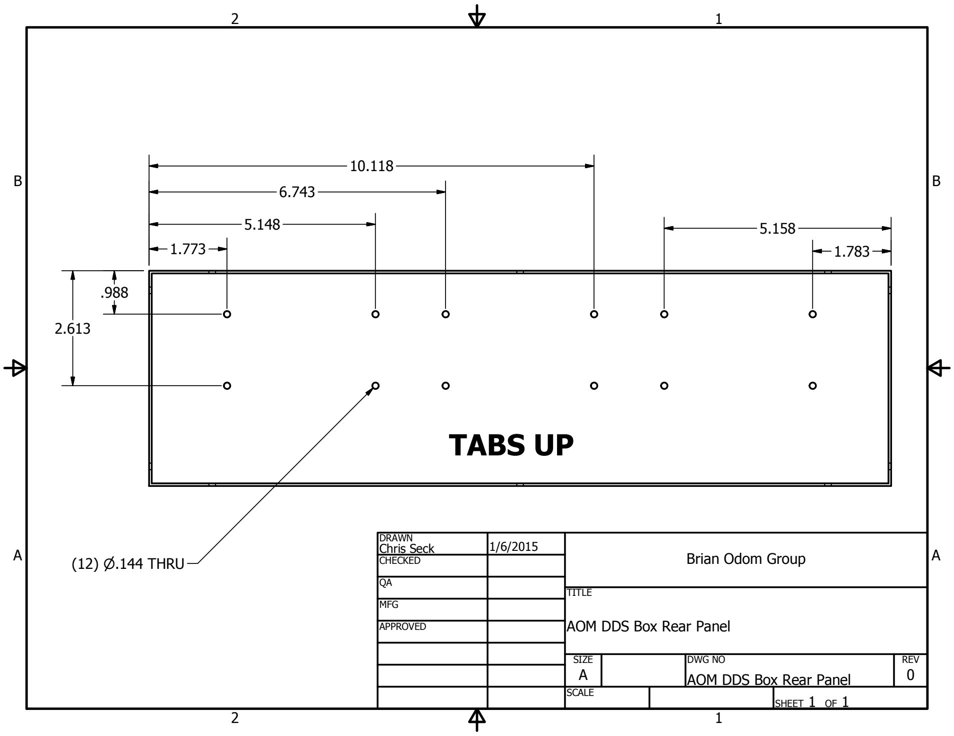
DRAWN Chris Seck	1/22/2015	Brian Odom Group		
CHECKED				
QA		TITLE		
MFG				
APPROVED		AOM DDS Box Front Panel		
		SIZE A	DWG NO AOM DDS Box Front Panel	REV 0
		SCALE		SHEET 3 OF 3



DRAWN Chris Seck	1/6/2015	Brian Odom Group		
CHECKED				
QA		TITLE		
MFG				
APPROVED		AOM DDS Box Left Side		
		SIZE A	DWG NO AOM DDS Box Left Side	REV 0
		SCALE		SHEET 1 OF 1



DRAWN Chris Seck	1/6/2015	Brian Odom Group
CHECKED		TITLE AOM DDS Box Right Side
QA		
MFG		
APPROVED		
		SIZE A — DWG NO AOM DDS Box Right Side — REV 0
		SCALE — SHEET 1 OF 1

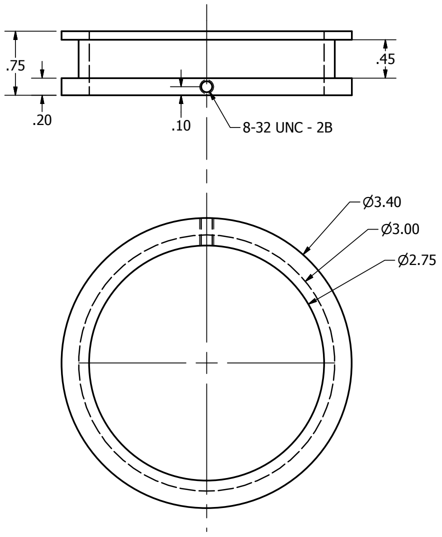


DRAWN Chris Seck	1/6/2015	Brian Odom Group
CHECKED		TITLE AOM DDS Box Rear Panel
QA		
MFG		
APPROVED		
		SIZE A — DWG NO AOM DDS Box Rear Panel — REV 0
		SCALE — SHEET 1 OF 1

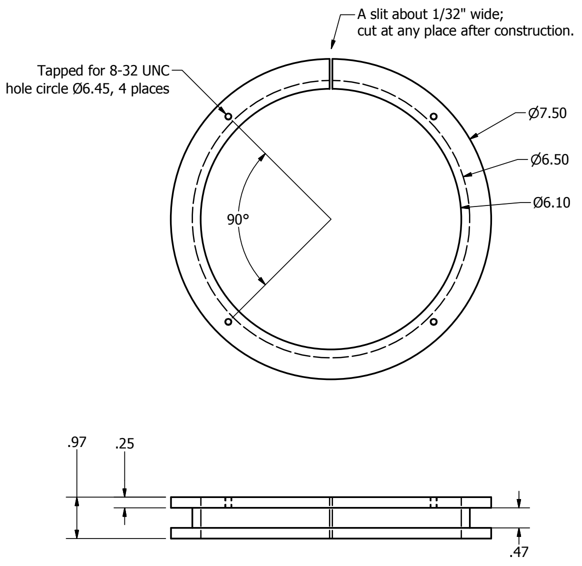
F.3.3. Magnetic Field Coil Forms

These forms hold the wire coils around or near vacuum system viewports.

Coil Form		Filename: Coil form_275.ipt
Units: inches/degree	Tolerances: ±0.01 unless specified	Yen-Wei Lin y-lin@u.northwestern.edu Tech FB31 / 7-1959 Date:
Quantity: 4 Material: Aluminum Page: 1 / 1	610-4011500-60026587-01-77535	

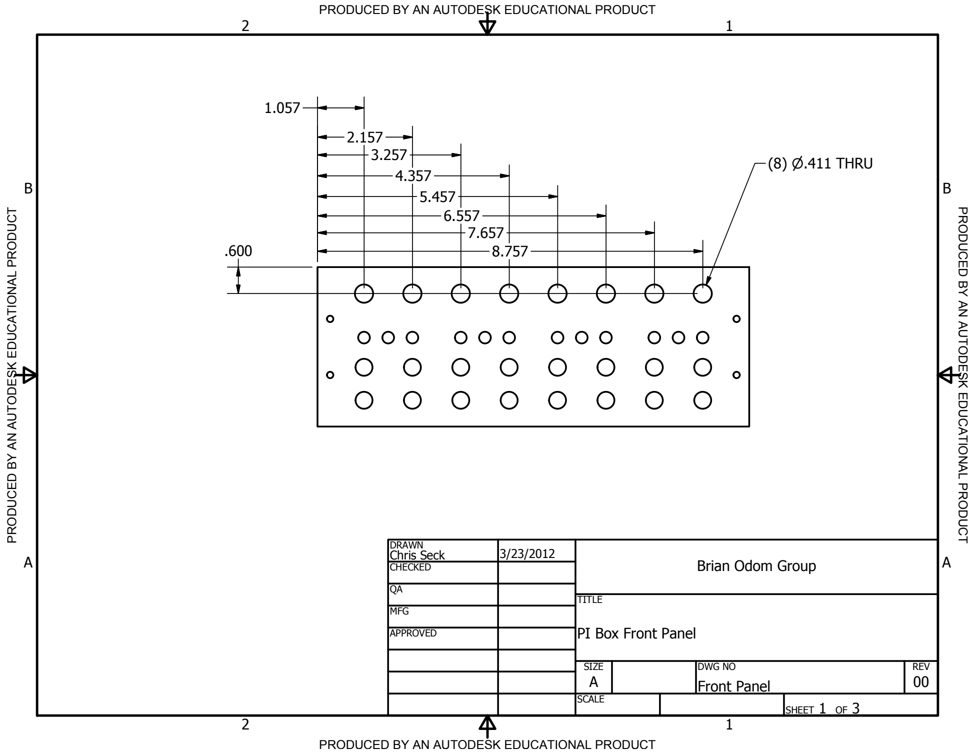


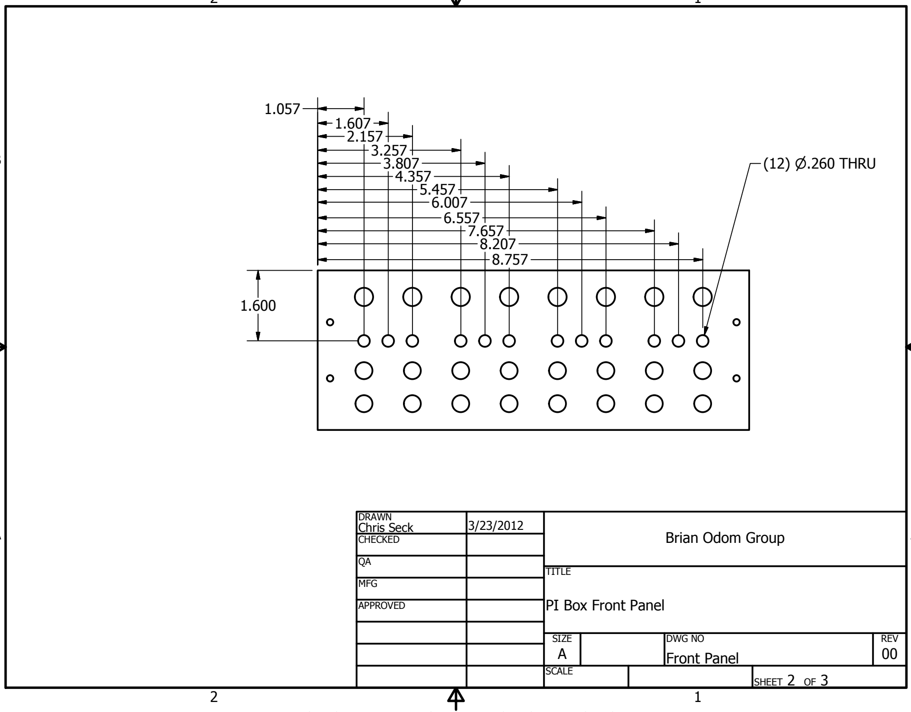
Coil form - 6CF		Filename: coil_form_6.ipt
Units: inches/degree	Tolerances: +/-0.01	Odom Group
Quantity: 2 Material: Aluminum Page: 1 / 1	Chart String: 610-4011500-60026587-01	y-lin@u.northwestern.edu Tech FB31 / 7-1959 Date: 6/17/2010

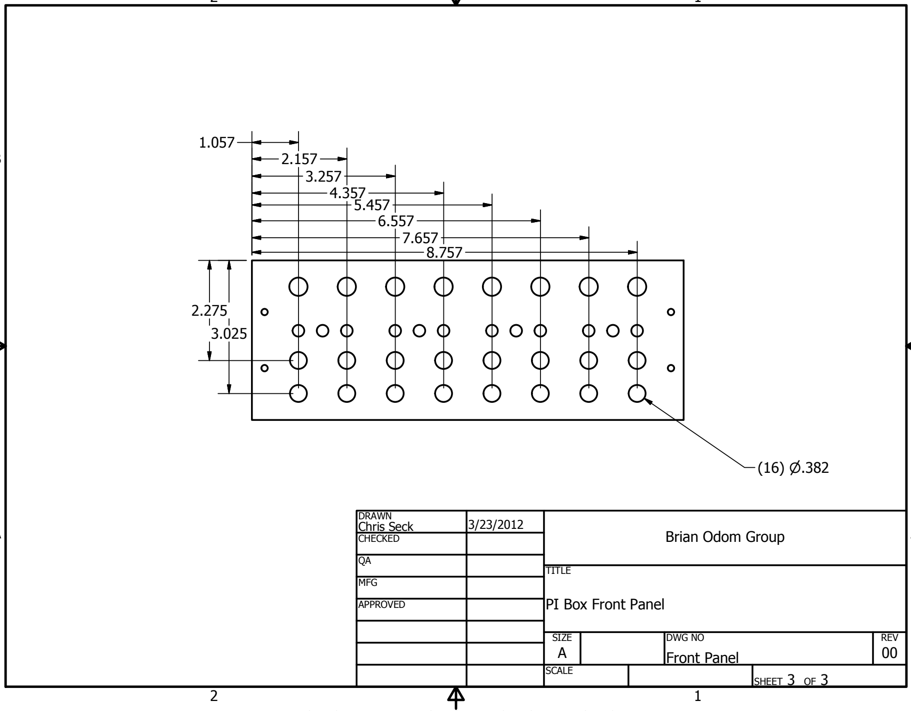


F.3.4. PID Box

This encloses (4) PIDv20 circuits with external set-point input only, and is used for the power-control servo loops. The base enclosure is a Hammond Manufacturing 1402HV. The paint is removed on the bottom and top panels such that the mounting screws ground them to the case.



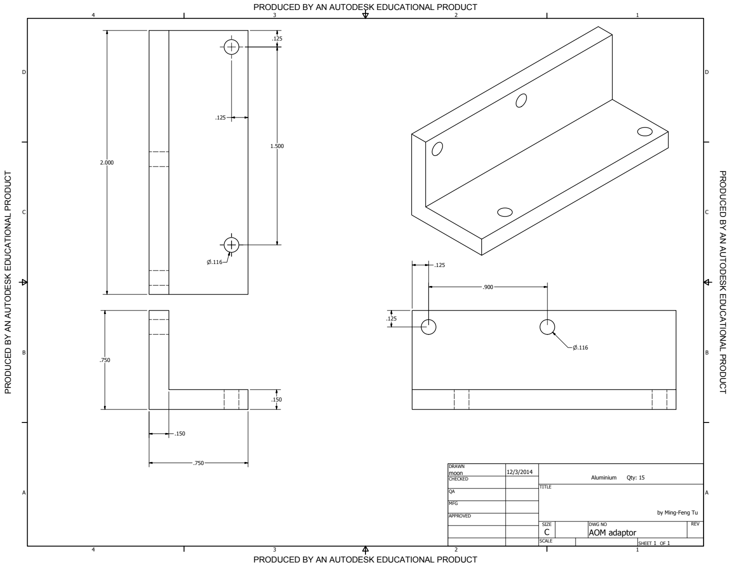

2
1
B
A
1.057
1.607
2.157
3.257
3.807
4.357
5.457
6.007
6.557
7.657
8.207
8.757
1.600
(12) ∅.260 THRU
DRAWN
Chris Seck
3/23/2012
CHECKED
QA
MFG
APPROVED
Brian Odom Group
TITLE
PI Box Front Panel
SIZE
A
DWG NO
Front Panel
REV
00
SCALE
SHEET 2 OF 3


2
1
B
A
1.057
2.157
3.257
4.357
5.457
6.557
7.657
8.757
2.275
3.025
(16) ∅.382
DRAWN
Chris Seck
3/23/2012
CHECKED
QA
MFG
APPROVED
Brian Odom Group
TITLE
PI Box Front Panel
SIZE
A
DWG NO
Front Panel
REV
00
SCALE
SHEET 3 OF 3

F.4. Miscellaneous

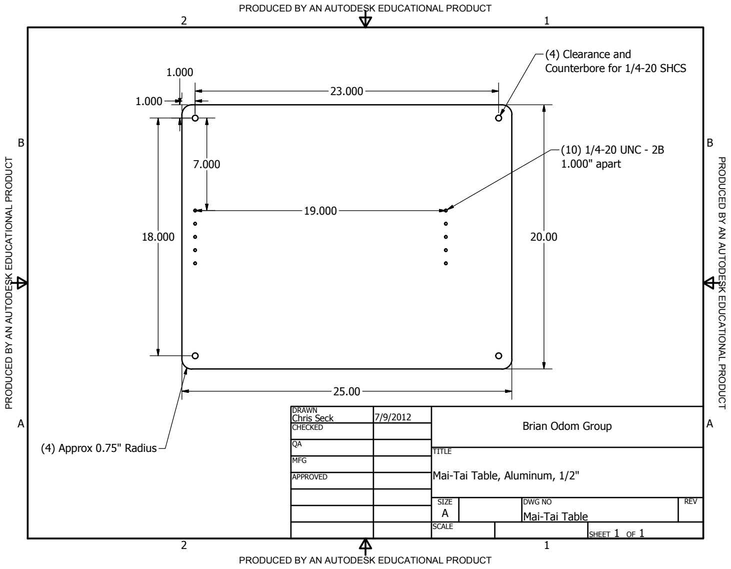
F.4.1. AOM Adapter Mount

This adapter mounts the ATM-series IntraAction AOMs to ThorLabs KM100PM mounts.



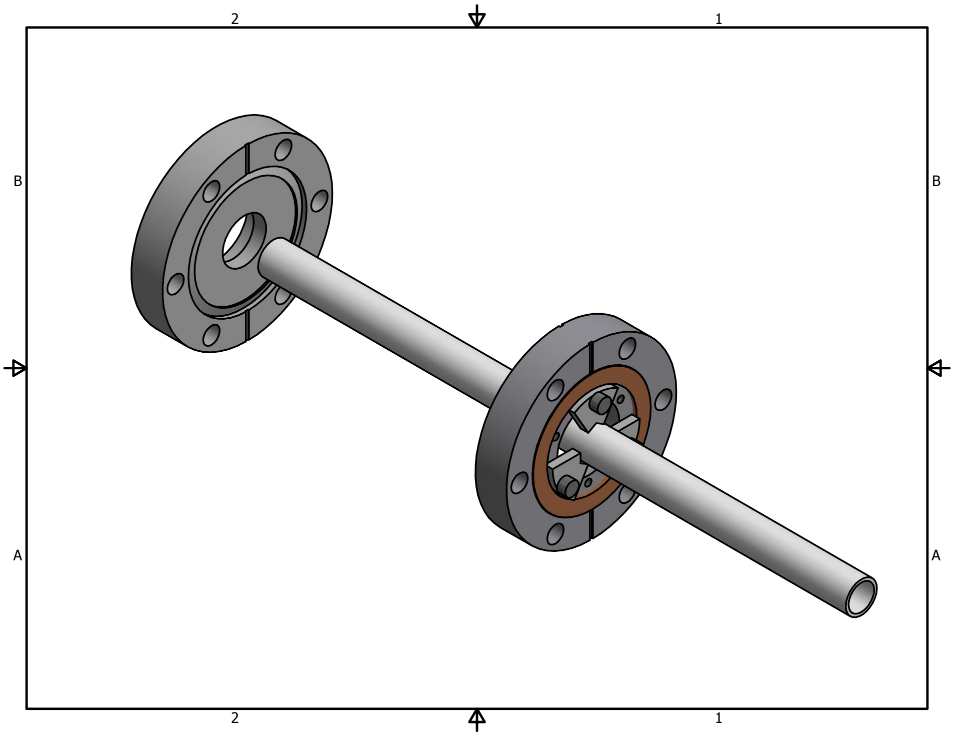
F.4.2. Mai-Tai Table

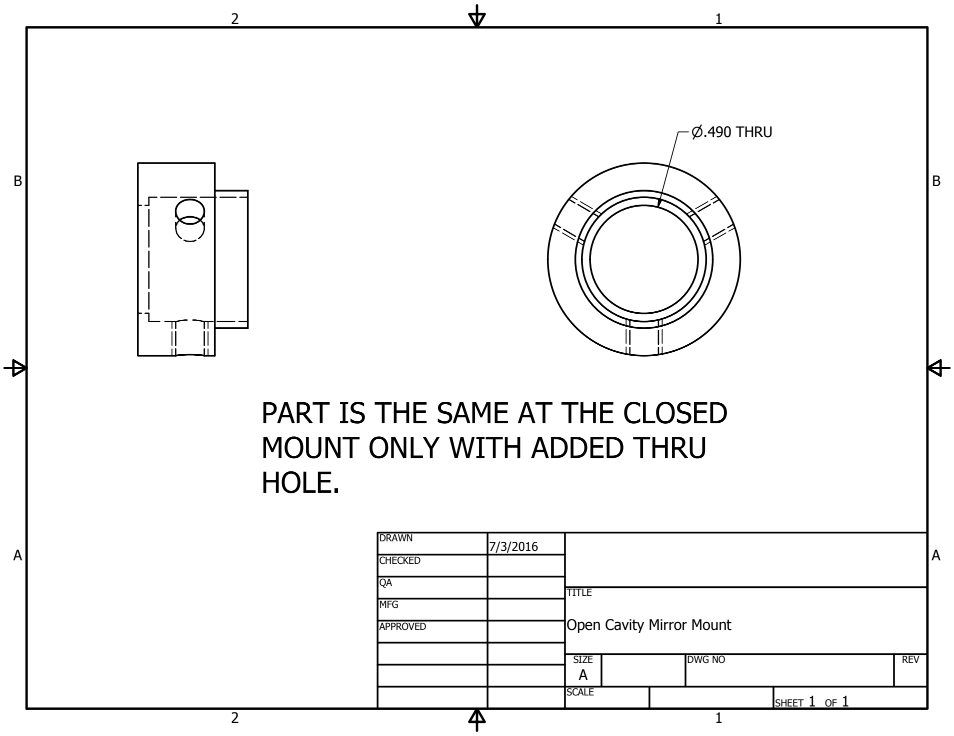
This table holds a Spectra Physics Mai Tai HP above an optical table with the use of ThorLabs 1.5”-diameter posts. If this is to be on top of another Mai Tai, it should be at least 12” above the table such that the Mai Tai cover can be removed for maintenance.



F.4.3. Temperature-stabilized Cavity

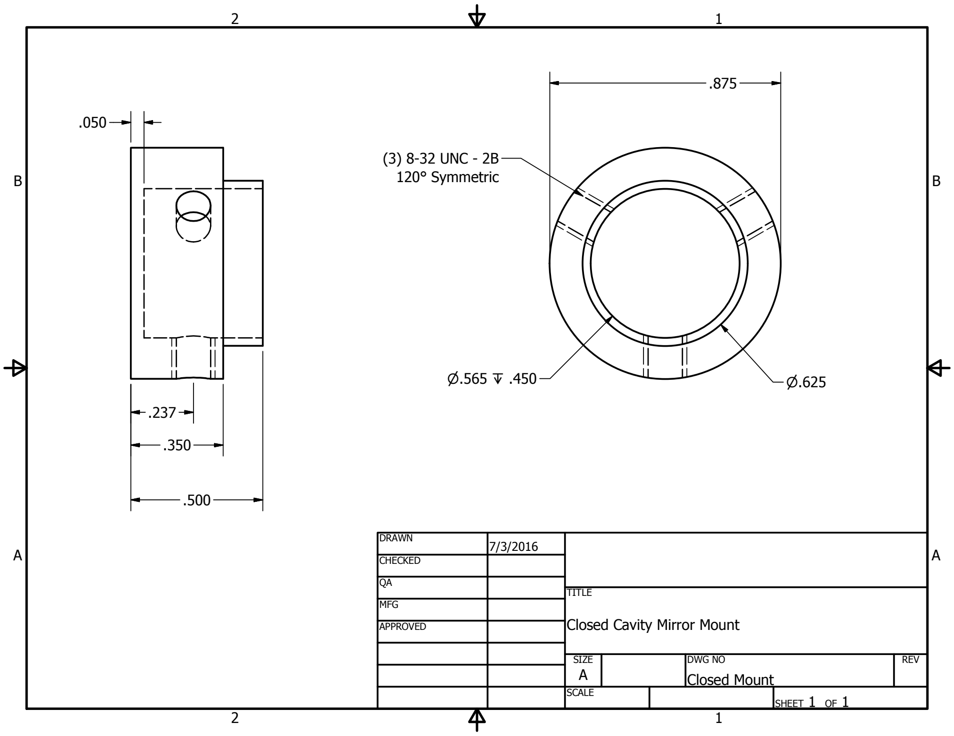
The mounts below are bonded to a fused silica tube and hold mirrors used for temperature stabilized cavities. Design by YWL.



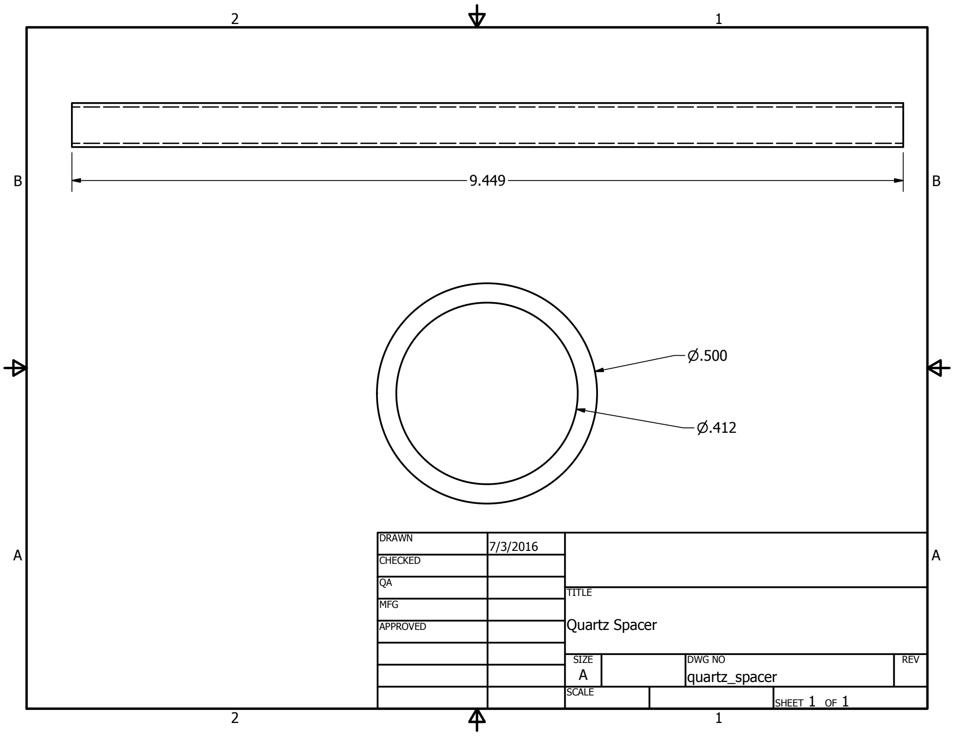


PART IS THE SAME AT THE CLOSED
MOUNT ONLY WITH ADDED THRU
HOLE.

DRAWN	7/3/2016	
CHECKED		
QA		TITLE
MFG		Open Cavity Mirror Mount
APPROVED		
		SIZE A · DWG NO · REV
		SCALE · SHEET 1 OF 1



DRAWN	7/3/2016	
CHECKED		
QA		TITLE
MFG		Closed Cavity Mirror Mount
APPROVED		
		SIZE A · DWG NO Closed Mount · REV
		SCALE · SHEET 1 OF 1



DRAWN 7/3/2016

CHECKED

QA

MFG

APPROVED

TITLE

Quartz Spacer

SIZE A

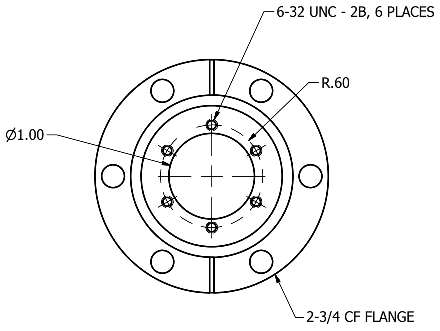
DWG NO quartz_spacer

REV

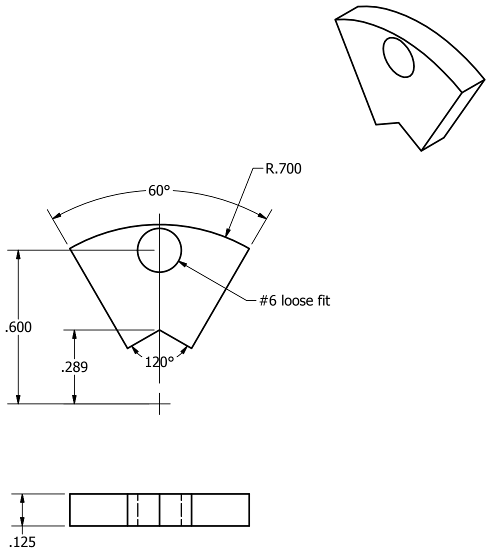
SCALE

SHEET 1 OF 1

2-3/4 CF Mounting Flange		Filename: double_side_flange_275CF.ipt
Units: inches/degree	Tolerances: +/-0.01 inches	Yen-Wei Lin
Quantity: 2 Material: 316SS Page: 1 / 1	614-4011500-60026587-01-77535 High vacuum machinng	y-lin@u.northwestern.edu Tech FB31 / 7-1959 Date:



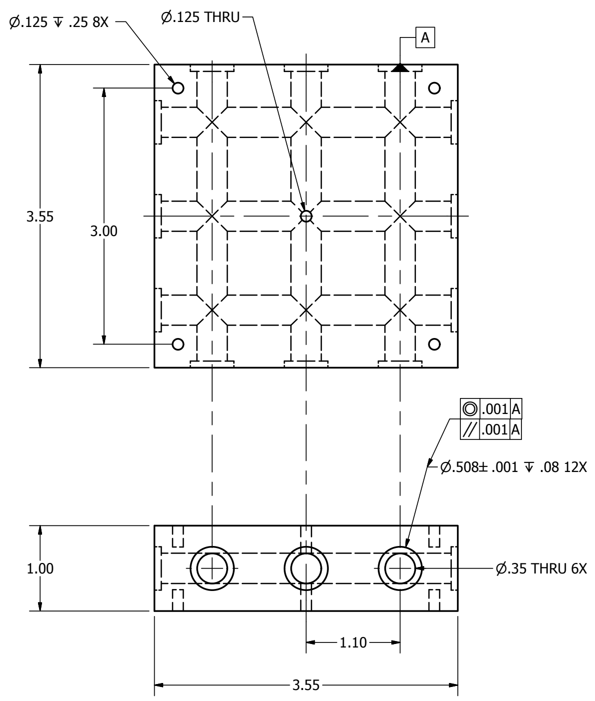
Tube Holder		Filename: tube_holder_proto.ipt
Units: inches/degree	Tolerances:	Yen-Wei Lin
Quantity: 12 Material: Teflon Page: 1 / 1	611-4011500-60024037-01	y-lin@u.northwestern.edu Tech FB31 / 7-1959 Date: 11/18/2010



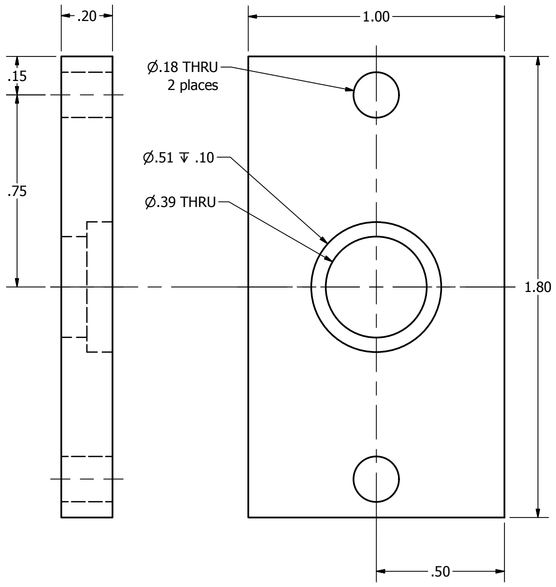
F.4.4. Temperature-stabilized Six Cavity Block

The cavity system below holds six cavities in a single Zerodur block with the mirrors held in place with a stainless steel plate and o-ring. Design by YWL and used in work by PRS [216].

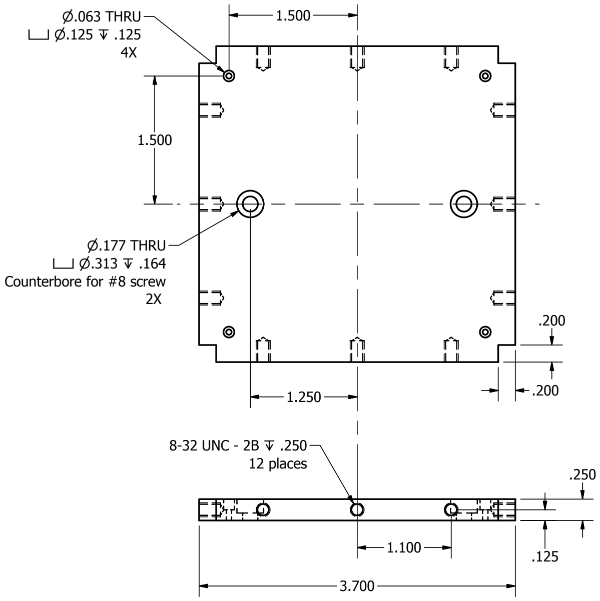
Cavity block		Filename: laser cavity 3.ipt
Units: in / deg	Tolerances: +/-0.01 unless specified	y-lin@u.northwestern.edu Tech FB31 / 7-1959 Date: 2015/10/1
Quantity: 1 Material: Zerodur glass Page: 1 / 1		



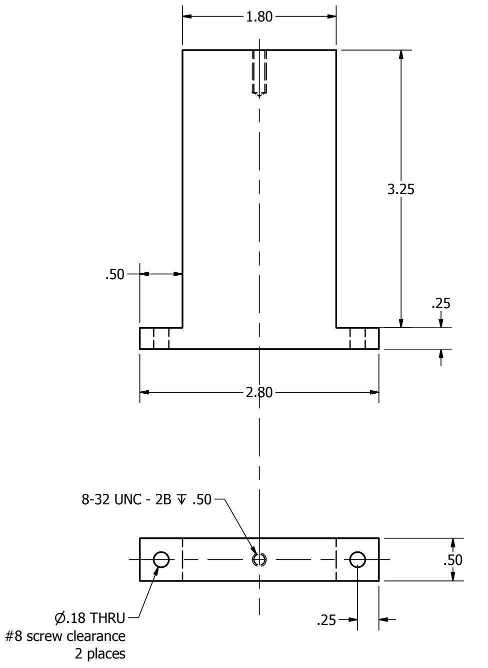
Mirror Clamp		Filename: mirror clamp 2.ipt
Units: inches/degree	Tolerances: +/-0.01	Yen-Wei Lin
Quantity: 12 Material: Stainless Steel Page: 1 / 1	171-4011500-10021816-01 High vacuum parts	yenweilin2009@u.northwestern.edu Tech FB31 / 7-1959 Date: 01/05/2016

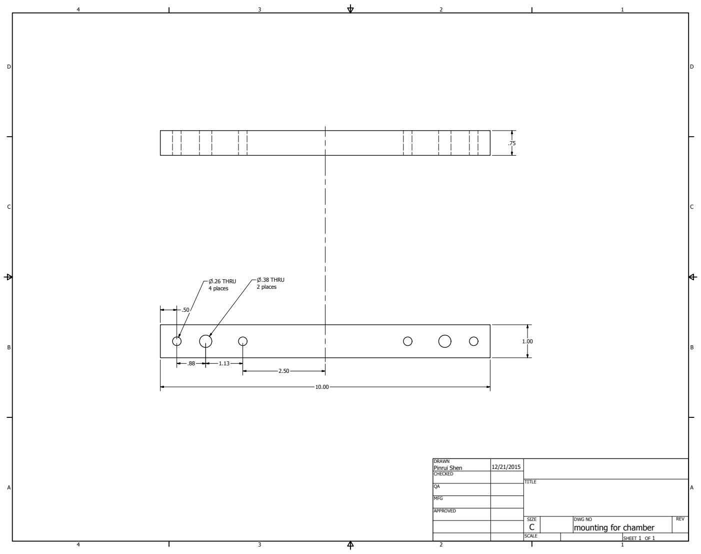


baseplate		Filename: stand_baseplate.ipt
Units: inches/degree	Tolerances: +/-0.01	Yen-Wei Lin
Quantity: 2 Material: SS Page: 1 / 1	171-4011500-10021816-01 High vacuum parts	y-lin@u.northwestern.edu Tech FB31 / 7-1959 Date: 2015-10-27



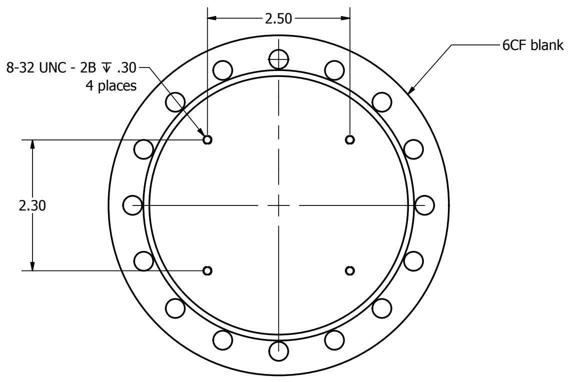
Stand		Filename: stand.ipt
Units: inches/degree	Tolerances: +/-0.01	Yen-Wei Lin
Quantity: 2 Material: SS Page: 1 / 1	171-4011500-10021816-01 High vacuum parts	y-lin@u.northwestern.edu Tech FB31 / 7-1959 Date: 2015-10-27





DRAWN Pinrui Shen	12/21/2015	
CHECKED		TITLE
QA		
MFG		
APPROVED		SIZE C · DWG NO mounting for chamber · REV
		SCALE · SHEET 1 OF 1

Base flange		Filename: 6CF_blank.ipt
Units: inches/degree	Tolerances: +/-0.01	Yen-Wei Lin
Quantity: 1	171-4011500-10021816-01	y-lin@u.northwestern.edu
Material:	High vacuum parts	Tech FB31 / 7-1959
Page: 1 / 1		Date: 2015/10/20



Vita

Christopher Seck was born in Southern California in October 1985. After finishing high school in 2004, he studied physics at California Polytechnic State University in San Luis Obispo, conducting research in DNA self-assembly of microspheres, bathymetric light detection and ranging (LIDAR), organic solar cells, and optical beam-shaping. He also served as the university Society of Physics Students President for the 2008-2009 academic year. He received a B.S. in physics with a minor in mathematics in June 2009. He then advanced to Northwestern University in Evanston, Illinois conducting research with trapped ions, specializing in molecular optical cooling and atomic sub-Doppler cooling methods. He completed all requirements for his Ph.D. in physics in August, 2016 and looks forward to his next adventure as a research scientist at the Georgia Institute of Technology.

Publications:

[401] G. D. Gillen, C. M. Seck, and S. Guha, ”Analytical beam propagation model for clipped focused-Gaussian beams using vector diffraction theory,” *Opt. Express*, vol. 18, no. 5, pp. 4023-4040, Mar 2010. Available: http://dx.doi.org/10.1364/OE.18.004023

[300] C. M. Seck, E. G. Hohenstein, C.-Y. Lien, P. R. Stollenwerk, and B. C. Odom, ”Rotational state analysis of AlH$^+$ by two-photon dissociation,” *Journal of Molecular Spectroscopy*, vol. 300, pp. 108-111, 2014. Available: http://dx.doi.org/10.1016/j.jms.2014.03.023

[263] C.-Y. Lien, C. M. Seck, Y.-W. Lin, J. H. Nguyen, D. A. Tabor, and B. C. Odom, ”Broadband optical cooling of molecular rotors from room temperature to the ground state,” *Nature Communications*, vol. 5, 2014. Available: http://dx.doi.org/10.1038/ncomms5783

[402] C. M. Seck, M. G. Kokish, M. R. Dietrich, and B. C. Odom, ”Raman sideband cooling of a ^{138}Ba^{+} ion using a Zeeman interval,” *Phys. Rev. A*, vol. 93, p. 053415, May 2016. Available: http://dx.doi.org/10.1103/PhysRevA.93.053415

[403] C. M. Seck, P. J. Martin, E. C. Cook, B. C. Odom, and D. A. Steck, ”Noise reduction of a Libbrecht-Hall style current driver,” *Review of Scientific Instruments*, vol. 87, no. 6, p. 064703, 2016. Available: http://dx.doi.org/10.1063/1.4953330