MEASUREMENTS OF THE TIME EVOLUTION OF COHERENT EXCITATION

by

HOWARD ALAN CAMP

B. S., University of Nebraska — Lincoln, 1999

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Physics
College of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas
2005

ABSTRACT

In recent years, coherent excitation techniques have focused on the ability to efficiently prepare atomic or molecular systems into a selected state. Such population control plays a key role in cutting-edge research taking place today, such as in the areas of quantum information and laser-controlled chemical reactions. Stimulated Raman adiabatic passage (STIRAP) is a widely-used coherent excitation technique that provides a relatively robust control mechanism for efficiently exciting a target population into a desired state. While the technique is well proven, current experimental techniques yield little information on the population dynamics taking place throughout the excitation process, and experimentalists rely solely on final excited-state measurements to determine the efficiency of population transfer. This dissertation presents a unique diagnostic tool to measure multilevel coherent population transfer on a short (nanosecond) timescale. The technique described here uses magneto-optical trap recoil ion momentum spectroscopy (MOTRIMS) as a noninvasive probe of a coherently-controlled system. It provides extremely detailed information about the excitation process, and highlights some important characteristics seen in excited populations that would otherwise be misleading or completely overlooked if one were to use more traditional diagnostic techniques. This dissertation discusses both the theoretical and experimental results applied to three-level coherently excited target populations of ^{87}Rb.

MEASUREMENTS OF THE TIME EVOLUTION OF COHERENT EXCITATION

by

HOWARD ALAN CAMP

B. S., University of Nebraska — Lincoln, 1999

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Physics
College of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas
2005

Approved by:

Major Professor
B. D. DePaola

ABSTRACT

In recent years, coherent excitation techniques have focused on the ability to efficiently prepare atomic or molecular systems into a selected state. Such population control plays a key role in cutting-edge research taking place today, such as in the areas of quantum information and laser-controlled chemical reactions. Stimulated Raman adiabatic passage (STIRAP) is a widely-used coherent excitation technique that provides a relatively robust control mechanism for efficiently exciting a target population into a desired state. While the technique is well proven, current experimental techniques yield little information on the population dynamics taking place throughout the excitation process, and experimentalists rely solely on final excited-state measurements to determine the efficiency of population transfer. This dissertation presents a unique diagnostic tool to measure multilevel coherent population transfer on a short (nanosecond) timescale. The technique described here uses magneto-optical trap recoil ion momentum spectroscopy (MOTRIMS) as a noninvasive probe of a coherently-controlled system. It provides extremely detailed information about the excitation process, and highlights some important characteristics seen in excited populations that would otherwise be misleading or completely overlooked if one were to use more traditional diagnostic techniques. This dissertation discusses both the theoretical and experimental results applied to three-level coherently excited target populations of ^{87}Rb.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

As in any work of this magnitude, this was not a solo effort. I appreciate my advisor, Dr. Brett DePaola's friendship, guidance, and doughnuts. Without any one of these, I would never have finished. Brett and Sue opened their home to me and my family for frequent meals and great conversation. Thank you both.

A great number of collaborators should be listed as having provided important insight to the work presented here. Dr. Larry Weaver was vital in helping all of us wrap our brains around density matrices. Thank you! I am grateful for Dr. Kevin Carnes and his never-ending answers to the many XSYS-related problems I kept throwing at him. Our system has been greatly improved due to his willingness to entertain questions. I appreciate the numerous times Dr. Charles Fehrenbach was called upon to "think outside the box". His insights helped us overcome a plethora of technical challenges. To that end, the technical staff have done likewise. Thank you, Al Rankin for a great new MOTRIMS chamber! Mike Wells has repeatedly helped in this research, especially his work with the cooling system in the MOT room. Now, it's too cold in there to take data...

There are a large number of post-docs and fellow students to whom I owe my appreciation. Hai Nguyen showed me the ropes, and tolerated my "newbie" mistakes. Chris Verzani was great comic relief. Both Xavier Fléchard and Richard Brédy (the Frenchmen) provided excellent leadership, and showed me how to be a better scientist. Alina Gearba proved to be a great friend, and I appreciate her tolerance, even when the experiment wasn't working. Mudessar Shah (Moody) and Marc Trachy (the Marcsman) have been extremely close friends throughout this sometimes-frustrating process, and I thank them both. Don't worry guys, most likely you've already taken the data you'll use to graduate.

Most of all, I am grateful for my wife, Kris, and both my children, Sam and Jonah, who supported me throughout these past five and-a-half years. Thank you!

DEDICATION

To my family.

Chapter 1

Introduction

Atomic, molecular and optical physics research has made great progress at a frenetic pace in the past thirty years, the results of which have had a far-reaching impact in every field of scientific research. The ability to understand, control, and measure atomic and molecular processes with today's unprecedented precision has opened frontier research areas such as atom optics, ultracold physics, nanotechnology, ultrafast laser research and quantum information. The quest for a deeper, more precise understanding of the physical world on an atomic level is what drives each of these research areas. This ambition paves the way for future research extending beyond limitations present even in today's cutting-edge AMO research.

Specific advancements in cold and ultracold research have provided a fertile breeding ground for studies of atomic and molecular phenomenon. The 1997 Nobel Prize in physics, awarded for achievements in cooling and trapping atoms with laser light,[1–5] reflects the importance of such contributions. Indeed, laser cooling and trapping is now a widespread technique used in a variety of applications, ranging from high-resolution spectroscopy,[6–10] to quantum state control.[11–13]

Research areas utilizing laser-assisted cooling and trapping, such as quantum information, require precise control and measurement of atoms and molecules. Coherent population control plays an important role in *efficiently* preparing and populating target states. In a two-level, non-degenerate system, using incoherent light as a means of population transfer

will yield, at best, 50% population in the upper state, whereas coherent transfer techniques prove to be much more efficient and can yield close to 100% population transfer.[14] Thus, controlling coherent excitation is a key element in one's ability to effectively manipulate atomic and molecular populations. The practice of using laser radiation fields to prepare and select particular states for population transfer has been done for more than thirty years. Currently, interest in coherent excitation and effective population transfer can be found in a sweeping range of applications, such as manipulating quantum bits of memory (qubits)[15–17] or producing efficient laser-controlled chemical reactions.[18,19]

Optical adiabatic rapid passage (ARP), a coherent population control technique, has been demonstrated to be an effective method of efficiently transferring atomic or molecular populations from an initial state to a selected final state.[20–22] There are a variety of ARP schemes, each with distinct advantages and disadvantages. A robust ARP technique that is relatively insensitive to experimental conditions is known as stimulated Raman adiabatic passage (STIRAP), and it plays an important role in today's frontier advances in AMO physics.[23–25] Because the STIRAP technique plays a central role in the work presented here, some background regarding it's role in the quest for efficient population transfer will be of benefit.

1.1 A Brief History of STIRAP

During the 1970's, state selection was used to prepare atoms and molecules for experiments such as spectroscopy[26] and collision studies.[18] Various methods of state selection allowed experiments to be performed involving otherwise-inaccessible energy levels. New optical pumping techniques were introduced, such as Franck-Condon pumping (FCP), a method providing access to higher vibrationally excited levels.[27–30] By exciting a beam of molecules into an upper rovibrational state, they will decay spontaneously to the electronic ground state within a few tens of nanoseconds (see Fig. 1.1a). Spontaneous decay transition probabilities are governed by the Franck-Condon overlap of the vibrational wave functions, and as

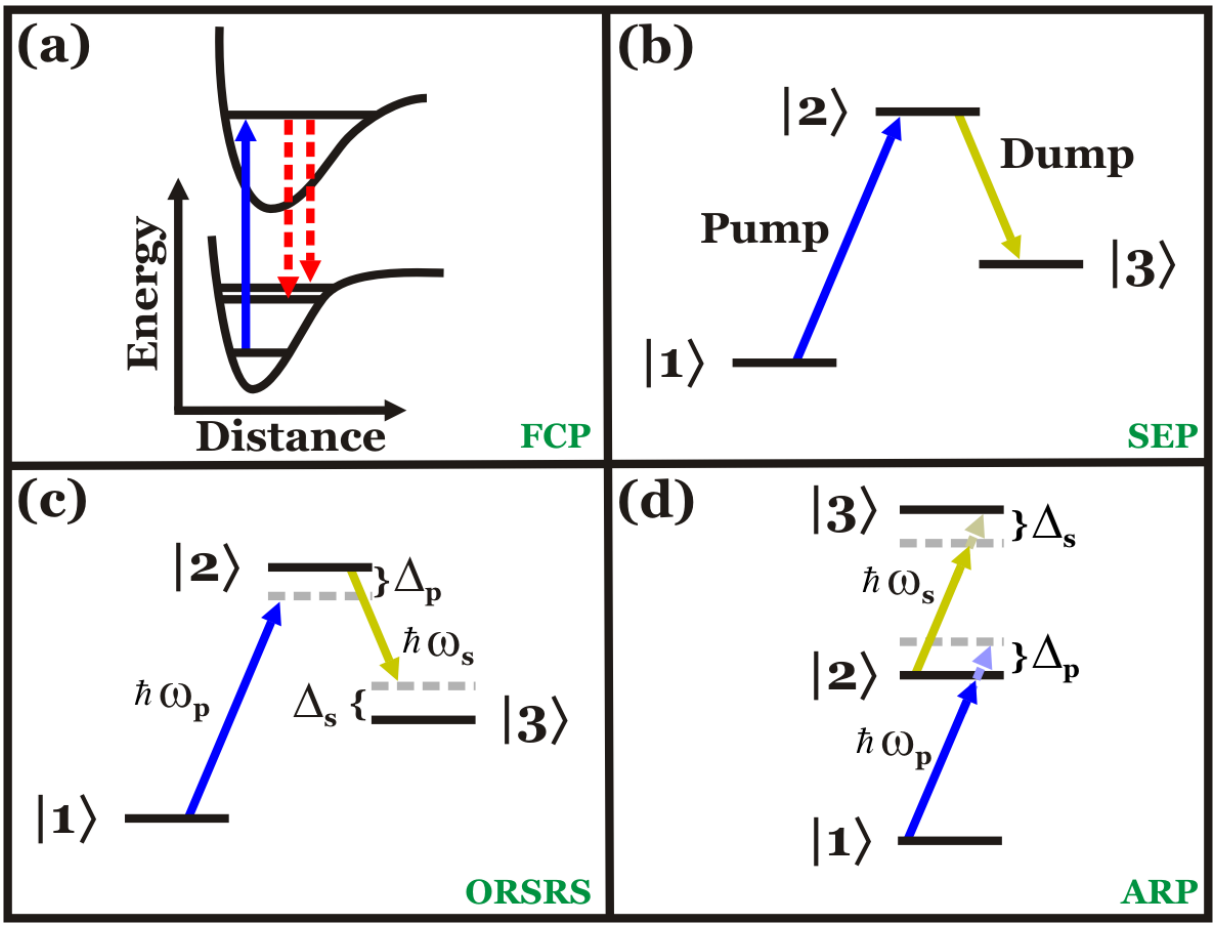

Figure 1.1: Widely-used coherent excitation methods. In these panels, $|1\rangle$ denotes the initial energy level where all population resides prior to the excitation process, $|2\rangle$ indicates the intermediate energy level, and $|3\rangle$ indicates the final terminal energy state after the population undergoes the indicated method of excitation. **(a)**, Franck-Condon pumping (FCP). **(b)**, Stimulated emission pumping (SEP). **(c)**, Off-resonant stimulated Raman scattering. **(d)**, Adiabatic rapid passage (ARP)

such, most transitions will occur at the outer turning points, causing much of the population to decay into a high vibrational level of the electronic ground state.[27]

In the early 1980's, a variety of state selection schemes were introduced. Stimulated emission pumping (SEP)[31–34] utilizes two lasers, one to "pump" population into an electronically excited intermediate state, and another to "dump" the excited population into a terminal state *via* stimulated emission, as shown in Fig. 1.1b. This "pump" and "dump" technique typically achieves population transfer of $\sim 10\%$, and provides more state-selectivity than

previous methods (*e.g.* Franck-Condon pumping).[31] SEP is used extensively in experiments such as stimulated emission studies,[35,36] molecular dynamics,[37–40] and high-resolution spectroscopy.[41–43] Similarly, overtone pumping (OTP)[44–47] is also used to populate vibrationally excited levels in the electronic ground state, and is used in experiments such as photoacoustic spectroscopy[48–50] and the study of optically pumped lasers.[51]

Off-resonance stimulated Raman scattering (ORSRS)[52–55] is similar to SEP, but the lasers are detuned from the resonant transitions (see Fig. 1.1c). ORSRS can be more efficient than SEP, yielding population transfers $\leq$50%. This optical pumping method can be used in experiments similar to those already mentioned (*e.g.* spectroscopy,[56]) and also has application in more unique experiments, such as the generation of sub-femtosecond pulse trains[57] or studying the properties of a Bose-Einstein condensate (BEC).[58]

While these experimental techniques were progressive at the time, each of these aforementioned methods suffer from general drawbacks, such as low population transfer efficiency, and from technical limitations. For example, the rapid falloff in overtone line strengths as vibrational excitation increases tends to reduce the general applicability of OTP.[36] During the 1990's, methods of adiabatic rapid passage (ARP) were recognized to be effective in state selective population transfer. ARP is achieved by starting with the transition laser detuned from resonance, and then sweeping the frequency through the resonance, as shown in Fig. 1.1d. Optical ARP has been used since the mid-1970's,[21,59] but more recent advances[60–63] in the control of experimental parameters such as pulse shape, relative pulse timing and frequency chirp have increased the success of using ARP to selectively transfer population.[64,65] ARP techniques routinely achieve $\sim$ 100% population transfer in some systems.

Two popular coherent excitation techniques include the use of π-pulses[66–69] to control population transfer, and stimulated Raman adiabatic passage (STIRAP).[23–25,70–73] The former scheme consists of carefully controlled laser fields directed at the target population for precisely one-half of a Rabi period, hence the name "π-pulse". During a single Rabi period,

the entire target population will be transferred from the initial state into the excited state, and back again. By precisely controlling the interacting laser field's intensity and duration, one can transfer the target population into the excited state, and then extinguish the laser light, thus "stranding" the population in the excited state until the atoms undergo spontaneous decay. A variety of related techniques also exist, such as 0π- and $\frac{\pi}{2}$-pulses.[74,75] Such schemes are used in a range of experiments, such as molecular spectroscopy,[76] multiphoton coupling studies,[77,78] and pulse-shaping techniques.[79–81] While π-pulse methods are highly efficient in population transfer, the difficulty in controlling the laser pulses on such a precise time scale can make the application of such a technique problematic. For example, averages over magnetic sublevels or intensity profiles prevent one from inducing the same excitation in each atom or molecule.[82]

STIRAP, on the other hand, is recognized to be a *highly robust* method of transferring population from the initial state to the terminal state of a system, with these same near-unity transfer efficiencies. STIRAP is a technique consisting of a series of laser pulses coupling three or more energy levels together. In the simplest three-level case, a Stokes laser is tuned into resonance, coupling the intermediate and terminal states of a system, while a pump laser is tuned into resonance from the ground state to the intermediate state of the system. The laser intensity can be pulsed while the laser frequency is held fixed, or the intensity can remain constant while the laser is chirped into resonance.

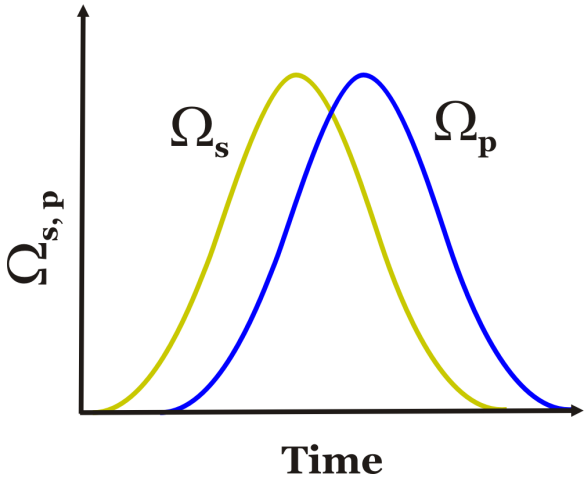

Figure 1.2: "Counter-intuitive" pulse order. The Stokes pulse (yellow line) preceeds the pump pulse (blue line) in such a case.

The pulses are arranged in a so-called "counter-intuitive" order, meaning the Stokes pulse precedes the pump pulse (see Fig. 1.2). The Stokes pulse dresses the system and, when the

pump pulse is applied and the Stokes field begins to diminish, population is transferred from the initial state directly to the terminal state with little or no population appearing in the intermediate energy level. STIRAP is known to be a relatively robust experimental technique. Comparatively large variations in the critical parameters governing the STIRAP process, such as the time delay between the Stokes and pump pulse or the relative intensities of each laser field, produce only small changes in the population transfer efficiency.

Variations on STIRAP exist, each with defining advantages or applications. While it is not the goal of this dissertation to cover in detail such exotic alternate transfer methods, it is of some value to note that the experimental techniques detailed herein are not the only ones available. In fact, any of the following variations, while perhaps not feasable with the experimental configuration relevant to this dissertation, could be studied in detail using the same general approach that will be discussed. Hence, some of these variations on the STIRAP method are briefly mentioned here.

Chirped adiabatic passage (CHIRAP)[61,62,83,84] uses frequency-swept ("chirped") laser pulses to selectively transfer populations into a terminal state. Such a scheme is depicted in Fig. 1.3a. The pulses used need not be perfectly Gaussian, nor is the frequency chirp required to be precisely linear.[83] This is one of the more common methods of STIRAP usually applied using narrow-bandwidth cw lasers, although broadband, short-pulse lasers can also be used to selectively populate closely-spaced energy levels.[61] The direction of chirp, in such cases, will determine which energy levels are populated.

Alternating stimulated Raman adiabatic passage (A-STIRAP)[82] is applied by fixing the frequency of the laser fields involved and temporally varying their intensities instead (see Fig. 1.3b). In an n-level system, laser light is applied first to odd-numbered energy level couplings, *e.g.* $|2\rangle \rightarrow |3\rangle$, $|4\rangle \rightarrow |5\rangle$, *etc.* Laser fields coupling the even-numbered states are applied next, as the odd-energy coupling fields are still present, but declining. Such a scheme is shown in Fig. 1.3b as pairs of pulses occurring at the same time, but coupling different energy states. This "alternating" approach can couple an arbitrary number of

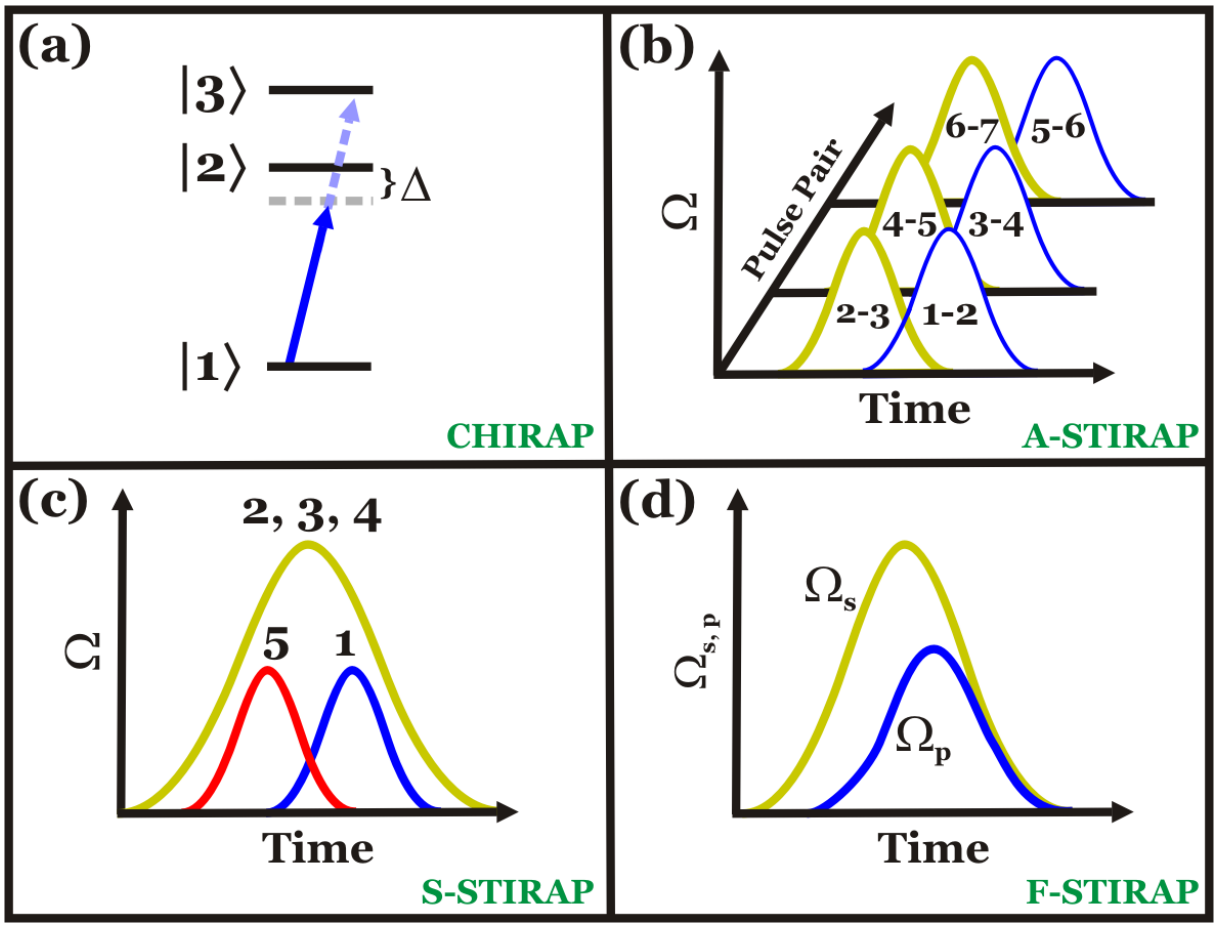

Figure 1.3: Typical variations on the STIRAP method. **(a)**, Chirped STIRAP (CHIRAP). **(b)**, Alternating-STIRAP (A-STIRAP). **(c)**, Straddling-STIRAP (S-STIRAP). **(d)**, Fractional-STIRAP (F-STIRAP).

energy levels together, selectively populating the terminal state. This dissertation discusses measurements of coherent population transfer in an experimental system employing the A-STIRAP technique in the most simple form involving three energy levels. In the following chapters, the term "STIRAP" will be used, implying this particular technique, and the experimental details of such a scheme will be discussed in detail.

Straddling stimulated Raman adiabatic passage (S-STIRAP)[85,86] is similar to A-STIRAP in that the laser fields are kept at a constant frequency and the intensities are swept, as shown in Fig. 1.3c. However, the initial (pump) and terminal (Stokes) fields, shown in the figure as blue and red curves, respectively, are straddled by all remaining intermediate laser

coupling fields, shown as a combined yellow curve. The intermediate laser pulse widths are wide enough to cover both the Stokes and pump laser fields. The Stokes and pump lasers are pulsed in the counter-intuitive order mentioned previously (see Fig. 1.2). A variation of the S-STIRAP technique simply keeps all of the intermediate coupling fields continuously present, while only the Stokes and pump lasers are pulsed.

Fractional stimulated Raman adiabatic passage (F-STIRAP),[87,88] is very similar to the traditional A-STIRAP technique in that the Stokes pulse preceeds the pump pulse temporally, however the Stokes and pump fields terminate simultaneously while maintaining a constant ratio of amplitudes as the field strength decreases (see Fig. 1.3d).[89] The advantage to this technique is that it creates a coherent superposition of the ground and terminal energy state during the entire time both Stokes and pump fields are present. This has application in several interesting fields of study, *e.g.* electromagnetically induced transparency (EIT).[90]

Stimulated hyper-Raman adiabatic passage (STIHRAP)[91–93] replaces the single-photon pump field with a two-photon resonant pump field, and either a one- or two-photon Stokes field. Thus, a traditional STIRAP approach where both the Stokes and pump lasers are single photon couplings is referred to as $(1 + 1)$ STIRAP. A two-photon pump and single photon Stokes system is denoted as $(2 + 1)$ STIHRAP, and a two-photon pump and two-photon Stokes scheme would be identified as $(2 + 2)$ STIHRAP.[91] Each of these cases are shown in Fig. 1.4. Many molecules of experimental interest require high energy fields to reach the first electronic excited state and, hence, such a two-photon approach would be useful.

1.2 Recent Work Involving STIRAP

Although STIRAP was initially implemented in the mid-1980's, it is still used heavily as a reliable and robust method of population transfer. New variations and deeper understanding involving adiabatic rapid passage continue to unfold. For example, in 2000, Ishkhanyan[94]

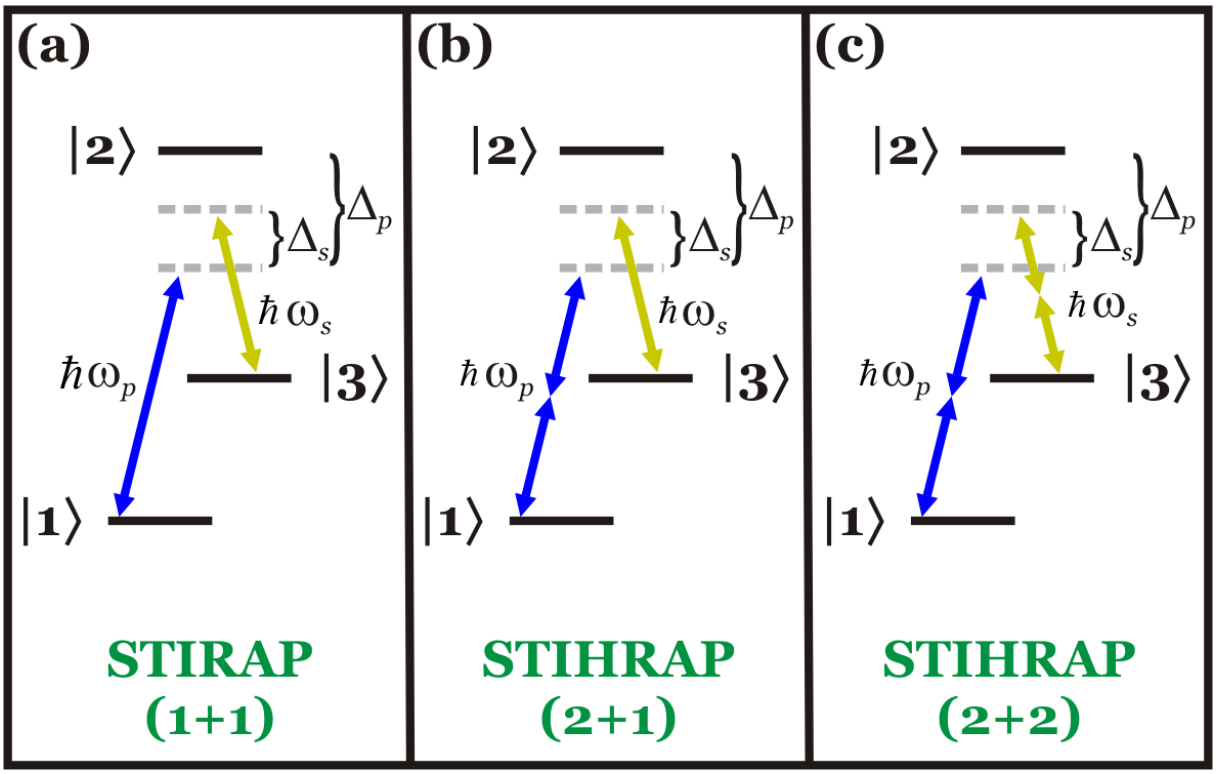

Figure 1.4: STIRAP *versus* STIHRAP. **(a)**, Typical (1+1) STIRAP. **(b)**, (2+1) STIHRAP, where the Stokes laser is still a single-photon transition. **(c)**, (2+2) STIHRAP, where both the Stokes and pump beams are now two-photon resonant transitions.

proposed the idea of using STIRAP as a means of preparing atoms for an atomic beam-splitting technique. This allows one to use standing waves to narrow the interference fringes in diffraction of such prepared atoms.

Three-level systems undergoing STIRAP have been studied in detail,[95,96] and n-level applications have also been widely discussed for some time.[85,97–105] For example, Sola *et al.*[86] have recently studied n-level systems and a corresponding variety of different mechanisms for population transfer, providing an excellent assortment of tools for exploring efficient population transfer. Also, Unanyan *et al.*[106] proposed an n-component maximal coherent superposition state *via* STIRAP techniques. This work has applications in areas such as atom interferometry.

The adiabatic limits of STIRAP have also been explored recently.[96,107,108] Such work has

provided a tool for designing pulse sequences which enable maximum population transfer where previously, using ordinary STIRAP pulse techniques, poor transfer efficiency was achieved. This work highlights a *disadvantage* of STIRAP, namely "the requirement that the average Rabi frequency of the pulses must be large compared to the radiative decay rates of the intermediate levels,"[108] and provides a solution in the form of "loop-STIRAP", a method to overcome the usual adiabatic conditions inherent in STIRAP.

A host of recent experiments have shown the importance of STIRAP in today's experimental research.[109–115] Several excellent overviews of coherent excitation and STIRAP now exist.[71–73,116] Furthermore, several recent experiments compare and contrast various coherent excitation techniques, providing great insights in the advantages and disadvantages of each method.[60,117,118]

1.3 Overview of Dissertation

The importance of coherent excitation is a recurring theme in the work described above. The inability to monitor atomic or molecular populations as they undergo coherent excitation adds a frustrating level of difficulty to the already-complicated task of efficiently populating a target state. A complete understanding of the population dynamics involved in these experiments is required in order to make efficient use of the techniques proposed for preparing and manipulating atoms or molecules.

This dissertation is a reflection of such a requirement. The work presented here illustrates a unique and powerful tool for measuring population dynamics by applying magneto-optical trap recoil ion momentum spectroscopy (MOTRIMS) to coherent excitation processes in order to measure population dynamics. The techniques presented here yield unprecedented measurements of dynamic population transfer on a very short (nanosecond) timescale. The information gleaned from these methods allows one to determine precisely how much population is in a given state at any time during the excitation process. The technique is not limited by the target species, lifetimes of the involved states, or even which states are popu-

lated, but only by the energy separation of the levels of interest, which must be greater than about 100 meV to be resolved. This method produces a quantitative dynamic measurement of coherent processes that otherwise cannot be observed.

Furthermore, many of the experiments using STIRAP for efficient population transfer rely on measurements of only a single, or at most, two levels, as a diagnostic for the efficiency of coherent population transfer. The techniques presented here will highlight the potentially misleading results derived by such measurements. The methods shown here do not suffer from a need to deduce populations in any unmeasured levels — all states can potentially be monitored.

A theoretical framework will first be presented in this dissertation, explaining the techniques and physical processes involved, and experimental results will then be shown supporting the claim that multilevel population dynamics can be measured as a function of time. The chapters in this dissertation are arranged as follows:

- **Chapter 2: Theoretical Modelling of Population Dynamics**

  Incoherent excitation will briefly be reviewed, along with varied treatments of 2-level coherent excitation. A theoretical model for the 3-level atomic structure studied in this dissertation will then be presented, using a density matrix approach. An extensive theoretical study of the experimental parameters will be presented, demonstrating the effect of certain critical variables on efficient population transfer.

- **Chapter 3: Experimental Setup**

  The experimental apparatus will be described in detail, including a brief overview of RIMS in general, and a specific review of MOTRIMS. A detailed explanation of laser control and data acquisition will also be presented. Technical details such as peak-locking methods and ion beam drift correction will be included in this chapter.

- **Chapter 4: Experimental Results**

  An explanation of how the data were analyzed will be presented, including details on measuring the $\frac{\sigma_{dd}}{\sigma_{ss}}$ relative cross-section. The experimental data themselves will be shown, along with discussions as to the significance of such measurements. Finally, resolution issues and error analysis will also be presented.

- **Chapter 5: Conclusion and Outlook**

  A general description of future experimental work based on the efforts presented here will be discussed. Some preliminary results from such spin-off projects will be shown without any detailed explanations as to method or technique.

The appendices at the end of the dissertation contain valuable reference material cited within the text, such as circuit diagrams, computer code, *etc.* As a whole, therefore, this dissertation describes the motivation for measuring population dynamics in such coherent excitation processes as previously described herein, and provides both a theoretical and experimental framework for carrying out such detailed measurements in the future.

Chapter 2

Theoretical Modelling of Population Dynamics

The goal of this dissertation, as mentioned in the previous chapter, is to present a framework for studying coherent excitation processes on short (few nanosecond) timescales. The technique presented in this thesis measures the coherent excitation of, and spontaneous emission from, any given state of a system. Supporting data are shown in Chap. 4 from a series of experiments designed to measure populations in a three-level system as a function of time.

This chapter presents a theoretical model used to calculate population dynamics under various experimental conditions. The methods of modeling such dynamics are shown, along with specific results for a variety of conditions. The mathematical machinery used in these calculations can be found in a number of texts describing incoherent and coherent population transfer,[116,119,120] as well as other resources describing density matrices.[101,121,122] For completeness, Appendix A contains a general overview of incoherent *versus* coherent excitation, as well as a more complete derivation of the equations used in this chapter.

2.1 Incoherent *vs.* Coherent Excitation

For a three-level system with equal numbers of degeneracies in each state, the classical rate equations can be written as

$$\dot{n}_1 = A_{21}n_2 + \frac{B_{21}I_1n_2}{c} - \frac{B_{12}I_1n_1}{c} \tag{2.1}$$

$$\dot{n}_2 = A_{32}n_3 - A_{21}n_2 + \frac{B_{12}I_1n_1}{c} + \frac{B_{32}I_2n_3}{c} - \frac{B_{21}In_2}{c} - \frac{B_{23}In_2}{c} \tag{2.2}$$

$$\dot{n}_3 = \frac{B_{23}I_2n_2}{c} - A_{32}n_3 - \frac{B_{32}I_2n_3}{c}, \tag{2.3}$$

where n_1, n_2, and n_3 represent the populations in levels $|1\rangle$, $|2\rangle$, and $|3\rangle$, respectively. The Einstein A- and B-coefficients mathematically introduce stimulated and spontaneous physical processes into the general incoherent rate equations. A_{nm}-coefficients represent spontaneous emission losses from energy level n to m. B_{nm}-coefficients where $n > m$ represent stimulated emission from state n to state m, while coefficients with $n < m$ represent stimulated absorption from state n to state m. Other loss terms, such as ionization, could be included, but for this treatment, the system is closed, and thus population is conserved between the three levels.

The steady-state solution places 1/3 of the population in each level. It is evident that incoherent excitation is a very inefficient transfer mechanism. At best, the population transfer into each energy state is $1/n$, where n is the number of energy levels involved in the excitation process (assuming an equal number of degeneracies for each state).

More importantly, this simple rate equation picture does not account for effects incurred by using a *coherent* radiation source. In order to understand coherent excitation processes, a different approach must be used.

2.2 2-Level Coherent Excitation

The time-dependent Schrödinger equation (TDSE) is a logical place to begin when including the effects of coherent excitation in such an atomic system. For the treatment of the 2-level atom, it will also be of benefit to simplify the study by temporarily neglecting spontaneous emission from the excited state. The TDSE and the Hamiltonian for the 2-level case are

$$i\hbar\frac{\partial}{\partial t}\Psi(t) = \mathscr{H}'(t)\Psi(t) \tag{2.4}$$

$$\mathscr{H}'(t) = \mathscr{H}_0 + \hat{V}(t), \tag{2.5}$$

where $\mathscr{H}_0$ is the Hamiltonian in the absence of any external fields, and $\hat{V}(t)$ is the time-dependent external radiation field, here taken as classical. In order to make use of the TDSE, one must find the time-derivative of $\Psi(t)$:

$$\mathscr{H}'\Psi(t) = \sum_n [\mathscr{H}_0 + \hat{V}(t)]c_n(t)\psi_n e^{-i\xi_n(t)}. \tag{2.6}$$

where ψ_n satisfies the time-independent Schrödinger equation, $\mathscr{H}_0\psi_n = E_n^0\psi_n$, and $\xi_n(t)$ represents a time-dependent phase. The presence of $\hat{V}(t)$ acting on ψ effectively redistributes the probability governing excitation:

$$\begin{aligned}\hat{V}(t)\psi_n &= V_{1n}(t)\psi_1 + V_{2n}(t)\psi_2 + \cdots \\ &= \sum_m V_{mn}(t)\psi_m,\end{aligned} \tag{2.7}$$

and can be expressed in conventional Dirac notation as

$$V_{qn}(t) = \int \psi_q^*\hat{V}(t)\psi_n d\tau = \langle\psi_q|\,\hat{V}(t)\,|\psi_n\rangle \equiv \langle q|\,\hat{V}(t)\,|n\rangle\,. \tag{2.8}$$

The original TDSE (Eqn. 2.4) can now be rewritten as

$$i\hbar\sum_n \psi_n[\dot{c}_n - i\dot{\xi}_n c_n]e^{-i\xi_n}\sum_n c_n[E_n^0\psi_n + \sum_m c_n\psi_m V_{mn}]e^{-i\xi_n}, \tag{2.9}$$

where the explicit time-dependence of $\dot{c}_n(t)$, $c_n(t)$, $\dot{\xi}_n(t)$, $\xi_n(t)$, and $V_{mn}(t)$ has been left off for simplicity. Utilizing the orthonormality of ψ_n,

$$\hbar\dot{c}_l = -i[(E_l^0 - \hbar\dot{\xi}_l)c_l + \sum_n c_n V_{ln}e^{-i(\xi_n - \xi_l)}] \tag{2.10}$$

2.2.1 First Approximation: Near-Resonance

Up to this point, such treatment of the Schrödinger equation has been exact. For the case of an atom in a radiation field, the the Schrödinger equation cannot be solved exactly. Here, however, the first of three approximations will be introduced, allowing us to solve the system of coupled differential equations. The system of interest is composed of only two energy levels coupled by the laser frequency. One approach to approximate a finite-level situation, first studied by I. I. Rabi,[173] is to truncate the sum in Eqn. 2.10, thus including only the same number of terms as there are energy levels. That is, for a two-level system, the indices n and l will range from 1 to 2. This is a reasonable approximation because the laser frequency used for excitation will typically be narrow-band and near-resonant, and thus the contribution to other energy levels will be negligible. It makes little sense, therefore, to keep track of interactions with energy levels far from resonance. Thus, the two terms $\hbar\dot{c}_1$ and $\hbar\dot{c}_2$ become:

$$\begin{aligned} \hbar\dot{c}_1 &= -i[(E_1^0 - \hbar\dot{\xi}_1)c_1 + V_{12}c_2e^{-i(\xi_2-\xi_1)}] \\ \hbar\dot{c}_2 &= -i[(E_2^0 - \hbar\dot{\xi}_2)c_2 + V_{21}c_1e^{+i(\xi_2-\xi_1)}]. \end{aligned} \tag{2.11}$$

It is more convenient to rewrite such terms in matrix form:

$$\hbar\dot{\mathbf{c}} = -i\begin{pmatrix} E_1^0 + V_{11} - \hbar\dot{\xi}_1 & V_{12}e^{-i(\xi_2-\xi_1)} \\ V_{12}^*e^{+i(\xi_2-\xi_1)} & E_2^0 + V_{22} - \hbar\dot{\xi}_2 \end{pmatrix}\mathbf{c}, \tag{2.12}$$

where

$$\mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}. \tag{2.13}$$

2.2.2 Second Approximation: Electric Dipole

Typically, the optical wavelength, λ, of the radiation interacting with the atom will be much larger than the distances involving the wavefunctions of the atom. For example, coherent

light with $\lambda \simeq 800$ nm interacts with an typical atom where the wavefunctions are contained within a sphere of radius < 1 nm. The electric field, $\mathbf{E}(\mathbf{r}, t)$, can therefore be treated as spatially uniform with respect to the atom. This is called the *dipole approximation*.[116,119,120] The electric field operator for a generalized plane wave has the form

$$\mathbf{E}(\mathbf{r}', t) = \frac{1}{2} E_0 [e^{i(\mathbf{k}\cdot\mathbf{r}' - \omega t)} + e^{-i(\mathbf{k}\cdot\mathbf{r}' - \omega t)}], \tag{2.14}$$

where E_0 is the maximum amplitude of the electric field, $\mathbf{k}$ is the propagation vector, and $\mathbf{r}'$ is the position vector. The magnitude of the propagation vector, $k = \frac{2\pi}{\lambda}$, is called the wave number. The exponential terms involving the propagation vector in Eqn. 2.14 can be expanded as

$$e^{i\mathbf{k}\cdot\mathbf{r}'} = 1 + (i\mathbf{k}\cdot\mathbf{r}') + \frac{1}{2!}(i\mathbf{k}\cdot\mathbf{r}')^2 + \ldots. \tag{2.15}$$

Applying the dipole approximation, therefore, truncates the exponential expansion such that $e^{i\mathbf{k}\cdot\mathbf{r}'} \simeq 1$. The dipole approximation can be applied so long as $ka \ll 1$, where a is the typical linear dimensions of the atomic wave functions. In the dipole approximation, then,

$$\hat{V} = -e\mathbf{E}(\mathbf{r}, t) \cdot \mathbf{r}, \tag{2.16}$$

and one can write $V_{mn}(t)$ explicitly as

$$\begin{aligned} V_{mn}(t) &= -e\mathbf{E} \langle m| \mathbf{r} |n\rangle \\ &= \frac{e}{2} E_0 (e^{i\omega t} + e^{-i\omega t}) \langle m| \mathbf{r} |n\rangle . \end{aligned} \tag{2.17}$$

Here, it is useful to introduce the Rabi frequency, defined as[23]

$$\Omega \equiv \frac{-eE_0}{\hbar} \langle e| \mathbf{r} |g\rangle , \tag{2.18}$$

where $\langle e|$ is the excited state, $|g\rangle$ is the ground state, and $\mathbf{r}$ is the electron coordinate. Substituting this definition of Ω into Eqn. 2.17 produces a more compact solution for $\hbar\dot{\mathbf{c}}$. One can reduce the complexity further by defining the energy levels

$$\begin{aligned} E_1 &\equiv E_1^0 + V_{11} \\ E_2 &\equiv E_2^0 + V_{22}. \end{aligned} \tag{2.19}$$

Now,

$$\hbar\dot{\mathbf{c}} =$$

$$-i \begin{pmatrix} E_1 - \hbar\dot{\xi}_1 & \frac{1}{2}\hbar\Omega(e^{-i(\xi_2-\xi_1-\omega t)} + e^{-i(\xi_2-\xi_1+\omega t)}) \\ \frac{1}{2}\hbar\Omega^*(e^{-i(\xi_2-\xi_1-\omega t)} + e^{-i(\xi_2-\xi_1+\omega t)}) & E_2 - \hbar\dot{\xi}_2 \end{pmatrix} \mathbf{c}.$$

(2.20)

The phases ξ_n and their time-derivatives are arbitrary. For convenience, one can select the arbitrary phases appropriately so as to simplify the problem:

$$\xi_2 - \xi_1 = \omega t, \tag{2.21}$$

$$\dot{\xi}_2 - \dot{\xi}_1 = \omega. \tag{2.22}$$

Therefore,

$$\hbar\dot{\mathbf{c}} = -i \begin{pmatrix} E_1 - \hbar\dot{\xi}_1 & \frac{1}{2}\hbar\Omega(e^{-2i\omega t} + 1) \\ \frac{1}{2}\hbar\Omega^*(e^{2i\omega t} + 1) & E_2 - \hbar\dot{\xi}_2 \end{pmatrix} \mathbf{c}, \tag{2.23}$$

which is certainly more aesthetically pleasing. More importantly, this choice of phase allows one to make an important approximation.

2.2.3 Third Approximation: Rotating Wave

The third and final approximation is now applied: the Rotating Wave Approximation (RWA). Here, because of the phase choice, one of the two exponential terms in Eqn. 2.20 was replaced with unity. The RWA, allows one to ignore high-frequency oscillations and replace

$$e^{\pm 2i\omega t} + 1 \simeq 1, \tag{2.24}$$

such that

$$\hbar\dot{\mathbf{c}} = -i \begin{pmatrix} E_1 - \hbar\dot{\xi}_1 & \frac{1}{2}\hbar\omega \\ \frac{1}{2}\hbar\Omega^* & E_2 - \hbar\dot{\xi}_2 \end{pmatrix} \mathbf{c}. \tag{2.25}$$

The laser field coupling energy states $|1\rangle$ and $|2\rangle$ can be detuned from resonance by some amount defined as

$$\begin{aligned} E_1 - \hbar\dot{\xi}_1 &\equiv \hbar\Delta_1 \\ E_1 - \hbar\dot{\xi}_2 &\equiv \hbar\Delta_2 \\ &= E_2 - E_1 + \hbar\Delta_1 - \hbar\Omega \end{aligned} \tag{2.26}$$

Defining the zero on the energy axis to be $\Delta_1 \equiv 0$ simplifies the picture somewhat. Including such detuning definitions, and dividing both sides by $\hbar$, provides a greatly simplified equation for $\dot{c}_1$:

$$\dot{\mathbf{c}} = -\frac{i}{2} \begin{pmatrix} 0 & \Omega \\ \Omega^* & 2\Delta_2 \end{pmatrix} \mathbf{c}. \tag{2.27}$$

Solving Eqn. 2.27 for c_1 and c_2 is now straightforward, given the initial boundary conditions $c_2(0) = 0$ and $c_1(0) = 1$. Such initial conditions indicate that all the population is in the ground state at time $t = 0$. Solving for c_2 yields

$$c_2 = -i\frac{\Omega}{\Omega'}\sin^2\left(\frac{\Omega' t}{2}\right)e^{-i\frac{1}{2}\Delta_2 t}. \tag{2.28}$$

Here, $\Omega' \equiv \sqrt{\Omega^2 + \Delta_2^2}$. Finally, the probability, $P_2(t)$, of finding an atom in state $|2\rangle$ is

$$\begin{aligned} P_2(t) &= c_2^* c_2 \\ &= \left(\frac{\Omega}{\Omega'}\right)^2 \sin^2\left(\frac{\Omega' t}{2}\right) \\ &= \frac{\Omega^2}{2\Omega'^2}[1 - \cos(\Omega' t)] \end{aligned} \tag{2.29}$$

The probability of placing an atom in (or removing it from) the excited state oscillates at a frequency of Ω' which is therefore known as the flopping frequency. Note that for resonant radiation, $\Delta_2 = 0$ and $\Omega' = \Omega$, the Rabi frequency. As Δ_2 increases, the frequency of oscillation also increases, although the amplitude decreases.

2.3 3-Level Coherent Excitation

Thus far, it has been shown that 2-level coherent excitation can be derived directly from Schrödinger's equation. Up to this point, however, spontaneous emission has been conspicuously absent from the treatment of coherently excited populations. As the concepts presented previously are expanded, and the 3-level atom is studied, it is no longer desirable to neglect such an effect.

It is beneficial to approach the 3-level atom by introducing density matrices. Many treatments of multi-level coherent excitation utilize density matrices,[116,119,131,176,177] and there are several advantages to such an approach. For example, it allows one to add decay terms into the system phenomenologically, and to estimate decoherence effects within the system. Such information is important in understanding the adiabatic *versus* diabatic processes taking place under different experimental conditions.

2.3.1 Density Matrices

As noted above, there have been several general treatments of density matrices applied to coherent excitation systems. A general overview[169] will be presented here, after which the 3-level system of interest throughout the remainder of this dissertation will be discussed.

The density matrix is defined to be

$$\rho(t) = |\psi\rangle \langle\psi| . \tag{2.30}$$

There are several important things to note about the density matrix. First, it is Hermitian. Second, because the wavefunction is normalized, the trace is unity. Third, one can diagonalize ρ with a unitary transformation. The diagonal terms represent the probabilities of the system states. If $Tr(\rho^2) = 1$, the system is said to be in a pure state, meaning one of the diagonal terms must be 1, and the rest zero. Conversely, if $Tr(\rho^2) < 1$, then the system is said to be in a mixed state, where it is represented by the density operator

$$\rho = \sum_i p_i |\Psi_i\rangle \langle\Psi_i| . \tag{2.31}$$

where p_i is the probability of finding the system in state Ψ_i. Mixed states will become important momentarily, since part of the motivation for using density matrices is to allow one to include the effects of spontaneous emission. The quantum Louisville equation is

$$i\hbar\dot{\rho}(t) = [\mathscr{H}, \rho(t)] . \tag{2.32}$$

This is the basis of our formalism, however no decay phenomena have been included yet. The complete quantum Louisville equation, including spontaneous emission, is written as

$$i\hbar\dot{\rho}_{ab}(t) = [\mathscr{H}_L, \rho(t)]_{ab} - i\hbar [\Gamma\rho(t)]_{ab} . \tag{2.33}$$

and details on how this equation is derived may be found in Appendix A.3.1.

It is useful to first assume the system is in two-photon resonance; that is, $\Delta_2 = 0$, and study the eigenvectors for the three-level case as done previously in the dressed-state approach for the two-level case. The angular relations are depicted in Fig. 2.1. Ω' is still defined to be the Rabi flopping frequency (see Fig. 2.1a), however, note that Δ_1 represents the single-photon detuning, and the angle involved is now labeled ϕ, while the angle θ now corresponds to the "two-photon Rabi frequency", as shown in Fig. 2.1b. ϕ is known as the "mixing angle", relating the component of the ground state mixed with the components of the excited states. This change of notation, moving from the 2-level to the 3-level treatment, will be used for the remainder of the dissertation, and is consistent with the literature.

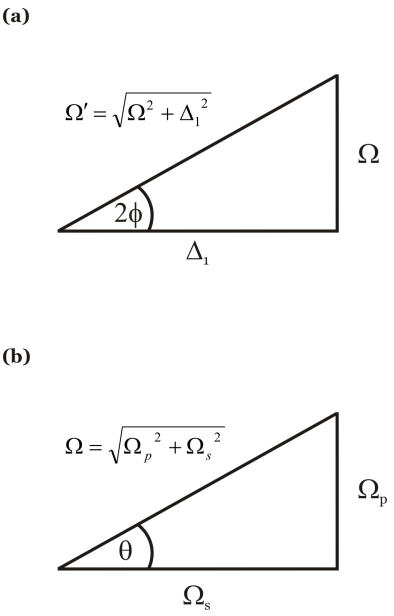

Figure 2.1: Visual interpretation of the angles ϕ and θ in the 3-level dressed-state approach.

Using a "dressed-state approach" (an example of which may be found in Appendix A.2.4) one obtains the normalized eigenvalues of the 3-level system to be

$$\lambda_- = -\frac{\hbar}{2}\Omega\tan(\phi) \tag{2.34}$$

$$\lambda_0 = 0 \tag{2.35}$$

$$\lambda_+ = -\frac{\hbar}{2}\Omega\cot(\phi), \tag{2.36}$$

and the corresponding eigenvectors are

$$\Phi_- = \begin{pmatrix} \sin\theta\cos\phi \\ -\sin\phi \\ \cos\theta\cos\phi \end{pmatrix} \tag{2.37}$$

$$\Phi_0 = \begin{pmatrix} -\cos\theta \\ 0 \\ \sin\theta \end{pmatrix} \tag{2.38}$$

$$\Phi_+ = \begin{pmatrix} \sin\theta\sin\phi \\ \cos\phi \\ \cos\theta\sin\phi \end{pmatrix}. \tag{2.39}$$

If the system state vector is represented as

$$\Psi(t) = A_-(t)\Phi_-(t) + A_0(t)\Phi_0(t) + A_+(t)\Phi_+(t), \tag{2.40}$$

then

$$\dot{\mathbf{A}} = -\frac{i}{\hbar}\begin{pmatrix} \lambda_- & -i\hbar\dot{\theta}\cos\phi & -i\hbar\dot{\phi} \\ i\hbar\dot{\theta}\cos\phi & \lambda_0 & i\hbar\dot{\theta}\sin\phi \\ i\hbar\dot{\phi} & -i\hbar\dot{\theta}\sin\phi & \lambda_+ \end{pmatrix}\mathbf{A} \tag{2.41}$$

Obtaining the 3-level Hamiltonian, $\mathscr{H}_L$, is done the same way as in the 2-level case (Eqn. 2.27) and results in

$$\mathscr{H}_L = \frac{\hbar}{2}\begin{pmatrix} 0 & \Omega_p & 0 \\ \Omega_p^* & 2\Delta_1 & \Omega_s \\ 0 & \Omega_s^* & 2\Delta_2 \end{pmatrix} \tag{2.42}$$

where Ω_p and Ω_s are defined to be the "pump" and "Stokes" Rabi frequencies for one- and two-photon resonances, respectively. Likewise, the detunings from such one- and two-photon resonant cases are Δ_1 and Δ_2, respectively, as shown in Fig. 2.2. The notation here is different than shown previously for the two-level case, conforming to the three-level nomenclature used throughout the literature. Placing this 3-level Hamiltonian in the quantum Louisville equation, one now has a complete description of the population dynamics for the system, including spontaneous emission losses from excited states.

This important difference from the previous two- and three-level cases shown allows one to gain a more complete picture of the dynamics taking place during transitions within the system. As mentioned previously (see Sec. 2.3.1), the density matrix produces $n^2 - 1$ coupled, first-order differential equations to describe such population dynamics within each level of the system, along with the coherence coupling between states. Therefore, the 3×3 matrix for the 3-level case produces 8 coupled, first-order differential equations to be solved.

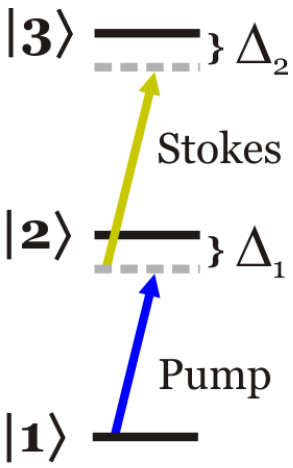

Figure 2.2: A generic energy level diagram showing a 3-level system.

2.4 Modeling the 3-Level Case

A theoretical model was created, allowing one to input a series of experimental conditions and plot the expected behavior of each atomic state as a function of time. The model was written to be interactive and visual, giving real-time results in the laboratory. The model was *not* written to be especially fast or optimized for efficiency. The goal was only to have a robust theoretical model into which one could input experimental parameters and obtain an intuitive feel for which of these population transfer variables might be more sensitive than others. This section describes the theoretical program and the results obtained from modeling 3-level behavior under typical laboratory conditions.

2.4.1 Modeling in Mathematica

The Mathematica[123] software was chosen for real-time modeling of the 3-level atomic behavior. Appendix D contains the complete Mathematica code, along with limited documentation. The Hamiltonian and Louisville equations are input into Mathematica, along with a list of impossible Einstein-A coefficients which eliminate population transfer in such non-physical cases as spontaneous emission *from* the ground state *to* an excited state. Initial

conditions are also set to ensure that all population starts in the ground state, although this is not a rigid requirement. Because the experimental conditions shown later (see Chap. 4) could conceivably start with a fraction of the population in some excited state, future cases where initial populations are not restricted to the ground state could easily be examined theoretically with this same framework.

With this information, Mathematica has six equations to work with, representing the density matrix elements ρ_{11}, ρ_{12}, ρ_{13}, ρ_{22}, ρ_{23}, and ρ_{33}. Note that, because the Hamiltonian is Hermitian, there is no need to solve for the complex conjugate elements ρ_{21}, ρ_{31}, and ρ_{32}. Mathematica is then used to numerically solve this system of six complex coupled, first-order differential equations. The results are treated as interpolated functions.[124]

The initial conditions declared in Mathematica govern the specific case to be studied, and can be selected for any physical system of interest. For the remainder of this dissertation, the system of interest will be the ^{87}Rb isotope. The excitation scheme will be a 3-level ladder system similar to the one shown (generically) in Fig. 2.2, where state $|1\rangle$ refers to $^{87}\mathrm{Rb}(5s)_{1/2}$ ($F = 2$), $|2\rangle$ is $^{87}\mathrm{Rb}(5p)_{3/2}$ ($F = 3$), and $|3\rangle$ is $^{87}\mathrm{Rb}(4d)_{5/2}$ ($F = 4$). The pump laser field coupling the $5s$ to $5p$ states has a wavelength of 780 nm, while the Stokes field coupling $5p$ to $4d$ has a wavelength of 1529 nm. The Einstein-A coefficients, A_{21} and A_{32}, are 26.6 ns and 84.0 ns, respectively.

Since the experimental conditions utilize near-Gaussian pulses, the theory also utilizes Gaussian pulses of the form

$$\Omega(t)_{s,p} = \Omega_{0s,p}\, e^{-2.773\left(\frac{(t-t_0)\pm\frac{1}{2}\tau}{w}\right)^2}. \tag{2.43}$$

where Ω_0 is given by the expression for Ω in Eqn. A.43, t_0 is an arbitrary time constant, and τ is the delay between the pump and Stokes laser pulses. The factor of $\frac{1}{2}$ defines τ to be the distance between the centroids of each pulse. The pulse width, w, represents the full-width, half-maximum of the Gaussian pulse. The factor $4\ln\frac{1}{2} \simeq 2.773$ defines this relationship, thus at a time t such that $t - t_0 \pm \tau = \frac{1}{2}w$, the amplitude of the pulse will be $\frac{1}{2}\Omega_0$.

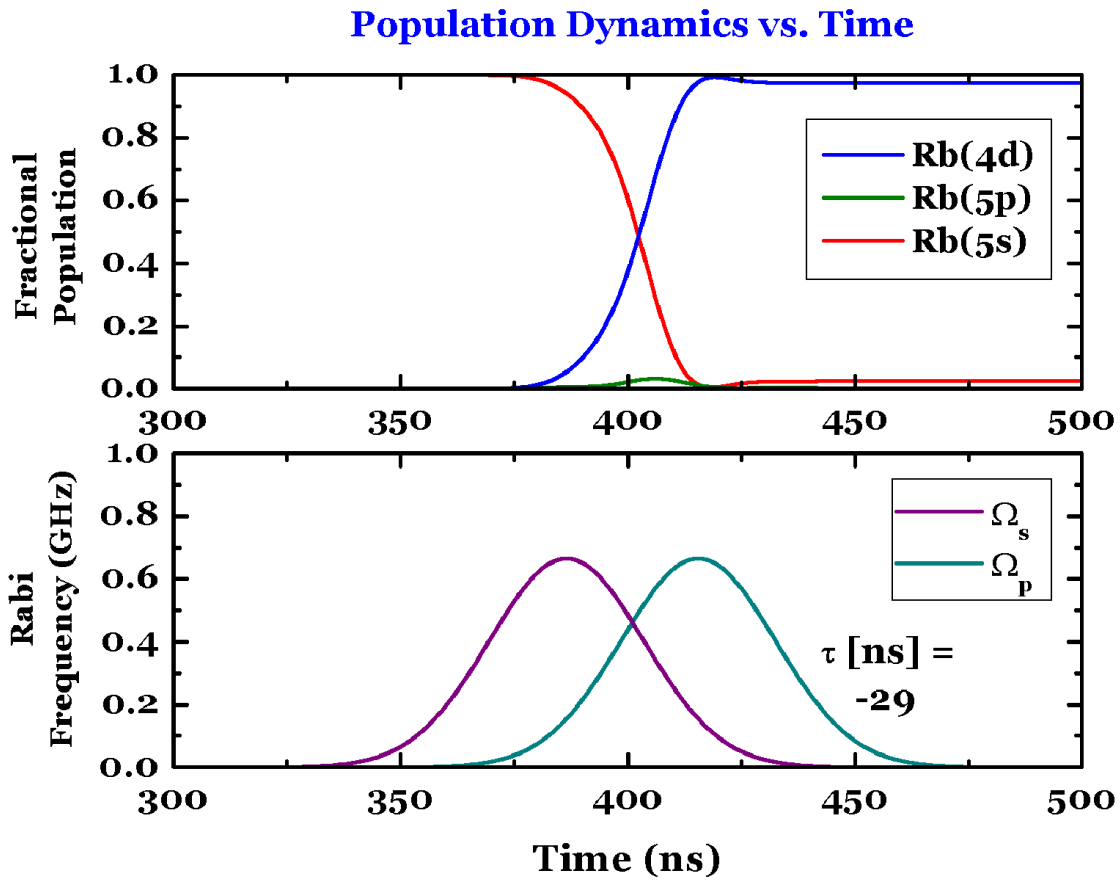

Figure 2.3: A typical theory plot showing population in the ^{87}Rb $5s$, $5p$, and $4d$ states, shown as red, green, and blue curves, respectively, where the spontaneous emission rates, A_{21} and A_{32} have been turned off. For this particular plot, parameters were $I_p = 1034\ mW/cm^2$, $I_s = 434\ mW/cm^2$, $w_p = w_s = 40$ ns, $\Delta_1/2\pi = 53$ MHz, $\Delta_2/2\pi = 0$ MHz, and $\tau = -29$ ns.

It should be noted that, while perfectly Gaussian pulses were chosen for study in this dissertation, this need not be the case.[125] Any pulse shape can be used in the code. However, as will be shown later, the experimental pulses are so nearly Gaussian, it is reasonable to model such an approximation in the theory.

With these definitions in place, Mathematica can solve the system of equations and produce interpolated functions representing the population dynamics and coherence effects as a function of time. First, the program can be tested by "turning off" spontaneous emission, as seen in Fig. 2.3. That is, the Einstein-A coefficients, A_{21} and A_{32}, that govern

the decay from the $5p$ and $4d$ energy levels, respectively, can be set to extremely small values, effectively removing decay from the excited states. Assuming the population is adiabatically transferred, Eqn. A.52 shows (for a two-level case) the entire ground state population should be placed in the terminal excited state, as indeed is the case in Fig. 2.3.

A typical output plot where decay is now present, Fig. 2.4, shows the population of the ^{87}Rb ground state, $(5s)$, middle excited state, $(5p)$, and the terminal state, $(4d)$, as red, green, and blue curves, respectively. Both Ω_p and Ω_s are also shown as functions of time. Here, τ was selected to be -29 ns.

The total population starts in the ground state. As Ω_2 increases, the system is dressed, coupling the $5p$ and $4d$ energy levels. Obviously, no population transfer occurs during this phase. After Ω_s peaks and begins to decrease, Ω_p is applied, and population is coherently transferred from the ground state (red curve) to the terminal excited state (blue curve). Initially, no population is placed in the intermediate $5p$ state, a clear indicator that the system is indeed undergoing coherent adiabatic population transfer *via* STIRAP. Later, the $5p$ state is fed by spontaneous emission loss from the $4d$ state. This characteristic lag in $5p$ population will also be seen experimentally in Sec. 4.1.2.

In this plot, about 83% of the population reaches the $4d$ state, while only about 18% reaches the $5p$ state, all of which is due to loss from the upper terminal state. In contrast, Fig. 2.5 shows a theoretical plot in the intuitive order, where the pulse delay is +35 ns. Here, it is evident that the $4d$ production is lessened considerably than that of the counterintuitive delay ordering, resulting in only about 57% $4d$-population. Also, the $5p$ population begins to increase immediately, indicating that it is being initially fed by excitation from the $5s$ ground state and not from the $4d$ terminal state. This is a clear indicator that the system is not undergoing coherent population transfer, but instead population is transferred conventionally, from the ground state to the middle state, and finally to the terminal excited state. It is evident, therefore, that τ has a great effect on the efficiency of population transfer.

The question naturally arises: what *combination* of parameters control the efficiency

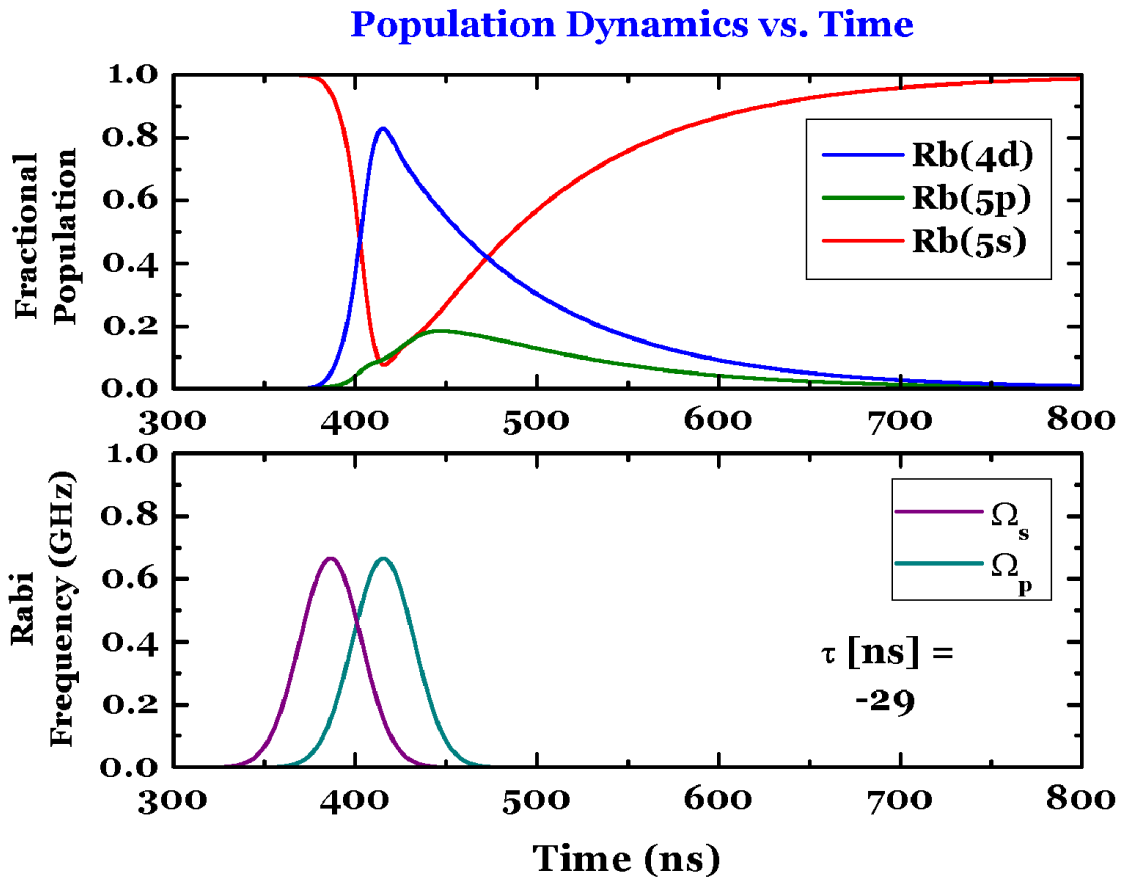

Figure 2.4: A typical theory plot showing population in the ^{87}Rb $5s$, $5p$, and $4d$ states, shown as red, green, and blue curves, respectively. The parameters used to generate this plot are identical to Fig. 2.3, except spontaneous emission is now present.

of STIRAP? This question is answered by studying the physical parameters involved in controlling the Rabi frequencies, namely, the intensities of the pump and Stokes lasers, I_p and I_s, respectively, the temporal widths of these pulses, w_p and w_s, the one- and two-photon detunings from resonance, Δ_1 and Δ_2, respectively, and the aforementioned pulse delay, τ.

These seven parameters are the critical variables governing the efficiency of STIRAP. Studying these seven parameters together is sufficient to generate a global picture of the general features involved in the STIRAP process. Each of these parameters will be studied in detail in the following sections. However, a more subtle (and much more difficult) question

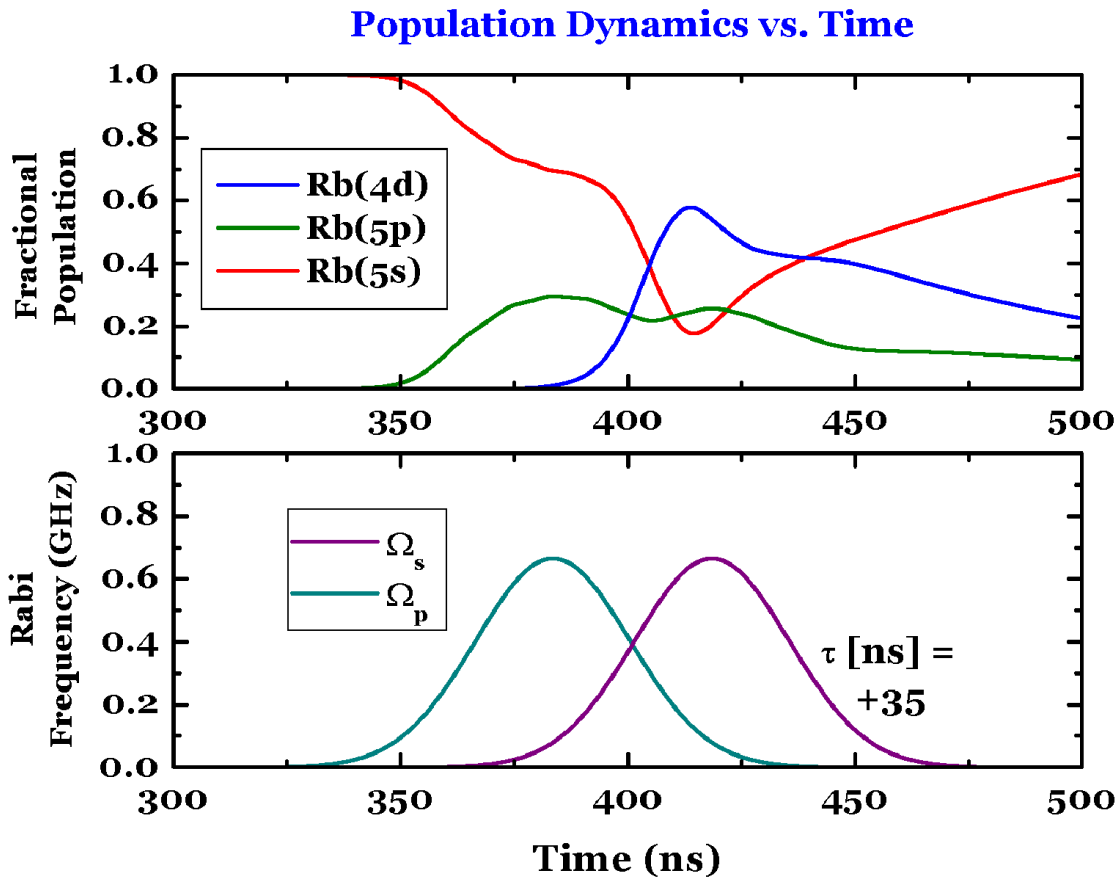

Figure 2.5: A typical positive pulse delay (intuitive order) plot. The parameters for this plot are the same as Figs. 2.3 and 2.4, except the pulse day was set to be +35 ns.

presents itself: How do these parameters *interrelate*? For example, how can one be sure that the reason for inefficient population transfer shown in Fig. 2.5 was due solely to the pulse delay being arranged in the intuitive order? Was it, perhaps, related to improper choices for two-photon detuning? Could the intensity of, say, the pump pulse have been weak enough to prevent efficient STIRAP?

The answer to such questions is not simple. Some relations can be easily detected by selecting reasonable values for six of the seven parameters, and varying the seventh, studying behavior as a function of time. However, "mapping out" the entire seven-dimensional parameter space is no simple task. While it is trivial to write a short program to step

through the entire 7D region of interest, analyzing the resulting data in a meaningful way is challenging to say the least.

2.4.2 Parameter Ranges

Some broad conclusions can be reached by studying the system when the parameters of interest are set in such a way to yield efficient population transfer. For example, the values used in Fig. 2.4 are reasonable to study general relationships between parameters. Before such a study, however, it is of benefit to delineate a reasonable range within which to study each variable of interest.

The pump and Stokes pulse widths, w_p and w_s, respectively, were studied for widths ranging from 33 ns to 100 ns. The lower limit was defined by experimental limitations, while the upper limit was selected because, experimentally, the interest was to show population transfer on the nanosecond timescale. Pulses larger than $\sim$ 100 ns, while technically feasable, would have required setting the experiment in a more coarse timescale, thus losing resolution. Such experimental details are discussed later. (See Chap. 3 and Sec. 4.3.)

The intensities for the pump and Stokes beams, I_p and I_s, were selected to range between 0 mW/cm^2 and 100 mW/cm^2. Intensities above 100 mW/cm^2 proved to be uninteresting, with population transfer already in saturation. One- and two-photon detunings, Δ_1 and Δ_2, were studied in a range -150 MHz to $+150$ MHz. The experimental technique used to achieve such detunings was the sole factor in selecting such a theoretical range of study. (See Secs. 3.2.2 and 3.2.3.) Other experimental techniques might be used, covering different ranges of detunings.

Finally, the pulse delay, τ, was selected to range from -200 ns to $+200$ ns. This corresponds to a reasonable overlap region between Ω_p and Ω_s for both intuitive and counter-intuitive ordering. Table 2.1 shows a summary of the ranges explored in this chapter.

To obtain a general feel for how certain parameters interact, one could monitor the efficiency of $4d$ production as a function of several parameter combinations. For example,

Parameter	Lower Limit	Upper Limit	Units
τ	-200	200	ns
I_p	0	100	mW/cm^2
I_s	0	100	mW/cm^2
$\Delta_p/2\pi$	-150	150	MHz
$\Delta_p/2\pi$	-150	150	MHz
w_p	33	100	ns
w_s	33	100	ns

Table 2.1: STIRAP Parameter Ranges

Fig. 2.6 is a plot of $4d$ production as a function of τ, *versus* time. Two distinct structures are visible, one in the counter-intuitive regime ($\tau < 0$), and one in the intuitive regime ($\tau > 0$). During the overlap time, two separate peaks occur, one at about 380 ns, and the other around 420 ns. This feature corresponds to coherent Rabi flopping of the population, as mentioned earlier in Sec. 2.2.3. In this particular case, the flopping is not dramatic, but if different parameters are selected for the other variables (intensity, detuning, and pulse width), a very different picture forms.

Figure 2.7 shows the fractional population in each level as a function of time. If the image is clicked, a short film is played in which τ is swept from -150 ns to $+150$ ns. When the pulses are nearly overlapped, the Rabi flopping is much more dramatic than in Fig. 2.6.

The conclusions reached, therefore, are limited to this *specific* choice of I_p, I_s, w_p, w_s, Δ_1 and Δ_2. Thus, while it is beneficial to study specific behavior under a specific set of initial conditions, the results cannot easily be applied in a more general sense. In other words, knowing how τ behaves in a specific case explains little of how τ might behave when other parameters are altered.

Analyzing the seven-dimensional data is not a trivial matter. A method has been developed that allows one to see interdependencies within the seven parameters, and this will be discussed in great detail in the following section.

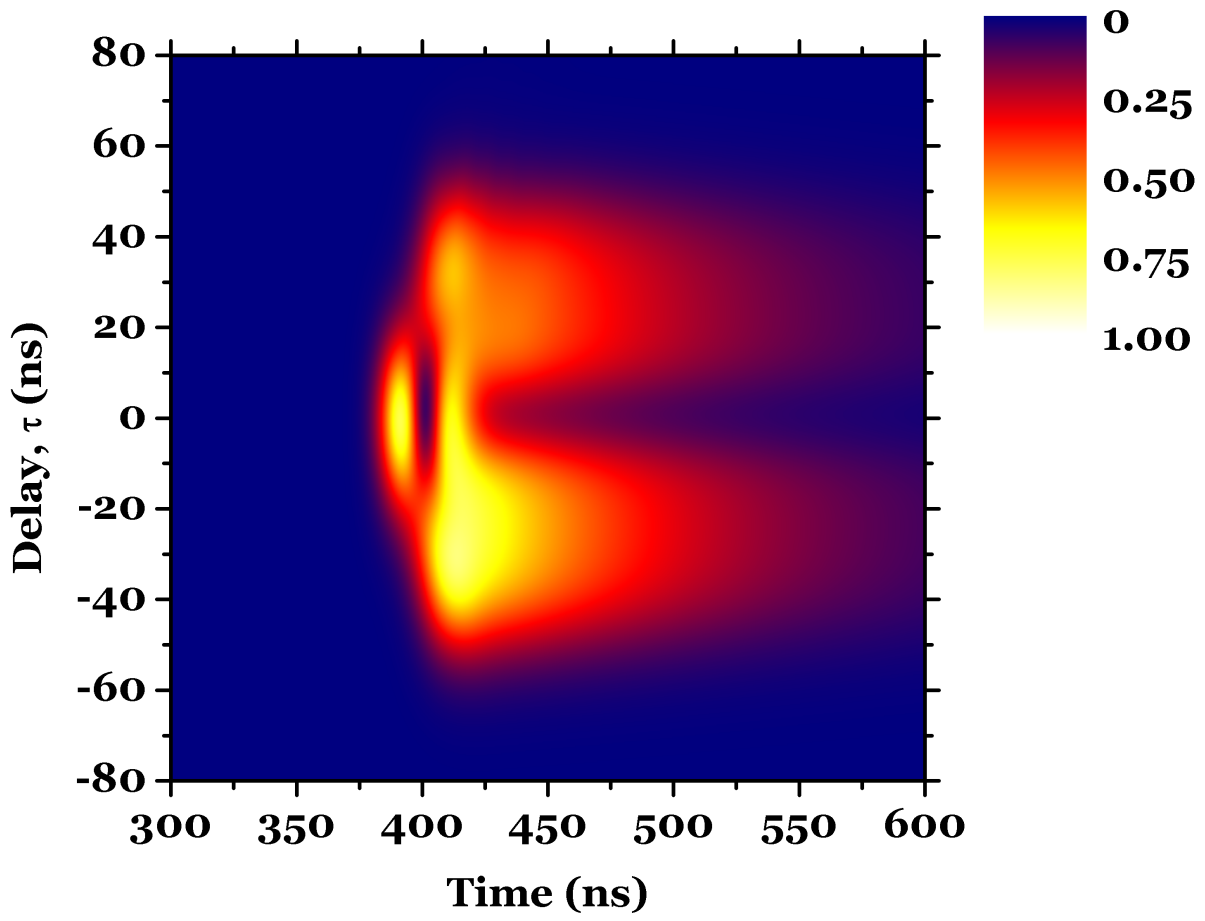

Figure 2.6: Rb($4d$) production is shown as a function of τ, plotted *versus* time. The other parameters are the same as in Fig. 2.5

2.5 Parameter Dependencies and Interrelations

A simple technique is used herein to analyze multi-dimensional data in a meaningful way. The goal is to understand how various STIRAP parameters interrelate, and what determines efficient population transfer. Thus, the diagnostic in determining whether a parameter is efficient or inefficient is the fractional $4d$ production. All seven parameters mentioned in Sec. 2.4.1 will be measured against this standard.

A short program was written (see Appendix D.2) to randomly select values for each of the seven parameters of interest. The parameter range was limited, as discussed in

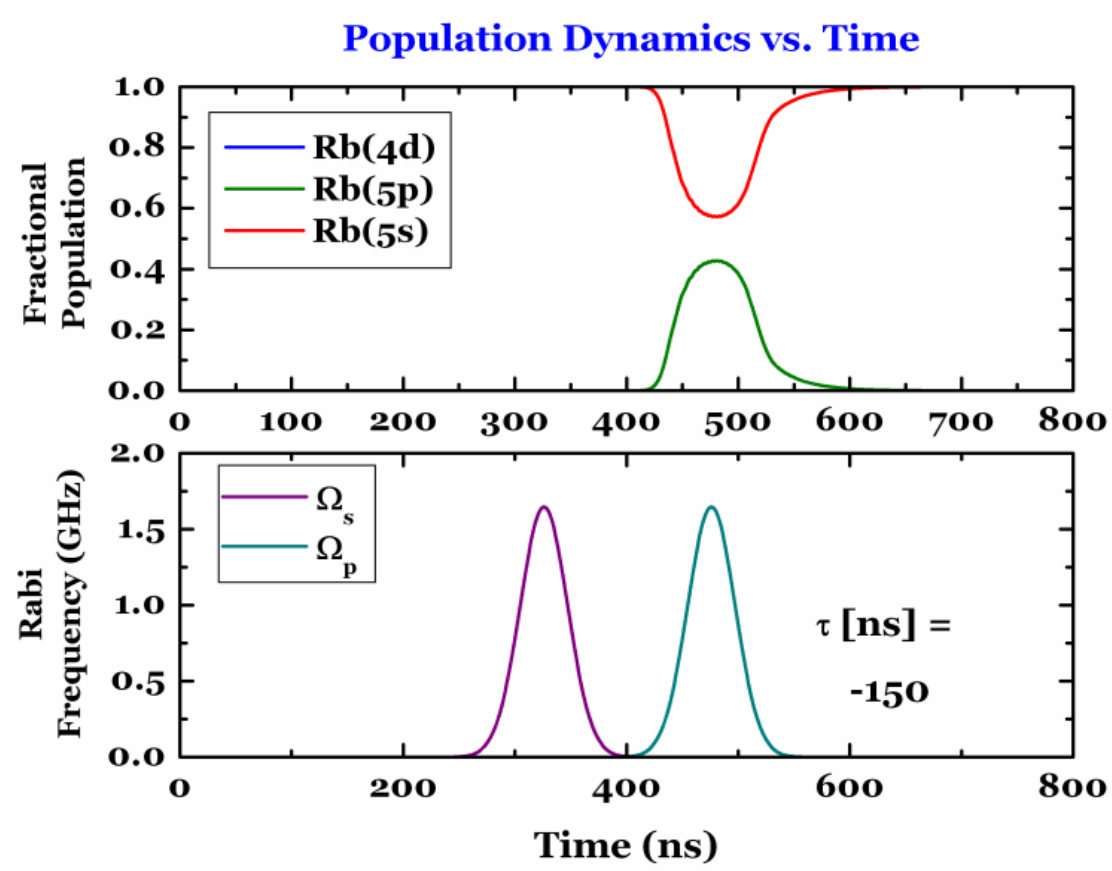

Figure 2.7: Pulse delay is explored by plotting fractional population *versus* time. The parameters used for this animation are listed elsewhere.[126] Click the image to view the film, or click here to launch an external media player to see the film at full scale.

Sec. 2.4.2. Based on those randomly selected parameters, Mathematica then calculated the interpolating functions governing the population as a function of time. The maximum $4d$ fraction was found for each set of parameters, along with its corresponding time, t_{max}. The populations in $5s$, $5p$, and $4d$ were then recorded at t_{max}, along with the seven parameters of interest, and the process was repeated.

This process was iterated 180,000 times. In order to develop an intuitive feeling for the interplay between the seven variables, a particular parameter of interest was plotted *versus* the maximum $4d$ population, ***keeping in mind that all other parameters were completely random***. This is an effective integration over the remaining six parameters. In

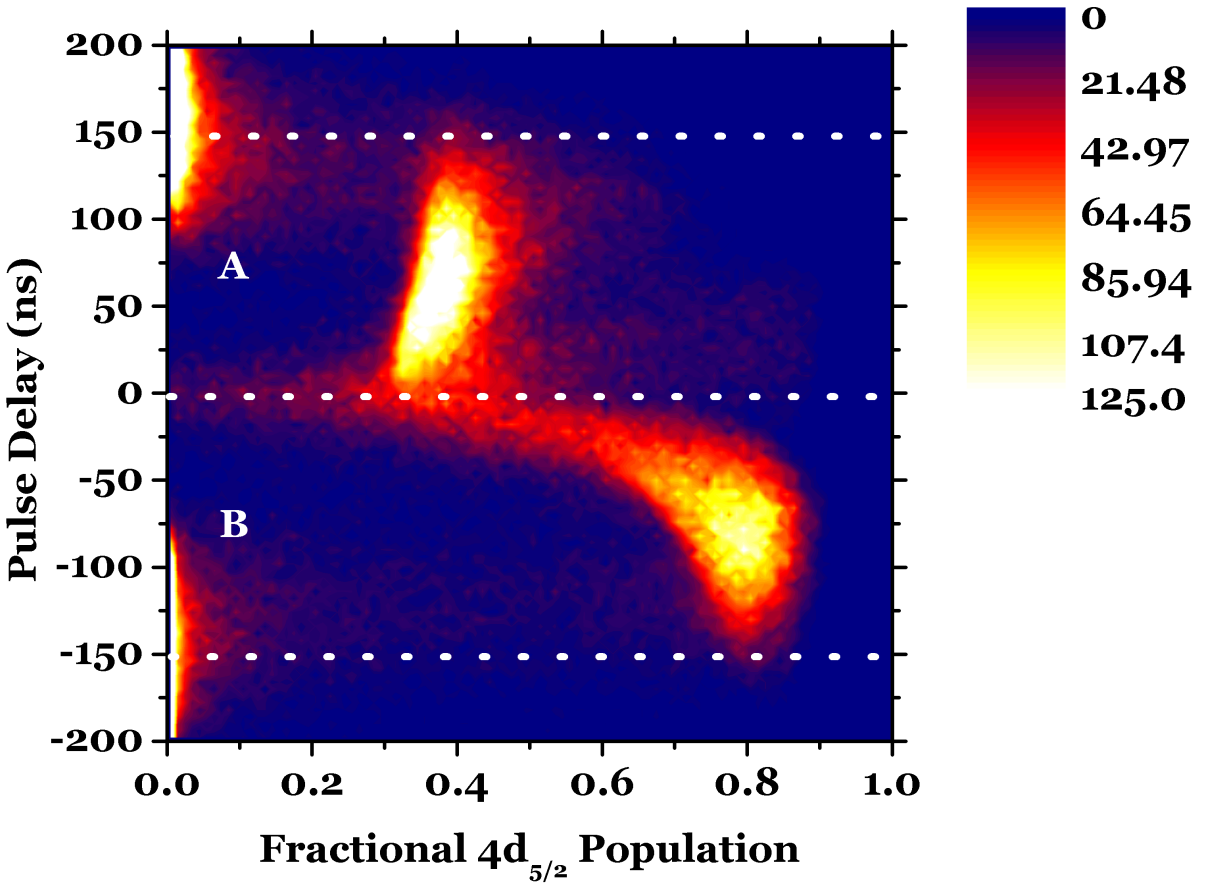

Figure 2.8: Fractional $4d_{5/2}$ production *versus* pulse delay. Two distinct regions exist, delineated by dotted lines, corresponding to intuitive pulse order (region 'A') and counter-intuitive pulse order (region 'B').

this manner, one can achieve a qualitative understanding of how a single parameter behaves with respect to maximum 4d production, yet one can also see how this single parameter couples to the projection of all other parameters in general. Such plots will be discussed in detail for each of the aforementioned seven parameter of interest.

2.5.1 Pulse Timing

Figure 2.8 shows a contour plot of τ *versus* fractional $4d$ population. The third dimension, color, indicates how many of the 180,000 runs resulted in a particular $4d$ fraction and a particular value of τ. That is, the color axis represents the number of data points that

correspond to a given τ and fractional $4d$ population combination, with all other parameters taking on random values. For example, the bright white area near $\tau = 75$ ns indicates that one is most likely to obtain a fractional $4d$ population of 0.40 throughout the random parameter space sampled by the remaining six parameters of interest. Conversely, in the dark blue areas it is not likely that any combination of the remaining variables will result in any population transfer into the $4d$ state. Thus, while one does not have specific information regarding the six remaining parameters, it is clear that in order to achieve population transfer above $\sim$ 0.30 into the $4d$ level, τ should range between -150 ns and $+150$ ns. General conclusions about the other parameters of interest can be reached from these results, such as the fact that, for example, larger pulse widths cannot compensate for τ values outside the aforementioned range.

Figure 2.8 also demonstrates the well-known idiosyncrasy of pulse order in STIRAP. Two regions, marked by dotted lines, show the intuitive and counter-intuitive delay regimes (areas labeled 'A' and 'B', respectively). It is more *likely* for a given set of random parameters to yield $4d$ population transfer in the intuitive regime, yet this intuitive configuration does not yield *efficient* population transfer, since the range of $4d$ fraction runs from 0.25 to 0.45 for this range of τ. On the other hand, for the counter-intuitive pulse ordering, population transfer is less-robust with respect to the remaining parameters, but the fraction of population transferred to the 4d state is *much* higher, up to 0.90. The advantage of this form of presentation is that it gives one the ability to see at a glance that pulse delay is a critical factor in determining efficient $4d$ population transfer, and that one can not compensate for a "bad" delay by adjusting any of the other parameters.

2.5.2 Intensity

Figure 2.9 is similar in construction to Fig. 2.8. Here, the parameters of interest, namely the intensities of the pump and Stokes laser light, I_p (Fig. 2.9a) and I_s (Fig. 2.9b), are plotted *versus* $4d$ population. Two distinct regions are apparent, one showing a robust but low

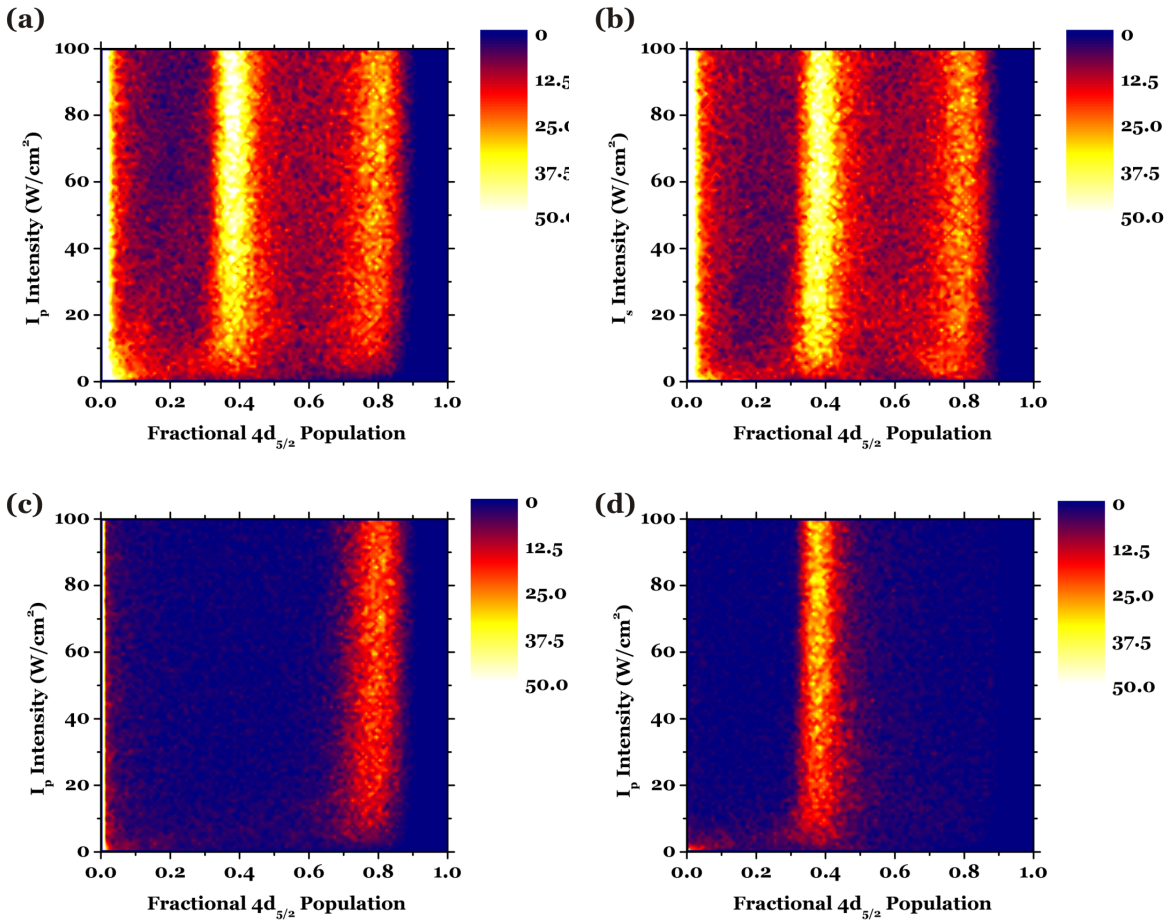

Figure 2.9: Fractional $4d_{5/2}$ production *versus* laser intensity. **(a)** Pump laser intensity I_p. (b) Stokes laser intensity I_s. **(c)** Only counterintuitive pulse order counts are plotted from (a). **(d)** Only the intuitive pulse order counts are plotted from (a).

transfer efficiency of population ($\sim$ 0.40), and the other showing a less-robust, but much more efficient transfer ($\sim$ 0.80). These bands are correlated with the state of the pulse delay. The band at 0.40 results from the intuitive pulse delay configuration, while the band at 0.80 corresponds to the counter-intuitive configuration. This conclusion can be verified by selectively plotting data for which τ takes on values in the intuitive or counter-intuitive regimes, as shown in Figs. 2.9c and 2.9d. In practice, this was done by taking "cuts" in the 7-dimensional parameter space for which τ lay either in region A or B of Fig. 2.8. The counterintuitive pulse order is plotted in Fig. 2.9c, where pulse delay was limited to a range

from -150 ns to -50 ns. Figure 2.9d shows results when the pulse delay ranged from 25 ns to 150 ns. Again, it is evident how critical τ can be as it gives rise to two distinct parameter regions in each of the other six variable plots.

Close inspection of Fig. 2.9 indicates that for $I_p > 20\ W/cm^2$, I_p does not seem to be a critical parameter for efficient production of $4d$. Intensities below $\sim 5\ mW/cm^2$ result in much less robust, but not much less efficient, $4d$ production. Figures 2.9a and 2.9b are almost identical, indicating that the robust nature of STIRAP is not strongly dependant on either the pump or Stokes intensity separately.

2.5.3 Single- and Double-Photon Detuning

Fractional $4d$ production as a function of detuning is shown in Fig. 2.10. At first glance, the results seen in Figs. 2.10a and 2.10b are not much different than that seen in, say, the intensity plots (see Fig. 2.9). Slight curvature can be detected in both cases, indicating some weak dependence on one- and two-photon resonance. Two bands are visible, each correlated to the intuitive and counterintuitive regimes, as before. In order to accentuate the relative importance of one- and two-photon detuning, however, data can be plotted for fixed selected delay, pump and Stokes pulse widths, and intensities,[130] while varying both Δ_1 and Δ_2. The results shown in Figs. 2.10c and 2.10d indicate that single-photon detuning is much less-important than two-photon detuning, a result consistent with previous calculations.[72,73] In a series of plots showing Δ_1 and Δ_2 *versus* fractional $4d$ population for different values of I_p and I_s, the curvature of the bands were seen to greatly increase as the intensities decreased. That is, for efficient population transfer, larger values of laser intensity can accomodate larger values of one- and two-photon detunings.

2.5.4 Pulse Width

Figure 2.11 shows the pump and Stokes laser pulse widths, w_p and w_s, *versus* fractional $4d$ population. As the widths decrease, population is transferred to the $4d$ level more efficiently in both the intuitive and counter-intuitive regimes. Also, both regime bands narrow as

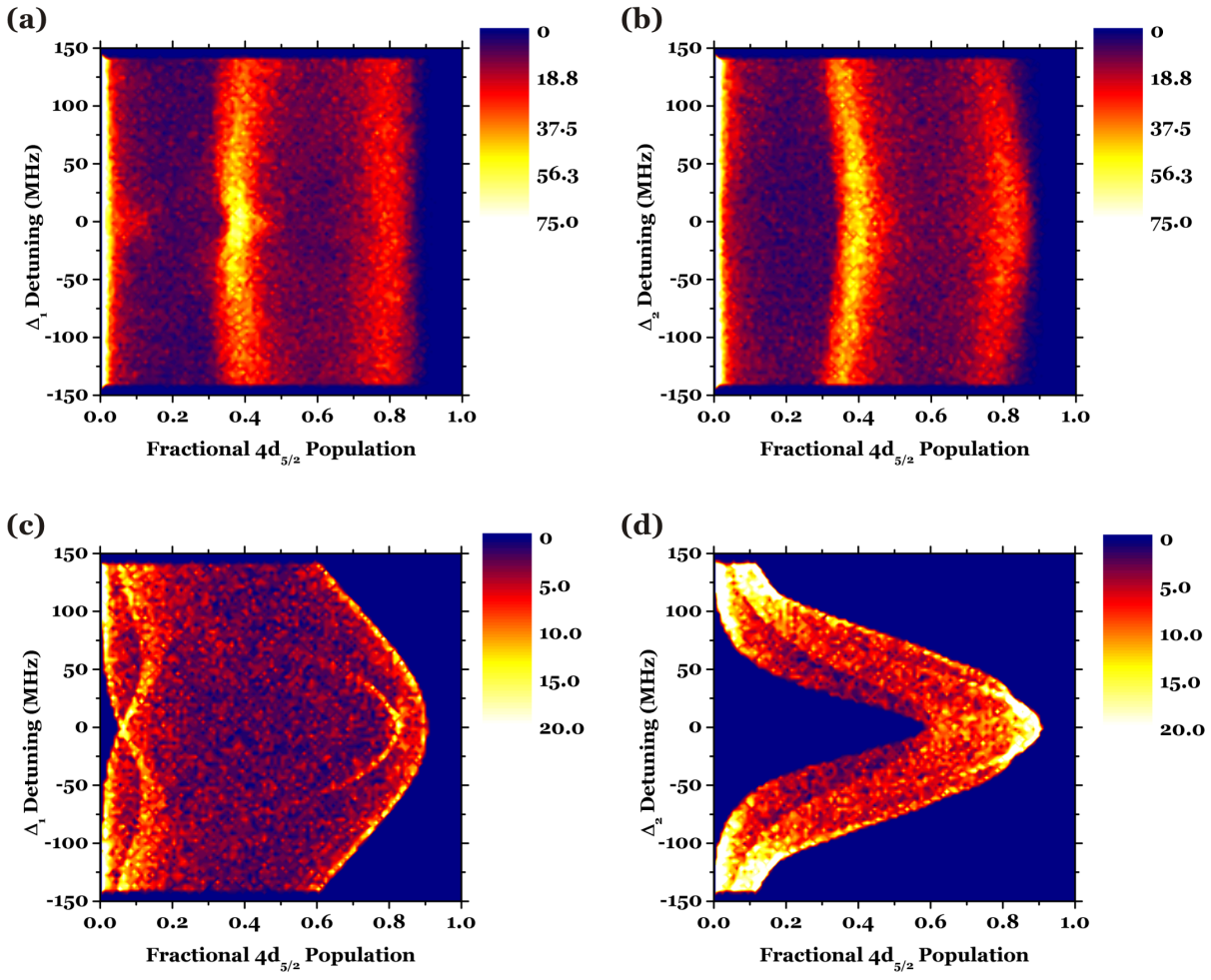

Figure 2.10: Fractional $4d_{5/2}$ production as a function of detuning. **(a)** Single-photon detuning, Δ_1. **(b)** Two-photon detuning, Δ_2. **(c)** Δ_1 is varied, while pump and Stokes widths, delay, and intensities are fixed.[130] **(d)** Δ_2 is varied, while the other parameters have the same values as in (c).

pulse widths decrease, indicating that the acceptable range of the other six parameters is decreasing. The counter-intuitive pulse regime appears to be affected by both pulse widths more than the intuitive pulse regime, as indicated in the sharpness of curvature evident in the bands at ~ 0.80.

Because no adiabatic approximation has been made, one could explore adiabatic or diabatic transfer regimes by selecting combinations of laser pulse widths and intensities. In general, the more narrow w_p and w_s become, the more diabatic the process becomes. Simi-

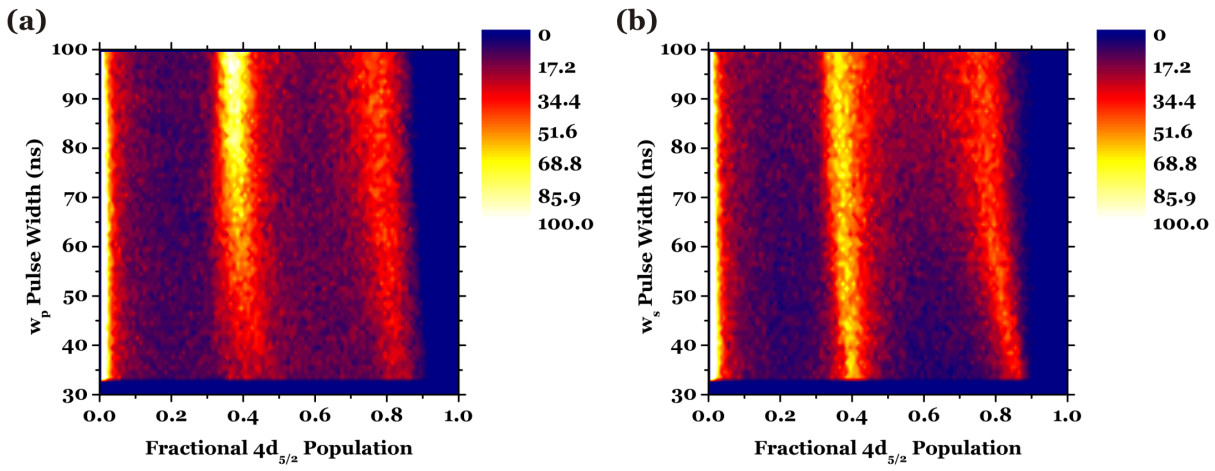

Figure 2.11: Fractional $4d_{5/2}$ production as a function of **(a)** pump, and **(b)** Stokes pulse widths.

larly, adiabaticity increases as intensities I_p and I_s increase, for fixed pulse widths. Exploration of adiabatic *versus* diabatic behavior can thus be achieved by selecting combinations of laser intensity and pulse width to study particular regions of interest.

2.5.5 Adiabatic and Diabatic Regimes

In Appendix A.2.5, the general requirements for adiabaticity are discussed. With the 3-level model described above, no adiabatic assumptions were made, as is generally the case in other treatments.[131–133] Here, use of the limiting adiabatic condition (Eqn. A.65) cannot be made, since the density matrix treatment for the three-level case did not make use of the adiabatic approximation. Instead, one can see the off-diagonal terms of the density matrix directly. The adiabatic nature of the three-level system can be seen by comparing the absolute value of the off-diagonal terms of the density matrix to the diagonal terms. **If the off-diagonal terms are much smaller than the diagonal terms, the system is considered to be in an adiabatic state.** Using the Mathematica code presented in this dissertation, one can explore the adiabatic limits of STIRAP by looking at such off-diagonal terms directly, which represent the coherent coupling of states. Using the terms shown in Eqn. 2.41, one can plot, for example, λ_-, *versus* terms such as $\dot{\theta}\cos\phi$, $\dot{\theta}\sin\phi$, and $\dot{\phi}$.

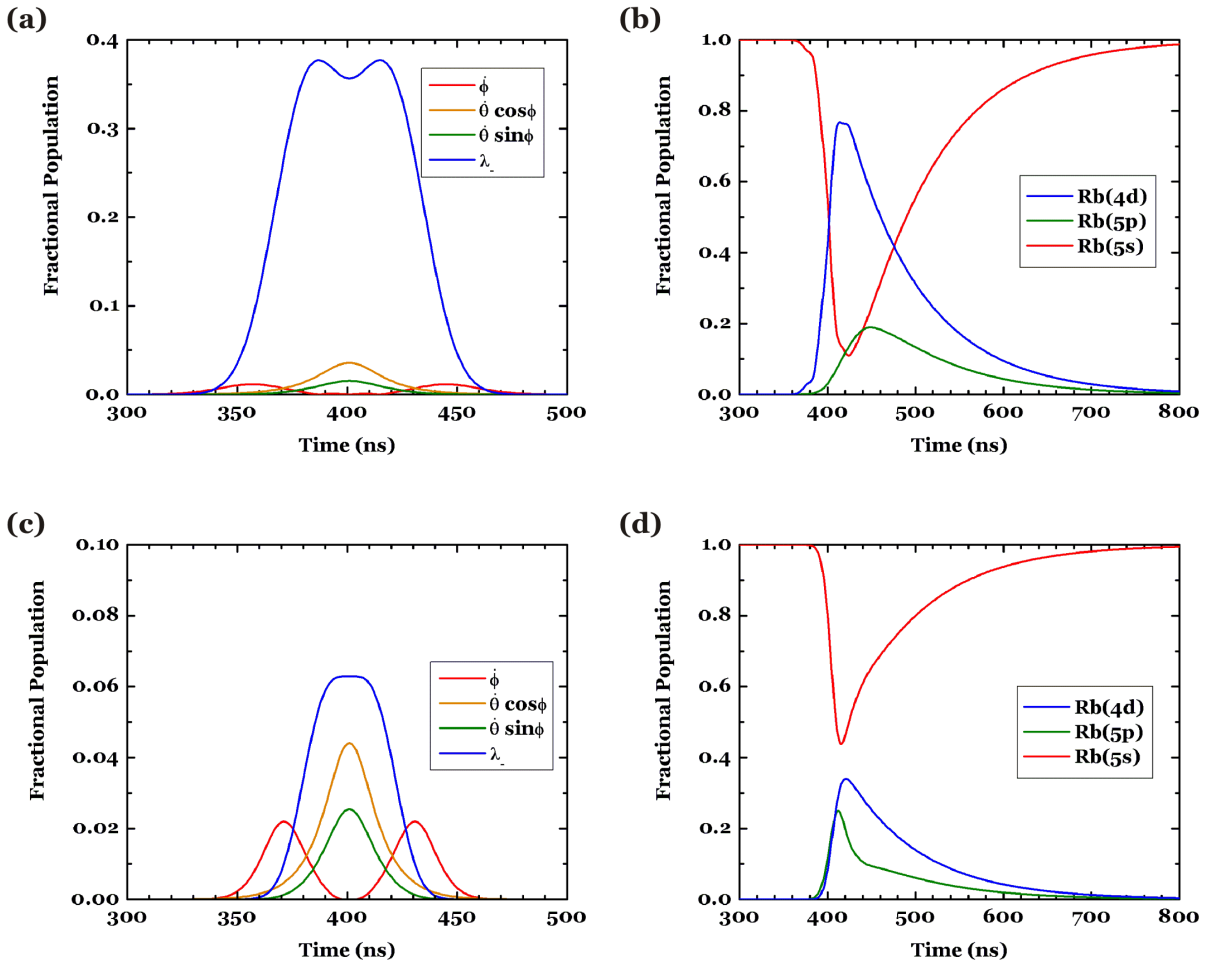

Figure 2.12: Diagonal matrix element, λ_-, and off-diagonal matrix elements, $\dot{\theta}\cos\phi$, $\dot{\theta}\sin\phi$, and $\dot{\phi}$, plotted *versus* time. **(a)** Adiabatic regime, indicative by $\lambda_- \gg A_{ij}$, where A_{ij} represents any of the off-diagonal elements. **(b)** Time evolution plot of populations in the $5s$, $5p$, and $4d$ states for the adiabatic regime. **(c)** Same as in (a), only showing the system to be diabatic. **(d)** Same as (c), but for the diabatic regime. The parameters used for both the adiabatic and diabatic cases are listed elsewhere.[134]

Figure 2.12a and b shows two plots, one for which the system is clearly in an adiabatic regime (Fig. 2.12a), and one where the system is in a diabatic regime (Fig. 2.12c), as is evident by comparing the amplitude of the diagonal term, λ_-, to the off-diagonal terms. The remaining two plots (Fig. 2.12b and 2.12d) show the time evolution of population in the adiabatic and diabatic regimes, respectively. An important feature to note is the difference in $5p$ population rise-time between the two regimes. When the system is diabatic, the $5p$

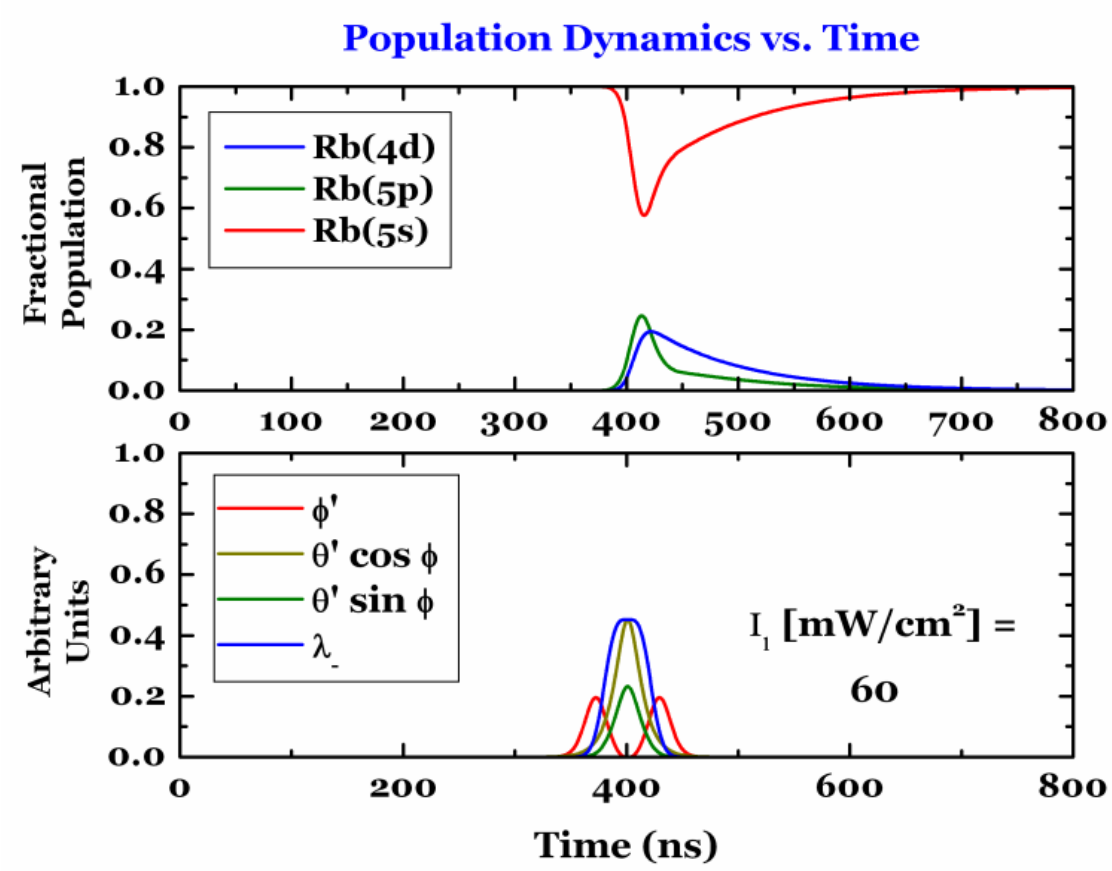

Figure 2.13: Adiabaticity and diabaticity can be seen by plotting fractional population *versus* time. The parameters used for this animation are listed elsewhere.[135] Click the image to view the film, or click here to launch an external media player to see the film at full scale.

state is fed directly from the $5s$ ground state. When the system is adiabatic, however, the $5p$ state is only fed by decay from the $4d$ state, and thus is not filled until later in time.

A more dramatic example of this behavior can be seen in Fig. 2.13. Here, a selected set of parameters yield very different results depending on the Rabi frequency of the light used in the system. For this demonstration, the pump intensity was swept from 60 mW/cm^2 to 1000 mW/cm^2. The Rabi frequencies for I_p and I_s were matched at all times. The lower graph shows λ_- and a variety of off-diagonal matrix elements. At low intensities, all of the values are comparable. Examining the resultant fractional population curves for low intensities, the system is clearly in a diabatic state, where population is sequentially placed

into the $5p$ and then $4d$ energy levels.

As the film of Fig. 2.13 plays on, and the intensity is increased, the diabaticity decreases. Note that λ_- must become much larger than the off-diagonal elements in order to avoid the diabatic regime. It is tempting to make the assumption that parameters yielding large fractional $4d$ populations is indicative of adiabatic population transfer. However, as can be seen here, this can be misleading. In the film, purely adiabatic transfer seems to be present somewhere around a Stokes intensity of 700 mW/cm^2, yet fractional $4d$ populations of 80% are reached as early as 300 mW/cm^2 when the system is in some state between clearly adiabatic and clearly diabatic.

Such murky areas can be explored theoretically in detail using a modeling system like the one presented here. However, it is far more interesting to look for behavior, such as adiabatic *versus* diabatic population transfer, or efficiency rates in the counterintuitive or intuitive regimes, by experimentally measuring the population dynamics as a function of time.

2.5.6 System Loss Mechanisms

Finally, a brief note on the robust nature of the simulation. Other loss mechanisms, such as Penning or associative ionization can be present in the system of interest, and can be a factor in determining coherent population transfer on nanosecond timescales.[7,10,128,129] Such loss terms can easily be included in the simulation code presented above, and in fact, the machinery for doing so is already in place. However, the rates for such losses are not known, and therefore, the system is treated as a closed system with a conservation of population.

In reality, preliminary measurements have been made for the system of interest discussed in this dissertation, and there is conflicting evidence indicating whether such loss mechanisms can be neglected. For example, with the Stokes laser continually present in the system, and the pump laser pulsed for 50 ns during a typical 5 μs period, a count rate of $\sim$ 10,000 counts per second is seen when the pulse timing ratios are taken into account. Detector efficiency

is near-unity, and the target typically consists of 10^8 atoms. Thus, even such a large count rate wouldn't necessarily seem to indicate significant loss from the system. However, it has been noted experimentally that the MOT target itself is visibly altered during such tests, indicating that the damage to the target may be more extensive. Future work will be done in this area, and eventually the loss rates will be measured and included in the theoretical simulations. (See Sec. 5.2.3.)

Chapter 3

Experimental Setup

With the theory well-established, the focus of this dissertation now turns to the experimental exploration of coherent population dynamics. This chapter will discuss the technical aspects of preparing a system of ^{87}Rb atoms for coherent excitation, and measuring the population transfer as the system undergoes excitation and relaxation. The actual experimental results will be shown in the next chapter.

The experimental setup can essentially be broken into 4 main topics:

- **Target Control**

  This includes the preparation and maintenance of the collision target. A general description regarding methods for cooling and trapping will be covered.

- **Diagnostic Control**

  The MOTRIMS diagnostic technique is discussed. This topic is blended with the previous due to the nature of MOTRIMS. Hence, target and diagnostic control are presented together as a general overview of the MOTRIMS technique.

- **Laser Control**

  The most difficult of all aspects with the experiment, laser control consists of a discussion on techniques and methods for frequency locking, saturated absorption spectroscopy, laser pulse timing, *etc.* This section has been broken into separate areas of

interest (namely, laser control, frequency detuning, saturated absorption spectroscopy, and laser dither locking) in order to make it more understandable.

- **Data Acquisition**

  Finally, a discussion of the actual data acquisition techniques is covered. Some of the more subtle aspects of dealing with MOTRIMS data are discussed, as these techniques have general application to data acquisition in other areas.

A fair amount of optics are employed for proper control and timing of the lasers used in this experiment. Very little of this experimental setup will be covered, except in the diagrams outlining the complete experimental apparatus. Any relevant details that may not be obvious, such as acousto-optical modulator (AOM) control, are discussed, while details, such as why a particular beamsplitter is inserted in a given beam path, are not discussed. A detailed explanation of the related MOTRIMS optical setup, however, can be found in the literature.[136,137]

The entire experimental process is a union of basic coherent excitation methods and a well-established spectroscopic technique, known as MOTRIMS. Technical aspects, including laser control and data acquisition, will also be discussed in detail, but the system is so complex that for many of the details one must consult the cited literature.

3.1 General Overview of MOTRIMS

Magneto optical trap recoil ion momentum spectroscopy (MOTRIMS)[137–140] is a technique used in ion-atom collision experiments to measure charge-transfer physics in a cold target. The advantages of using a magneto optical trap (MOT) to produce a cold, localized target are three-fold. First, because of the extremely low temperature of the target, the recoil ion momentum resolution is 0.03 a.u.[141] The decrease in initial momentum spread provides, *for the equivalent target mass*, about an order of magnitude improvement in momentum resolution compared to typical recoil ion momentum spectroscopy (RIMS)[142,143] techniques.

Such high-resolution spectroscopy allows one to resolve the initial and final states involved in the collision that cannot be determined in traditional cold target recoil ion momentum spectroscopy (COLTRIMS)[144–147] using, for example, a supersonic gas-jet expansion technique.

The second advantage in using a MOT for a RIMS target is the ability to study excited target states. Cooling and trapping atoms naturally prepares the target in some mixture of the ground and first excited state, allowing one to measure charge-transfer with excited atoms. Additionally, with the introduction of an external excitation laser field, the population can be prepared in almost any configuration of initial excited states, or can be aligned with respect to the laboratory frame. The inherent excitation of the cooling and trapping methods provide such excited states without the need for an elaborate or overly-complicated experimental apparatus.

Finally, MOTRIMS offers one the ability to study a wider range of possible target species. Most alkali and earth alkali atoms can be trapped in a MOT, while such species cannot be efficiently cooled in a supersonic gas-jet. This opens the door for the study of atoms with a single optically active electron, which is of interest due to the similarities and features common with atomic hydrogen.

MOTRIMS is the "engine" behind the experiment that allows one to measure the population dynamics of the system with such high resolution. Therefore, a brief introduction to MOTRIMS will be presented here in the context of measuring population dynamics, while more extensive details can be found elsewhere.[136]

3.1.1 The Target — MOT

Magneto optical trapping methods are covered thoroughly throughout the literature,[148–150] thus only a cursory explanation of a MOT related to the system used in this dissertation will be given. A MOT generally consists of a laser radiation field and an inhomogeneous magnetic field. The radiation field provides a method of radiatively cooling the atoms by

presenting a velocity-dependent force, while the magnetic field induces a spatially dependent Zeeman shift in the atomic transition frequencies. Together, these two effects cool and trap atoms in a localized region (usually a few millimeters in diameter), at temperatures below 1 mK.

Due to the finite transition linewidths, the trapped target selected for this dissertation is ^{87}Rb, which has a nuclear spin of $I = 3/2$. The hyperfine structure of the $5s_{1/2}$ ground state, $5p_{3/2}$ excited state, and $4d_{5/2}$ terminal state are shown in Fig. 3.1. Two laser fields are present in the MOT to provide the velocity-dependent cooling forces. The "trapping" laser field couples the $F = 2$ hyperfine level of the $5s$ ground state to the $F' = 3$ of the $5p$ middle excited state. This trapping laser has a wavelength of 780.03 nm. Population is excited from the ground state into the intermediate $5p$ state, which has a lifetime of 26.63 ns.

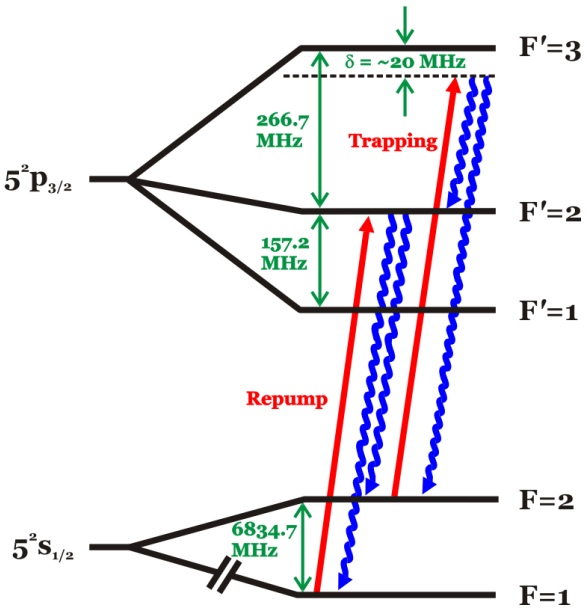

Figure 3.1: The energy level diagram for ^{87}Rb, showing the $5s_{1/2}$, and $5p_{3/2}$ energy states. Trapping and repump laser transitions (straight arrows) and spontaneous decay channels (rippled arrows) are also shown.

The trapping laser frequency is actually detuned slightly from perfect resonance with the $5s - 5p$ transition. This detuning, labeled "δ" in Fig. 3.1, is typically ~ 20 MHz to the red of the $5s - 5p$ transition. Atoms in a particular velocity group such that they are Doppler-shifted into resonance with this transition are therefore trapped. This small detuning from resonance is required for trapping — if the lasers were tuned precisely to the $5s - 5p$ transition, then the only atoms to be excited would be those that are completely stationary, at the zero-point of the magnetic field gradient.

The trapping laser field has a 0.5% chance of exciting ground state population into the

$F' = 2$ state. When this occurs, the resultant excited population in $F' = 2$ has a 50% probability of decaying either back to the $F = 2$ ground state, where it will be back under the influence of the trapping field, or decaying down to the $F = 1$ ground state, at which point it will no longer interact with the trapping laser field. If this optical pumping effect is not corrected, the ground state population will quickly be pumped into the $F = 1$ hyperfine level, and the atoms will not be trapped.

Therefore, a "repump" laser field is introduced, in order to couple the $5s_{1/2}, F = 1$ ground state and the $5p_{3/2}, F' = 2$ middle state. Because the rate at which the atoms are optically pumped into the $F = 1$ ground state is relatively slow compared to the rate at which this population is excited back into the $F' = 2$ state, this effectively keeps the population in the trapping and cooling cycle. The MOT will thus have a mixture of population in the $5s_{1/2}, F = 2$ ground state and the $5p_{3/2}, F' = 3$ excited state at any given time.

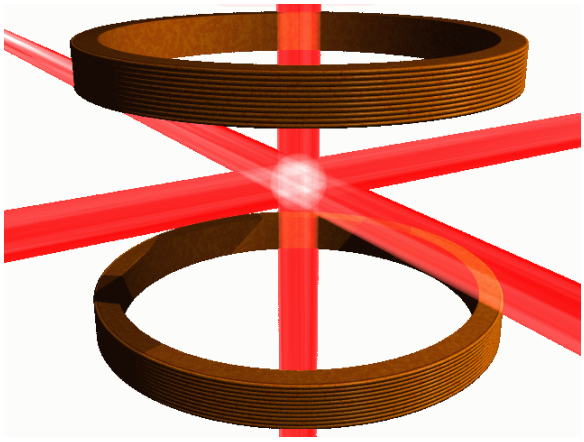

Figure 3.2: Conceptual picture of the magnetic and optical fields used for cooling and trapping.

Without a position-dependent force, the cooled atoms can still slowly wander away; they are not confined spatially. An inhomogeneous magnetic field is therefore present to supply the necessary position-dependent trapping force. This magnetic field is provided by a pair of copper wires wound into coils, arranged in the anti-Helmholtz configuration. The magnetic field in such a configuration provides a gradient twice as steep in the plane perpendicular to the coils than parallel to them. The magnitude of the field is zero in the center. Because of this spatially-dependent magnetic field gradient, the Zeeman shift induced in the atomic transitions is also spatially dependent. As an atom wanders farther from the zero-point of the magnetic field gradient, the Zeeman shift will cause the atom to

Parameter	Value	Units
B-Field Gradient	~ 5	Gauss/cm
MOT Temperature	130 ± 100	μK
MOT Density	$\sim 10^8$	atoms/cm^3
Vacuum Pressure	10^{-10}	Torr

Table 3.1: Typical MOT Characteristics

move closer to resonance with the photons moving toward the center of the trap. Together, the magnetic and optical fields cool and trap the atoms to produce a dense, cold target, shown conceptually in Fig. 3.2. For the experiments presented here, the magnitude of the magnetic field gradient is ~ 5 Gauss/cm. Table 3.1 shows this, and other typical values for the MOT target used herein.

3.1.2 The Technique — RIMS

The MOT described in the last section is used as a target in a typical recoil ion momentum spectroscopy (RIMS) experiment. In brief, RIMS consists of measuring the time-of-flight (TOF) of a recoil ion involved in a charge-transfer collision. Here, a singly-charged projectile ion induces charge-transfer from the target neutral atom, rendering the projectile neutral while the target atom becomes ionized. To within a constant offset, the TOF of the recoil ion is equal to the difference in flight times between the projectile and recoil.

This recoil ion is extracted using a momentum spectrometer, and both particles are independently detected downstream of the collision region. Because the TOF is dependent on the initial and final states of the two atoms involved in the collision, one can determine the initial and final capture channels of the projectile and recoil ions by measuring this TOF. Such a RIMS technique is shown in Fig. 3.3, which can be clicked to view a short animation depicting the interaction process. When the projectile particle is detected, a timer is started (shown as a green stopwatch in the film), and the recoil particle signals the timer to stop. This time-of-flight difference is recorded for every projectile-recoil pair event that takes place.

Figure 3.3: A schematic representation of the RIMS technique. Click the image to view a short animated film, or click here to launch an external media player to see the film at full scale.

The momentum spectrometer produces an electric field designed to extract the ions and then detects them using position-sensitive channel-plate detectors. The spectrometer consists of a solid plate and 34 rings, shown in Fig. 3.4. The plate and rings are connected together with 1 MΩ resistors. The solid plate, referred to as the "pusher" plate, has a voltage applied to it (typically $\sim 70 - 100$ V), while an independent "focus" voltage is applied to the 16^{th} ring from the pusher plate. The resulting field gradients focus the recoil ions in mometum space, ensuring that ions with the same recoil momentum will strike the downstream detector with the same flight time. The spectrometer also focuses the recoil ions spatially, reducing any effect the finite target thickness has on the momentum resolution.[151]

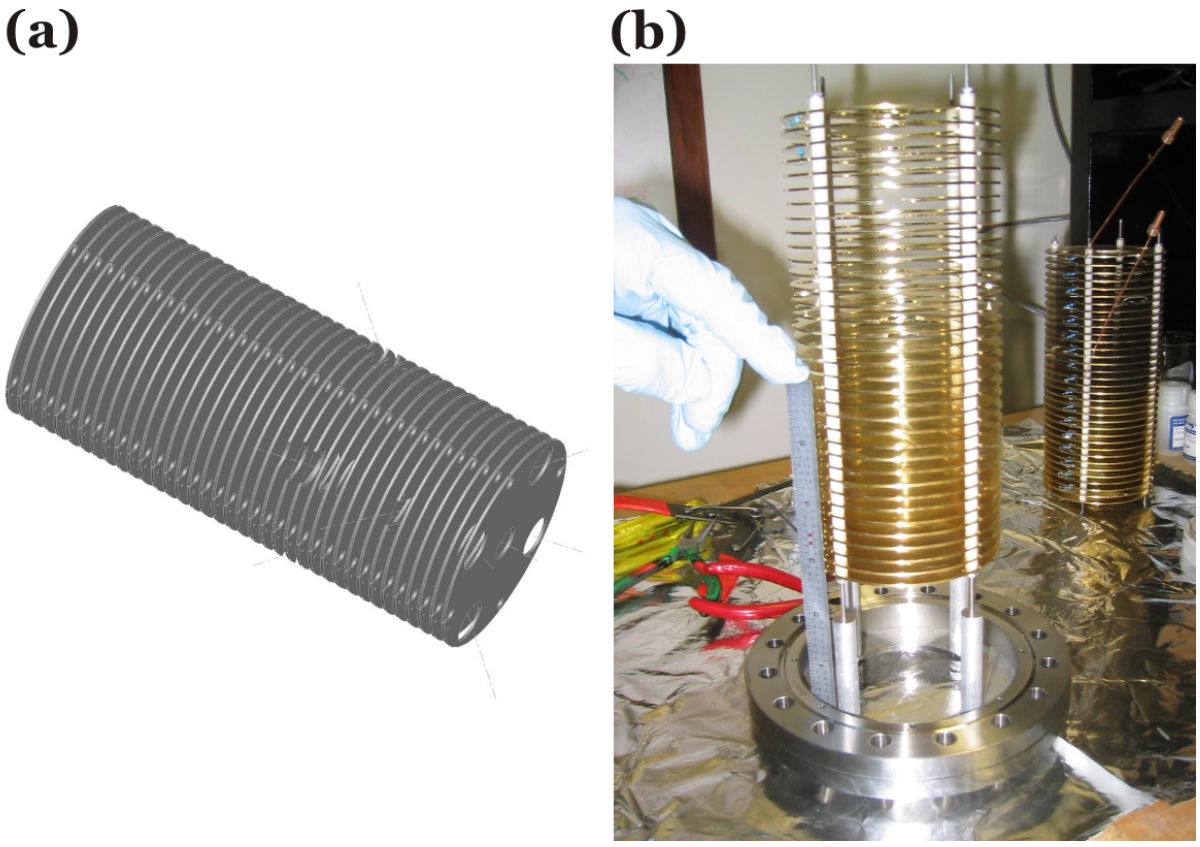

Figure 3.4: A typical MOTRIMS momentum spectrometer. **(a)** A computer-generated model, showing the rings and the holes that allow laser light to pass into the spectrometer. **(b)** A photograph of the actual spectrometer. An older version is seen in the background.

The intent in such a measurement is to utilize the TOF information in order to deduce the energy difference between the initial projectile ion and the final recoil ion. This energy defect, known as the Q-Value, is defined as

$$Q = (E_i - E_f), \tag{3.1}$$

where E_i and E_f are defined to be the initial and final binding energies for the projectile and recoil ions, respectively. It can further be shown[137,147] that

$$Q = -\frac{1}{2}m_e v_p^2 - v_p P_R, \tag{3.2}$$

where m_e is the mass of an electron, v_p is the initial velocity of the projectile ion, and P_R is

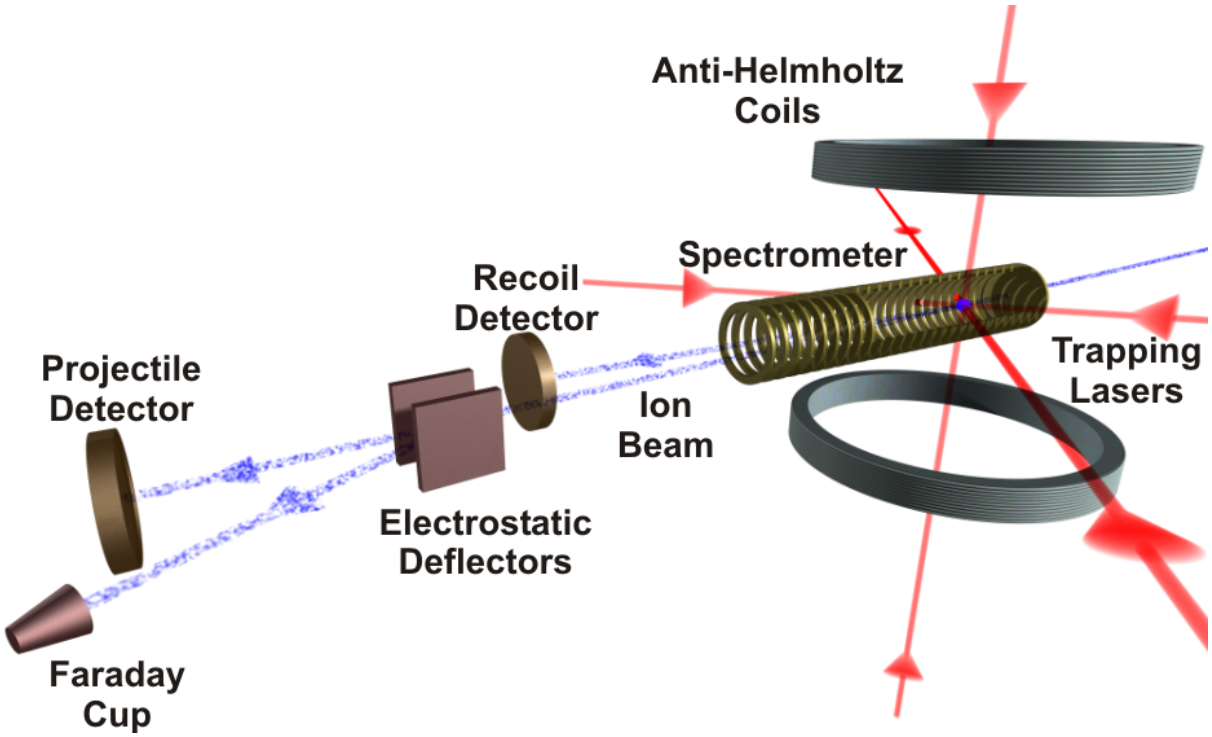

Figure 3.5: The MOTRIMS experimental apparatus.

the momentum component of the recoil ion parallel to the extraction direction. From this relationship between the projectile velocity and the recoil momentum, and because the time-of-flight between the projectile and recoil ions is a direct measurement of the momentum difference between the two ions, the TOF measurements in RIMS directly give rise to a Q-Value spectrum.

3.1.3 The Combination — MOTRIMS

Combining the RIMS method of measuring Q-Values with MOT techniques to create an extremely cold target is called MOTRIMS. The full MOTRIMS experimental apparatus is depicted in Fig. 3.5. An ion gun supplies a 7 keV beam of Na^+ ions, shown in the figure as a blue line. The projectile ions are tuned to pass through the target, and are detected[152] downstream after passing through a pair of electrostatic deflectors. Any projectile ions that do not undergo charge-transfer are deflected into a Faraday cup. The recoil ions are

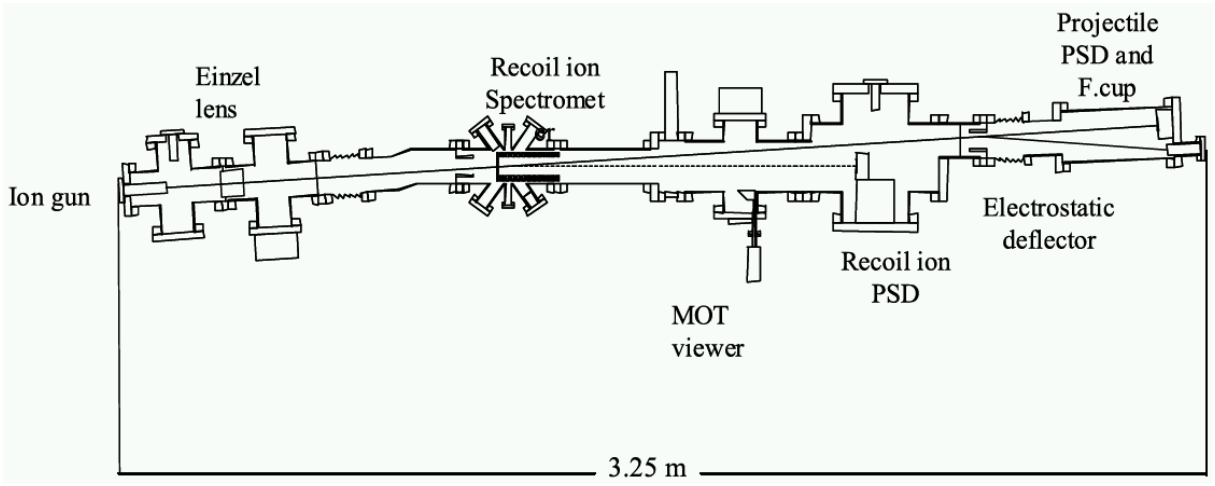

Figure 3.6: The physical MOTRIMS beamline.

extracted by the momentum spectrometer, pass through a field-free drift region, and are detected on a second detector downstream roughly one meter from the collision area, as shown in Fig. 3.6.

The six counter-propagating trapping lasers are directed into the experimental chamber, which is intentionally left out of Fig. 3.5 for clarity, but is modelled graphically in Fig. 3.7. The chamber consists of a section of 4-inch tubing having a set of ten ports designed to cross at the center of the beamline. Three pairs of ports are orthogonal, and are used for the trapping laser light. The remaining four ports are used to pass the excitation lasers into the chamber. The magnetic field gradient discussed in Sec. 3.1.1 is provided by a pair of anti-Helmholtz coils, each with 655 turns of wire and a measured resistance of 9.1 ohms. Dimensions for the coils are given in the cross-sectional diagram shown in Fig. 3.8.

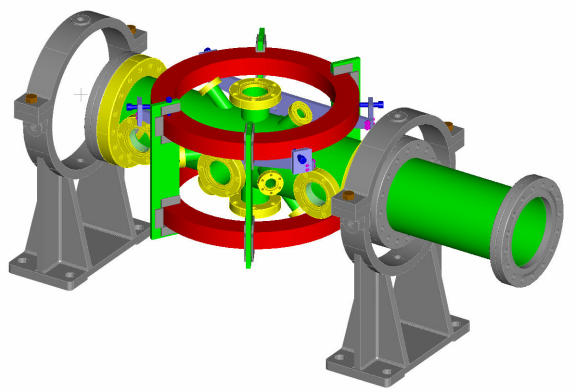

Figure 3.7: A computer-generated model of the MOTRIMS chamber. B-field coils are shown in red.

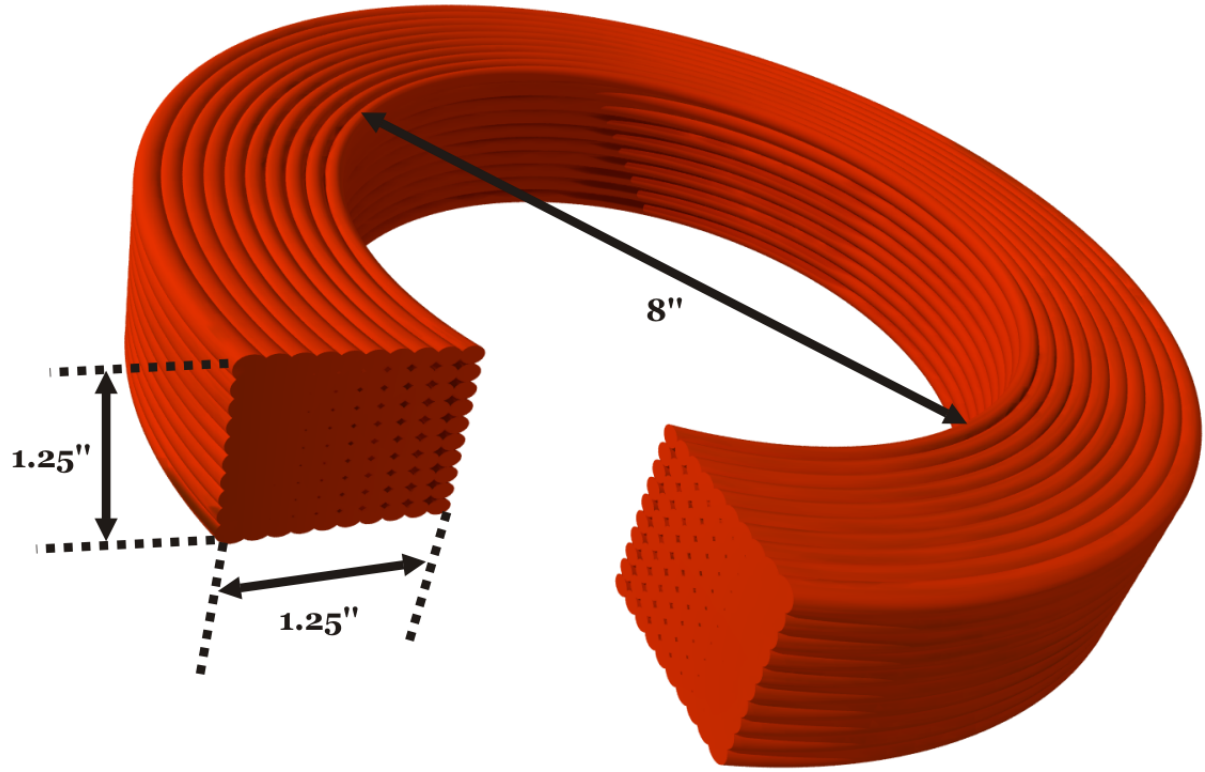

Figure 3.8: Magnetic field coil dimensions.

The time-of-flight between the detection of a neutral projectile and a recoil ion is measured *via* data acquisition electronics which are explained in Sec. 3.3. A typical TOF spectrum using this spectrometer and measured for a collision involving Na^+ on ^{87}Rb is shown in Fig. 3.9, where counts are plotted *versus* Q-Value. Each channel is labeled, showing the initial and final capture states involved in the collision.

3.2 Expansion to Coherent Excitation

MOTRIMS is only the diagnostic tool used here in measuring the coherent excitation processes within the target. Measuring population dynamics relies on accurately knowing the relative number of atoms in the ground and excited states within the target (in

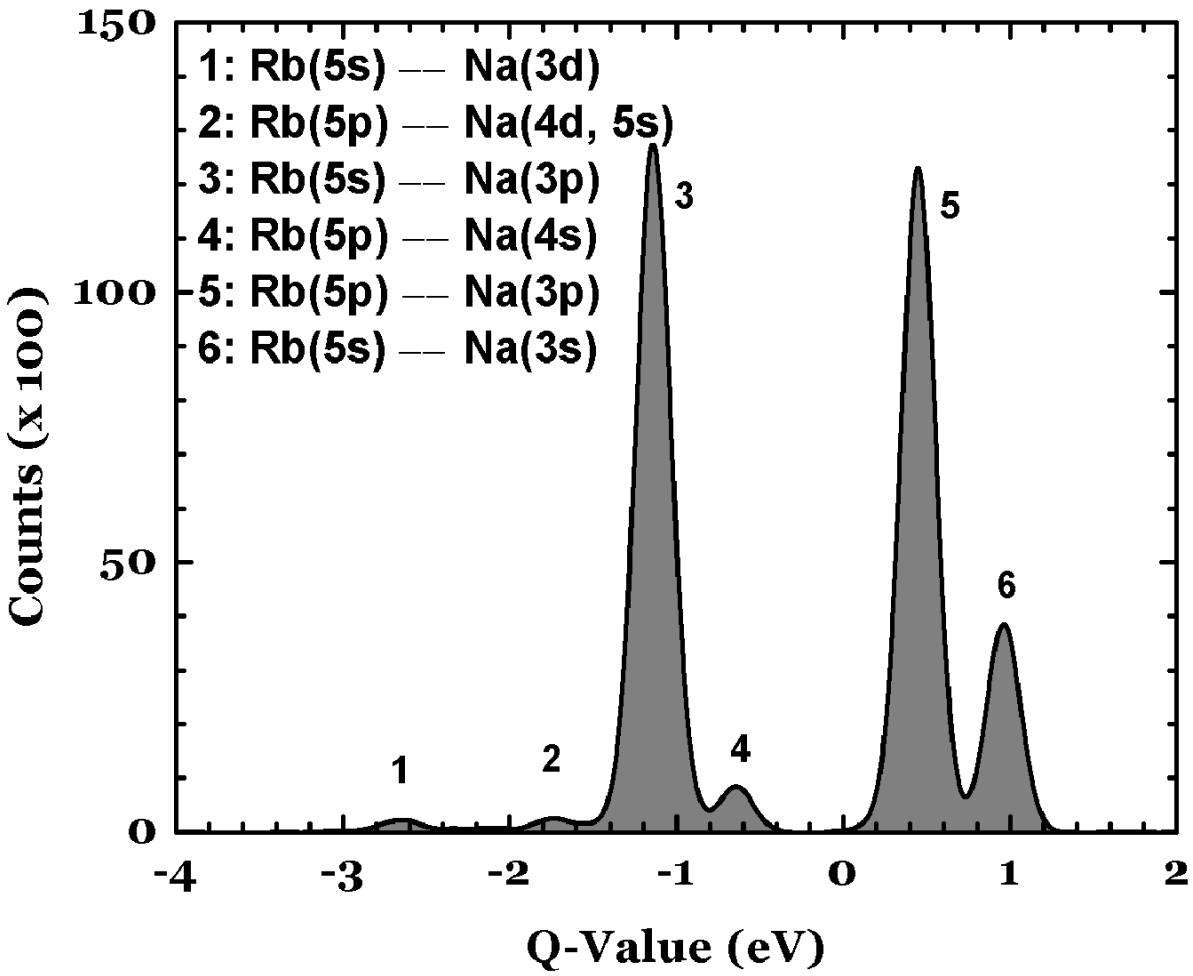

Figure 3.9: A typical Q-Value spectrum measured using the MOTRIMS technique.

this case, the $5s$, $5p$, and $4d$ levels). A method of measuring the excited state fraction, f, within a MOT has been described in detail elsewhere,[153] and this technique will be briefly discussed.

3.2.1 Measuring Excited State Fractions

The Q-Value spectrum shown in Fig. 3.9 is an example of one used in determining the excited state fractional populations within the MOT. The area, A, under a given capture channel curve is proportional to the number of atoms, n, within the initial state of the charge transfer channel. That is,

$$A_i \propto n_i \sigma_i, \tag{3.3}$$

where σ_i is the cross section for that charge-transfer channel, and $i = s, p, d$ represents the particular channel, $5s$, $5p$, or $4d$, respectively. Both theoretical and experimental measurements of most relevant cross sections for the Na^+ on Rb system are available in the literature.[137,154] The proportionality constant contains geometric factors, target thickness, and acquisition time. Using this relationship, the excited state fraction can be given by

$$f_x = \frac{A_x/(\sigma_x/\sigma_s)}{A_x/(\sigma_x/\sigma_s) + A_y/(\sigma_y/\sigma_s) + A_z/(\sigma_z/\sigma_s)}, \tag{3.4}$$

where $x, y, z = s, p, d$ (in any order, but all are used). Using this relation, and assuming the relative cross-sections are known, a single Q-Value spectrum with sufficient resolution will yield the excited state fractions in the target MOT. Because the projectile ion beam continuously (but randomly in time) samples the MOT and charge-transfer occurs, one can measure the excited state fraction of the target as a function of time. For example, in the case of a two-level system, if the laser field is turned off for a short time, the population will decay to the ground state, and channels involving capture to or from $5p$ will disappear.

Figure 3.10 shows such a measurement. Charge-transfer coincidence events are plotted as a function of laser period *versus* Q-Value. The laser is turned off for 500 ns out of a 5 μs total period. During this "trap-off" time, the atoms relax to the $5s$ ground state, and the $5p$ capture channels disappear.

Such a measurement is valid so long as the time the trapping lasers are turned off is much less than the time it takes for the previously-trapped atoms to be affected by gravitational forces and fall out of the trapping region. Typically, the atoms will fall a distance comperable to the ion beam's diameter on a millisecond timescale, so turning off the trapping lasers for tens of microseconds has a negligible effect on the measured population dynamics. Turning off the trapping laser before attempting coherent excitation is a useful technique used throughout this dissertation to ensure the target population starts in a single

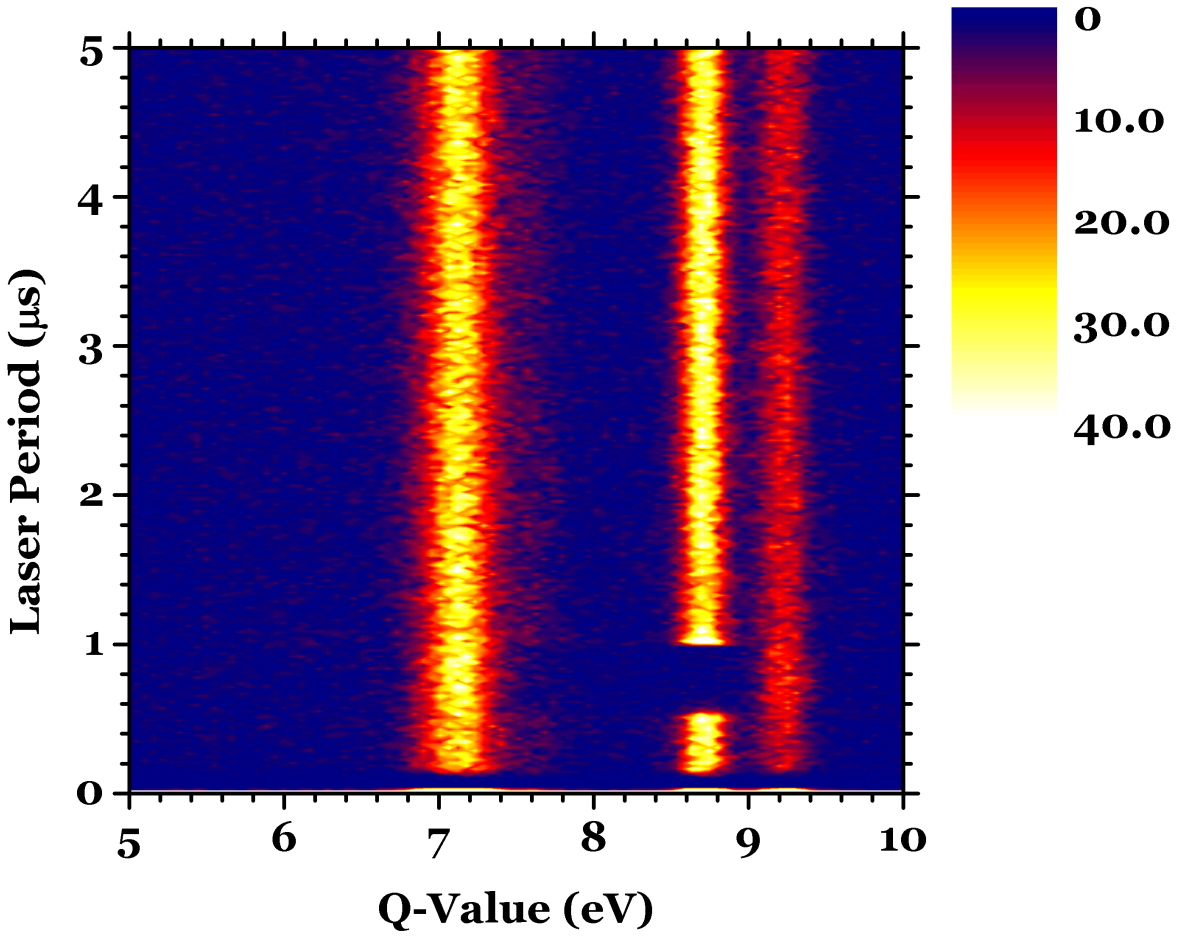

Figure 3.10: A typical time-to-amplitude converted (TAC) spectra, showing the laser period as a function of Q-Value. The third dimension (false color) represents the number of counts.

state rather than a mixture of ground and excited states.

3.2.2 Laser Control

Coherently exciting the MOT target requires two additional laser fields, pulsed at appropriate times. This added complication to the already-complex MOTRIMS scheme requires one to align six trapping laser beams (not including the repump beam, which is aligned separately), the projectile ion beam, and two coherent excitation laser fields to coincide at the same physical position in space at which the magnetic field is zero. The coherent excitation laser beams must also be pulsed appropriately in the STIRAP configuration, and must

be timed accurately to within a few nanoseconds. Such a mixture of spatial and temporal control is difficult, at best.

In order to make clean, unconvoluted measurements of coherent excitation, the trap laser beams are turned off before the coherent excitation phase, and turned back on again after the population in the MOT has had time to relax back into the ground state. A typical laser cycle is shown in Fig. 3.11. The trap laser (shown in red) is turned off for 500 ns, while the Stokes and pump lasers (shown as yellow and blue lines, respectively) are both pulsed for 50 ns. The trap is turned on again for a total period of 5 μs.

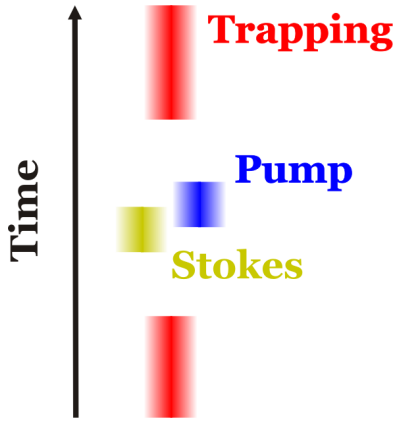

Figure 3.11: A typical laser cycle for a system undergoing coherent excitation *via* STIRAP. In this particular example, $\tau = w_p = w_s$.

The laser fields must be turned on and off on a timescale that is small compared to the excitation and decay timescales of interest. This is accomplished by using acousto-optical modulators (AOMs) that use sound waves to diffract electromagnetic waves. As the laser light passes through the AOM crystal, it is deviated both physically and in frequency, and leaves the crystal on a different physical trajectory than the incoming light. Thus, the experiment can be arranged in such a manner that when the AOM has radio-frequency (RF) signal applied to it, the light is deviated into the experimental chamber. By rapidly switching the RF signal on and off, one can turn the laser field on and off within the experimental chamber. The AOM controllers were designed and constructed in-house. More details on the control electronics for the AOMs can be found in Appendix B.2.

Figure 3.12 shows the entire operational setup for controlling the laser fields. Three cw diode lasers are employed, along with a tapered amplifier (TA). The initial trapping laser is referred to as the "master" laser, producing about 30 mW, and is injected into the

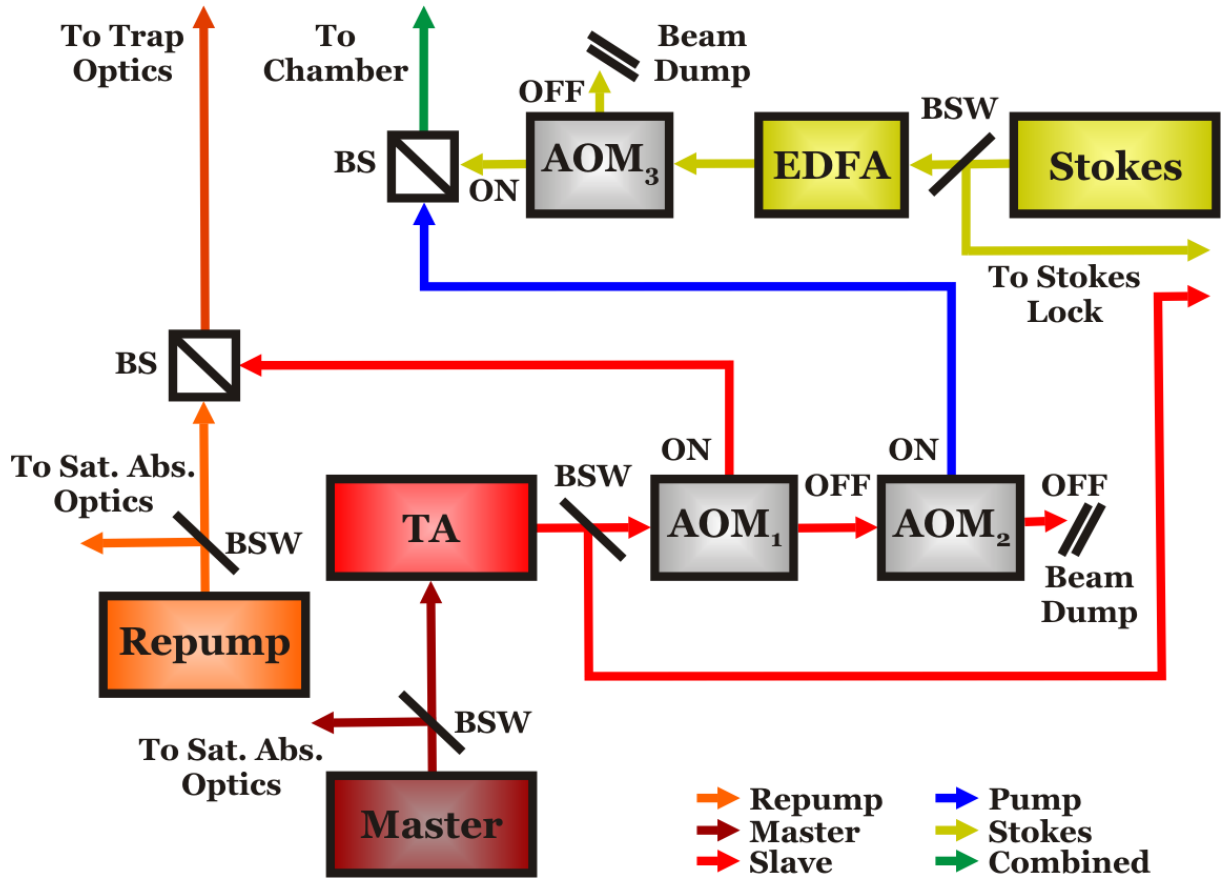

Figure 3.12: The experimental setup for controlling multiple pulsed laser fields within the experimental chamber. The red line shows the trapping (and repump) laser fields, while the yellow and blue lines show the Stokes and pump laser beams. The green light represents the combined excitation fields. The label "BS" refers to a polarizing beamsplitter. "BSW" indicates a beam splitter wedge.

TA which has an output power of ~ 450 mW. The TA acts as a "slave" laser, and its output is then used as the trapping field. The TA is necessary because the slave laser light will serve as both the trapping field and the pump excitation field. Also, some of the initial master laser light is used for locking purposes, as discussed in Sec. 3.2.3. Because the light will be fed through two AOMs in succession, the resultant pump laser field would be too weak to perform efficient coherent excitation without the enhanced power output from the TA.

The trapping laser field is combined with the repump light prior to entering the chamber.

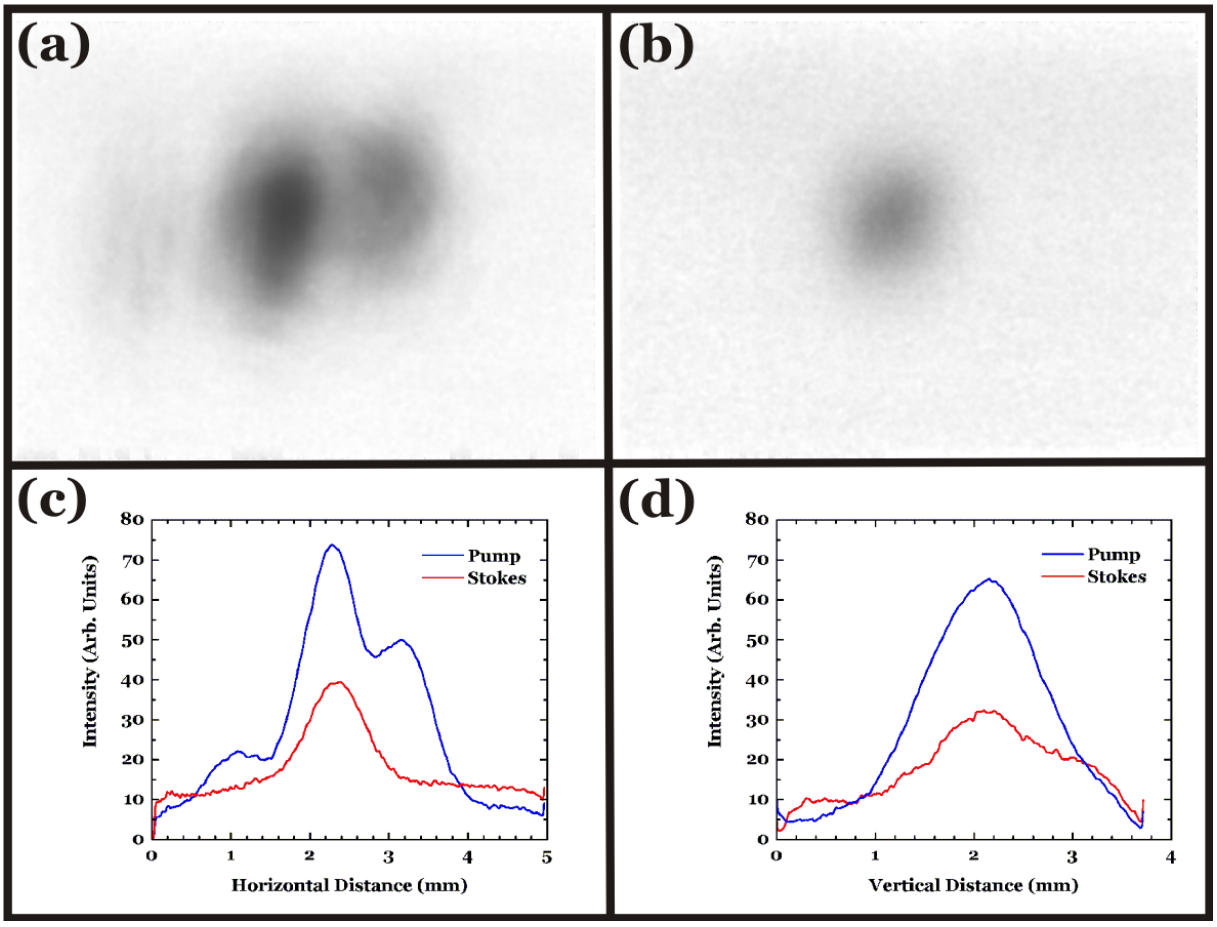

Figure 3.13: Typical beam profile measurements are shown for the Stokes and pump laser fields. **(a)** Pump laser field. **(b)** Stokes laser field. **(c)** Vertical beam projection, taken from the 2D images by integrating vertically. The FWHM in this particular case is $w_p = 1.47$ mm, $w_s = 0.77$ mm. **(d)** Horizontal beam projection, obtained in the same manner as (c). Here, the FWHM is $w_p = 1.19$ mm, $w_s = 1.37$ mm.

When coherent excitation is desired, AOM_1 is turned off, thus redirecting 100% of the laser light into AOM_2. This second AOM independently controls the pump field detuning, Δ_1. When the trap is turned off, the repump is not affected, since the trapping and repump light are combined well downstream of AOM_1. This is not important, since without the trapping transition active, the MOT population will immediately relax into the ground state ($F = 2$) where the repump has no effect.

To initiate coherent excitation, AOM_2 and AOM_3 are pulsed on. Note that the Stokes

field is passed through an erbium-doped fiber amplifier (EDFA) prior to passing into AOM_3, and thus is amplified by a factor of 1.66. The Stokes and pump laser light are combined using a long-pass filter prior (not shown in Fig. 3.12) to entering the chamber, and thus good spatial overlap is achieved. Both beams can be deflected into photodiodes to measure their temporal shape, or deflected onto a charge-coupled device (CCD) camera to measure their spatial profiles. Figure 3.13 shows a sample of a typical beam profile measurement. Such measurements are necessary to determine the intensity (and hence, the Rabi frequency) of the Stokes and pump fields. The full-width, half-maximum (FWHM) of the 2D laser field is assessed by integrating the beam profile first vertically and then horizontally, and finding the FWHM for each resultant peak. The average of the two measurements was used as an estimate of the beam diameter, and then, after measuring the beam power with a power meter, the total intensity was found by the relation

$$I = \frac{P}{(\pi \frac{d}{2})^2}, \tag{3.5}$$

where d is the diameter of the laser beam.[155]

The laser timing is controlled by a programmable board, installed on a PC with custom software written[156] in LabVIEW[157] allowing one to select various pulse configurations. Figure 3.14 shows a sample of the program, called the "***Arbitrary Waveform Generator***", or AWG. Up to 32 channels are available for controlling multiple TTL pulses, each with time step control down to 50 ns. For Fig. 3.14, channels 1, 2, and 3 control AOM_1, AOM_2, and AOM_3, respectively. The total time period is 5 μs. It is important to note that the absolute timing between channels is not uniform, meaning that differing optical path lengths, control cables, *etc.* must be taken into account in order to calculate when the laser pulses will actually arrive at the chamber. Thus, it appears from Fig. 3.14 that the delay between the Stokes and pump pulse is 200 ns, whereas in reality τ was measured optically *via* photodetectors to be 50 ns. The relative timing, however, is accurate. That is, once an absolute time relation has been determined, one can, for example, shift the Stokes pulse by

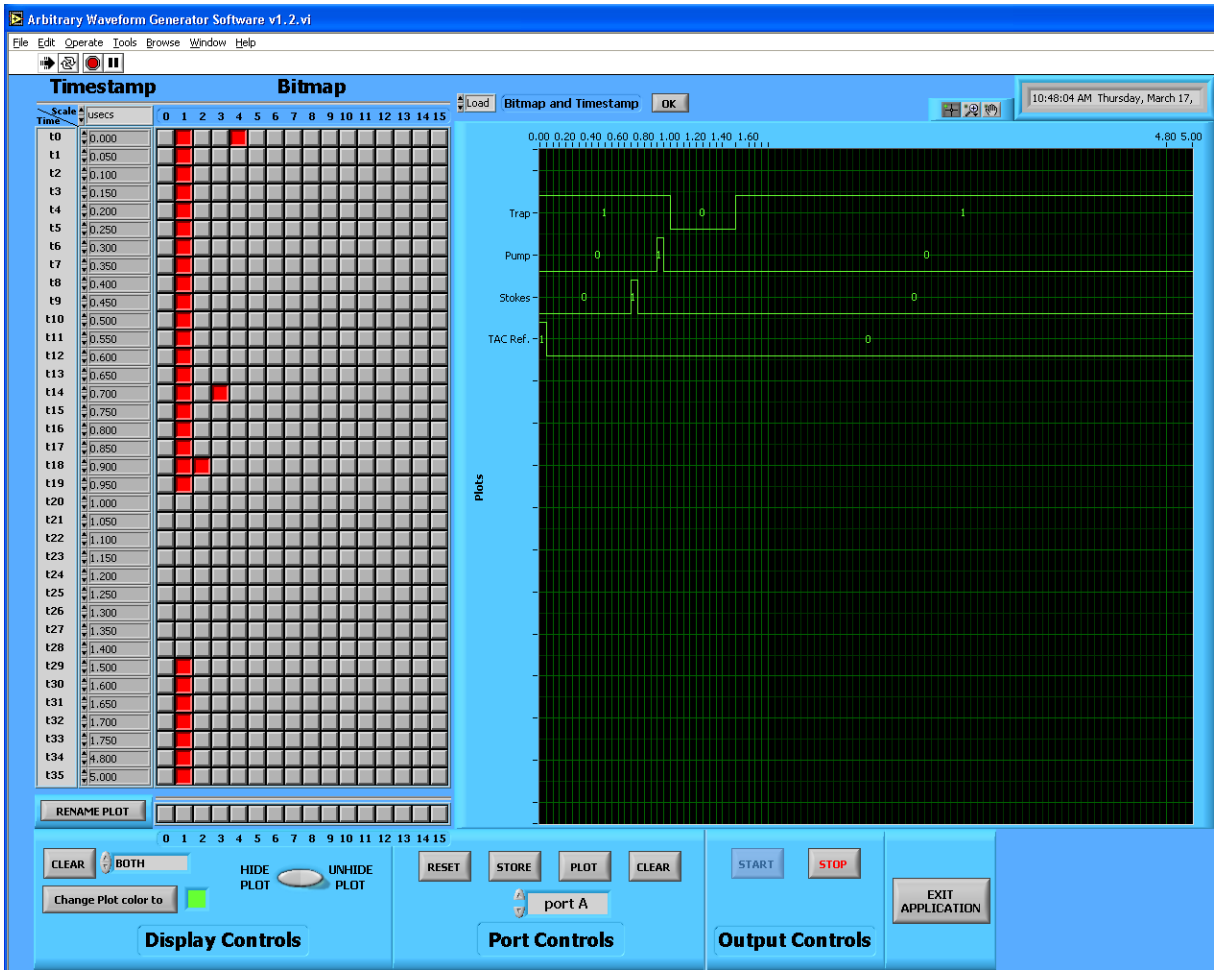

Figure 3.14: A screenshot of the AWG laser control software.

50 ns, and see a corresponding 50 ns shift optically. The minimum temporal step size using the AWG software is 50 ns, while the precision is better than 1 ns. With the additional use of passive delay electronics, the minimum temporal step size is 0.5 ns.

3.2.3 Frequency Detuning

Along with precise timing of the lasers, the frequencies must be precisely controlled. As mentioned above, the AOMs used to switch the laser fields on and off also induce a frequency shift in the laser light. This frequency shift can be tuned *via* the AOM controlling electronics, which are included in Appendix B.2. AOMs typically allow one to tune the frequency of the

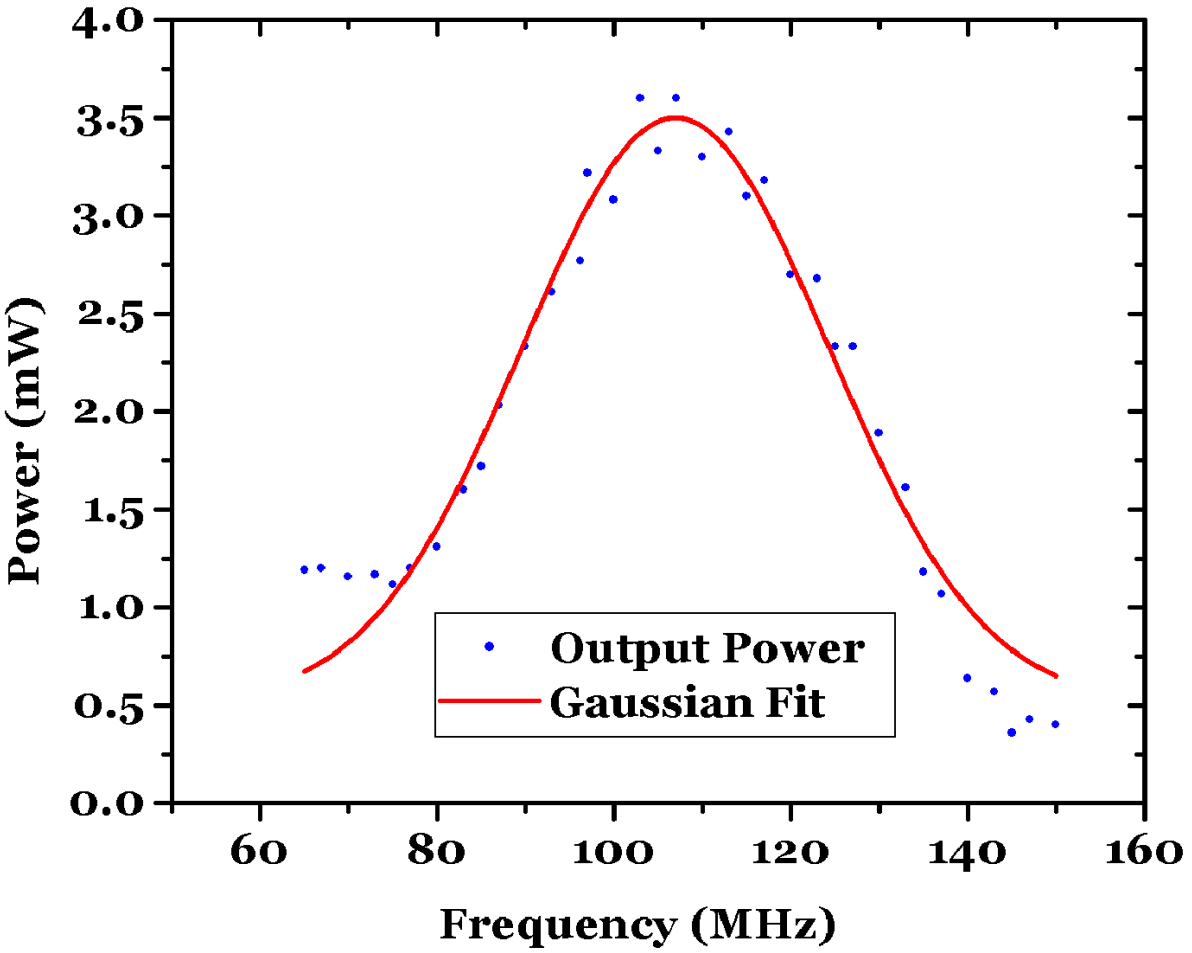

Figure 3.15: AOM_3 operating range. Power is plotted as a function of applied frequency.

propagated light across 40 or 50 MHz. For example, Fig. 3.15 shows the typical operating frequency range for AOM_3. Measured output power through the AOM is plotted *versus* applied frequency. The optimal transmission frequency is ~ 107 MHz.

Figure 3.16 shows the frequency relationship between the master and Stokes laser fields, as well as the frequency applied to each AOM. The trapping laser field is locked 133 MHz to the red of the $F' = 3$ transition using saturated absorption spectroscopy, as discussed in Sec. 3.2.4. Yet, as mentioned in Sec. 3.1.1, the trapping lasers should be detuned by $\delta \simeq 20$ MHz from resonance with the $5s - 5p$ transition in order to optimally cool and trap the target population. AOM_1, therefore, is used to blue-shift the trapping lasers closer to

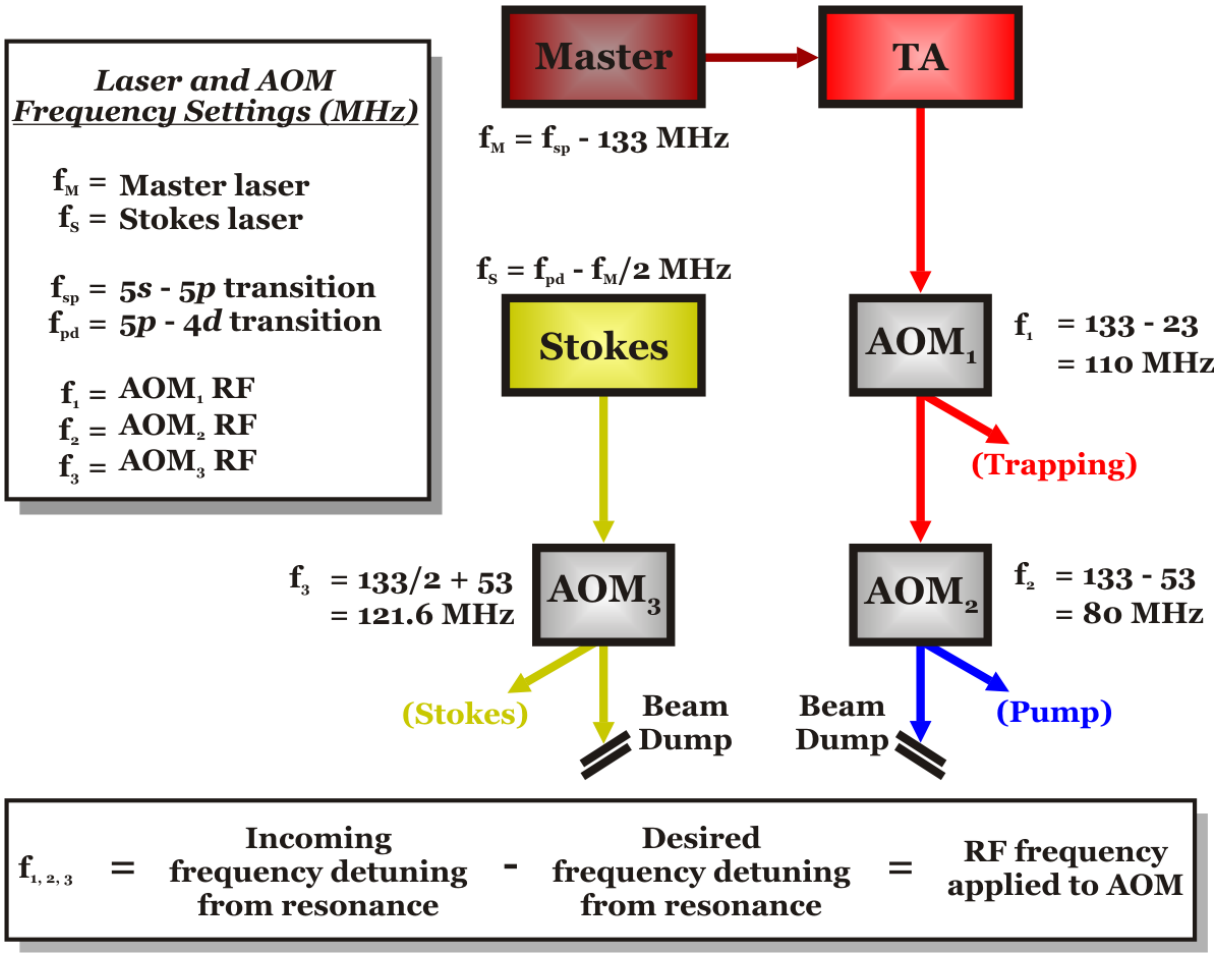

Figure 3.16: Diagram showing frequency relationships between the Master and Stokes lasers, as well as the applied RF for AOM_1, AOM_2, and AOM_3.

resonance, and is usually tuned to ~ 110 MHz. This provides $133 - 110 = 20$ MHz detuning to the red of the $5s - 5p$ resonance, as required for trapping.

When AOM_1 is turned off, the undeflected (and unshifted) slave laser light enters AOM_2 to be properly detuned (and physically deflected) for use as the pump laser field during coherent excitation. AOM_2 shifts the light to the blue by 80 MHz, such that the pump laser field is ultimately 53 MHz below the $5s - 5p$ transition, as seen in Fig. 3.17. The pump laser is purposefully far from resonance because it is not desirable that pump laser light alone should excite population into the $5p$ state. Because the optimum coherent excitation for this system is a two-photon resonant process, as shown in Fig. 3.17, the Stokes laser must

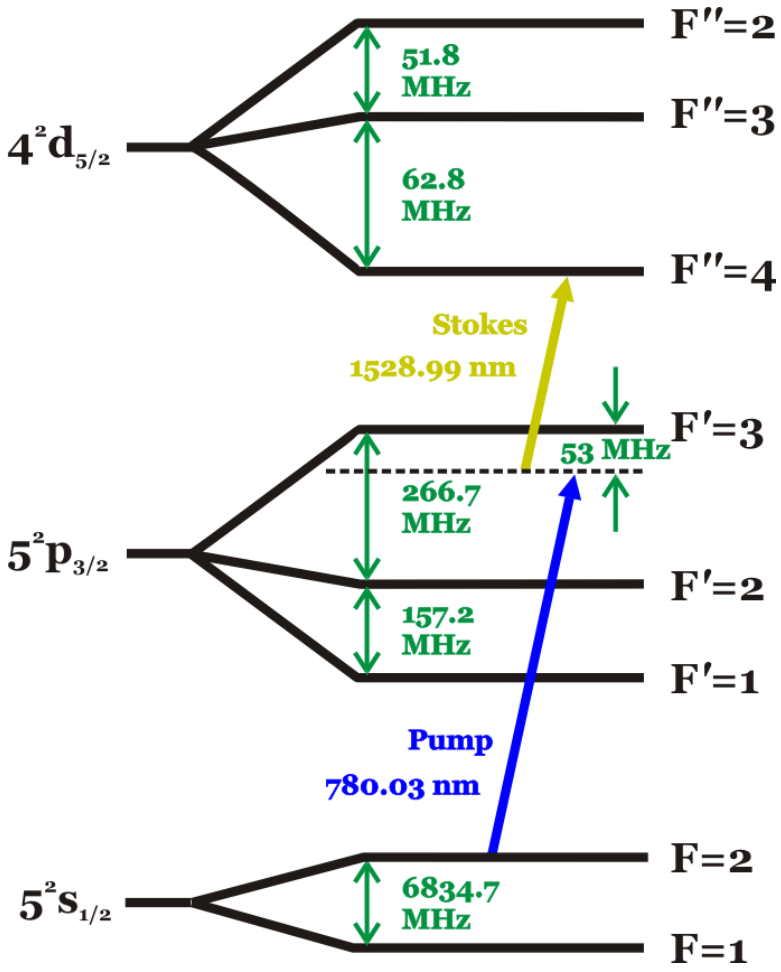

Figure 3.17: Two-photon resonant coupling of the $5s$ and $4d$ levels for coherent excitation *via* STIRAP.

make up the remaining frequency range in order to couple the $5s$ ground state and the $4d$ excited state. The Stokes light is initially locked to a frequency 133/2 MHz to the red of the $5s - 5p$ transition. (See the end of Sec. 3.2.4 for details on locking, and an explanation of this factor of 1/2 difference from the master lock.) It is then passed through AOM_3 and detuned to the blue by 121.6 MHz. This detuning configuration for the pump and Stokes fields ensures two-photon resonance between the $5s$ and $4d$ energy states.

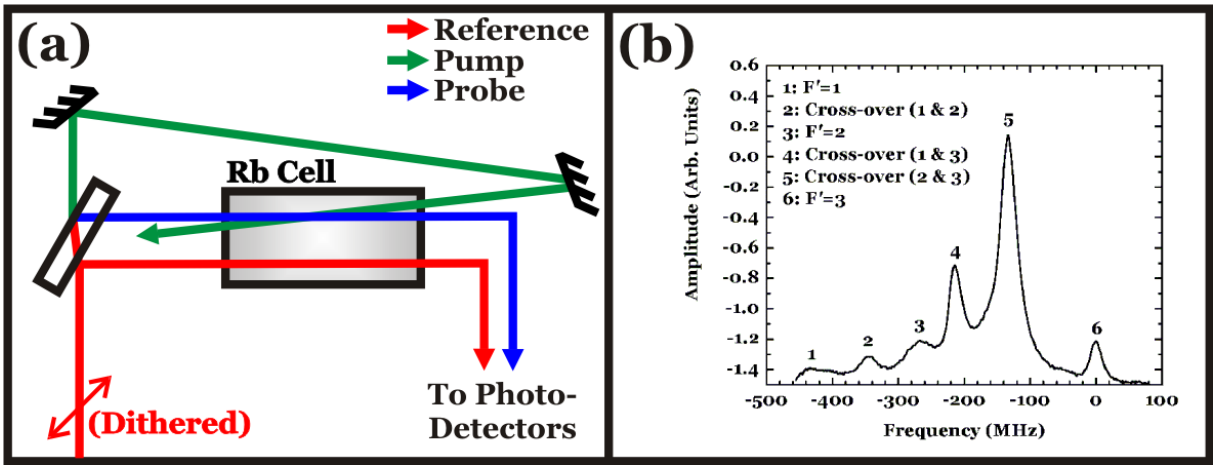

Figure 3.18: Saturated absorption spectroscopy used for laser locking. **(a)** Schematic diagram of laser dither locking technique. The laser is dithered to produce an absorption spectra. **(b)** Saturated absorption spectrum produced from such a technique.

3.2.4 Laser Dither Locking *vs.* B-Field Locking

In order for the laser field frequencies in all cases (trap, repump, pump and Stokes) to be useful, one must first lock the lasers to a known frequency reference. This is done *via* saturated absorption spectroscopy, a common spectroscopic technique used in studying atomic structure.[158,159] The general technique is described elsewhere,[137] and a schematic diagram of the saturated absorption optics are shown in Fig. 3.18a. The green line is the pump beam, while the blue line is the probe beam. The pump beam will be absorbed by the group of atoms in the cell whose velocity distribution is such that the laser frequency is resonant with an available atomic transition. Similarly, the probe beam also interacts with a group of atoms having a particular velocity distribution. Because the two beams are passing through the Rb cell in nearly opposite directions, they interact with two different velocity groups. Only when the atoms are nearly stationary will both beams interact with the same velocity group.

The red line in Fig. 3.18a is a reference beam. Because the probe and reference beams are propagating in the same direction, subtracting the reference beam from the probe beam yields zero except for frequencies where the pump beam saturates the transition of the

stationary atoms. By dithering the laser across a range of frequencies, one may produce a Doppler-free saturated absorption spectrum. Figure 3.18b shows such a Doppler-free absorption spectrum, for the ^{87}Rb transition ($5s_{1/2}$, $F = 2$ to $5p_{3/2}$, $F' = 1 - 3$). All states shown are caused by the trapping light tuned into resonance from $F = 2$ to the indicated F' energy level. For example, peak 3 is the resonant transition from $F = 2$ to $F' = 2$.

Cross-over peaks also occur in saturated absorption spectra when the laser frequency is halfway between two transitions originating from the same lower level. A particular velocity group of atoms will be red-shifted into resonance with the pump laser field, while this same velocity group will be blue-shifted into resonance with the probe laser field. The cross-over peak, labeled "5", is used as a reference peak. This peak sits 133.35 MHz below the $F' = 3$ trapping transition, labeled "6". Peak 5 was chosen for locking purposes due to its dominating size. Two different techniques were used for locking the lasers to specific transition frequencies. The generic process, termed "laser dither locking" [160] is accomplished by applying a low frequency ($\sim$ 15 Hz) dither to the laser field. By selecting the dither amplitude, one effectively selects the range of the absorption spectrum to sweep. As the amplitude is decreased, a smaller range of the spectrum is sampled. In this manner, one can "zoom in" on a particular peak, in this case, peak 5.

Appendix B.1 shows the locking electronics used to generate a feedback signal to the lasers in order to keep the frequency stable. This is explained graphically in Fig. 3.19. In the case where the peak is unshifted (Fig. 3.19a), the lock electronics dither the laser frequency, the saturated absorption signal being shown in the upper part of Fig. 3.19a. The corresponding square-wave reference signal, shown in red, is multiplied by the absorption signal, resulting in the curves of Fig. 3.19b. If the peak is centered on the reference signal, exactly half of the peak will be inverted. This results in an integrated signal of zero produced as output from the lock box, and thus no feedback is sent to the laser for correction.

If, on the other hand, the signal drifts in frequency (Fig. 3.19b), the resulting inversion will not happen at the peak's center, and thus the integrated signal will not be zero. This

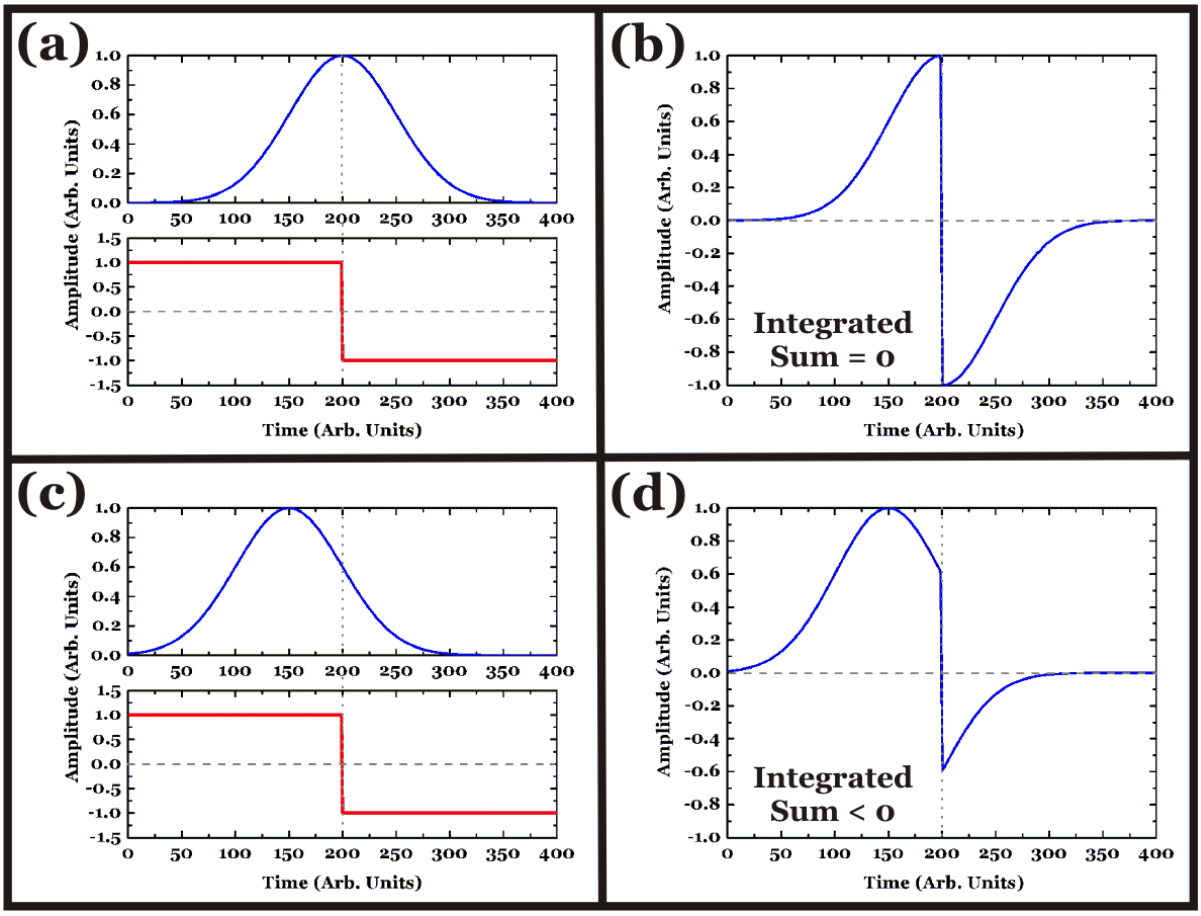

Figure 3.19: A graphical representation of the laser dither locking process. **(a)** The desired peak for locking is shown in blue, along with the reference signal used by the locking electronics, shown in red. **(b)** The product signal from part (a). **(c)** A slight shift in peak position is shown, relative to the reference voltage. **(d)** The product signal from part (c).

positive (or negative) output signal is fed back into the laser, producing a frequency shift that returns the laser frequency to the center of the peak. In this manner, one can lock the lasers to an absolute reference peak, continually correcting for frequency drifts caused by laser instability, temperature changes, *etc.*

The advantage to laser dither locking is its robust nature. The lasers will remain locked even with acoustical noise present, large ambient temperature variations, or other interference sources. The disadvantage to laser dither locking is the fact that it inherently requires enough dither amplitude to detect a peak of some sort for locking. Experimentally, the

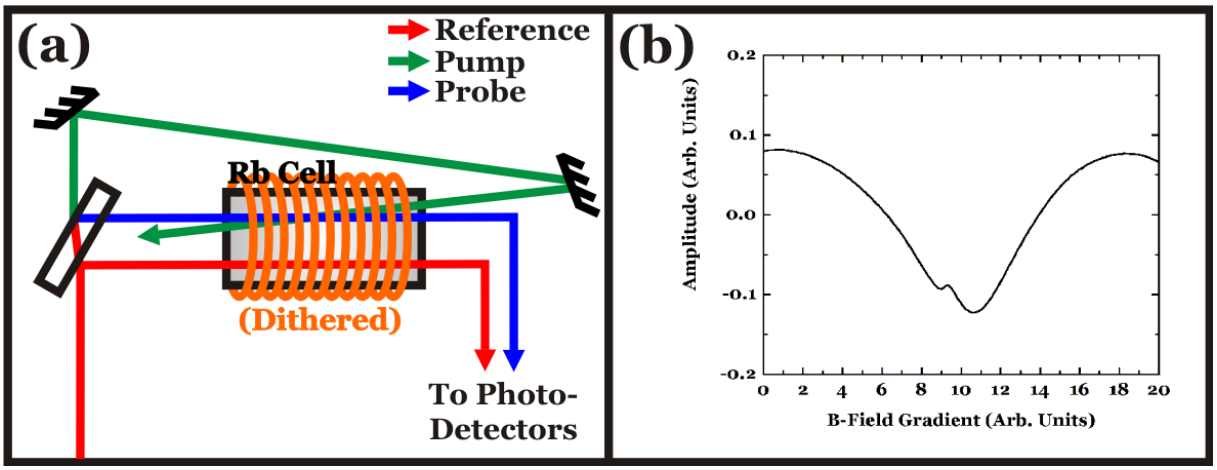

Figure 3.20: Saturated absorption spectrum used for B-field locking. **(a)** Schematic diagram of B-field locking technique. The laser is kept at a stable frequency, and the B-field is dithered to produce an absorption spectrum. **(b)** Saturated absorption spectrum produced from such a technique. Only the $2-3$ crossover peak is shown here.

smallest dither amplitudes achievable while still remaining under locking conditions results in a large enough frequency dither to pass through the entire trapping range of the MOT. In other words, if the MOT is detuned too far from resonance (either to the red or blue), atoms will not be trapped. With too large a locking dither amplitude, the lasers continually sweep through the optimum trapping frequency, thus trapping and releasing the population. This causes instability in the MOT target and a smaller average number of target atoms. Furthermore, as will be shown, the slave laser field is used to lock the Stokes laser to the proper resonant frequency for two-photon resonance. Since the Stokes laser is also controlled by this laser dither locking technique, a "double sweep" effect, where both the master and Stokes frequencies used for the Stokes lock are dithered at slightly different rates, destroying any reasonable control of the Stokes laser frequency.

Laser dither locking was used to control the repump laser, since its laser frequency was not critical. Laser dither locking also continued to be used for the Stokes laser because, as shown in Sec. 2.5.3, the precise frequency of the pump and Stokes fields does not greatly affect STIRAP efficiency. However, a different technique was eventually employed to stabilize the master laser and, consequently, the trapping and pump lasers.

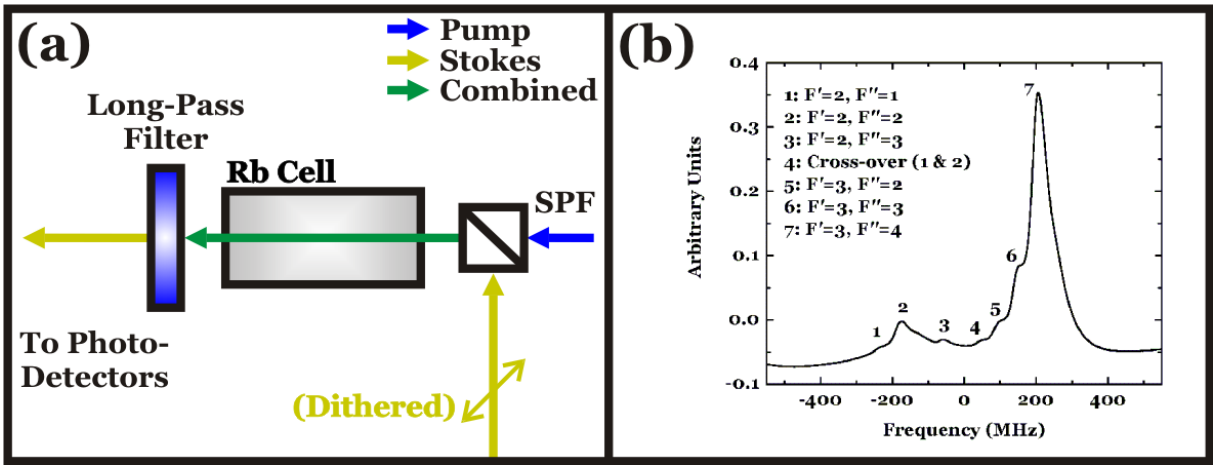

Figure 3.21: The Stokes laser field is locked using two-photon absorption spectroscopy. **(a)** Schematic showing co-propagating pump and probe beams. The label "SPF" represents a short-pass filter. **(b)** Resultant absorption peak used for locking.

Magnetic field locking[161,162] carries many similarities compared to simple peak-locking as can be seen in Fig. 3.20a. The optical setup is unchanged, except for the addition of a magnetic field coil wrapped around the rubidium cell. The basic electronics used are the same as well, except the frequency of the laser field itself is not dithered. Instead, a weak magnetic field produced from a coil around the rubidium cell is dithered. This B-field dithering results in Zeeman shifts of the laser light, causing similar absorption peaks to appear in the absorption spectrum as when one frequency dithers the laser light directly, as shown in Fig. 3.20b. The same technique of integrating over a peak is employed, and the lock electronics function identically to the laser dither locking description given above.

The advantage to B-field locking, however, is noteworthy. Since the laser frequency itself is not dithered, the laser field is much more stable. Only when the locking electronics detect that the absorption spectrum has changed (again, due to ambient temperature variations, acoustical noise, *etc.*) is there any correction applied to the laser frequency. This eliminates the stability problems inherent in simple laser dither locking, and provides an essentially noise-free laser field.

With the master laser field locked *via* this B-field locking technique, the remaining

problem is to lock the Stokes field to a similar absolute reference peak, as with the previous two lasers. This is done by utilizing a piece of the slave laser field prior to passing through AOM_1. That is, the slave light is locked 133 MHz below the $5p, F' = 3$ transition in the manner described above, and fed into a second room-temperature rubidium cell, as shown in Fig. 3.21. A portion of the Stokes laser light is also passed through this cell in the same direction as the pump. Because both the pump and Stokes fields are passing through the cell in the same direction, they both interact with the same velocity group of atoms that are Doppler-shifted into resonance with the laser fields. However, the Stokes laser wavelength is a factor of two larger than the pump light. Since frequency is inversely proportional to the wavelength ($\nu = c/\lambda$), the Stokes light interacts with atoms that are Doppler-shifted 133/2 MHz from resonance.

Because the pump field is independently locked, dithering the Stokes laser field amplitude will once again cause an absorption peak to appear at the two-photon resonant frequency of the combined pump and Stokes light. Employing laser dither locking techniques, this two-photon resonance is maintained by locking to the $F' = 3 \rightarrow F'' = 4$ transition, labeled as peak 7 in Fig. 3.21. Thus the Stokes field is locked to the proper frequency to ensure two-photon resonance with the $4d$ state. Laser dither locking (as opposed to magnetic field locking) is sufficient to keep the Stokes laser locked at a known frequency. With a different excitation technique (say, π-pulses), it is likely that a different locking scheme would be necessary to ensure a more rigorous two-photon resonant frequency state.

3.3 Data Acquisition

The data acquisition process is broken into two major events: the detection of a projectile particle and the detection of a recoil ion. As mentioned in Sec. 3.1.2, these two events are the heart of the RIMS spectroscopy method. Essentially, the acquisition system records the time difference between projectile and recoil events, along with the relevant 2D positions on the corresponding detectors. At the time such a coincidence event takes place, the status

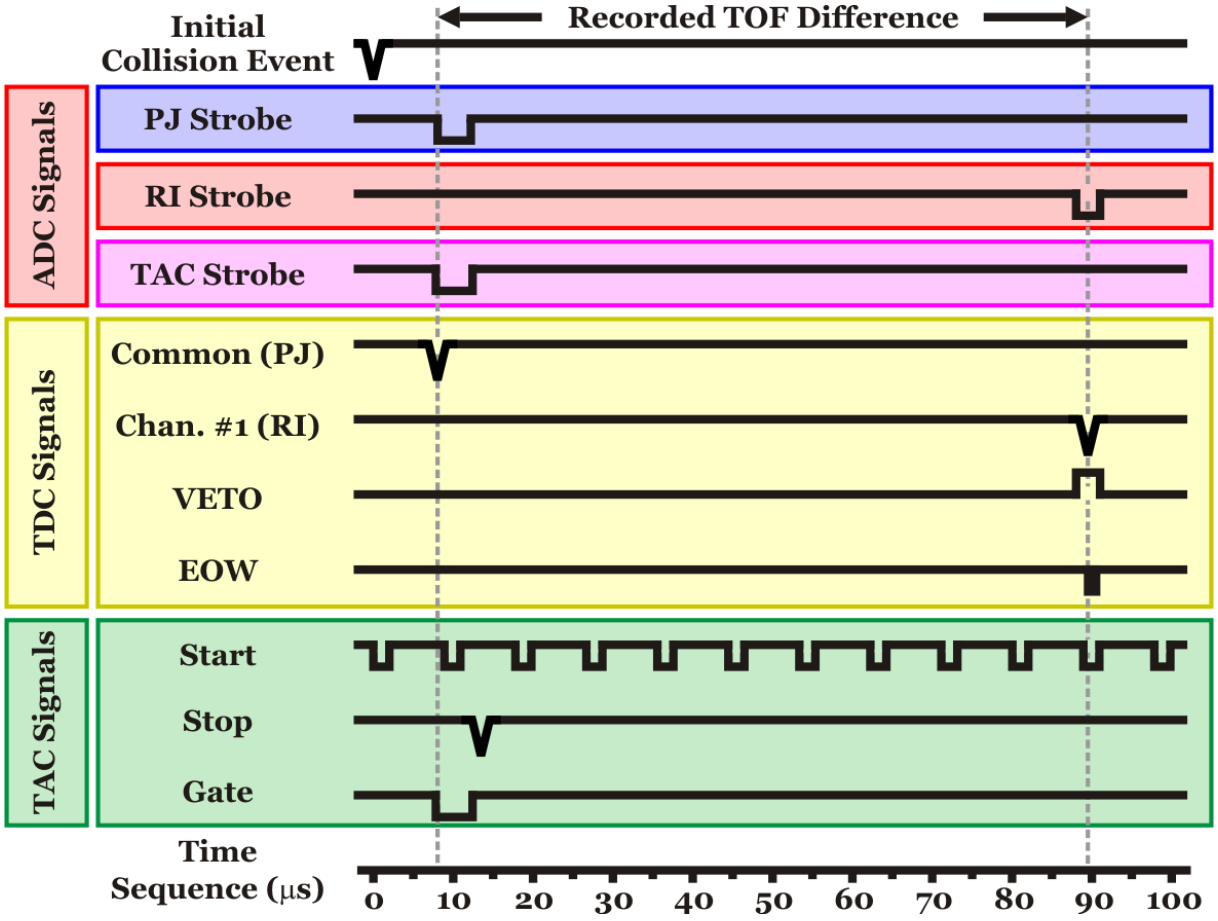

Figure 3.22: Data acquisition time line showing a typical event cycle, and the corresponding important events. The Projectile signals are labeled "PJ", while the recoil ion signals are labeled "RI".

of the excitation laser fields is also recorded such that every event can be correlated to a given laser configuration. It is essential to know whether the Stokes laser was present, for instance, in order to correctly interpret the acquired data.

Figure 3.22 shows the acquisition time line and the important events that take place in a typical detection cycle. The entire acquisition process begins with the detection of a projectile particle. The flight time of the projectile is about 8 μs. Upon detection of the projectile, one can deduce that charge transfer must have taken place, and thus a corresponding recoil ion will eventually reach its detector. The flight time of the recoil ion is about 90 μs. Because of the geometrical arrangement of the beamline, the recoil detector

tends to have a higher count rate compared to the projectile detector due to "spray" from the projectile ions that inevitably strike the recoil detector downstream as shown conceptually in Fig. 3.23. In order to eliminate an overwhelming number of false coincident detections, a 3 μs hardware window is set 90 μs after the projectile is detected. Ion events detected by the recoil detector prior to the activation of this window are ignored; thus, obviously false coincidences are rejected.

When the window is active, the first recoil ion to be detected causes a time-to-digital converter (TDC) to record the time difference between the projectile and recoil events, as shown in Fig. 3.22, labeled "Recorded TOF Difference". The "veto" signal shown in Fig. 3.22 is the event window mentioned above, while the "end-of-window" (EOW) signal controls the operation of the TDC. Thus, the TDC measures the TOF difference between the projectile and recoil events, and hence measures the Q-Value for the every collision event. The TDC employed in the experiments presented in Sec. 4 has a time resolution of 1 ns. However, the energy spread of the projectile ion beam, a couple of eV in 7 keV, degrades the time resolution to about $1.5 - 2.0$ ns.

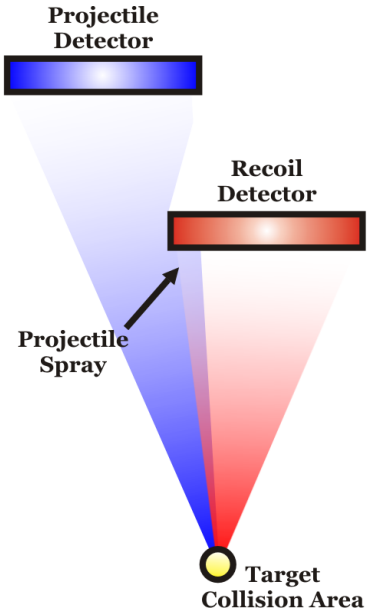

Figure 3.23: Conceptual picture of the detector "spray".

At the time of the projectile event, the status of the lasers is also recorded *via* a time-to-amplitude converter (TAC). A pulse is generated from the AWG software (channel #4 in Fig. 3.14) at the beginning of each laser cycle. This signal is used to start the TAC timer. Meanwhile, a hardware gate is generated at the time of the projectile event detection, along with a pulse used to stop the TAC. This last pulse has been delayed until the end of the hardware gate, as shown in Fig. 3.22. Incoming start signals from the AWG software are ignored by the TAC unless the gate is activated. The gate size is the same length as a single laser period, thus only a single start/stop combination is

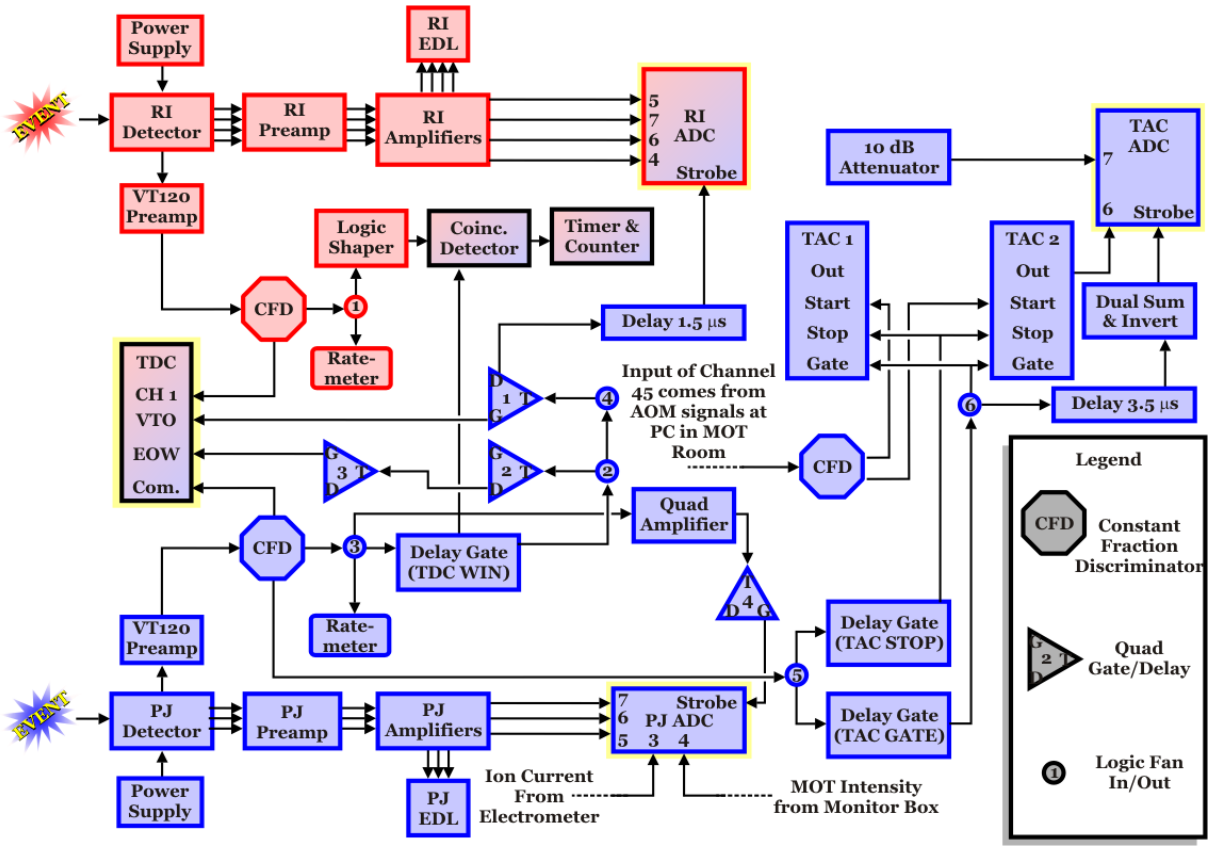

Figure 3.24: The data acquisition hardware in schematic form.

detected per projectile event.

3.3.1 Hardware Configuration

A detailed schematic of the data acquisition hardware is shown in Fig. 3.24. Electronics shown in blue are activated by a projectile event, while electronics shown in red are activated by the recoil event. Those electronics shown in both red and blue are indicative of signals received from both projectile and recoil events. Incoming signals begin in two places, shown as "events" on the diagram, corresponding to the projectile and recoil events. The signals terminate in four places: the projectile analog-to-digital converter (ADC), the recoil ADC, the TAC ADC, and the TDC, each shown with a faint yellow box around them in Fig. 3.24.

When a projectile event is detected the TDC common is immediately activated, starting the TDC timer. The corresponding hardware gate for the TDC is generated after passing

through a series of delay/gate generators, and arrives at the TDC almost 90 μs later. The projectile ADC records the position information of the projectile detector, and waits to be read with the rest of the electronics.

A separate signal derived from the projectile event is sent to a pair of TACs in order to record the status of the lasers in the experiment. Two identical TACs are employed, labeled TAC_1 and TAC_2. The TACs are configured in the same manner, except TAC_1 has a 10 dB attenuated output signal. The attenuator converts the $0 - 10$ V output of the TAC into a $0 - 2$ V input for the ADC, thus allowing one to measure the entire laser time period. TAC_2, however, is read directly without attenuation. All voltages above 2V are read as 2V by the ADC. This yields a high-resolution measurement of the first 20% of the TAC range. Therefore, if the lasers are arranged in such a way where the pump and Stokes pulses arrive during the first 20% of the laser period, the measured resolution of the population dynamics will be improved by a factor of 5.

The recoil event triggers the stop to the TDC, recording the TOF difference between the projectile and recoil events. The 2D detector information is also measured by the recoil ADC. At this point, when the recoil ADC is strobed, all four terminal acquisition electronics (projectile, recoil and TAC ADCs, and the TDC) are polled by the XSYS acquisition software. The electronics are cleared, and the system is reset for the next event.

3.3.2 XSYS Data Acquisition System

The XSYS data acquisition software, run on a VAX with VMS operating system, controls the reading and writing of data from the acquisition electronics discussed in the previous section. An event language file, along with supporting header files, directs XSYS on how to record and analyze incoming data. The relevant files can be found in Appendix C. When a session is begun, all events are recorded to disk as they arrive from the experimental apparatus. When a session is rerun, the data are regenerated as if originating from the experiment. Thus, data can be taken and re-analyzed several times with various software

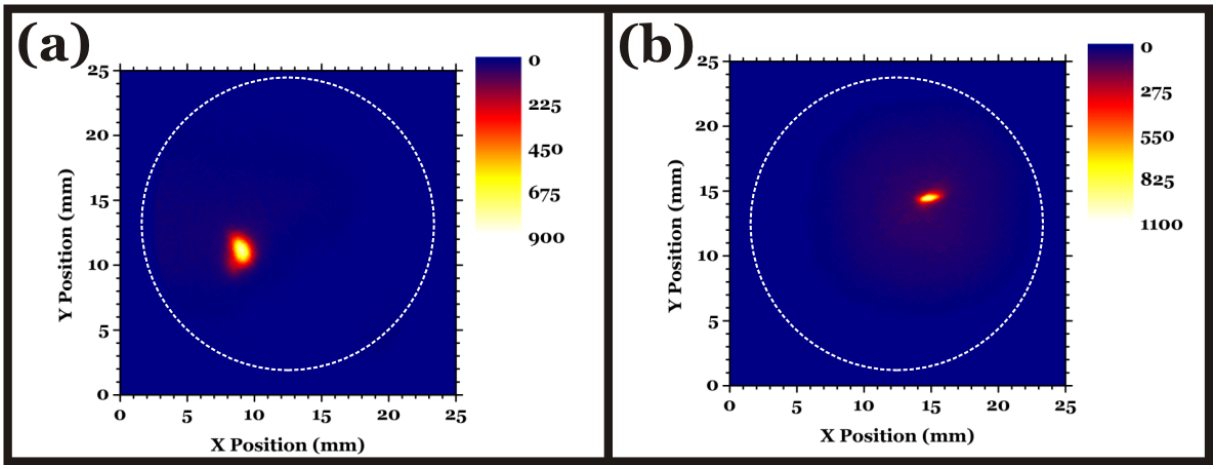

Figure 3.25: Typical images obtained from the **(a)** projectile and **(b)** recoil detectors. The white dotted lines are visual representations of the actual detector size.

settings. This is called "list-mode" acquisition.

Figure 3.25 shows typical images measured on the projectile (Fig. 3.25a) and recoil (Fig. 3.25b) detectors. The active diameter of each detector is 40 mm. When charge-transfer is taking place, a single well-defined spot appears on each detector. Elective 2D software gates can be set in both the projectile and recoil detector data areas to discriminate against recording counts in certain regions, if desired. This allows one to select regions of interest for analysis, or reduce background events and improve the signal-to-noise ratio (which is already quite low, as seen in the figure). Gates can also be set in the TOF data area. More importantly, cuts can be made of each distinct Q-Value channel (seen as dark vertical stripes on TAC_1 and TAC_2 in Fig. 3.26) or cuts can be made over a specific time range within the TACs. The ability to manipulate the data in such a manner makes the XSYS software a powerful acquisition tool.

Figure 3.26 shows a typical XSYS data acquisition session with 12 active data areas. In this particular example, no coherent excitation lasers are present. The trapping lasers are turned off for 500 ns out of a total laser period of 5 μs. The four left-most data areas (labeled $51-54$) are typically used for on-the-fly data analysis. In this particular example, cuts of the $Rb(5s) \rightarrow Na(3p)$ (view 51), $Rb(5p) \rightarrow Na(3p)$ (view 52), and $Rb(5s) \rightarrow Na(3s)$

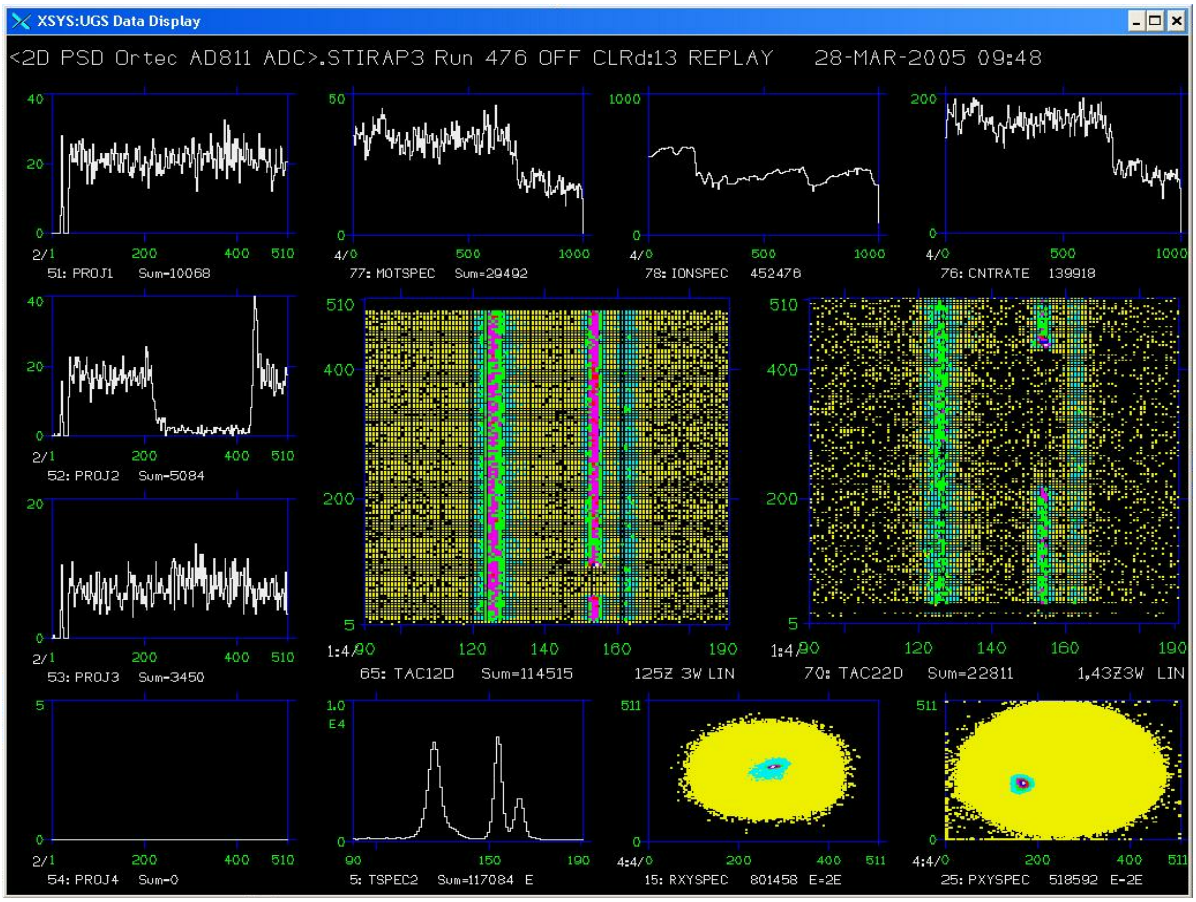

Figure 3.26: A typical XSYS data acquisition session. Click the image to view the film, or click here to launch an external media player to see the film at full scale.

(view 53) channels are shown. The bottom-right three windows (labeled 5, 15, and 25) measure the time-of-flight, recoil detector, and projectile detector information, respectively. The top-right three windows (labeled $76 - 78$) monitor the total count rate, the condition of the ion beam, and the condition of the MOT, respectively. Finally, the central large windows show the TAC_1 and TAC_2 measurements (labeled 65 and 70, respectively).

Typical operating conditions allow one to take data indefinitely, so long as the MOT target remains stable, the ion source is available, and the computer hard disk has available space. However, corrections must be made for projectile ion beam drift.

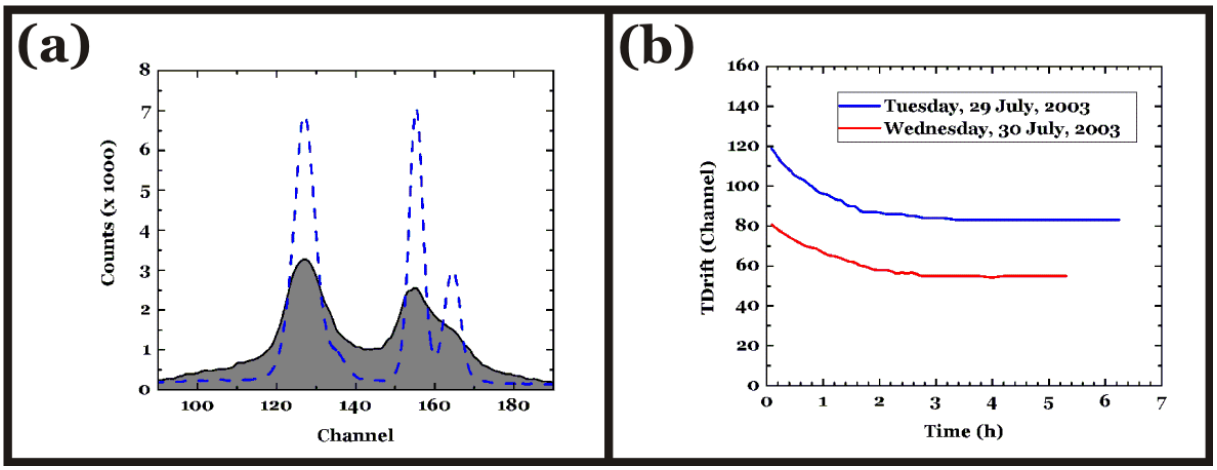

Figure 3.27: Projectile ion beam drift as a function of time. **(a)** Counts *versus* Q-value. The gray shaded peaks show a data session with no drift control. The blue dotted line spectra shows the same data taken with the drift lock active. **(b)** Plot showing peak position *versus* time.

3.3.3 Drift Corrections

During the course of an acquisition session, data are collected for several hours. Over such long periods of time, the recorded time-of-flight spectra becomes blurred, as shown in Figure 3.27a. The gray spectrum is obtained without any compensation for drift effects, while the blue dotted line shows the same data acquired with a drift correction present. It is believed this blurring is caused by small changes to the initial kinetic energy of the projectile beam. Figure 3.27b shows the centroid of the $5s - 3p$ Q-Value (called the TDrift channel) as a function of time. Initially, the drift is quite large, while after several hours have past, the beam has stabilized.

This drift can be corrected as data are acquired, or re-read. Appendix C.3.9 shows a short program written in VMS ***DIGITAL command language*** (DCL) to correct for such drifts. In effect, the program samples the centroid position of a selected peak (usually the $5s - 3p$ peak) in the TOF spectra. If the centroid of this peak changes by more than one channel, the acquisition program shifts all incoming TOF data by one channel in the opposite direction. This effectively minimizes the ion drift.

Chapter 4

Experimental Results

The experimental data shown in this chapter are presented as evidence that the theory and experimental techniques thus far described are not only accurate and valid, but are also the first high-resolution measurements of coherent population dynamics to-date. As mentioned previously (Sec. 3.3), the TDC resolution is 2 ns. The shortest decay time involved in the three-level system of interest is 27 ns (from $5p$ to $5s$), and thus the experimental technique yields ample resolution to measure such coherent population dynamics in detail. This chapter will present the results obtained from such experiments measuring coherent population transfer from the $5s$ to $4d$ states in various configurations and regimes, as well as discuss system resolution and error analysis methods in detail.

4.1 Data Analysis Methods

The TAC_1 and TAC_2 spectra obtained in a typical acquisition session contain most of the information necessary to deduce the population dynamics present in the system. Along with

Cross Section Ratio (σ)	Value	Error
$\frac{4d-5d}{5s-3s}$	12.86	1.68
$\frac{4d-5d}{4d-3s}$	2.12	0.26
$\frac{4d-4s}{5s-3s}$	6.06	0.26

Table 4.1: Relative capture cross sections from $^{87}\mathrm{Rb}(4d_{5/2})$

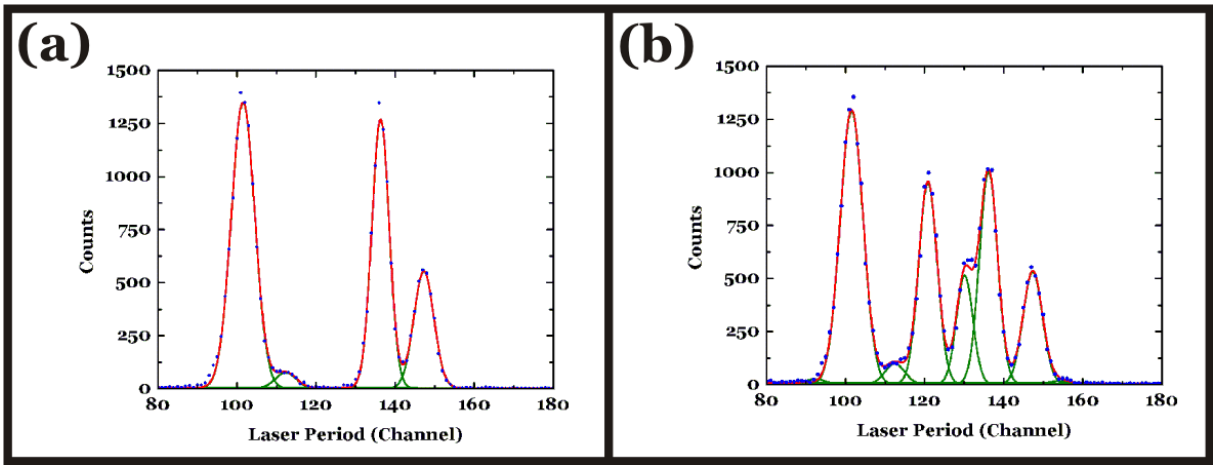

Figure 4.1: Comparison of the two different Q-value spectra for determining the σ_{ds}/σ_{ss} cross section. **(a)** Only the trapping laser field is present. **(b)** Stokes laser field is now present.

measurements of initial conditions, such as laser intensities, the only external experimental data required for analysis are the relative cross-sections involved in each of the relevant capture channels. Analyzing a data set requires knowledge of relative cross sections for four charge-transfer channels: σ_{dd}/σ_{ss} for Rb($4d$) $\rightarrow$ Na($3d$), σ_{ds}/σ_{ss} for Rb($4d$) $\rightarrow$ Na($4s$), σ_{pp}/σ_{ss} for Rb($5p$) $\rightarrow$ Na($3p$), and σ_{ss} for Rb($5s$) $\rightarrow$ Na($3s$), where σ_{ss} refers to the Rb($5s$ $\rightarrow$ Na($3p$) charge transfer channel. In fact, for *relative* populations, only the cross section ratios σ_{dd}/σ_{ss}, σ_{ds}/σ_{ss}, and σ_{pp}/σ_{ss} are required. Two of these, σ_{dd}/σ_{ss} and σ_{pp}/σ_{ss}, have previously been measured, and the results published, along with experimental details describing how such measurements were made.[137,141] Using a similar method, the cross sections involving the Rb($4d$) state were obtained, and are shown in Table 4.1. The remaining cross section, however, had to be determined prior to analyzing the newly acquired data. While the measurements have been published elsewhere,[163] they are presented here in detail.

4.1.1 Determining ds/ss Cross-Section

In order to measure σ_{ds}/σ_{ss}, Q-value spectra were taken first by exciting population in the target from the $5s$ ground state to the $5p$ intermediate state, and then separately by adding a Stokes laser field to excite population from this $5p$ intermediate state into the $4d$ terminal

state. Under these two experimental conditions the trapping lasers were left on continually, ensuring a high-quality MOT target. During excitation into the terminal ($4d$) state, the Stokes laser was tuned as close to resonance as allowed *via* AOM_3 ($f \simeq 103$ MHz) to ensure the greatest likelihood of excitation from the $5p$ to the $4d$ excited states. Note that no pump laser is used in these measurements, since the trapping lasers naturally excite population into the $5p$ intermediate state.

Figure 4.1 compares the Q-value spectra obtained under these two different experimental conditions. When only the trapping laser field is present in the system (Fig. 4.1a), four clear charge-transfer peaks are evident, indicating transfer involving only the $5s$ and $5p$ initial states in rubidium. When the Stokes field is present (Fig. 4.1b), three additional peaks appear, making a total of seven resolved capture channels. Additionally, the entire Q-value spectrum sits upon a small but noticeable background level that is not present when the Stokes light is blocked.

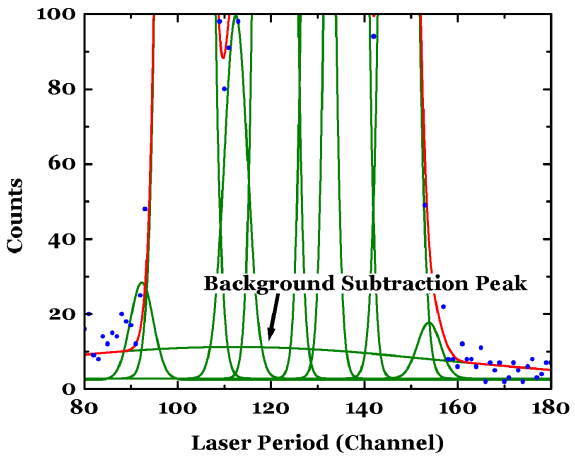

Figure 4.2: A typical background curve fitting result.

As mentioned previously (Sec. 3.2.1), $A \propto n\sigma$. Because the total number of atoms in the target is *assumed* to remain constant regardless of whether the Stokes laser is present or not, comparing the difference in areas when the Stokes is present *versus* absent yields the relative capture cross section:[141]

$$\frac{\sigma_k}{\sigma_s} = \Delta A_k \left(\sum_{i=1}^{k-1} \Delta A_i \frac{\sigma_s}{\sigma_i} \right)^{-1} . \tag{4.1}$$

where k is the excited index, either p or d in this work. Therefore, without any need for measuring the number of atoms in any of the related excited states, one can deduce the relative cross sections based on the areas under the Q-value capture channel peaks.

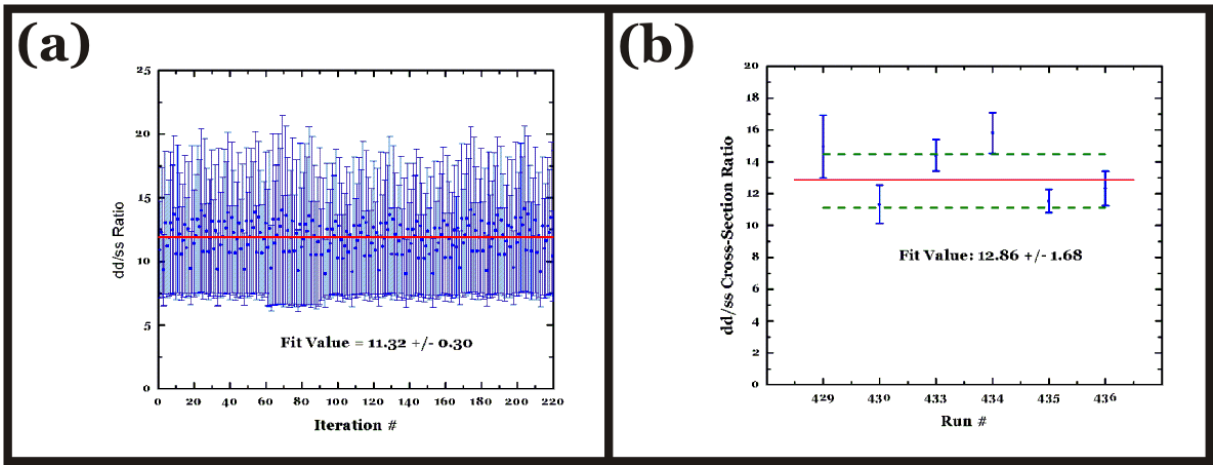

Figure 4.3: A sample σ_{dd}/σ_{ss} fit session. **(a)** several hundred cross section fits are plotted with calculated fit errors (shown in blue), along with the calculated average (shown as a red line). **(b)** The calculated averages for several data sets are plotted (shown in blue) along with the average value (shown as a red line) and the corresponding error bars for the final figure (shown as green dashed lines).

Subtle differences, however, make this a somewhat more difficult task than it has been for previous cases using this method. The background subtraction when measuring cross sections involving the $4d$ energy level is critical. Small changes in such a fit yield widely varying results because the $Rb(5s) \rightarrow Na(3s)$ peak changes very little when comparing the spectrum when the Stokes field is present to the spectrum when the Stokes field is absent. Figure 4.2 shows a typical Gaussian fit to the nonlinear background present under the capture channel peaks. The source of this background is as-yet unknown. Due to the nature of this fit, it is initially somewhat subjective as to what values seem to produce valid results.

An attempt at removing the subjectiveness was made by writing a short fitting program. This program selected random fit values within a realistic range for the background subtraction curve. The remaining peaks were fit automatically, and the final result was manually checked for validity. This automated process made it possible to attempt several hundred iterations of the fit routine, and measure the average fit result. The cross section results for a given data set were plotted, as shown in Fig. 4.3a, and this process was performed on each

of several data sets. The inherent error in such measurements is due to the extremely small change in peak size within the Rb(5s) → Na(3s) capture channel. The final value obtained is shown in Fig. 4.3b, and all relevant capture cross sections involving the terminal $4d$ state are shown in Table 4.1. Finally, the actual fitting routines, written for the Origin software package,[164] can be found in Appendix E.

4.1.2 Time Evolution Visualization: Population Dynamics

Figure 4.4 shows the laser control configuration used for data acquisition. The entire pulse sequence has a period of 5 μs, during which time the trapping lasers are present for all but 500 ns. During this time when the trapping field is not active, the Stokes and pump pulses were arranged in a variety of configurations, each of which will be described in this chapter.

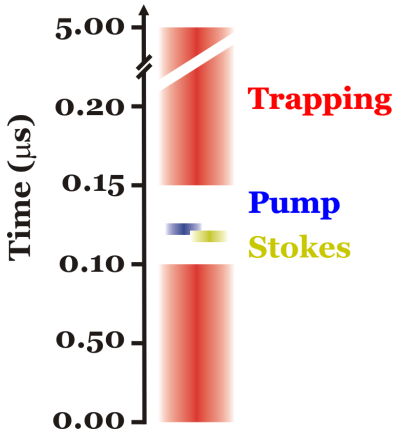

Figure 4.4: The laser configuration controlled by the AWG software for data acquisition.

Armed with all the cross section information necessary, a typical data acquisition session is analyzed in the following manner. While the trap lasers are on, the counts measured in the Rb($5p$)−Na($3p$) and Rb($5s$)−Na($3s$) capture channels from TAC_1 (see Fig. 4.5a) are summed together after first dividing the counts in those channels by the corrseponding cross section ratios. This results in a number proportional to the total number of atoms in the MOT. Knowing that TAC_1 has 10 ns/channel and TAC_2 has 2 ns/channel (as discussed in Sec. 4.3.1), the total number of atoms per time channel in TAC_1 can be converted into a value applied to the data measured from TAC_2.

For each time channel from TAC_2, counts from each of the relevant charge-transfer capture channels are measured (see Fig. 4.5b). Weighting each capture channel by the appropriate cross section ratios, and dividing them by the total MOT number obtained

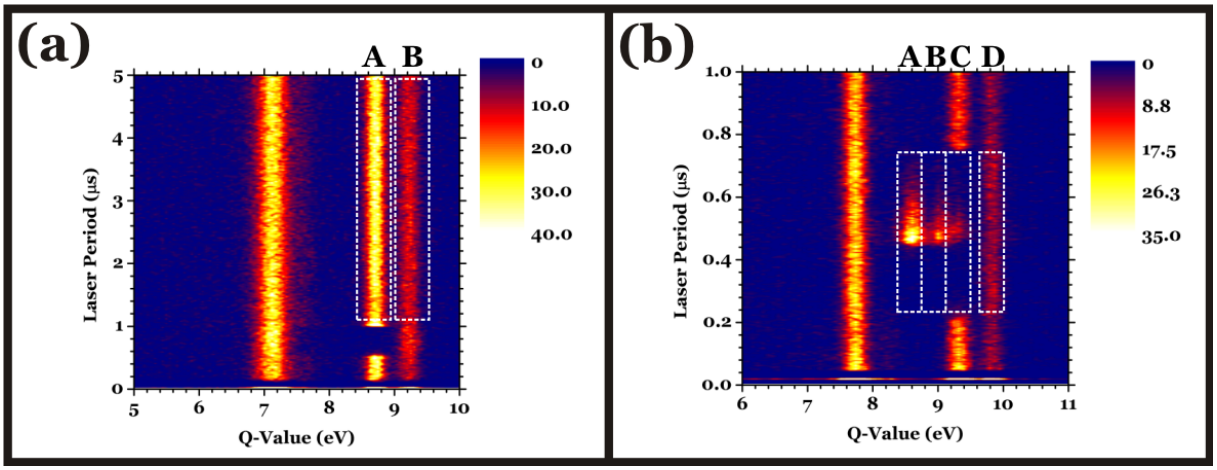

Figure 4.5: Population dynamics *versus* time. **(a)** TAC_1 shows the entire 5 μs period. **(b)** TAC_2 shows a high-resolution close-up of the first 2 μs.

from TAC_1, one can plot the fractional population in each excited level of the target as a function of time, normalized to the total MOT population.

Such a plot is shown in Fig. 4.6. The graph shows the target population divided into the $5s$ and $5p$ states prior to the trap laser field being turned off. After a time, the trapping laser field is turned off, and the population in the $5p$ intermediate state decays down to the $5s$ ground state. The decay rate can be measured from this graph to be 27.4 ns, consistent with the known decay lifetimes measured elsewhere.[119]

The Stokes laser is applied, followed by the pump laser in the counter-intuitive order typical of STIRAP excitation. Population can be seen moving directly from the $5s$ ground state directly into the $4d$ terminal state, as evidenced by a lack of population appearing in the intermediate $5p$ state. About 50 ns later, population begins to appear in the intermediate state, fed by natural decay from the upper $4d$ level. As was mentioned earlier (see Sec. 2.4.1), this is a hallmark indicator of the STIRAP process. When the pump laser field is turned off, population then decays naturally into the $5p$ and $5s$ states, and at a later time, the trapping field is turned back on.

Plots such as those shown in Fig. 4.6 are used throughout the remainder of this chapter to describe the internal dynamics taking place within the target. The total time scale shown

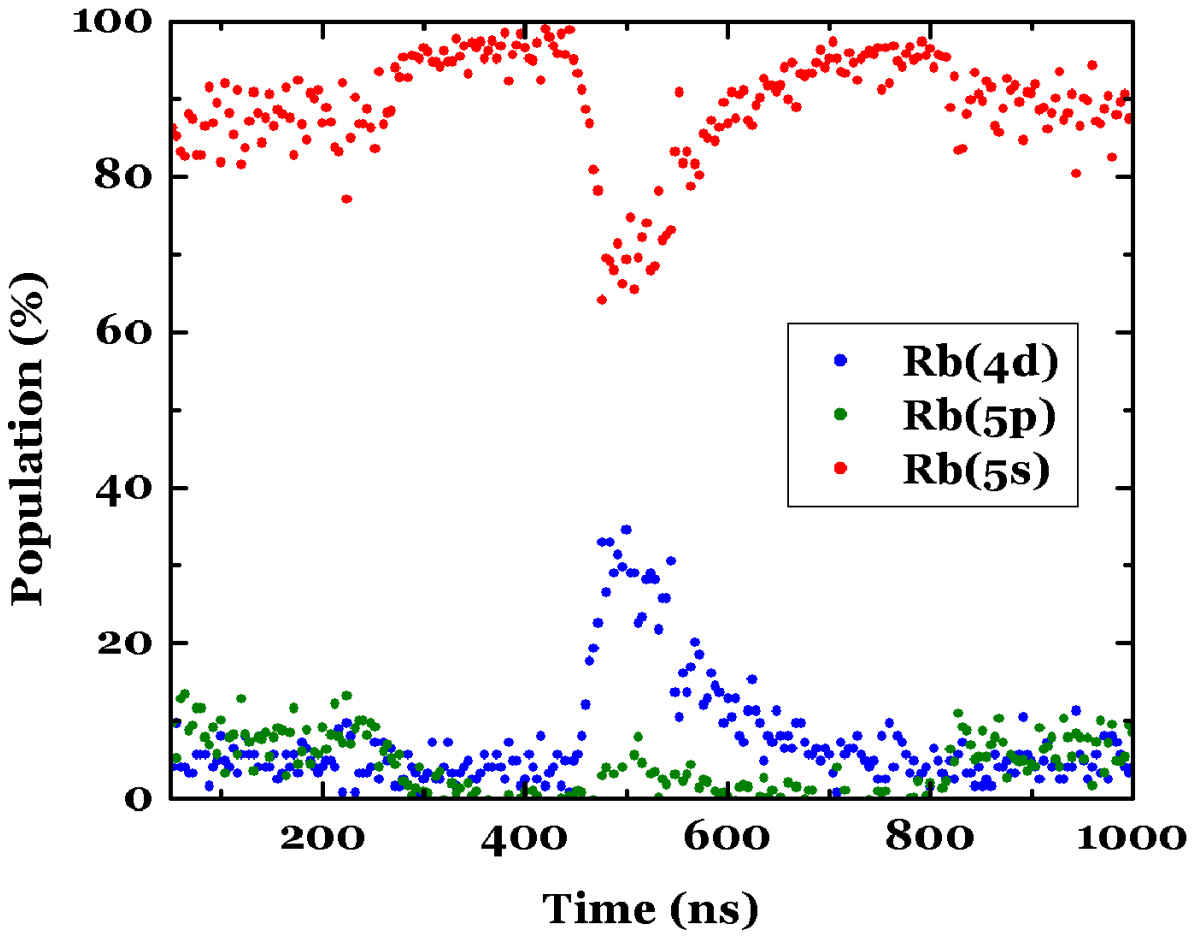

Figure 4.6: MOT population dynamics as a function of time. Specific parameter settings are listed elsewhere.[165]

in such figures corresponds to the high-resolution 2 ns/channel time scale from TAC_2, thus the entire 5 μs laser period is not shown. For the remainder of the time outside of that recorded by TAC_2, the trapping laser field is present to rejuvenate the target.

4.2 Transfer Characteristics

Various excitation regimes that depend on the initial parameters used in STIRAP were discussed earlier in Sec. 2.4. Some of these parameters were studied experimentally, and will be presented here. *The goal in presenting these data is to verify the validity of the technique for measuring population dynamics of a system on a nanosecond timescale, not to*

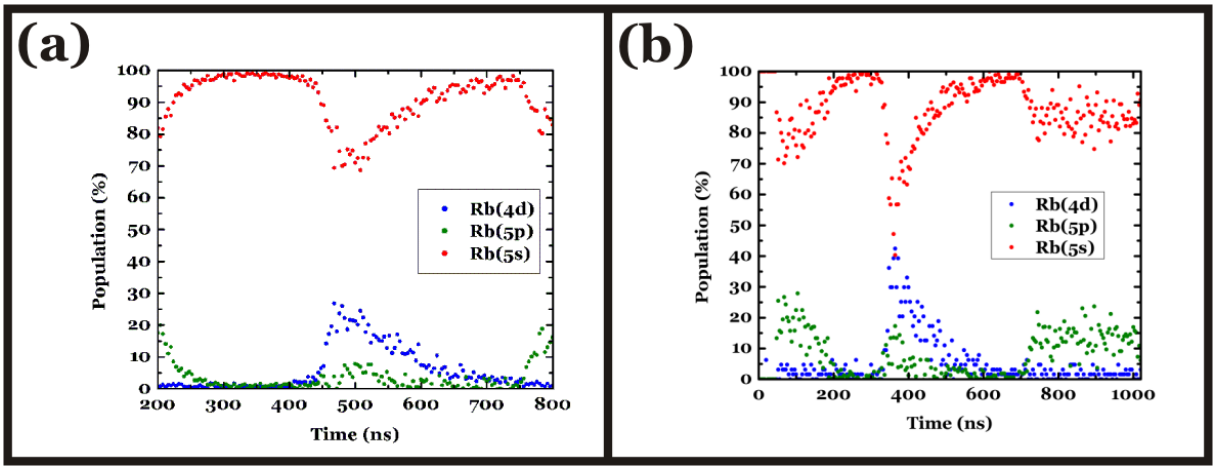

Figure 4.7: Population dynamics for the adiabatic and diabatic regimes are shown. **(a)** Adiabatic case. **(b)** Diabatic case. The parameters used for these plots are listed elsewhere.[166]

exhaustively search through the seven-dimensional parameter space.

4.2.1 Adiabatic *vs.* Diabatic Regimes

In order to illustrate the usefulness of the experimental technique presented in this dissertation, one can look at the adiabatic *versus* diabatic regimes in coherent excitation. As is discussed in Appendix 2.5.5, the clearest indicator of whether a system is in an adiabatic or diabatic regime is found in the comparison of the diagonal to off-diagonal matrix elements. In terms of experimental measurements, this is not feasable. Instead, one can qualitatively measure the adiabaticity of the system by measuring the time at which population is fed into the $5p$ intermediate state. Figure 4.7 shows two experimental cases yielding very different results. Fig. 4.7a presents the adiabatic regime, indicated by the time lag in population fed into the $5p$ state. The system undergoes STIRAP excitation, and then the population relaxes back into the ground state prior to the return of the trapping laser field. Fig. 4.7b, on the other hand, shows the diabatic regime, as population is simultaneously moved into the $5p$ and $4d$ states. Ground state population is directly placed into the intermediate state, indicative of non-adiabatic transfer.

An important distinction should be made by looking at these two plots which reaches to

the heart of this dissertation. One would expect that by measuring which parameters yielded the most population transfer into the $4d$ terminal state, one would know which configuration yielded true coherent excitation *via* STIRAP. However, this is clearly misleading in the examples shown. Figure 4.7a is the adiabatic regime, indicative of STIRAP transfer, yet the maximum population placed in the $4d$ state is only 40%. Fig. 4.7b shows a much higher $4d$ fractional population ($\sim 60\%$), yet it is clearly a diabatic regime.

If one's goal is to truly maximize population transfer, this is a critical distinction to be made. Most experimental methods discussed previously in Sec. 1.2 measure, at most, the initial and final states involved in the excitation and normalize the measurements to theory in order to determine the efficiency of population transfer.[114] Without measuring the temporal evolution of the intermediate state(s) of the system, one cannot definitively deduce that the system is indeed approaching the most optimal configuration for efficient population transfer. While the use of an optimization method (for example, a genetic algorithm) might be of benefit in scanning through the vast parameter space to find the optimal conditions, it is not clear that such a technique would produce results more expediently than utilizing the information gleaned from measuring the time evolution of the system.

4.2.2 Counterintuitive, Overlapping, and Intuitive Regimes

Another example of how this method of measuring population dynamics provides useful information can be seen by looking at three related parameter regimes, namely, the counterintuitive, overlap, and intuitive pulse delays. As shown previously in Fig. 2.8, the most critical experimental parameter to efficient coherent excitation *via* STIRAP is the pulse delay, τ. If $\tau < 0$, the system is in the counterintuitive pulse order (see Fig. 1.2), where the Stokes pulse preceeds the pump pulse. It is in this configuration that the most efficient population transfer is likely to occur. Figure 4.8 shows such a counterintuitive pulse delay. In this case, $\tau = -30$ ns. The maximum population transfer into the $4d$ terminal state is 60%, indicating relatively efficient transfer characteristics for experiments performed

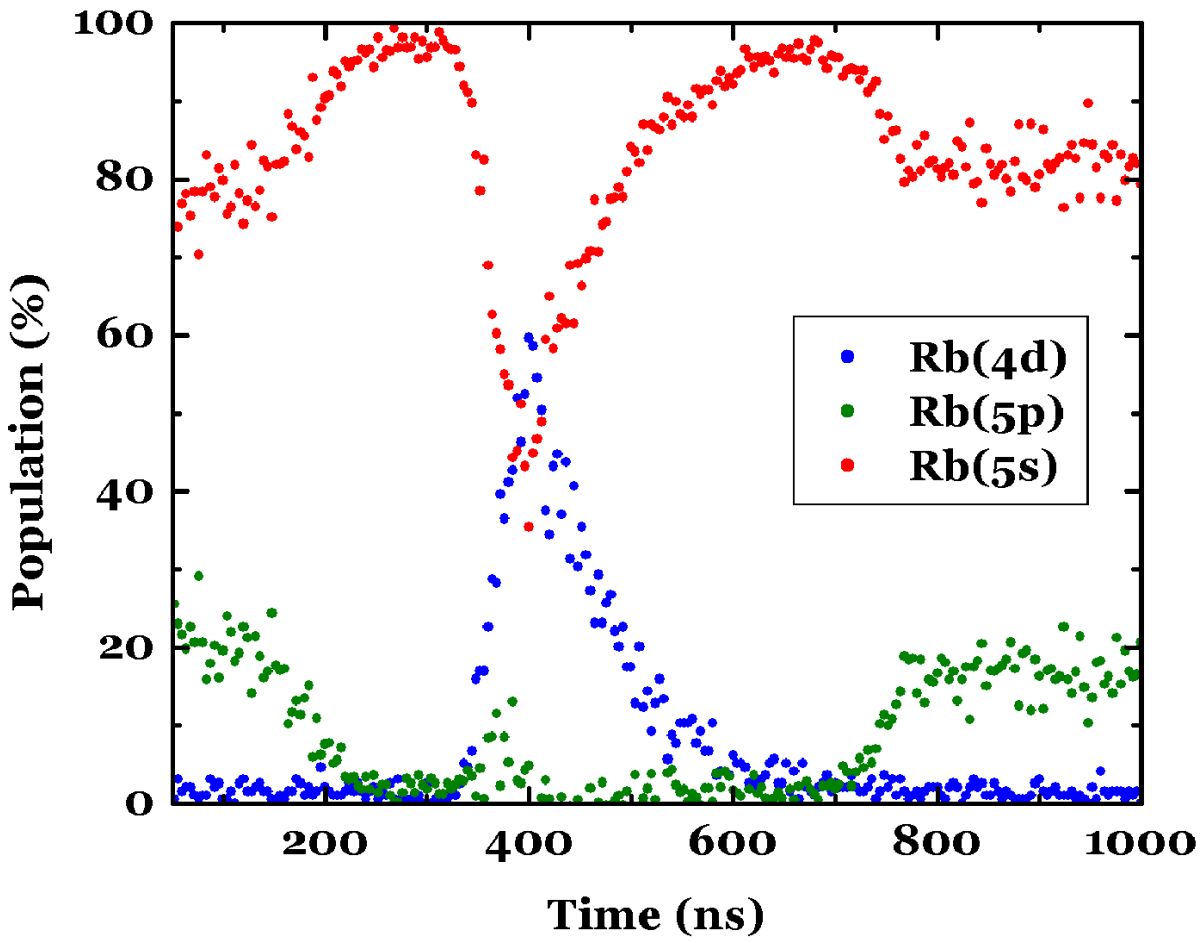

Figure 4.8: Population dynamics for the counterintuitive regime is shown. Here, $\tau = -50$ ns.

to-date. Some indication of adiabaticity is present in these data, hence it is possible this configuration could yield an even greater population transfer efficiency if it were moved into an indisputably adiabatic state.

Comparatively, Fig. 4.9 shows the system in an intuitive regime, where $\tau = +50$ ns. The maximum fractional population placed in the $4d$ state is only about 20%. This is clearly less efficient than the counterintuitive configuration, and it is consistent with the theoretical exploration shown earlier in Sec. 2.5.1.

Finally, an intermediate "overlap" regime is depicted in Fig. 4.10, where $\tau = 0$ ns. Here, population transfer is somewhat efficient, yielding more fractional population in the $4d$ state than the intuitive order, yet not as efficient as can be seen in the counterintuitive case. Here,

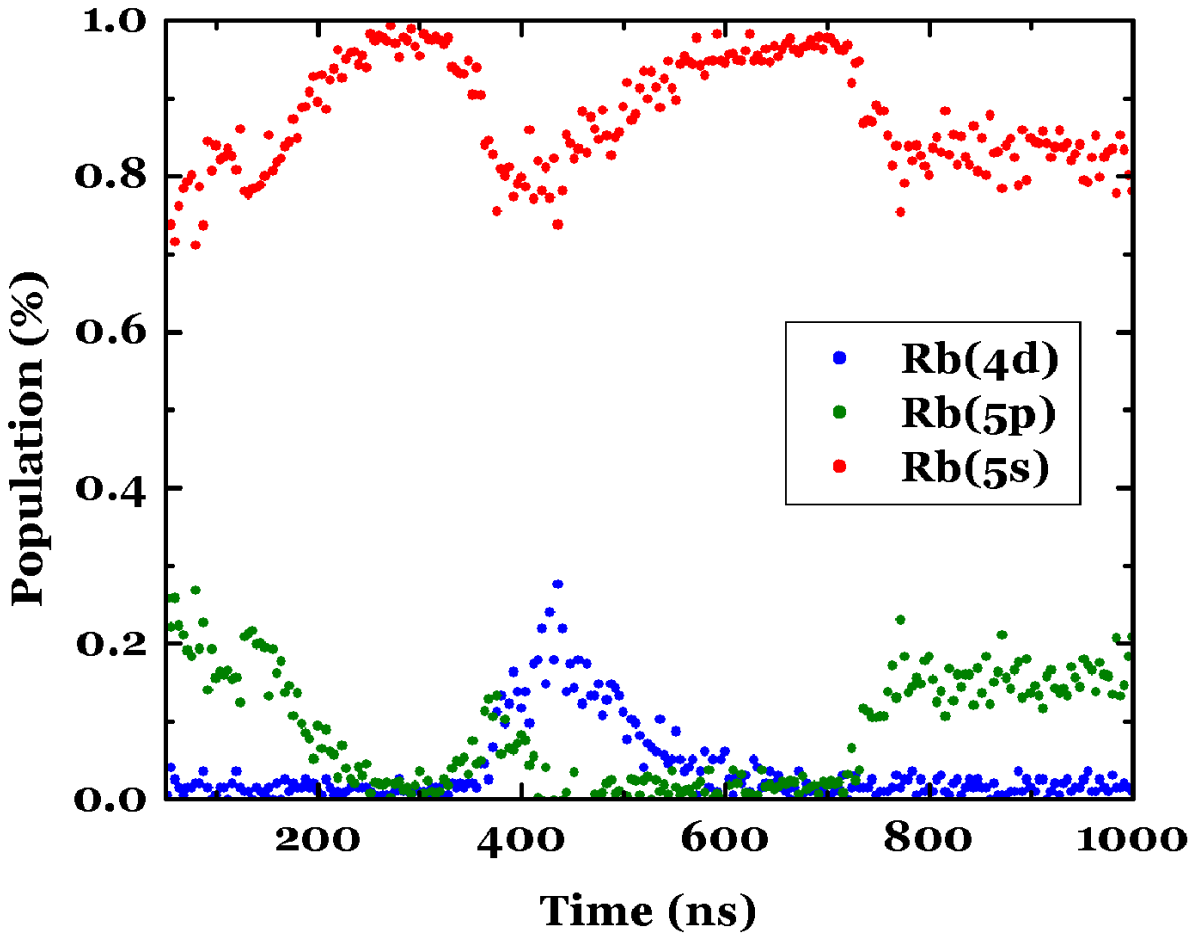

Figure 4.9: Population dynamics for the intuitive regime is shown. The parameters used to collect these data were the same as in Fig. 4.8, except, in this case, $\tau = +50$ ns.

however, the system appears to be in a diabatic state, as indicated by the $5p$ population being fed at the same time as the $4d$ population. This probably indicates that parameters other than τ were not correctly controlled.

Such depictions of population transfer greatly increase one's understanding of what is taking place during the excitation process. Without such knowledge, it is much more difficult to assert that the system is, for instance, in a diabatic or adiabatic regime, or that a counterintuitive pulse order has indeed yielded a maximally efficient population transfer. The dynamics of such interactions are complex, and the need to measure the complete system's population on a short timescale is vital to understanding how to efficiently control

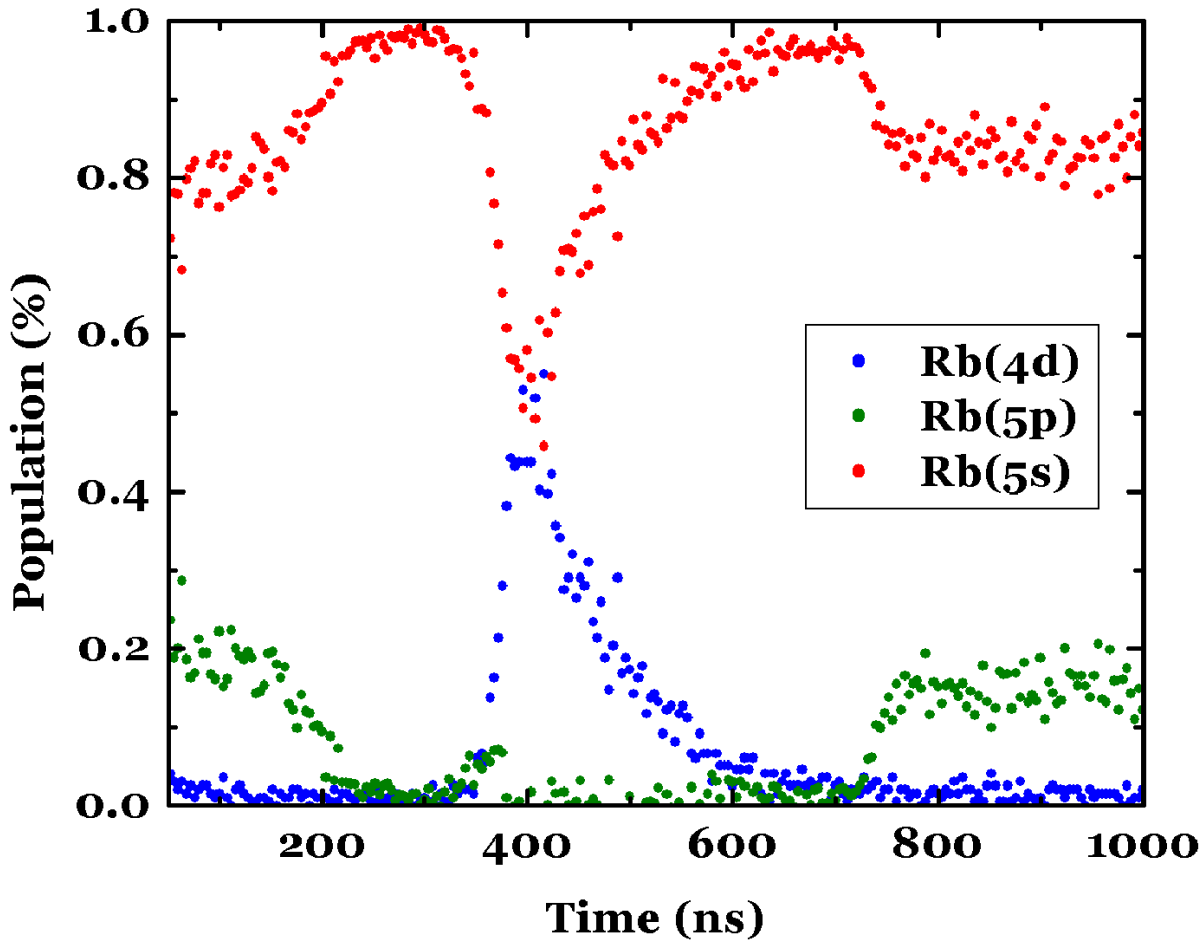

Figure 4.10: Population dynamics for the overlap regime is shown. Here, $\tau = 0$ ns, and the remaining parameters are the same as in Fig. 4.8.

the population.

4.3 Resolution and Error Analysis

The resolution of the system is dependant on two major factors: the time resolution of the TAC used for taking measurements correlating the laser period to the collision events, and the time resolution of the TDC used to measure the Q-value spectrum. The error bars for these data are dominated by the cross section measurement error bars, most notably the σ_{ds}/σ_{ss} cross section, for reasons described in Sec. 4.1.1. Both the system resolution and the measurement error will be outlined in this section.

4.3.1 System Resolution

The system studied in this dissertation, namely $Na^+ \rightarrow Rb$, is by no means the only system possible. As mentioned previously in Sec. 3.1, any trappable species can be used as a target, and any ion source can be used. Indeed, more elaborate MOTRIMS schemes, where projectiles are not produced solely by a thermionic source, but instead produced by other means, such as an electron beam ion source (EBIS), widen the possible projectile species even further.

The limiting factor, however, lies in the resolution of the resultant Q-value, or more specifically, whether the charge-transfer channels are resolvable. The TDC together with the collision system used in these experiments have a time-of-flight resolution of 2 ns/channel, which is sufficient for resolving the three-level case. The width of the Q-value peaks obtained using this system are consistent with a slight spread in the projectile energy, resulting in a Q-value resolution of 0.03 a.u. For the three-level case studied herein, this is not an issue, however for future multilevel cases of interest, this limiting resolution can be an issue, as discussed in Sec. 5.

The time resolution of the measurements described in the previous section is limited by the resolution of TAC_2. This TAC was set for a period of 5 μs, corresponding to the laser period of 5 μs provided by the AWG control software. Figure 4.11 shows the callibration curve for TAC_2, where the experimentally measured resolution is 2.1 ns/channel. In principle, this resolution could be improved by simply using a finer TAC scale. The ADC used to measure the TAC_2 output has a voltage range of 0 to +2 V. One would need to use multiple TACs for such an improved resolution, however, in order to cover the entire 500 ns trap-off region where the coherent population dynamics occur. Thus, in theory, one could improve the measurement to the electronic resolution limits of the TAC, which is at least a factor of 10 greater than currently used. It is therefore conceivable that, using this improved method of measuring population dynamics, one can be able to directly measure Rabi flops, as noted in Sec. 5.2.1.

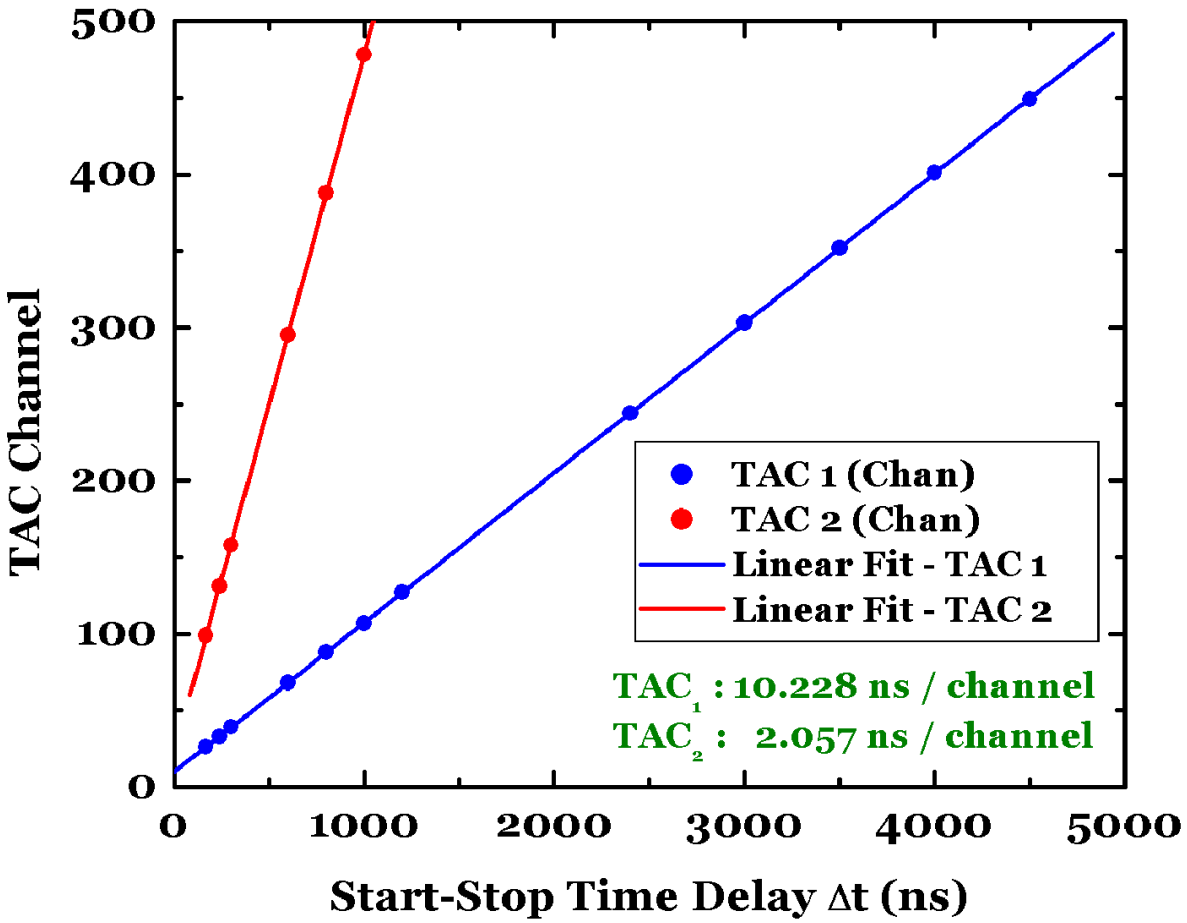

Figure 4.11: Plot showing TAC_1 (blue) and TAC_2 (red) callibration.

4.3.2 Measurement Error

As shown in Table 4.1, the σ_{ds}/σ_{ss} cross section error bars are large, compared to the other cross section measurements used in calculating the excited state fraction of the target. Figure 4.12 shows a sample population *versus* time plot with error bars included on a single point to show the absolute error for these data. In this graph, the error in $5s$ is 2.5%, while $5p$ is 1.6%, and $4d$ is 2.4%. This absolute error is directly linked to the uncertainty of the relative cross section measurements. With a better measured value for the σ_{ds}/σ_{ss} cross section, this error would be reduced. The relative error of these data is best described by the scatter in the data themselves.

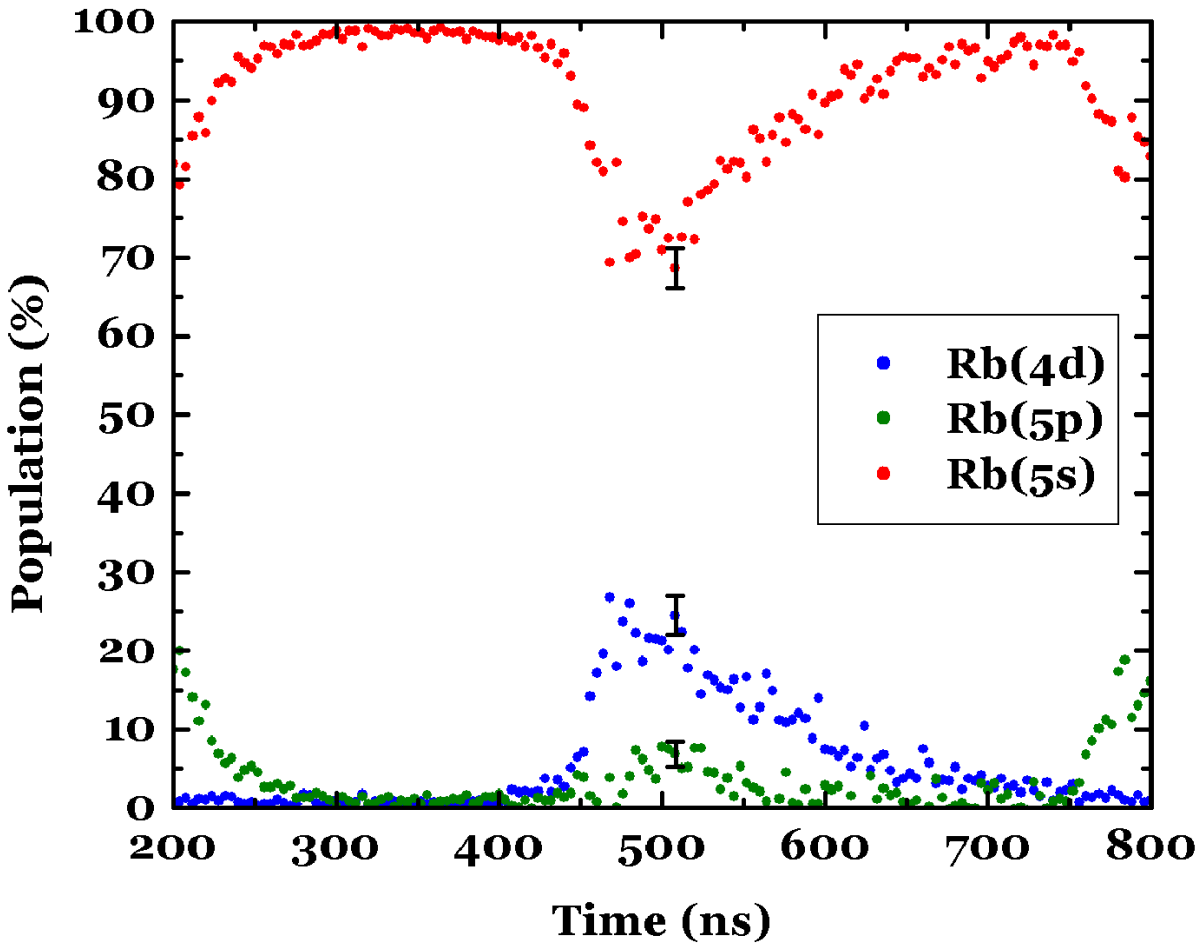

Figure 4.12: Typical error bars for a population *versus* time measurement.

Sources of systematic error include the assumption that the ion beam is sampling the same volume of MOT target that the combination of pump and Stokes lasers are exciting. The general practice is to align the ion beam to produce as much current downstream on the Faraday cup as possible, then adjust the MOT target's position until a maximum count rate is obtained. This ensures the best overlap between the projectile ions and the stationary target. The ion beam has a diameter of ~ 1 mm, while the target is typically $\sim 0.5 - 1.0$ mm. The pump and Stokes lasers are typically ~ 1.0 mm in diameter. Figure 3.13, for example, shows average beam diameters of $w_p = 1.42$ mm and $w_s = 0.98$ mm. The combined pump and Stokes beam is aligned by maximizing production of dimer ions

(formed through associative ionization involving Rb($4d$)) detected on the downstream recoil detector when the projectile ion beam is blocked. Given the comparable beam sizes and the independent maximization of target overlap, the error introduced from improper beam overlap is minimized.

The only time such overlap issues are of great concern is when the MOT target is much larger than the pump and Stokes beams. It is possible to sample too many atoms in the $5s$ ground state that never interact with the excitation lasers, thus weighing the ground state population incorrectly. Such an improper configuration is depicted in Fig. 4.13, and can be detected experimentally by measuring the beam size at the focal point, as well as the MOT size using standard CCD camera measurements. The solution to such a problem is either to reduce the size of the target (which, in practice, is difficult), or increase the pump and Stokes beam diameters, thus reducing their intensities and consequently decreasing the Rabi frequencies.

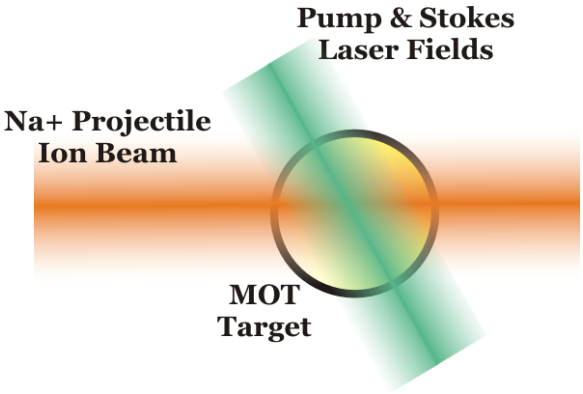

Figure 4.13: A conceptual diagram showing a poor overlap configuration in the system.

Chapter 5

Conclusion and Outlook

This chapter provides a brief conclusion to the work presented in this dissertation, as well as a discussion of the outlook for future research in this area. The work completed thus far is by no means comprehensive. There is plenty of work remaining in order to better understand the population dynamics involved in coherent processes. After a brief conclusion, some current and possible future projects will be discussed, along with preliminary work in these areas.

5.1 Conclusion

In conclusion, the goal of this dissertation was to present a new technique allowing one to measure population dynamics occurring in a system undergoing coherent population transfer. The technique consists of combining well-proven methods of spectroscopy to cold targets (MOTRIMS) with widely-used methods of coherent excitation (STIRAP). The MOTRIMS technique used here for probing the MOT target yields a high-resolution measurement of each energy state involved in the collision process, and provides a dynamic, noninvasive method of measuring fractional populations in all resolvable capture channels.

While the results of this dissertation were not designed to show the robustness of STIRAP, or provide highly efficient population transfer, they were designed to highlight the ability of the measurement technique to provide a more comprehensive picture of dynamics taking place during coherent population transfer. While other measurement methods indi-

cate the efficiency of population transfer equally well, the ability to measure the system more completely provides one with a better understanding of system characteristics. This provides insight as to which parameters should be manipulated to improve population transfer efficiency further, or, equally important, which parameters will *not* improve transfer efficiency. Such techniques were not only presented, but the theoretical adiabatic and diabatic predictions of populations undergoing STIRAP were verified by direct measurement.

5.2 Outlook: Beyond 3-Level STIRAP

The data presented here are only a beginning. With the technique shown to be valid, there remains the exploration of the multidimensional parameter space discussed earlier in Sec. 2.5. Such a study would provide insight to the characteristics of adiabatic and diabatic limits as applied to certain parameters. For example, it would be of benefit to experimentally map out the adiabaticity of a three-level excitation system as a function of Stokes or pump laser field intensities. Further, the two-photon resonance conditions imposed by STIRAP could be explored in further detail, measuring efficiency as a function of diabaticity.

5.2.1 Direct Measurement of Rabi Flopping

The resolution of the system presented in this work is borderline for directly measuring Rabi flops taking place within a system. If the time-resolution (that is, the resolution provided by TAC_2) were to be improved by a factor of two, which is certainly an achievable goal as discussed earlier in Sec. 4.3.1, then it should be feasable to directly measure such flopping behavior in the coherently excited population. Figure 5.1 shows a theoretical example of Rabi flopping, showing the population being driven back and forth from the $5s$ ground state to the $4d$ terminal state at the two-photon Rabi frequency.

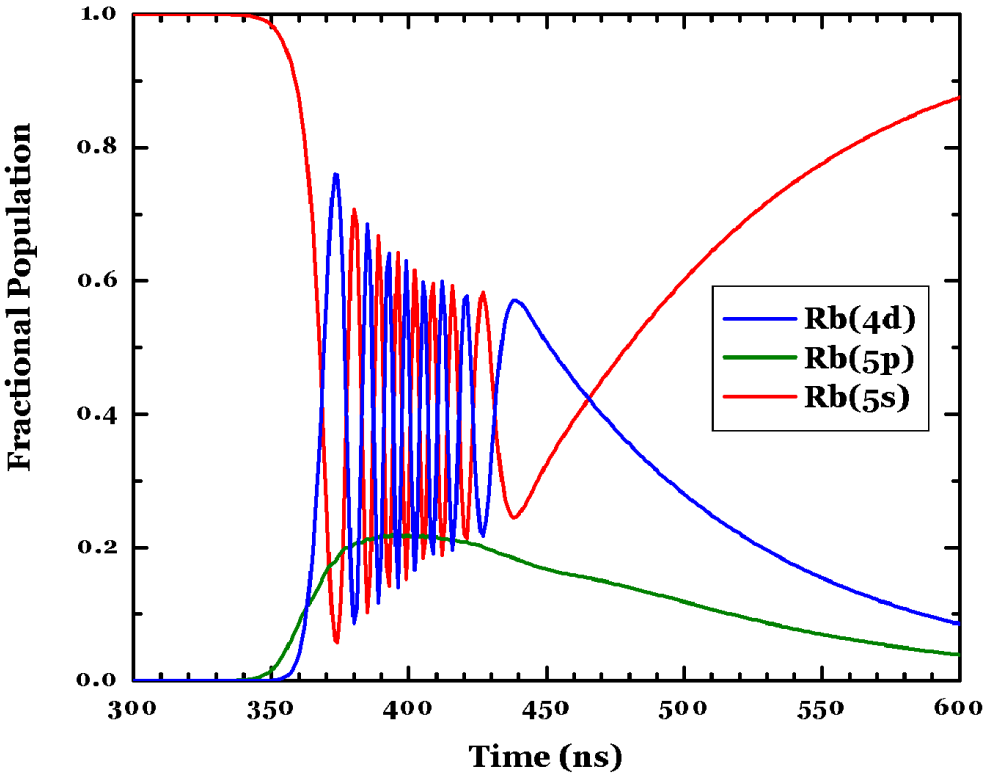

Figure 5.1: Plot showing calculated two-photon Rabi flops within a three-level system.

5.2.2 4-Level STIRAP — Rydberg Studies

Now that the population dynamics for a three-level coherently excited system have begun to be studied, it would be useful to the quantum information community to perform a study of the dynamics involved in efficiently preparing Rydberg states. Preliminary efforts have shown it is possible to use a four-level system based on the three-level structure studied here to produce highly-excited Rydberg targets.

Figure 5.2 shows some of the earliest measurements of the feasability of such a four-level system. In Fig. 5.2a, a Q-value spectrum is shown, where the additional capture channel $Rb(9f) \rightarrow Na(nl)$ is shown. Figure 5.2b shows these same data plotted *versus* laser period. The point labeled "A" shows the Rydberg channel. It is interesting to note that

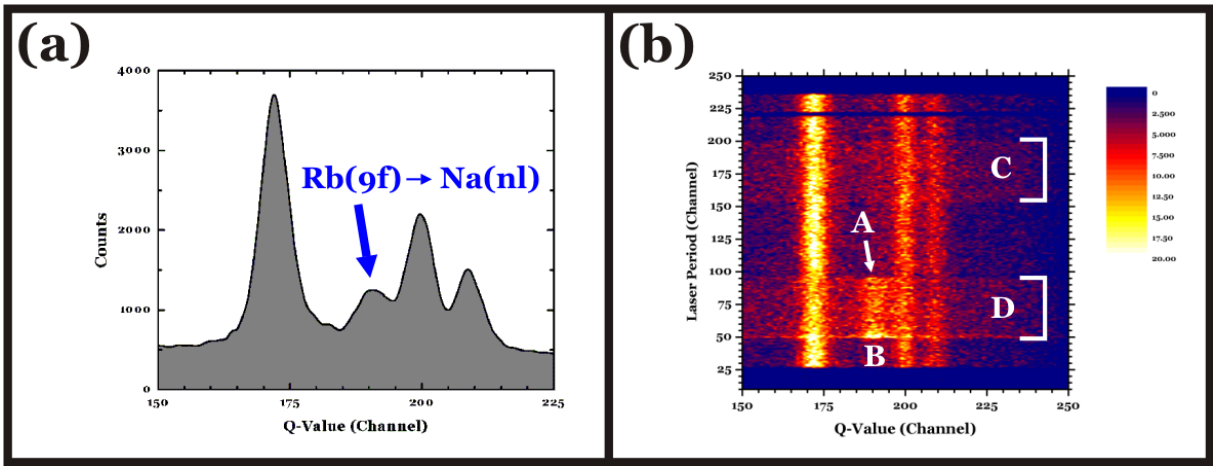

Figure 5.2: Plot showing charge transfer from a 4-level excited target. **(a)** The resultant Q-value spectrum for a four-level Rydberg system. **(b)** TAC plot for these same data. The points labeled with letters are discussed in the text.

point "B" seems to indicate possible 4-level STIRAP is present, due to the large amount of population placed into the Rydberg state. Points labeled "C" and "D" indicate bands of counts present due to dimer and monomer ion production, respectively. Immediately, the question of resolution arises. It is unclear whether the specific case of $Na^+ \rightarrow Rb$ is the best system to study such excitation, however a variety of other schemes have been proposed (for example, $H^+ \rightarrow Rb$), and the issue of poor resolution is not a limiting factor in such an experimental endeavor. To date, no one has experimentally studied a four-level, coherently excited Rydberg system. This can potentially provide vital insights into the efficient production of Rydberg targets *via* STIRAP.

5.2.3 Associative and Penning Ionization

The techniques presented here are not only limited to the study of coherent excitation methods. A variety of additional experiments can also be of benefit. For example, during the course of the work presented in this dissertation, it was discovered that one can also study associative and Penning ionization, that is, the production of monomer ($Rb(4d) + Rb(4d) \rightarrow Rb^+ + e^- + Rb(5s)$) and dimer ($Rb(4d) + Rb(4d) \rightarrow Rb_2^+ + e^-$ and $Rb(4d) + Rb(4p) \rightarrow Rb_2^+ + e^-$) ions can be studied. Figure 5.3 shows some preliminary time-of-flight measurements showing

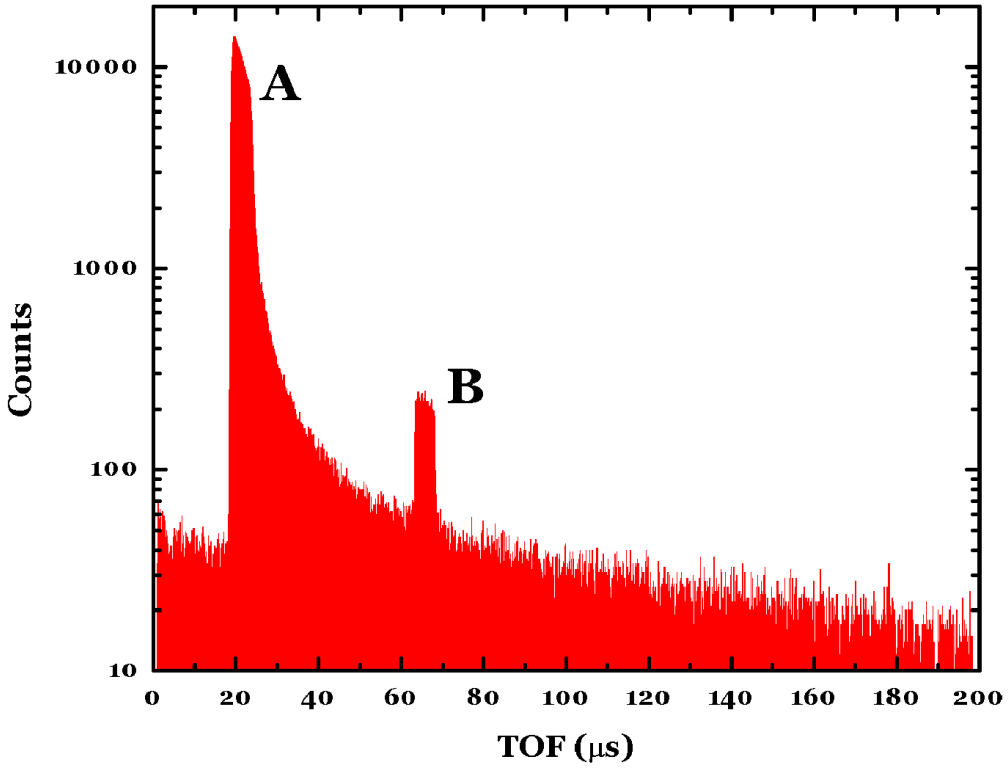

Figure 5.3: Time-of-flight plot showing the production of monomer and dimer molecules. The peak labeled "A" indicates the presence of dimers, while the peak labeled "B" indicates the presence of monomers.

the creation of monomer and dimer ions. Currently, the largest difficulty in exploring the molecular dynamics of such systems is the lack of information on the relative cross sections. Using only time-of-flight, the two dimer channels are not distinguishable from one another. Various schemes, however, are currently being devised that would allow one to measure the cross sections, and thus the measurements made *via* the population dynamics methods of MOTRIMS would allow one to study the collisional processes present in such systems.

These are just a few examples of the possible experimental research areas opened up by the techniques presented within this dissertation. As the experimental techniques become more robust, the experimental possibilities will also broaden. In effect, the research pre-

sented here opens a door to a rich environment of future experimental work in population dynamics and the understanding of coherent excitation.

Bibliography

[1] S. Chu, Rev. Mod. Phys. **70**, 685 (1998).

[2] C. Cohen-Tannoudji, Rev. Mod. Phys. **70**, 707 (1998).

[3] W. D. Phillips, Rev. Mod. Phys. **70**, 721 (1998).

[4] W. D. Phillips, J. V. Prodan, and H. J. Metcalf, J. Opt. Soc. Am. B **2**, 1751 (1985).

[5] J. Weiner, V. S. Bagnato, S. Zilio, and P. S. Julienne, Rev. Mod. Phys. **71**, 1 (1999).

[6] J. Dalibard and C. Cohen-Tannoudji, J. Opt. Soc. Am. B **2**, 1707 (1985).

[7] P. L. Gould, P. D. Lett, P. S. Julienne, and W. D. Phillips, Phys. Rev. Lett. **60**, 788 (1988).

[8] P. S. Julienne, Phys. Rev. Lett. **61**, 698 (1988).

[9] P. S. Julienne and R. Heather, Phys. Rev. Lett. **67**, 2135 (1991).

[10] P. D. Lett et al., Phys. Rev. Lett. **67**, 2139 (1991).

[11] G. K. Brennen, C. M. Caves, P. S. Jessen, and I. H. Deutsch, Phys. Rev. Lett. **82**, 1060 (1999).

[12] A. Hemmerich, Phys. Rev. A **60**, 943 (1999).

[13] L. Tian and P. Zoller, Phys. Rev. A **68**, 042321 (2003).

[14] B. W. Shore and M. A. Johnson, J. Chem. Phys. **68**, 5631 (1978).

[15] T. Brandes, F. Renzoni, and R. H. Blick, Phys. Rev. B **64**, 035319 (2001).

[16] A. D. Greentree, J. H. Cole, A. R. Hamilton, and L. C. L. Hollenberg, Phys. Rev. B **70**, 235317 (2004).

[17] I. Roos and K. Mølmer, Phys. Rev. A **69**, 022321 (2004).

[18] R. B. Kurzel and J. I. Steinfeld, J. Chem. Phys. **53**, 3293 (1970).

[19] A. D. Bandrauk and E. S. Sedik, J. Chem. Phys. **121**, 7764 (2004).

[20] J. S. Melinger, A. Hariharan, S. R. Gandhi, and W. S. Warren, J. Chem. Phys. **95**, 2210 (1991).

[21] M. M. T. Loy, Phys. Rev. Lett. **32**, 814 (1974).

[22] M. M. T. Loy, Phys. Rev. Lett. **41**, 473 (1978).

[23] U. Gaubatz, P. Rudecki, S. Schiemann, and K. Bergmann, J. Chem. Phys. **92**, 5363 (1990).

[24] Y. Ohta, K. Hoki, and Y. Fujimura, J. Chem. Phys. **116**, 7509 (2002).

[25] Z. Kis, A. Karpati, B. W. Shore, and N. V. Vitanov, Phys. Rev. A **70**, 053405 (2004).

[26] H. M. Gibbs and G. G. Churchill, Phys. Rev. A **3**, 1617 (1971).

[27] M. Fuchs and J. P. Toennies, J. Chem. Phys. **85**, 7062 (1986).

[28] H. G. Rubahn and J. P. Toennies, J. Chem. Phys. **89**, 287 (1988).

[29] M. Külz, A. Kortyna, M. Keil, B. Schellhaaß, and K. Bergmann, J. Chem. Phys. **89**, 287 (1988).

[30] M. Külz et al., Phys. Rev. A **53**, 3324 (1996).

[31] C. Kittrell et al., J. Chem. Phys. **75**, 2056 (1981).

[32] W. D. Lawrance and A. E. W. Knight, J. Chem. Phys. **77**, 570 (1982).

[33] D. E. Reisner et al., J. Chem. Phys. **77**, 573 (1982).

[34] J. P. Pique, F. Stoeckel, and A. Campargue, Appl. Opt. **26**, 3103 (1987).

[35] E. Abramson, R. W. Field, D. Imre, K. K. Innes, and J. L. Kinsey, J. Chem. Phys. **83**, 453 (1985).

[36] S. H. Kable, J. J. W. Thoman, and A. E. W. Knight, J. Chem. Phys. **88**, 4748 (1988).

[37] Y. S. Choi, P. Teal, and C. B. Moore, J. Opt. Soc. Am. B **7**, 1829 (1990).

[38] G. Stock and W. Domcke, Phys. Rev. A **45**, 3032 (1992).

[39] J. A. Syage and J. E. Wessel, Appl. Opt. **26**, 3573 (1987).

[40] R. Wester, A. V. Davis, A. E. Bragg, and D. M. Neumark, Phys. Rev. A **65**, 051201R (2002).

[41] S. A. Kovalenko and J. Ruthmann, J. Chem. Phys. **109**, 1894 (1998).

[42] S. A. Reid, H. L. Kim, and J. D. McDonald, J. Chem. Phys. **92**, 7079 (1990).

[43] D. E. Reisner, R. W. Field, J. L. Kinsey, and H. L. Dai, J. Chem. Phys. **80**, 5968 (1984).

[44] B. R. Foy, M. P. Casassa, J. C. Stephenson, and D. S. King, J. Chem. Phys. **90**, 7037 (1989).

[45] K. J. Rensberger, J. B. Jeffries, and D. R. Crosley, J. Chem. Phys. **90**, 2174 (1988).

[46] M. P. Schmid, P. Maroni, R. D. Beck, and T. R. Rizzo, Rev. Sci. Instrum. **74**, 4110 (2003).

[47] R. L. Swofford, M. E. Long, and A. C. Albrecht, J. Chem. Phys. **65**, 179 (1976).

[48] J. R. Fair, O. Votava, and D. J. Nesbitt, J. Chem. Phys. **108**, 72 (1997).

[49] E. A. Rohlfing, J. Gelfand, and R. B. Miles, J. Chem. Phys. **75**, 4893 (1981).

[50] E. A. Rohlfing, J. Gelfand, and R. B. Miles, J. Appl. Phys. **53**, 5240 (1982).

[51] H. C. Miller, J. McCord, and G. D. Hager, J. Appl. Phys. **84**, 3467 (1988).

[52] A. C. Tam, Phys. Rev. A **19**, 1971 (1979).

[53] B. J. Herman and J. H. Eberly, Phys. Rev. A **39**, 3447 (1989).

[54] C. Warner and B. Bobbs, J. Opt. Soc. Am. B **3**, 1345 (1986).

[55] P. R. Hemmer and M. G. Prentiss, J. Opt. Soc. Am. B **5**, 1613 (1988).

[56] W. B. Bosma and S. Mukamel, Phys. Rev. Lett. **68**, 2456 (1992).

[57] J. Wu and H. Zeng, Opt. Lett. **28**, 1052 (2003).

[58] Y. Yoshikawa, T. Sugiura, Y. Torii, and T. Kuga, Phys. Rev. A **69**, 041603R (2004).

[59] D. Grischkowsky, Phys. Rev. A **7**, 2096 (1973).

[60] B. W. Shore, K. Bergmann, A. Kuhn, S. Schiemann, and J. Oreg, Phys. Rev. A **45**, 5297 (1992).

[61] J. S. Melinger, S. R. Gandhi, A. Hariharan, J. X. Tull, and W. S. Warren, Phys. Rev. Lett. **68**, 2000 (1992).

[62] B. Broers, H. B. van Linden van den Heuvell, and L. D. Noordam, Phys. Rev. Lett. **69**, 2062 (1992).

[63] S. Chelkowski, A. D. Bandrauk, and P. B. Corkum, Phys. Rev. Lett. **65**, 2355 (1990).

[64] A. G. Adam, T. E. Gough, N. R. Isenor, and G. Scoles, Phys. Rev. A **32**, 1451 (1985).

[65] A. G. Adam, T. E. Gough, N. R. Isenor, G. Scoles, and J. Shelley, Phys. Rev. A **34**, 4803 (1986).

[66] G. L. Lamb, Jr., Phys. Rev. A **12**, 2052 (1975).

[67] J. Oreg and G. Hazak, Phys. Rev. A **32**, 2776 (1985).

[68] U. Boscain, G. Charlot, J. Gauthier, S. Guérin, and H. Jauslin, J. Math. Phys. **43**, 2107 (2002).

[69] M. Holthaus, Phys. Rev. A **49**, 1950 (1994).

[70] U. Gaubatz, P. Rudecki, S. Schiemann, M. Külz, and K. Bergmann, Chem. Phys. Lett. **149**, 463 (1988).

[71] K. Bergmann, H. Theuer, and B. W. Shore, Rev. Mod. Phys. **70**, 1003 (1998).

[72] N. V. Vitanov, M. Fleischhauer, B. W. Shore, and K. Bergmann, Adv. At., Mol., Opt. Phys. **46**, 1 (2001).

[73] N. V. Vitanov, T. Halfmann, B. W. Shore, and K. Bergmann, Annu. Rev. Phys. Chem. **52**, 763 (2001).

[74] D. Felinto, L. H. Acioli, and S. S. Vianna, Phys. Rev. A **70**, 043403 (2004).

[75] H. Friedmann and A. D. Wilson, Appl. Phys. Lett. **28**, 270 (1976).

[76] J. Cao, C. J. Bardeen, and K. R. Wilson, J. Chem. Phys. **113**, 1898 (2000).

[77] G. N. Gibson, Phys. Rev. Lett. **89**, 263001 (2002).

[78] G. N. Gibson, Phys. Rev. A **67**, 043401 (2003).

[79] N. Dudovich, D. Oron, and Y. Silberberg, Phys. Rev. Lett. **88**, 123004 (2002).

[80] J. E. Rothenberg, D. Grischkowsky, and A. C. Balant, Phys. Rev. Lett. **53**, 552 (1984).

[81] J. Oreg and G. Hazak, J. Opt. Soc. Am. B **3**, 488 (1986).

[82] B. W. Shore, K. Bergmann, J. Oreg, and S. Rosenwaks, Phys. Rev. A **44**, 7442 (1991).

[83] Y. B. Band, Phys. Rev. A **50**, 5046 (1994).

[84] Y. B. Band and O. Magnes, Phys. Rev. A **50**, 584 (1994).

[85] V. S. Malinovsky and D. J. Tannor, Phys. Rev. A **56**, 4929 (1997).

[86] I. R. Solá, V. S. Malinovsky, and D. J. Tannor, Phys. Rev. A **60**, 3081 (1999).

[87] P. Marte, P. Zoller, and J. L. Hall, Phys. Rev. A **44**, R4118 (1991).

[88] T. Wang, M. Kostrun, and S. F. Yelin, Phys. Rev. A **70**, R4118 (1991).

[89] N. V. Vitanov, K. A. Suominen, and B. W. Shore, J. Phys. B **32**, 4535 (1999).

[90] K. J. Boller, A. Imamoğlu, and S. E. Harris, Phys. Rev. Lett. **66**, 2593 (1991).

[91] L. P. Yatsenko et al., Phys. Rev. A **58**, 4683 (1998).

[92] S. Guérin, L. P. Yatsenko, T. Halfmann, B. W. Shore, and K. Bergmann, Phys. Rev. A **58**, 4691 (1998).

[93] K. Böhmer, T. Halfmann, L. P. Yatsenko, B. W. Shore, and K. Bergmann, Phys. Rev. A **64**, 023404 (2001).

[94] A. M. Ishkhanyan, Phys. Rev. A **61**, 063609 (2000).

[95] J. R. Kuklinski, U. Gaubatz, F. T. Hioe, and K. Bergmann, Phys. Rev. A **40**, 6741 (1989).

[96] T. A. Laine and S. Stenholm, Phys. Rev. A **53**, 2501 (1996).

[97] J. Oreg, F. T. Hioe, and J. H. Eberly, Phys. Rev. A **29**, 690 (1984).

[98] G. W. Coulston and K. Bergmann, J. Chem. Phys. **96**, 3467 (1991).

[99] J. Oreg, K. Bergmann, B. W. Shore, and S. Rosenwaks, Phys. Rev. A **45**, 4888 (1992).

[100] J. Martin, B. W. Shore, and K. Bergmann, Phys. Rev. A **52**, 583 (1995).

[101] B. W. Shore, J. Martin, M. P. Fewell, and K. Bergmann, Phys. Rev. A **52**, 566 (1995).

[102] M. Shapiro, Phys. Rev. A **54**, 1504 (1996).

[103] N. V. Vitanov, Phys. Rev. A **58**, 2295 (1998).

[104] R. G. Unanyan, B. W. Shore, and K. Bergmann, Phys. Rev. A **59**, 2910 (1999).

[105] N. V. Vitanov and S. Stenholm, Phys. Rev. A **60**, 3820 (1999).

[106] R. G. Unanyan, B. W. Shore, and K. Bergmann, Phys. Rev. A **63**, 043401 (2001).

[107] R. G. Unanyan, L. P. Yatsenko, K. Bergmann, and B. W. Shore, Opt. Commun. **139**, 48 (1997).

[108] M. Fleischhauer, R. Unanyan, B. W. Shore, and K. Bergmann, Phys. Rev. A **59**, 3751 (1999).

[109] S. Schiemann, A. Kuhn, S. Steuerwald, and K. Bergmann, Phys. Rev. Lett. **71**, 3637 (1993).

[110] J. Martin, B. W. Shore, and K. Bergmann, Phys. Rev. A **54**, 1556 (1996).

[111] W. Süptitz, B. C. Duncan, and P. L. Gould, J. Opt. Soc. Am. B **14**, 1001 (1997).

[112] K. Bergmann, H. Theuer, and B. W. Shore, Rev. Mod. Phys. **70**, 1003 (1998).

[113] S. Ghosh, S. Sen, S. S. Bhattacharyya, and S. Saha, Phys. Rev. A **59**, 4475 (1999).

[114] B. C. Duncan, V. Sanchez-Villicana, and P. L. Gould, Phys. Rev. A **63**, 043411 (2001).

[115] C. Y. Ye, A. Sautenkov, and Y. V. Rostovtsev, Opt. Lett. **28**, 2213 (1996).

[116] B. W. Shore, *The Theory of Coherent Atomic Excitation*, volume 2, John Wiley and Sons, New York, 1990.

[117] G. Z. He, A. Kuhn, S. Schiemann, and K. Bergmann, J. Opt. Soc. Am. B **7**, 1960 (1990).

[118] V. O. Nesterenko, P. G. Reinhard, W. Kleinig, and D. S. Dolci, Phys. Rev. A **70**, 023205 (2004).

[119] H. J. Metcalf and P. van der Straten, *Laser Cooling and Trapping*, Springer-Verlag, New York, 1999.

[120] B. H. Bransden and C. J. Joachain, *Physics of Atoms and Molecules*, volume 2nd Ed., Prentice Hall, New York, 2003.

[121] B. D. DePaola, Lectures on coherent excitation, Kansas State University, Manhattan, KS, 2003, (Unpublished).

[122] O. L. Weaver, Kansas State University, Manhattan, KS, 2005, (Private Communication).

[123] Wolfram Research, Inc., Mathematica, Version 5.1, Champaign, IL, 2004.

[124] The Mathematica documentation explains the `InterpolatingFunction` in the following manner: "Mathematica represents numerical approximations to functions as `InterpolatingFunction` objects. These objects are functions which, when applied to a particular x, return the approximate value of $y(x)$ at that point. The `InterpolatingFunction` effectively stores a table of values for $y(x_i)$, then interpolates this table to find an approximation to $y(x)$ at the particular x you request. `InterpolatingFunction` uses divided differences to construct Lagrange or Hermite interpolating polynomials. `InterpolatingFunction[...][x]` finds the value of an approximate function with a particular argument x. If you supply arguments outside of the domain, a warning is generated, and then an extrapolated value is returned.".

[125] C. E. Carroll and F. T. Hioe, Phys. Rev. A **42**, 1522 (1990).

[126] For the film in Fig. 2.7, the following parameters were used: $I_p = 6323\ mW/cm^2$, $I_s = 2652\ mW/cm^2$, $w_p = 50$ ns, $w_s = 50$ ns, $\Delta_1/2\pi = 53$ MHz, $\Delta_2/2\pi = 0$ MHz, and $\tau = -35$ ns. Likewise, the diabatic case used the following parameters: $I_p = 91.97\ mW/cm^2$, $I_s = 38.58\ mW/cm^2$, $w_p = 33$ ns, $w_s = 33$ ns, $\Delta_1/2\pi = 20$ MHz, $\Delta_2/2\pi = 0$ MHz, and τ ranged from -150 ns to $+150$ ns.

[127] The parameters used in Figs. D.1 and D.2 were $I_p = 1035\ mW/cm^2$, $I_s = 434\ mW/cm^2$, $w_p = w_s = 40$ ns, $\Delta_1/2\pi = 250$ MHz, $\Delta_2/2\pi = 0$ MHz, and $\tau = -20$ ns.

[128] L. Barbier and M. Chéret, J. Phys. B **20**, 1229 (1987).

[129] M. Chéret, L. Barbier, W. Lindinger, and R. Deloche, J. Phys. B **15**, 3463 (1982).

[130] For Fig. 2.10, the following parameters were used: $I_p = 20$ W/cm^2, I_s was set such that the Rabi frequencies, Ω_p and Ω_s were equal, $w_p = w_s = 50$ ns and $\tau = -80$ ns. Both Δ_1 and Δ_2 (one- and two-photon resonance, respectively) were allowed to vary.

[131] Y. B. Band, Phys. Rev. A **45**, 6643 (1992).

[132] J. C. Light, J. Chem. Phys. **66**, 5241 (1977).

[133] M. E. Crenshaw and C. D. Cantrell, Phys. Rev. A **66**, 5241 (1988).

[134] For Fig. 2.12, the following parameters were used in the adiabatic case: $I_p = 6323\ mW/cm^2$, $I_s = 2652\ mW/cm^2$, $w_p = 50$ ns, $w_s = 50$ ns, $\Delta_1/2\pi = 250$ MHz, $\Delta_2/2\pi = 0$ MHz, and $\tau = -35$ ns. Likewise, the diabatic case used the following parameters: $I_p = 92\ mW/cm^2$, $I_s = 39\ mW/cm^2$, $w_p = 33$ ns, $w_s = 33$ ns, $\Delta_1/2\pi = 20$ MHz, $\Delta_2/2\pi = 0$ MHz, and $\tau = -20$ ns.

[135] For the film in Fig. 2.13, the following parameters were used: I_p ranged from $60\ mW/cm^2$ to $6000\ mW/cm^2$, I_s was selected such that the Rabi frequencies, Ω_p and Ω_s matched at all times, $w_p = 3$ ns, $w_s = 33$ ns, $\Delta_1/2\pi = 20$ MHz,

$\Delta_2/2\pi = 0$ MHz, and $\tau = -35$ ns. Likewise, the diabatic case used the following parameters: $I_p = 92\ mW/cm^2$, $I_s = 39\ mW/cm^2$, $w_p = 33$ ns, $w_s = 33$ ns, $\Delta_1/2\pi = 20$ MHz, $\Delta_2/2\pi = 0$ MHz, and $\tau = -20$ ns.

[136] H. Nguyen, X. Fléchard, R. Brédy, H. A. Camp, and B. D. DePaola, Rev. Sci. Instrum. **75**, 2638 (2004).

[137] H. Nguyen, *MOTRIMS*, PhD thesis, Kansas State University, 2003.

[138] X. Fléchard, H. Nguyen, E. Wells, I. Ben-Itzhak, and B. D. DePaola, Phys. Rev. Lett. **87**, 123203 (2001).

[139] M. van der Poel, C. V. Nielsen, M. A. Gearba, and N. Andersen, Phys. Rev. Lett. **87**, 123201 (2001).

[140] J. W. Turkstra et al., Phys. Rev. Lett. **87**, 123202 (2001).

[141] X. Fléchard et al., Phys. Rev. Lett. **91**, 243005 (2003).

[142] J. Ullrich et al., J. Phys. B **30**, 2917 (1997).

[143] T. Kambara et al., J. Phys. B **28**, 4593 (1995).

[144] R. Dörner et al., Phys. Rep. **330**, 95 (2000).

[145] H. Schmidt-Böcking, R. Dörner, and J. Ullrich, Europhys. News **33/6**, 210 (2002).

[146] J. Ullrich et al., Rep. Prog. Phys. **66**, 1463 (2003).

[147] V. Mergel et al., Nucl. Instrum. Methods Phys. Res., Sect. B **98**, 593 (1995).

[148] E. L. Raab, M. Prentiss, A. Cable, S. Chu, and D. E. Pritchard, Phys. Rev. Lett. **59**, 2631 (1987).

[149] S. Chu and C. Wieman, J. Opt. Soc. Am. B **6**, 2020 (1989).

[150] C. Wieman, G. Flowers, and S. Gilbert, Am. J. Phys. **63**, 317 (1995).

[151] In reality, a typical Wiley-McLaren spectrometer[167] could have been used, but the MOTRIMS spectrometer was originally designed for precision measurements of both Q-Value and scattering angle.

[152] The type of detectors used is not critical, but details can be seen elsewhere.[136,137,168].

[153] M. A. Gearba et al., Opt. Lett. (2005), (submitted).

[154] T. G. Lee, H. Nguyen, X. Fléchard, B. D. DePaola, and C. D. Lin, Phys. Rev. A **66**, 042701 (2002).

[155] This is technically incorrect. A more accurate measurement would be to calculate an intensity based on elliptically-shaped beams. However, because the STIRAP technique is relatively robust, it is not necessary to know the Rabi frequencies with such precision, and thus assuming the beams to be circular is a close enough approximation.

[156] Special thanks to Jonathan Sabah who designed and wrote the LabVIEW interface. The version used for this dissertation is v. 1.2. We lovingly refer to it as "The Johnny-Box".

[157] National Instruments, Corp., Labview, Version 6.1, Austin, TX, 2001.

[158] K. B. MacAdam, A. Steinbach, and C. Wieman, Am. J. Phys. **60**, 1098 (1992).

[159] W. Lu et al., Rev. Sci. Instrum. **67**, 3003 (1996).

[160] B. T. H. Varcoe et al., Meas. Sci. Technol. **11**, N111 (2000).

[161] T. Petelski, M. Fattori, G. Lamporesi, J. Stuhler, and G. M. Tino, Eur. Phys. J. D **22**, 279 (2003).

[162] G. Wąsik, W. Gawlik, J. Zachorowski, and W. Zawadzki, Appl. Phys. B **75**, 613 (2002).

[163] M. H. Shah et al., Phys. Rev. A (2005), (submitted).

[164] OriginLab, Corp., Origin, Version 6.1, Northhampton, MA, 2001.

[165] For Fig. 4.6, the following parameters were used: $I_p = 623\ mW/cm^2$, $I_s = 214\ mW/cm^2$, $w_p = 50$ ns, $w_s = 50$ ns, $\Delta_1/2\pi = 53$ MHz, $\Delta_2/2\pi = 0$ MHz, and $\tau = -25$ ns.

[166] For Fig. 4.7, the following parameters were used in the adiabatic case: $I_p = 14.8\ W/cm^2$, $I_s = 36.9\ W/cm^2$, $w_p = 50$ ns, $w_s = 50$ ns, $\Delta_1/2\pi = 53$ MHz, $\Delta_2/2\pi = 0$ MHz, and $\tau = -25$ ns. Likewise, the diabatic case used the following parameters: $I_p = 142\ W/cm^2$, $I_s = 56\ W/cm^2$, $w_p = 50$ ns, $w_s = 50$ ns, $\Delta_1/2\pi = 53$ MHz, $\Delta_2/2\pi = 0$ MHz, and $\tau = -30$ ns.

[167] W. C. Wiley and I. H. McLaren, Rev. Sci. Instrum. **26**, 1150 (1955).

[168] H. C. Straub, M. A. Mangan, B. G. Lindsay, K. A. Smith, and R. F. Stebbings, Rev. Sci. Instrum. **70**, 4238 (1999).

[169] Special thanks to Dr. O. L. Weaver who spent a great deal of time and effort explaining density matrix derivations to the MOTRIMS group. The physical insights gained by his lectures were of great help.

[170] A. Einstein, Phys. Z. **18**, 121 (1917).

[171] R. C. Tolman, Phys. Rev. **23**, 693 (1924).

[172] R. C. Hilborn, Am. J. Phys. **50**, 982 (1981).

[173] I. I. Rabi, Phys. Rep. **51**, 652 (1937).

[174] C. Cohen-Tannoudji, B. Diu, and F. Laloë, *Quantum Mechanics*, Wiley-Interscience, New York, 1977.

[175] R. T. Robiscoe, Phys. Rev. A **25**, 1178 (1982).

[176] Y. B. Band and P. S. Julienne, J. Chem. Phys. **94**, 5291 (1991).

[177] Y. B. Band and P. S. Julienne, J. Chem. Phys. **95**, 5681 (1991).

Appendix A

Derivation of 2- and 3-Level Systems

This appendix presents the theoretical equations governing coherent population transfer in three-level systems, including population losses from spontaneous emission due to the finite lifetimes of each state. The incoherent and coherent population dynamics derivations presented here can be found in many texts,[116,119,120] while the majority of the density matrix treatment was compiled from other resources.[101,121,122,169]

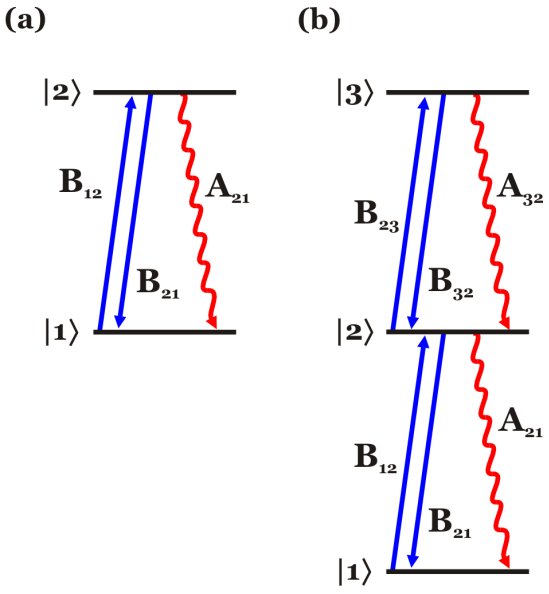

Figure A.1: Schematic diagram showing an atom interacting with incoherent radiation. **(a)** 2-Level system, **(b)** 3-level system. Stimulated (straight arrows) and spontaneous (rippled arrows) radiation are portrayed.

In order to better understand this complete theoretical picture, a short introduction is presented, describing the differences between atomic or molecular systems undergoing incoherent *versus* coherent excitation, both neglecting spontaneous emission. A two-level system will be utilized for simplicity, followed by an expanded derivation of coherent processes in a three-level system more similar to the experimental conditions shown in subsequent chapters. Finally, spontaneous emission will be included and a general theoretical treatment of the system will be presented. This step-by-step approach allows one to

appreciate the subtle behavior of coherent excitation processes.

A.1 Incoherent *vs.* Coherent Excitation

Albert Einstein was the first to propose three fundamental radiative processes governing how atoms can interact with radiation.[170–172] These three processes, spontaneous emission, stimulated absorption and stimulated emission, are schematically shown for a generic two-level system in Fig. A.1a. Two energy levels, $|1\rangle$ and $|2\rangle$ are portrayed, with stimulated (straight arrows) and spontaneous (rippled arrows) radiation passing between the states. The light that initiates stimulated emission and absorption has intensity, I, and all the spontaneous radiation has energy $\hbar\omega$.

The Einstein A- and B-coefficients mathematically introduce such stimulated and spontaneous physical processes into general incoherent rate equations. A_{nm}-coefficients represent spontaneous emission losses from energy level n to m. B_{nm}-coefficients where $n > m$ represent stimulated emission from state n to state m, while coefficients with $n < m$ represent stimulated absorption from state n to state m. Other loss terms, such as ionization, could be included, but for this treatment, the system is closed, and thus population is conserved between the two levels. Hence,

$$n_1 + n_2 = 1, \tag{A.1}$$

where n_1 and n_2 represent the populations in levels $|1\rangle$ and $|2\rangle$, respectively. The rate equations for the two levels can be written as

$$\dot{n}_1 = A_{21}n_2 + \frac{B_{21}In_2}{c} - \frac{B_{12}In_1}{c} \tag{A.2}$$

$$\dot{n}_2 = -\dot{n}_1, \tag{A.3}$$

where $B_{12} = B_{21} \equiv B$, assuming that $|1\rangle$ and $|2\rangle$ have the same degeneracy. Solving these coupled sets of equations yields populations in the two energy states as functions of time:

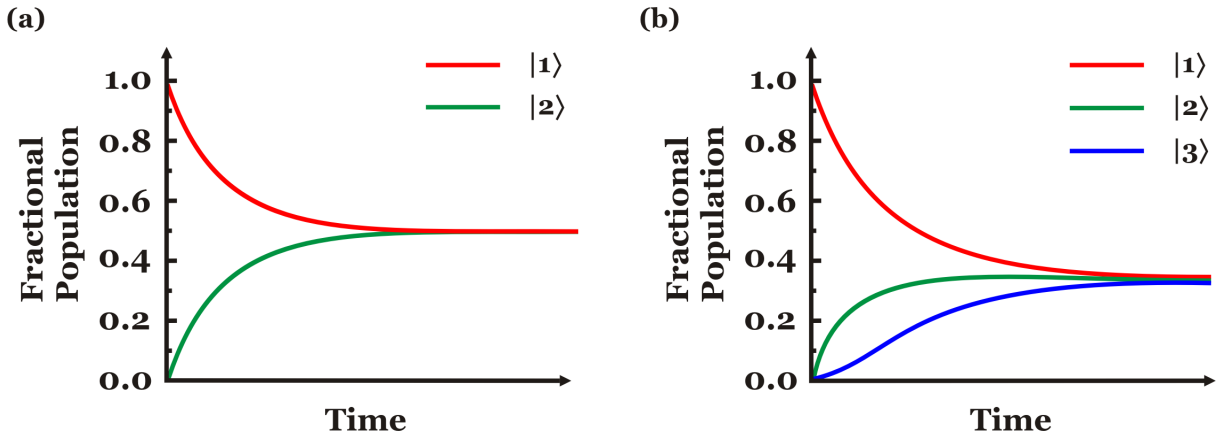

Figure A.2: Energy level populations shown as a function of time. **(a)** The red curve shows population in state $|1\rangle$, while the green curve shows population in state $|2\rangle$. **(b)** Populations in a 3-Level system shown as a function of time. The red curve shows population in state $|1\rangle$, the green curve shows population in state $|2\rangle$, and the blue curve shows population in state $|3\rangle$.

$$n_1(t) = \frac{A + BI/c}{A + 2BI/c}[1 + e^{-(A+2BI/c)t}] \tag{A.4}$$

$$n_2(t) = \frac{BI/c}{A + 2BI/c}[1 - e^{-(A+2BI/c)t}]. \tag{A.5}$$

It is useful to note that if $BI/c \gg A$ then, as $t \to \infty$, the populations in both n_1 and n_2 will approach 1/2. Hence, the time-average (steady-state) of the incoherent excitation process places 50% of the population in the excited state, as shown in Fig. A.2a.

The same treatment can be applied to the 3-level "ladder" system of Fig. A.1b, where energy levels $|1\rangle < |2\rangle < |3\rangle$, and transitions can only occur between levels $|1\rangle$ and $|2\rangle$, and levels $|2\rangle$ and $|3\rangle$, but not levels $|1\rangle$ and $|3\rangle$. Suppose the product $I_1 B_{12} = I_2 B_{23}$, where I_1 is the radiation connecting $|1\rangle$ and $|2\rangle$, and I_2 is the radiation connecting $|2\rangle$ and $|3\rangle$. Then, the rate equations for such a system are as follows:

$$\dot{n}_1 = A_{21}n_2 + \frac{B_{21}I_1n_2}{c} - \frac{B_{12}I_1n_1}{c} \tag{A.6}$$

$$\dot{n}_2 = A_{32}n_3 - A_{21}n_2 + \frac{B_{12}I_1n_1}{c} + \frac{B_{32}I_2n_3}{c} - \frac{B_{21}In_2}{c} - \frac{B_{23}In_2}{c} \tag{A.7}$$

$$\dot{n}_3 = \frac{B_{23}I_2n_2}{c} - A_{32}n_3 - \frac{B_{32}I_2n_3}{c}. \tag{A.8}$$

Figure A.2b shows incoherent population transfer as a function of time in a 3-level ladder system, where, once again, $BI/c \gg A$. Here, the steady-state solution places 1/3 of the population in each level. It is evident from these two simple cases that incoherent excitation is a very inefficient transfer mechanism. At best, the population transfer into each energy state is $1/n$, where n is the number of energy levels involved in the excitation process (assuming an equal number of degeneracies for each state).

More importantly, this simple rate equation picture does not account for effects incurred by using a *coherent* radiation source. In order to understand coherent excitation processes, a different approach must be used.

A.2 2-Level Coherent Excitation

The time-dependent Schrödinger equation (TDSE) is a logical place to begin when including the effects of coherent excitation in such an atomic system. For the treatment of the 2-level atom, it will also be of benefit to simplify the study by temporarily neglecting spontaneous emission from the excited state. Later, it will be shown how the Einstein-A coefficients can be included in the more comprehensive treatment of the 3-level atom. The TDSE and the Hamiltonian for the 2-level case are

$$i\hbar\frac{\partial}{\partial t}\Psi(t) = \mathscr{H}'(t)\Psi(t) \tag{A.9}$$

$$\mathscr{H}'(t) = \mathscr{H}_0 + \hat{V}(t), \tag{A.10}$$

where $\mathscr{H}_0$ is the Hamiltonian in the absence of any external fields, and $\hat{V}(t)$ is the time-dependent external radiation field, here taken as classical. In order to make use of the

TDSE, one must find the time-derivative of $\Psi(t)$. To that end,

$$\Psi(t) = \sum_n c_n(t)\psi_n e^{-i\xi_n(t)}, \tag{A.11}$$

where ψ_n satisfies the time-independent Schrödinger equation, $\mathscr{H}_0\psi_n = E_n^0\psi_n$, and $\xi_n(t)$ represents a time-dependent phase. Because it is not an observable quantity, an expression for $\xi_n(t)$ will be selected later for convenience. The time-derivative of $\Psi(t)$ is

$$\frac{\partial}{\partial t}\Psi(t) = \sum_n \psi_n[\dot{c}_n(t) - i\dot{\xi}_n(t)c_n(t)]e^{-i\xi_n(t)}. \tag{A.12}$$

From Eqns. A.10 and A.11, one can write

$$\mathscr{H}'\Psi(t) = \sum_n [\mathscr{H}_0 + \hat{V}(t)]c_n(t)\psi_n e^{-i\xi_n(t)}. \tag{A.13}$$

The presence of $\hat{V}(t)$ acting on ψ effectively redistributes the probability governing excitation:

$$\begin{aligned} \hat{V}(t)\psi_n &= V_{1n}(t)\psi_1 + V_{2n}(t)\psi_2 + \cdots \\ &= \sum_m V_{mn}(t)\psi_m. \end{aligned} \tag{A.14}$$

In order to obtain a measurable probability, one must multiply Eqn. A.14 from the left by ψ_q^* and integrate over all space, making use of the orthonormality characteristics of ψ_n. This can be expressed in conventional Dirac notation as

$$V_{qn}(t) = \int \psi_q^*\hat{V}(t)\psi_n d\tau = \langle\psi_q|\,\hat{V}(t)\,|\psi_n\rangle \equiv \langle q|\,\hat{V}(t)\,|n\rangle\,. \tag{A.15}$$

Substituting this result into Eqn. A.13 yields

$$\mathscr{H}'\Psi(t) = \sum_n c_n[E_n^0\psi_n + \sum_m \psi_m V_{mn}(t)]e^{-i\xi_n}. \tag{A.16}$$

The original TDSE (Eqn. A.9) can now be rewritten, substituting the solutions for $\frac{\partial}{\partial t}\Psi(t)$ (Eqn. A.12) and $\mathscr{H}'(t)\Psi(t)$ (Eqn. A.16) as

$$i\hbar \sum_n \psi_n[\dot{c}_n - i\dot{\xi}_n c_n]e^{-i\xi_n} \sum_n c_n[E_n^0\psi_n + \sum_m c_n\psi_m V_{mn}]e^{-i\xi_n}, \tag{A.17}$$

where the explicit time-dependence of $\dot{c}_n(t)$, $c_n(t)$, $\dot{\xi}_n(t)$, $\xi_n(t)$, and $V_{mn}(t)$ has been left off for simplicity. Once again, one can multiply by ψ_l^* and integrate over all space,

$$\begin{aligned} i\hbar \int \sum_n \psi_l^*\psi_n(\dot{c}_n - i\dot{\xi}_n c_n)e^{-i\xi_n}d\tau &= \int \sum_n c_n E_n^0 \psi_l^*\psi_n e^{-i\xi_n}d\tau \\ &+ \int \sum_n \sum_m c_n\psi_l^*\psi_m V_{mn}e^{-i\xi_n}d\tau. \end{aligned} \tag{A.18}$$

Utilizing the orthonormality of ψ_n,

$$\begin{aligned} i\hbar(\dot{c}_l - \dot{\xi}_l c_l)e^{-i\xi_l} &= c_l E_l^0 e^{-i\xi_l} + \sum_n c_n V_{ln}e^{-i\xi_n} \\ \Longrightarrow \qquad \hbar\dot{c}_l &= -i[(E_l^0 - \hbar\dot{\xi}_l)c_l + \sum_n c_n V_{ln}e^{-i(\xi_n - \xi_l)}] \end{aligned} \tag{A.19}$$

A.2.1 First Approximation: Near-Resonance

Up to this point, such treatment of the Schrödinger equation has been exact. For the case of an atom in a radiation field, the the Schrödinger equation cannot be solved exactly. Here, however, the first of three approximations will be introduced, allowing us to solve the system of coupled differential equations. The system of interest is composed of only two energy levels coupled by the laser frequency. One approach to approximate a finite-level situation, first studied by I. I. Rabi,[173] is to truncate the sum in Eqn. A.19, thus including only the same number of terms as there are energy levels. That is, for a two-level system, the indices n and l will range from 1 to 2. This is a reasonable approximation because the laser frequency used for excitation will typically be narrow-band and near-resonant, and thus the

contribution to other energy levels will be negligible. It makes little sense, therefore, to keep track of interactions with energy levels far from resonance. Thus, the two terms $\hbar\dot{c}_1$ and $\hbar\dot{c}_2$ become:

$$\begin{aligned} \hbar\dot{c}_1 &= -i[(E_1^0 - \hbar\dot{\xi}_1)c_1 + V_{12}c_2e^{-i(\xi_2-\xi_1)}] \\ \hbar\dot{c}_2 &= -i[(E_2^0 - \hbar\dot{\xi}_2)c_2 + V_{21}c_1e^{+i(\xi_2-\xi_1)}]. \end{aligned} \tag{A.20}$$

It is more convenient to rewrite such terms in matrix form:

$$\hbar\dot{\mathbf{c}} = -i \begin{pmatrix} E_1^0 + V_{11} - \hbar\dot{\xi}_1 & V_{12}e^{-i(\xi_2-\xi_1)} \\ V_{12}^*e^{+i(\xi_2-\xi_1)} & E_2^0 + V_{22} - \hbar\dot{\xi}_2 \end{pmatrix} \mathbf{c}, \tag{A.21}$$

where

$$\mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}. \tag{A.22}$$

A.2.2 Second Approximation: Electric Dipole

Typically, the optical wavelength, λ, of the radiation interacting with the atom will be much larger than the distances involving the wavefunctions of the atom. For example, coherent light with $\lambda \simeq 800$ nm interacts with an typical atom where the wavefunctions are contained within a sphere of radius < 1 nm. The electric field, $\mathbf{E}(\mathbf{r}, t)$, can therefore be treated as spatially uniform with respect to the atom. This is called the *dipole approximation.*[116,119,120] The electric field operator for a generalized plane wave has the form

$$\mathbf{E}(\mathbf{r}', t) = \frac{1}{2}E_0[e^{i(\mathbf{k}\cdot\mathbf{r}'-\omega t)} + e^{-i(\mathbf{k}\cdot\mathbf{r}'-\omega t)}], \tag{A.23}$$

where E_0 is the maximum amplitude of the electric field, $\mathbf{k}$ is the propagation vector, and $\mathbf{r}'$ is the position vector. The magnitude of the propagation vector, $k = \frac{2\pi}{\lambda}$, is called the wave number. The exponential terms involving the propagation vector in Eqn. A.23 can be expanded as

$$e^{i\mathbf{k}\cdot\mathbf{r}'} = 1 + (i\mathbf{k}\cdot\mathbf{r}') + \frac{1}{2!}(i\mathbf{k}\cdot\mathbf{r}')^2 + \dots . \tag{A.24}$$

Applying the dipole approximation, therefore, truncates the exponential expansion such that $e^{i\mathbf{k}\cdot\mathbf{r}'} \simeq 1$. The dipole approximation can be applied so long as $ka \ll 1$, where a is the typical linear dimensions of the atomic wave functions. In the dipole approximation, then,

$$\hat{V} = -e\mathbf{E}(\mathbf{r}, t)\cdot\mathbf{r}, \tag{A.25}$$

and one can write $V_{mn}(t)$ explicitly as

$$\begin{aligned} V_{mn}(t) &= -e\mathbf{E}\langle m|\,\mathbf{r}\,|n\rangle \\ &= \frac{e}{2}E_0(e^{i\omega t} + e^{-i\omega t})\langle m|\,\mathbf{r}\,|n\rangle\,. \end{aligned} \tag{A.26}$$

Here, it is useful to introduce the Rabi frequency, defined as[23]

$$\Omega \equiv \frac{-eE_0}{\hbar}\langle e|\,\mathbf{r}\,|g\rangle\,, \tag{A.27}$$

where $\langle e|$ is the excited state, $|g\rangle$ is the ground state, and $\mathbf{r}$ is the electron coordinate. The physical significance of the Rabi frequency is that it describes the coupling strength between the atom and the radiation field. This will be seen in greater detail later on. Substituting this definition of Ω into Eqn. A.26 and combining this with Eqn. A.21 produces a more compact solution for $\hbar\dot{\mathbf{c}}$. One can reduce the complexity further by defining the energy levels

$$\begin{aligned} E_1 &\equiv E_1^0 + V_{11} \\ E_2 &\equiv E_2^0 + V_{22}. \end{aligned} \tag{A.28}$$

Now, Eqn. A.21 becomes

$$\hbar\dot{\mathbf{c}} =$$

$$-i\begin{pmatrix} E_1 - \hbar\dot{\xi}_1 & \frac{1}{2}\hbar\Omega(e^{-i(\xi_2-\xi_1-\omega t)} + e^{-i(\xi_2-\xi_1+\omega t)}) \\ \frac{1}{2}\hbar\Omega^*(e^{-i(\xi_2-\xi_1-\omega t)} + e^{-i(\xi_2-\xi_1+\omega t)}) & E_2 - \hbar\dot{\xi}_2 \end{pmatrix}\mathbf{c}. \tag{A.29}$$

The phases ξ_n and their time-derivatives are arbitrary. For convenience, one can select the arbitrary phases appropriately so as to simplify the problem:

$$\xi_2 - \xi_1 = \omega t, \tag{A.30}$$

and therefore:

$$\dot{\xi}_2 - \dot{\xi}_1 = \omega. \tag{A.31}$$

This choice allows one to write Eqn. A.29 as

$$\hbar\dot{\mathbf{c}} = -i\begin{pmatrix} E_1 - \hbar\dot{\xi}_1 & \frac{1}{2}\hbar\Omega(e^{-2i\omega t} + 1) \\ \frac{1}{2}\hbar\Omega^*(e^{2i\omega t} + 1) & E_2 - \hbar\dot{\xi}_2 \end{pmatrix}\mathbf{c}, \tag{A.32}$$

which is certainly more aesthetically pleasing. More importantly, this choice of phase allows one to make an important approximation.

A.2.3 Third Approximation: Rotating Wave

The third and final approximation is now applied: the Rotating Wave Approximation (RWA). Here, because of the phase choice, one of the two exponential terms in Eqn. A.29 was replaced with unity. Why is this beneficial? Shore[116] states it most clearly, saying, "we wish to choose the phase difference that, while eliminating an exponential time variation,

also produces the smallest diagonal elements... [T]he addition of a constant to all diagonal elements is equivalent to the introduction of a state vector phase that has no effect on observable properties. By making diagonal elements small we avoid unnecessary labor of computing rapid oscillations shared by all amplitudes."

The RWA, therefore, allows one to ignore high-frequency oscillations and replace

$$e^{\pm 2i\omega t} + 1 \simeq 1, \tag{A.33}$$

making Eqn. A.32 become

$$\hbar\dot{\mathbf{c}} = -i \begin{pmatrix} E_1 - \hbar\dot{\xi}_1 & \frac{1}{2}\hbar\omega \\ \frac{1}{2}\hbar\Omega^* & E_2 - \hbar\dot{\xi}_2 \end{pmatrix} \mathbf{c}. \tag{A.34}$$

The laser field coupling energy states $|1\rangle$ and $|2\rangle$ can be detuned from resonance by some amount defined as

$$\begin{aligned} E_1 - \hbar\dot{\xi}_1 &\equiv \hbar\Delta_1 \\ E_1 - \hbar\dot{\xi}_2 &\equiv \hbar\Delta_2 \\ &= E_2 - E_1 + \hbar\Delta_1 - \hbar\Omega \end{aligned} \tag{A.35}$$

Defining the zero on the energy axis to be $\Delta_1 \equiv 0$ simplifies the picture somewhat, and is not unreasonable. With $\Delta_1 = 0$, the coupling radiation is defined to start from the ground state, $|1\rangle$, while Δ_2 represents the angular frequency detuning from resonance with the excited state, $|2\rangle$.

Figure A.3 shows such an energy-level diagram with the energy states thus defined. The coupling radiation field between levels $|1\rangle$ and $|2\rangle$ has an energy $\hbar\omega$. Rewriting Eqn. A.34 to include such detuning definitions, and dividing both sides by $\hbar$, provides a greatly simplified equation for $\dot{c}_1$:

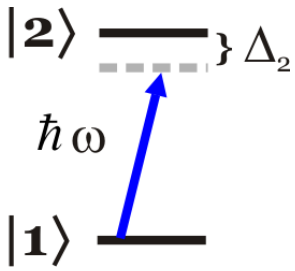

Figure A.3: Energy level diagram showing a 2-level system, where the laser field is detuned from $|2\rangle$ by Δ_2.

$$\dot{\mathbf{c}} = -\frac{i}{2}\begin{pmatrix} 0 & \Omega \\ \Omega^* & 2\Delta_2 \end{pmatrix}\mathbf{c}. \tag{A.36}$$

Solving Eqn. A.36 for c_1 and c_2 is now straightforward, given the initial boundary conditions $c_2(0) = 0$ and $c_1(0) = 1$. Such initial conditions indicate that all the population is in the ground state at time $t = 0$. Solving for c_2 yields

$$c_2 = -i\frac{\Omega}{\Omega'}\sin^2\left(\frac{\Omega' t}{2}\right)e^{-i\frac{1}{2}\Delta_2 t}. \tag{A.37}$$

Here, $\Omega' \equiv \sqrt{\Omega^2 + \Delta_2^2}$. Finally, the probability, $P_2(t)$, of finding an atom in state $|2\rangle$ is

$$\begin{aligned} P_2(t) &= c_2^* c_2 \\ &= \left(\frac{\Omega}{\Omega'}\right)^2 \sin^2\left(\frac{\Omega' t}{2}\right) \\ &= \frac{\Omega^2}{2\Omega'^2}[1 - \cos(\Omega' t)] \end{aligned} \tag{A.38}$$

The probability of placing an atom in (or removing it from) the excited state oscillates at a frequency of Ω' which is therefore known as the flopping frequency. Note that for resonant radiation, $\Delta_2 = 0$ and $\Omega' = \Omega$, the Rabi frequency. As Δ_2 increases, the frequency of oscillation also increases, although the amplitude decreases. Figure A.4 shows such behavior. The probability of excitation $P_2(t)$ as a function of time is plotted *versus* detuning Δ_2. When the laser field is exactly tuned to be resonant with $|2\rangle$, that is $\Delta_2 = 0$, 100% of the population oscillates between the ground state and the excited state.

As mentioned in Sec. 1.1, π-pulses are a common technique to achieve efficient population transfer. The ability to control experimental conditions is related to Ω', that is, laser intensity and detuning, must be controlled to much better than half a Rabi period. This is, in practice, a difficult task. For example, for a laser field with a spatial Gaussian intensity profile, Ω', and therefore P_2 varies considerably over the extent of the beam spot.

The definition of the Rabi frequency presented in Eqn. A.27 can be explained in terms of experimental parameters in order to provide a better understanding of the experimental

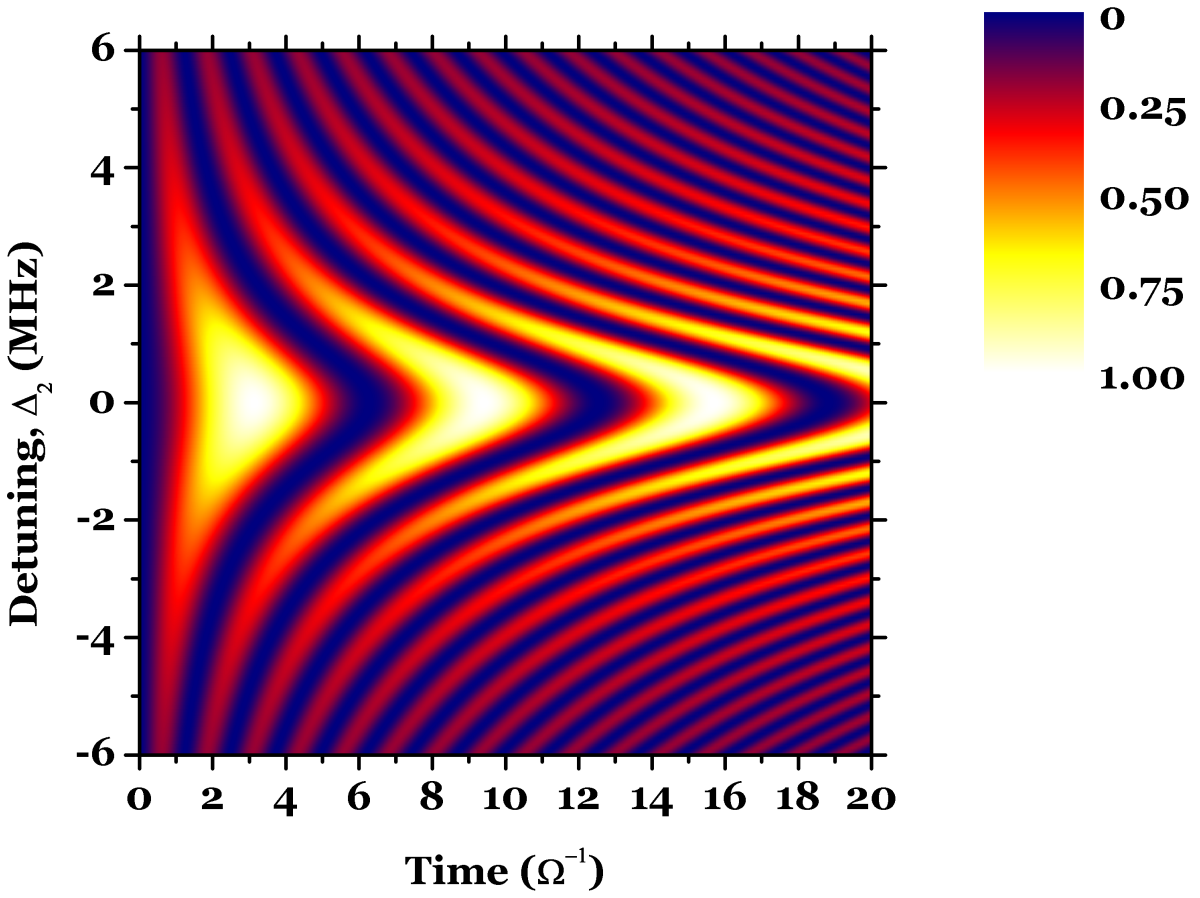

Figure A.4: Probability of atomic excitation $P_2(t)$ from the ground state to the excited state as a function of time, plotted *versus* detuning Δ_2. The false color represents this probability.

control necessary in such π-pulse techniques. The purpose here is not to belabor the assertion that such an experimental method is difficult, but to ground the reader in an understanding as to why coherent excitation techniques can be sensitive to experimental conditions. This illustrates why the choice of using STIRAP for coherent excitation is an inviting solution to such experimental difficulties, and will hopefully instill an appreciation for the work presented in this dissertation.

The Rabi frequency, as defined previously, is written again for convenience:

$$\Omega \equiv \frac{-eE_0}{\hbar} \langle e| \mathbf{r} |g\rangle \,, \tag{A.39}$$

The dipole moment, μ_{eg}, is dependant on the ground state and excited state wavefunctions, and is written as

$$\mu_{eg} = -e \langle e | \mathbf{r} | g \rangle , \tag{A.40}$$

while the decay rate of the population from the excited state to the ground state, γ, is

$$\gamma = \frac{\omega^3 \mu^2}{3\pi\epsilon_0 \hbar c^3}. \tag{A.41}$$

The intensity, I, of the coherent light is written as

$$I = \frac{c\epsilon_0}{2} E_0^2, \tag{A.42}$$

With these definitions, one can represent Ω in parameters more convenient to measure in the laboratory, namely, I, γ, and λ. From Eqn. A.39, then, one can write

$$\begin{aligned} \Omega &= -eE_0 \langle e | \mathbf{r} | g \rangle \sqrt{\frac{1}{\hbar^2}} \\ &= \sqrt{\frac{2I\mu^2}{c\epsilon_0 \hbar^2}} \\ &= \sqrt{\frac{3\lambda^3 I\gamma}{2\pi hc}}. \end{aligned} \tag{A.43}$$

Because the intensity is proportional to the square of the Rabi frequency, variations in intensity can dramatically affect the overall population transfer when using methods that rely on precise control of Ω. The treatment of 2-level coherent excitation presented here has been beneficial in contrasting the behavior previously seen in the classical incoherent treatment. Also, the approximations used in the 2-Level Schrödinger approach have been delineated, and these same approximations will be used in the following sections.

A.2.4 ARP: The Dressed-State Approach

In order to gain some insight into adiabatic rapid passage (ARP), one can study two-level coherent excitation in a different manner than prescribed above. Starting with the time-independent Schrödinger equation,

$$\mathscr{H}\phi = \lambda\phi, \tag{A.44}$$

where λ is the eigenvalue and ϕ is the eigenvector of the Hamiltonian. This stationary state is called a "dressed state" in the Hamiltonian and contains the effects of the laser field. To see this, one can solve the eigenvalue equation:

$$\frac{\hbar}{2}\begin{pmatrix} 0 & \Omega \\ \Omega & 2\Delta \end{pmatrix}\begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix} = \lambda\begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix}. \tag{A.45}$$

The eigenvalues for this equation are

$$\lambda_\pm = \frac{\hbar}{2}(\Delta \pm \Omega'). \tag{A.46}$$

The associated eigenvectors are

$$\phi_\pm = \phi_\pm^0\left(1, \quad \tfrac{\Delta\pm\Omega'}{\Omega}\right), \tag{A.47}$$

where $\phi_\pm^0$ are normalization constants.

A useful visual interpretation of the eigenvectors can be made at this point by introducing the angle, θ, relating the Rabi frequency, the flopping frequency, and the detuning.[116,174] Figure A.5 shows such a relationship. Notice that when 2θ approaches even multiples of π, the detuning is large and positive, called "blue detuning". Similarly, when 2θ approaches odd multiples of π, the detuning is large and negative, called "red detuning". Rewriting the eigenvectors in terms of θ, therefore, yield

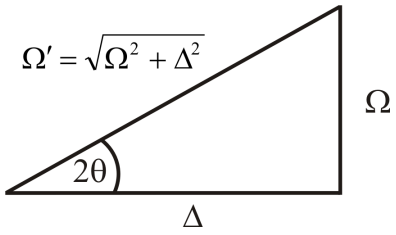

Figure A.5: Visual interpretation of the angle θ in the 2-level dressed-state approach.

$$\phi_{\pm} = \phi_{\pm}^{0}\left(1, \quad \tfrac{\cos(2\theta)\pm 1}{\sin(2\theta)}\right). \tag{A.48}$$

One can normalize the eigenvectors to give

$$\phi_{+} = \left(\sin\theta, \quad \cos\theta\right) \tag{A.49}$$

$$\phi_{-} = \left(\cos\theta, \quad -\sin\theta\right). \tag{A.50}$$

Therefore,

$$\begin{aligned} \phi_{+} &= \psi_1 \sin\theta + \psi_2 \cos\theta \\ \phi_{-} &= \psi_1 \sin\theta - \psi_2 \cos\theta, \end{aligned} \tag{A.51}$$

where ψ_1 and ψ_2 are the eigenvector of the field-free system. The probabilities for observing an atom in the $\pm$ states are then

$$\begin{aligned} P_{+} &= \phi_{+}^{*}\phi_{+} = |\psi_1|^2 \sin^2\theta + |\psi_2|^2 \cos^2\theta \\ P_{-} &= \phi_{-}^{*}\phi_{-} = |\psi_1|^2 \cos^2\theta + |\psi_2|^2 \sin^2\theta. \end{aligned} \tag{A.52}$$

Assuming that, at time $t = 0$, the entire population is found in the ground state, then $P_1 = |\psi_1|^2 = 1$ (and, conversely, $P_2 = |\psi_2|^2 = 0$). If one starts with a large red detuning, that is, $|\Delta| \gg \Omega$ and $\Delta < 0$, then $\theta \simeq \frac{\pi}{2}$. This must mean that the population is in the ϕ_+ state, since $P_+ = \sin(\frac{\pi}{2})^2 = 1$. Now, if one changes the detuning of the system slowly, without leaving the ϕ_+ state (that is, $P_+ = 1$, always), this is termed an adiabatic transfer. If the detuning is swept from red to blue adiabatically, passing through $\Delta = 0$, the entire population will be transferred into ψ_2, since $\theta \simeq 0$ when $|\Delta| \gg \Omega$ and $\Delta > 0$. If, on the other hand, one begins with a large blue detuning, the population will start with $P_- = 1$, and, after adiabatically sweeping the detuning to the red, the system will also terminate in the ψ_2 state.

Figure A.6 shows the dressed-state picture of coherent population transfer as a function of detuning under adiabatic conditions. Note that spontaneous emission is neglected. The population is completely swept from the ground state, P_1, into the excited state, P_2. Also note that, unlike the earlier excitation scheme, total population transfer occurs *independent of laser intensity* so long as adiabatic conditions are maintained. This dressed state conceptualization provides one with a framework in which to quantitatively define what is meant by "adiabatic" when referring to ARP, as will be discussed next.

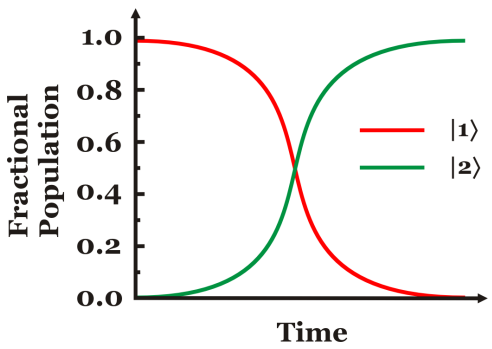

Figure A.6: The dressed-state picture of coherent population transfer as a function of detuning. Spontaneous emission is neglected.

A.2.5 The Adiabatic Approximation

In order to describe adiabatic interactions in some detail, the Rabi frequency and the angle θ can be considered as slowly time-varying functions.[116] Thus,

$$\begin{aligned} \phi_+(t) &= \psi_1 \sin[\theta(t)] + \psi_2 \cos[\theta(t)] \\ \phi_-(t) &= \psi_1 \cos[\theta(t)] - \psi_2 \sin[\theta(t)], \end{aligned} \tag{A.53}$$

where (as deduced from Fig. A.5)

$$\sin[2\theta(t)] = \frac{\Omega(t)}{\sqrt{\Omega(t)^2 + \Delta^2}} \equiv \frac{\Omega(t)}{\Omega'(t)}, \tag{A.54}$$

are the time-dependent eigenvectors of the Hamiltonian, $\mathscr{H}(t)$. To make notation more compact, one can denote the symbol ν to represent the labels $+$ and $-$ shown on the eigenvectors and eigenvalues. With this new notation,

$$\mathscr{H}(t)\phi_\nu(t) = \frac{\hbar}{2}(\Delta \pm \Omega'_\nu(t))\phi_\nu(t) \tag{A.55}$$

where $\Omega'_\nu \equiv \Omega'_\pm = \pm\Omega'$. One can define the interaction, $V(t)$, to be 0 prior to time $t = 0$. Other initial conditions include

$$\begin{array}{rclcrcl} \mathscr{H}(0) & = & \mathscr{H}^0 & & \theta(0) & = & 0 \\ \Omega(0) & = & 0 & & \Phi_-(0) & = & \psi_1 \\ \Omega'(0) & = & \Delta & & \Phi_+(0) & = & \psi_2. \end{array} \tag{A.56}$$

The vectors $\Phi_\nu(0)$ shown above are called adiabatic states. The goal, here, is to find the state vector, $\Psi(t)$, such that

$$\hbar\frac{\partial}{\partial t}\Psi(t) = -i\mathscr{H}(t)\Psi(t), \tag{A.57}$$

and the initial conditions are also satisfied. Writing $\Psi(t)$ in the adiabatic state basis,

$$\Psi(t) = A_+(t)\Phi_+(t) + A_-(t)\Phi_-(t), \tag{A.58}$$

subject to the condition, $\Psi(0) = \psi_1$. Then, $A_\nu(t)$ must satisfy the initial conditions

$$\begin{array}{rcl} A_-(0) & = & 1 \\ A_+(0) & = & 0. \end{array} \tag{A.59}$$

The time derivatives of the adiabatic basis vectors are

$$\begin{array}{rcl} \dfrac{d}{dt}\Phi_-(t) & = & -[\psi_1\sin\theta(t) - \psi_2\cos\theta(t)]\dfrac{d}{dt}\theta(t) \\ & = & -\Phi_+(t)\dfrac{d}{dt}\theta(t), \\ \dfrac{d}{dt}\Phi_+(t) & = & \Phi_-(t)\dfrac{d}{dt}\theta(t), \end{array} \tag{A.60}$$

whereupon the time derivative of $\Psi(t)$ is obtained:

$$\hbar\frac{d}{dt}\Psi(t) = \hbar\left[\dot{A}_+(t) - \dot{\theta}(t)A_-(t)\right]\Phi_+(t) + \hbar\left[\dot{A}_-(t) + \dot{\theta}(t)A_+(t)\right]\Phi_-(t). \tag{A.61}$$

Utilizing the orthogonality of the adiabatic basis vectors,

$$\frac{d}{dt}\begin{bmatrix} A_+(t) \\ A_-(t) \end{bmatrix} = \frac{-i}{2}\begin{bmatrix} \Omega'(t) & 2i\dot{\theta}(t) \\ -2i\dot{\theta}(t) & -\Omega'(t) \end{bmatrix}\begin{bmatrix} A_+(t) \\ A_-(t) \end{bmatrix}. \tag{A.62}$$

Neglecting the $\dot{\theta}(t)$ terms in Eqn. A.62 is equivalent to the adiabatic approximation made earlier, and produces a set of uncoupled equations for the amplitudes $A_\nu(t)$:

$$A_\pm(t) \simeq \exp\left[\mp\frac{i}{2}\int_0^t dt'\Omega'(t)\right] A_\pm(0). \tag{A.63}$$

For the initial assumptions made throughout this derivation, applying the adiabatic approximation means the amplitude $A_+(t)$ will remain zero, while $A_-(t)$ changes as a function of time. The $\dot{\theta}$ term can be usefully rewritten as a function of the Rabi flopping frequency and the detuning:

$$\begin{aligned} 2\dot{\theta}(t) &= \frac{\dot{\Omega}(t)\Delta}{[\Omega(t)^2+\Delta^2]} \\ &= \frac{\dot{\Omega}(t)\Delta}{\Omega'(t)^2} \end{aligned} \tag{A.64}$$

The adiabatic condition used[116,175] to determine whether a system will remain in a dressed state is then

$$|\dot{\Omega}(t)\Delta| \ll |\Omega'(t)|^3. \tag{A.65}$$

Adiabatic assumptions are made frequently[132,133] when studying the two-level system, as has been presented here. Equation A.65 provides a quantitative measurement for the adiabaticity of the excitation process. Adiabatic Rapid Passage relies on such slow time-varying changes. If changes in $\Omega(t)$ were to occur more rapidly than the relationship shown in Eqn. A.65, the atom would be unable to respond adiabatically and oscillations in populations would occur.[116]

A.3 3-Level Coherent Excitation

Thus far, it has been shown that 2-level coherent excitation can be derived directly from Schrödinger's equation, and an alternative dressed state picture enhances one's physical understanding, especially by providing an explicit definition of what is meant by adiabatic conditions. Up to this point, however, spontaneous emission has been conspicuously absent from the treatment of coherently excited populations. As the concepts presented previously are expanded, and the 3-level atom is studied, it is no longer desirable to neglect such an effect.

It is beneficial to approach the 3-level atom by introducing density matrices. Many treatments of multi-level coherent excitation utilize density matrices,[116,119,131,176,177] and there are several advantages to such an approach. For example, it allows one to add decay terms into the system phenomenologically, and to estimate decoherence effects within the system. Such information is important in understanding the adiabatic *versus* diabatic processes taking place under different experimental conditions.

A.3.1 Density Matrices

As noted above, there have been several general treatments of density matrices applied to coherent excitation systems. A general overview[169] will be presented here, after which the 3-level system of interest throughout the remainder of this dissertation will be discussed.

The density matrix is defined to be

$$\rho(t) = |\psi\rangle \langle\psi| . \tag{A.66}$$

There are several important things to note about the density matrix. First, it is Hermitian. Second, because the wavefunction is normalized, the trace is unity. Third, one can diagonalize ρ with a unitary transformation. The diagonal terms represent the probabilities of the system states. If $Tr(\rho^2) = 1$, the system is said to be in a pure state, meaning one of the diagonal terms must be 1, and the rest zero. Conversely, if $Tr(\rho^2) < 1$, then the system

is said to be in a mixed state, where it is represented by the density operator

$$\rho = \sum_i p_i \left|\Psi_i\right\rangle \left\langle\Psi_i\right| . \tag{A.67}$$

where p_i is the probability of finding the system in state Ψ_i. Mixed states will become important momentarily, since part of the motivation for using density matrices is to allow one to include the effects of spontaneous emission. Now,

$$i\hbar \frac{d}{dt}\rho(t) = i\hbar \left[\frac{d\left|\psi\right\rangle}{dt} \left\langle\psi\right| + \left|\psi\right\rangle \frac{d\left\langle\psi\right|}{dt} \right] . \tag{A.68}$$

One can recognize that the time-dependent Schrödinger equation is

$$\begin{aligned} i\hbar \frac{d}{dt}\left|\psi\right\rangle &= \mathscr{H}\left|\psi\right\rangle \\ -i\hbar \frac{d}{dt}\left\langle\psi\right| &= \left(\mathscr{H}\left|\psi\right\rangle\right)^\dagger \\ &= \left\langle\psi\right| \mathscr{H}^\dagger \end{aligned} \tag{A.69}$$

Combining Eqns. A.68 and A.69 yields

$$\begin{aligned} i\hbar\dot{\rho}(t) &= \mathscr{H}\left|\psi\right\rangle \left\langle\psi\right| - \left|\psi\right\rangle \left\langle\psi\right| \mathscr{H}^\dagger \\ &= \mathscr{H}\rho(t) - \rho(t)\mathscr{H}^\dagger \\ &\equiv \left[\mathscr{H}, \rho(t)\right] . \end{aligned} \tag{A.70}$$

This solution is known as the quantum Louisville equation. However, no decay phenomena have been included yet.

A.3.2 Including Spontaneous Emission: The Hamiltonian

At this point, one should be more careful in describing the Hamiltonian of the system. In particular, if one defines the system S to be the atom plus the laser modes, and S interacts with a reservoir, R, then the Hamiltonian is

$$\mathscr{H} = \mathscr{H}_S + \mathscr{H}_L + \mathscr{H}_R + \mathscr{H}_I \tag{A.71}$$

where $\mathscr{H}_S$ describes interactions with S, $\mathscr{H}_L$ represents the time-dependent interaction of the laser field and atom, $\mathscr{H}_R$ describes the interaction with the free radiation field (and hence will not affect the state of the atom), and $\mathscr{H}_I$ describes the interaction between the non-lasing modes of the radiation field and the atom. $\mathscr{H}_I$ is

$$\mathscr{H}_I = V_S^\dagger a + V_S a^\dagger, \tag{A.72}$$

where V_S interacts with the atomic variables. a and $a^\dagger$ are photon destruction and creation operators, respectively. Thus, $\mathscr{H}_I$ will always change the photon number by ± 1.

The complete equation of motion for the entire density matrix is therefore,

$$i\hbar\dot{\rho}(t) = [\mathscr{H}_S + \mathscr{H}_L + \mathscr{H}_R + \mathscr{H}_I, \rho(t)] \tag{A.73}$$

It will be helpful to address the problem in terms of the interaction picture by making a unitary matrix transformation of ρ. Thus,

$$\rho(t) = e^{-i(\mathscr{H}_S+\mathscr{H}_R)t}\tilde{\rho}(t)e^{+i(\mathscr{H}_S+\mathscr{H}_R)t}, \tag{A.74}$$

where $\tilde{\rho}$ is the density matrix in the interaction picture. The time derivative of $\tilde{\rho}(t)$ is

$$i\hbar\frac{d}{dt}\tilde{\rho}(t) = \left[\tilde{\mathscr{H}_L}(t) + \tilde{\mathscr{H}_I}(t), \tilde{\rho}(t)\right], \tag{A.75}$$

and the Hamiltonian components, $\mathscr{H}_L$ and $\mathscr{H}_I$, in the interaction picture are

$$\begin{aligned} \tilde{\mathscr{H}_L} &= e^{+i(\mathscr{H}_S+\mathscr{H}_R)t}\mathscr{H}_L e^{-i(\mathscr{H}_S+\mathscr{H}_R)t}, \\ \tilde{\mathscr{H}_I} &= e^{+i(\mathscr{H}_S+\mathscr{H}_R)t}\mathscr{H}_I e^{-i(\mathscr{H}_S+\mathscr{H}_R)t}. \end{aligned} \tag{A.76}$$

Now, one can replace $\tilde{\rho}(t)$ in the second term of Eqn. A.75 (leaving the first term alone) with

$$\tilde{\rho}(t) = \tilde{\rho}(0) - i\int_0^t \left[\tilde{\mathscr{H}_L}(t') + \tilde{\mathscr{H}_I}(t'), \tilde{\rho}(t')\right] dt', \tag{A.77}$$

which is simply Eqn. A.75 rewritten by integrating both sides. Then, the equation of motion becomes

$$\begin{aligned} &i\hbar\tfrac{d}{dt}\tilde{\rho}(t) = \\ &\left[\tilde{\mathscr{H}_L}(t), \tilde{\rho}(t)\right] + \left[\tilde{\mathscr{H}_I}(t), \tilde{\rho}(0)\right] - i\left[\tilde{\mathscr{H}_I}(t), \textstyle\int_0^t \left[\tilde{\mathscr{H}_L}(t') + \tilde{\mathscr{H}_I}(t'), \tilde{\rho}(t')\right] dt'\right] \end{aligned} \tag{A.78}$$

The last term in this solution is called the dissipative term, and includes the loss terms from spontaneous emission. Now, in order to obtain an equation for the system density matrix, ρ_S, one must take the trace over the reservoir variables, as shown below. Taking the trace *only over the reservoir variables* will be denoted as Tr_R. In other words, $\rho_S = Tr_R(\rho)$. For the remainder of the derivation, the tilde notation will be suppressed, yet the equations still refer to the interaction picture. Also, $\rho(t)$ can be written only as ρ in some instances, to avoid confusing notation issues. This is done simply for clarity; ρ is still understood to be an explicit function of t. Starting from the left-hand side of Eqn. A.78, one finds

$$\begin{aligned} \left[Tr_R(i\dot{\rho})\right]_{ab} &= \sum_r i\dot{\rho}_{ar,br} \\ &= i\frac{\partial}{\partial t}\sum_r \rho_{ar,br} \\ &= i\left(\dot{\rho}_S\right)_{ab}. \end{aligned} \tag{A.79}$$

The first term on the right-hand side of Eqn. A.78 becomes

$$\left(Tr_R\left[\mathscr{H}_L, \rho(t)\right]\right)_{ab} = \sum_r \left(\langle ra|\,\mathscr{H}_L\,|r'c\rangle\,\langle r'c|\,\rho\,|r,b\rangle - \langle ra|\,\rho\,|r'c\rangle\,\langle r'c|\,\mathscr{H}_L\,|rb\rangle\right). \tag{A.80}$$

This can be simplified by the following relations

$$\begin{aligned} \langle ra| \mathscr{H}_L |r'c\rangle &= \delta_{rr'} \langle a| \mathscr{H}_L |c\rangle \\ \langle r'c| \mathscr{H}_L |rb\rangle &= \delta_{rr'} \langle c| \mathscr{H}_L |b\rangle , \end{aligned} \tag{A.81}$$

where $\delta_{rr'}$ is the Kronecker delta. This reduces Eqn. A.80 to be

$$\begin{aligned} \left(Tr_R \left[\mathscr{H}_L, \rho(t)\right]\right)_{ab} &= \sum_r \left[\left(\mathscr{H}_L\right)_{ac} \rho_{rc,rb} - \rho_{ra,rc} \left(\mathscr{H}_L\right)_{cb}\right] \\ &= \left(\left[\mathscr{H}_L, \rho_S\right]\right)_{ab} . \end{aligned} \tag{A.82}$$

Now, the second term on the right hand side of Eqn. A.78 deals with the density matrix when time $t = 0$. It is assumed that at $t = 0$, however, that the atom and reservoir are uncorrelated. Furthermore, the radiation field is in the zero temperature state. This means

$$\begin{aligned} \rho(0) &= \rho_R(0)\rho_S(0) \\ &= |0\rangle \langle 0| \rho_S(0), \end{aligned} \tag{A.83}$$

and because $\mathscr{H}_I$ is purely off-diagonal within the radiation field, this second term is equal to zero.

The last term of Eqn. A.78 requires an assumption to be made, namely

$$\rho(t') \simeq \rho_R(0)\rho_S(t), \tag{A.84}$$

meaning that ρ does not change very much over a small time interval $t' \rightarrow [0, t]$, and thus $\rho_S(0) \simeq \rho_S(t)$. Breaking down the double commutator, the first term inside the integral is

$$\left[\mathscr{H}_I(t), \left[\mathscr{H}_L(t', \rho(t')\right]\right] . \tag{A.85}$$

However, since $\rho_R = |0\rangle \langle 0|$ (from Eqn. A.83), a typical term looks like

$$\langle ra| \mathscr{H}_I \mathscr{H}_L \rho_S \rho_R |rb\rangle = \langle 0a| \mathscr{H}_I \mathscr{H}_L \rho_S |0b\rangle . \tag{A.86}$$

Because neither $\mathscr{H}_L$ nor ρ_S changes the photon number,

$$\langle 0a| \mathscr{H}_I |0n\rangle = 0, \tag{A.87}$$

and thus the first term of the double commutator equals zero. From this same logic, one can simplify this double commutator term further:

$$\langle ra| \mathscr{H}_I(t)\mathscr{H}_I(t')\rho_R\rho_S |rb\rangle = \langle 0a| \mathscr{H}_I(t)\mathscr{H}_I(t')\rho_S |0b\rangle . \tag{A.88}$$

Because ρ_S does not affect the radiation field states, it can be rewritten as

$$\langle 0a| \mathscr{H}_I(t)\mathscr{H}_I(t')\rho_S |0b\rangle = \langle 0a| \mathscr{H}_I(t)\mathscr{H}_I(t') |0c\rangle (\rho_S)_{cb} . \tag{A.89}$$

Furthermore, when the same photon is emitted and then absorbed, no new atomic state can be reached, except the initial one. Thus,

$$\langle 0a| \mathscr{H}_I(t)\mathscr{H}_I(t') |0c\rangle (\rho_S)_{cb} = \langle 0a| \mathscr{H}_I(t)\mathscr{H}_I(t') |0a\rangle (\rho_S)_{ab} . \tag{A.90}$$

Finally, writing the allowed intermediate states between the two interactions yields

$$\langle ra| \mathscr{H}_I(t)\mathscr{H}_I(t')\rho_R\rho_S |rb\rangle = \left(\sum_{\gamma,c} \langle 0a| \mathscr{H}_I(t) |1\gamma c\rangle \langle 1\gamma c| \mathscr{H}_I(t') |0a\rangle \right) (\rho_S)_{ab} , \tag{A.91}$$

$$\langle ra| \rho_R\rho_S\mathscr{H}_I(t')\mathscr{H}_I(t) |rb\rangle = (\rho_S)_{ab} \left(\sum_{\gamma,c} \langle 0b| \mathscr{H}_I(t') |1\gamma c\rangle \langle 1\gamma c| \mathscr{H}_I(t) |0b\rangle \right) . \tag{A.92}$$

Note, here, that if the indices a and b are the same, then Eqn. A.92 is the complex conjugate of Eqn. A.91. The remaining two terms in the double commutator are

$$\langle ra|\,\mathscr{H}_I(t)\rho_R\rho_S\mathscr{H}_I(t')\,|rb\rangle \;=\; \langle ra|\,\mathscr{H}_I(t)\,|0c\rangle\,(\rho_S)_{cd}\,\langle 0d|\,\mathscr{H}_I(t')\,|rb\rangle\,, \tag{A.93}$$

$$\langle ra|\,\mathscr{H}_I(t')\rho_R\rho_S\mathscr{H}_I(t)\,|rb\rangle \;=\; \langle ra|\,\mathscr{H}_I(t')\,|0c\rangle\,(\rho_S)_{cd}\,\langle 0d|\,\mathscr{H}_I(t)\,|rb\rangle\,. \tag{A.94}$$

Once more, things can be simplified by recognizing that the same photon, r, cannot be in both states $r + a \rightarrow c$ and $r + b \rightarrow d$ unless both $a = b$ and $c = d$. This means

$$\sum_\gamma \langle 1\gamma a|\,\mathscr{H}_I(t)\rho_R\rho_S\mathscr{H}_I(t')\,|1\gamma b\rangle \;=\; \delta_{ab}\sum_{\gamma,c}\langle 1\gamma a|\,\mathscr{H}_I(t)\,|0c\rangle\,\langle 0c|\,\mathscr{H}_I(t')\,|1\gamma a\rangle\,(\rho_S)_{cc}\,, \tag{A.95}$$

$$\sum_\gamma \langle 1\gamma a|\,\mathscr{H}_I(t')\rho_R\rho_S\mathscr{H}_I(t)\,|1\gamma b\rangle \;=\; \delta_{ab}\sum_{\gamma,c}\langle 1\gamma a|\,\mathscr{H}_I(t')\,|0c\rangle\,\langle 0c|\,\mathscr{H}_I(t)\,|1\gamma a\rangle\,(\rho_S)_{cc}\,. \tag{A.96}$$

Again, it is of benefit to note that Eqn. A.96 is simply the complex conjugate of Eqn. A.95. The combination of the four equations, A.91, A.92, A.95, and A.96, can be combined into a single relation:

$$\sum_\gamma \int_0^t \langle 0x|\,\mathscr{H}_I(t)\,|1\gamma y\rangle\,\langle 1\gamma y|\,\mathscr{H}_I(t')\,|0x\rangle\,dt' = \langle x|\,V_S\,|y\rangle\,\langle y|\,V_S^\dagger\,|x\rangle\,e^{i(E_x-E_y-\omega)t}\int_0^t e^{i(\omega-E_x+E_y)t'}\,dt', \tag{A.97}$$

where x and y are atomic states. If ω_{ba} is defined to be

$$\omega_{ba} \equiv E_x - E_y, \tag{A.98}$$

then the right-hand side of Eqn. A.97 can be written as

$$\sum_\gamma \int_0^t \langle 0x| \mathscr{H}_I(t) |1\gamma y\rangle \langle 1\gamma y| \mathscr{H}_I(t') |0x\rangle \, dt' = \sum_\gamma |\langle x| V_S |y\rangle|^2 \frac{1 - e^{-i(\omega-\omega_{ba})t}}{i(\omega - \omega_{ba})}. \tag{A.99}$$

By converting the sum over the states into an integral over the density of states,

$$\begin{aligned} \sum_\gamma |\langle x| V_S |y\rangle|^2 \frac{1 - e^{-i\beta t}}{i\beta} &= |\langle x| V_S |y\rangle|^2 \int \rho_\gamma(\omega) \left(\frac{\sin(\beta t) - i\,[1 - \cos(\beta t)]}{\beta} \right) d\omega \\ &= |\langle x| V_S |y\rangle|^2 \pi \rho_\gamma(\omega_{ba}) \\ &= \frac{1}{2} A_{xy} \end{aligned} \tag{A.100}$$

where $\beta \equiv \omega - \omega_{ba}$, ρ_γ is the photon density, A_{xy} is the Einstein-A coefficient describing spontaneous emission from state x to state y, and the imaginary term is ignored because it is the integral of an odd function over an even interval (and thus vanishes).

A.3.3 The Quantum Louisville Equation Revisited

Armed with this information, the attention is returned to Eqn. A.70. Focusing first on the off-diagonal terms where $a \neq b$, Eqns. A.95 and A.96 vanish, leaving

$$\dot{\rho}_{ab}(t) = -i \langle a| [\mathscr{H}_L(t), \rho(t)] |b\rangle - \rho_{ab}(t) \sum_k \frac{1}{2} (A_{ak} + A_{bk}), \tag{A.101}$$

and in a similar approach, the diagonal terms (where $a = b$) reduce to

$$\dot{\rho}_{aa}(t) = -i \langle a| [\mathscr{H}_L(t), \rho(t)] |a\rangle - \sum_k [\rho_{aa}(t) A_{ak} - \rho_{kk}(t) A_{ka}]. \tag{A.102}$$

From Eqn. A.101, one can see that the radiative decay rates destroy coherence between states expressed in the off-diagonal density matrix elements. Indeed, these off-diagonal elements are termed "coherences", and describe the coherence and decoherence effects coupling the states. Equation A.102 shows that population decay from a higher state feeds population into lower states, as is expected.

Finally, using Eqns. A.101 and A.102, one can write the quantum Louisville equation (from Eqn. A.70) to include the decay rates of a system. It should be evident that, for diagonal elements, the second term of Eqn. A.101 is equal to the second term of Eqn. A.102. That is, for $a = b$,

$$\rho_{ab}(t) \sum_k \frac{1}{2} \left(A_{ak} + A_{bk}\right) = \sum_k \rho_{aa}(t) A_{ak}. \tag{A.103}$$

A single expression can be written, to account for both the diagonal and off-diagonal decay terms, namely

$$\left[\Gamma \rho(t)\right]_{ab} = \rho_{ab} \sum_k \frac{1}{2} \left(A_{ak} + A_{bk}\right) - \delta_{ab} \sum_k \rho_{kk} A_{ka}. \tag{A.104}$$

This allows one to modify the quantum Louisville equation to include the spontaneous decay rates as follows

$$i\hbar \dot{\rho}_{ab}(t) = \left[\mathscr{H}_L, \rho(t)\right]_{ab} - i\hbar \left[\Gamma \rho(t)\right]_{ab}. \tag{A.105}$$

It is useful to first assume the system is in two-photon resonance; that is, $\Delta_2 = 0$, and study the eigenvectors for the three-level case as done previously in the dressed-state approach for the two-level case. The angular relations are slightly different in definition than in Fig. A.5, and are depicted in Fig. A.7. Ω' is still defined to be the Rabi flopping frequency (see Fig. A.7a), however, note that Δ_1 represents the single-photon detuning, and the angle involved is now labeled ϕ, while the angle θ now corresponds to the "two-photon Rabi frequency", as shown in Fig. A.7b. ϕ is known as the "mixing angle", relating the component of the ground state mixed with the components of the excited states. This

change of notation, moving from the 2-level to the 3-level treatment, will be used for the remainder of the dissertation, and is consistent with the literature.

With the same dressed-state treatment as in Sec. A.2.4, one obtains the normalized eigenvalues of the 3-level system to be

$$\lambda_- = -\frac{\hbar}{2}\Omega\tan(\phi) \quad \text{(A.106)}$$

$$\lambda_0 = 0 \quad \text{(A.107)}$$

$$\lambda_+ = -\frac{\hbar}{2}\Omega\cot(\phi), \quad \text{(A.108)}$$

and the corresponding eigenvectors are

$$\Phi_- = \begin{pmatrix} \sin\theta\cos\phi \\ -\sin\phi \\ \cos\theta\cos\phi \end{pmatrix} \quad \text{(A.109)}$$

$$\Phi_0 = \begin{pmatrix} -\cos\theta \\ 0 \\ \sin\theta \end{pmatrix} \quad \text{(A.110)}$$

$$\Phi_+ = \begin{pmatrix} \sin\theta\sin\phi \\ \cos\phi \\ \cos\theta\sin\phi \end{pmatrix}. \quad \text{(A.111)}$$

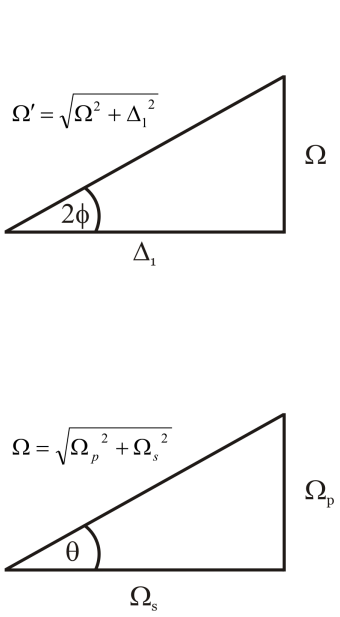

Figure A.7: Visual interpretation of the angles ϕ and θ in the 3-level dressed-state approach.

If the system state vector is represented as

$$\Psi(t) = A_-(t)\Phi_-(t) + A_0(t)\Phi_0(t) + A_+(t)\Phi_+(t), \quad \text{(A.112)}$$

then

$$\dot{\mathbf{A}} = -\frac{i}{\hbar}\begin{pmatrix} \lambda_- & -i\hbar\dot{\theta}\cos\phi & -i\hbar\dot{\phi} \\ i\hbar\dot{\theta}\cos\phi & \lambda_0 & i\hbar\dot{\theta}\sin\phi \\ i\hbar\dot{\phi} & -i\hbar\dot{\theta}\sin\phi & \lambda_+ \end{pmatrix}\mathbf{A} \quad \text{(A.113)}$$

Obtaining the 3-level Hamiltonian, $\mathscr{H}_L$, is done the same way as in the 2-level case (Eqn. A.36) and results in

$$\mathscr{H}_L = \frac{\hbar}{2}\begin{pmatrix} 0 & \Omega_p & 0 \\ \Omega_p^* & 2\Delta_1 & \Omega_s \\ 0 & \Omega_s^* & 2\Delta_2 \end{pmatrix} \tag{A.114}$$

where Ω_p and Ω_s are defined to be the "pump" and "Stokes" Rabi frequencies for one- and two-photon resonances, respectively. Likewise, the detunings from such one- and two-photon resonant cases are Δ_1 and Δ_2, respectively, as shown in Fig. A.8. The notation here is different than shown previously for the two-level case, conforming to the three-level nomenclature used throughout the literature. Placing this 3-level Hamiltonian in the quantum Louisville equation, one now has a complete description of the population dynamics for the system, including spontaneous emission losses from excited states.

This important difference from the previous two- and three-level cases shown allows one to gain a more complete picture of the dynamics taking place during transitions within the system. As mentioned previously (see Sec. A.3.1), the density matrix produces $n^2 - 1$ coupled, first-order differential equations to describe such population dynamics within each level of the system, along with the coherence coupling between states. Therefore, the 3×3 matrix for the 3-level case produces the 8 coupled, first-order differential equations from Eqn. A.105 to be solved.

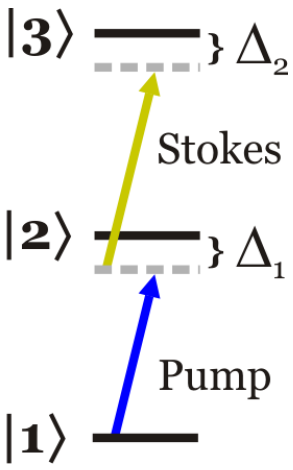

Figure A.8: A generic energy level diagram showing a 3-level system.

Appendix B

Electronic Circuit Diagrams

Most of the in-house electronic circuitry used for construction or control of the MOT and associated lasers has been published elsewhere.[137] Two new circuits, however, have been designed since this publication, and are included in this appendix. They are:

- **Peak Locking**

  Electronic diagram showing the latest peak locking circuitry.

- **AOM Controller**

  The AOM controller circuit diagram is shown, along with the external control hardware associated with the AOM.

B.1 Peak Locking

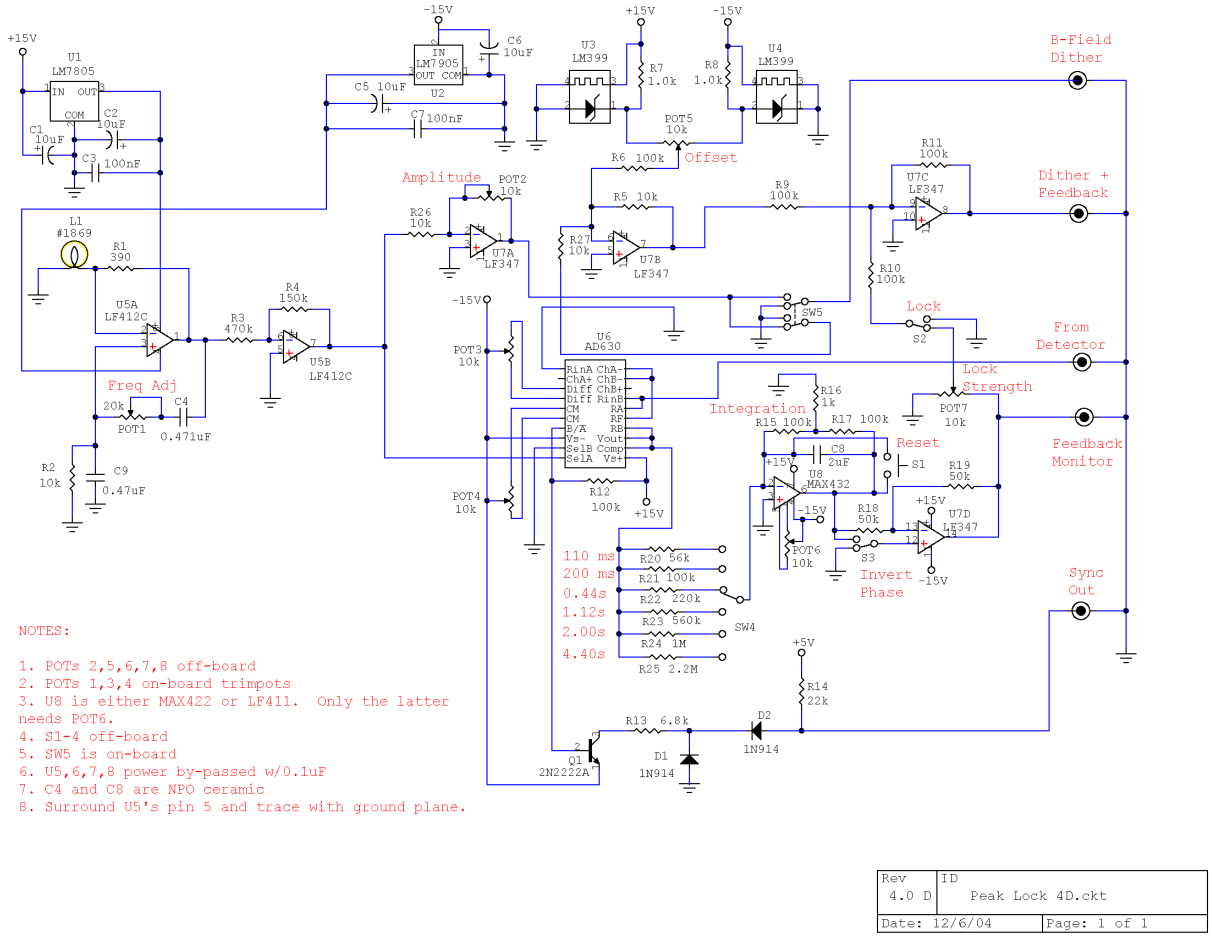

Figure B.1: Peak locking circuit diagram.

B.2 AOM Controller

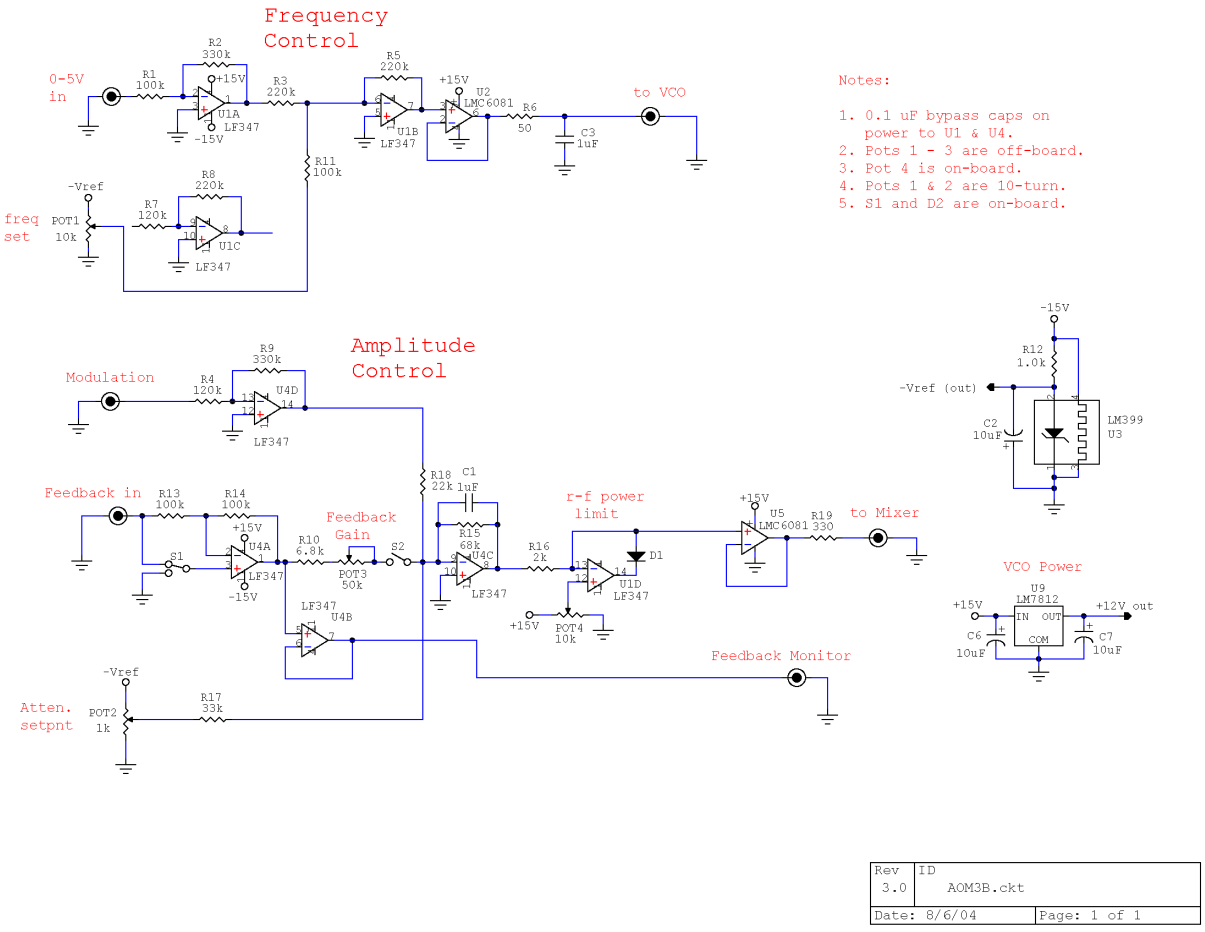

Figure B.2: AOM control circuit diagram.

The associated AOM hardware allows one to control the amplitude and detuned frequency of the diffracted light passing through the AOM, and includes an RF switch to rapidly turn the AOM on or off. The incoming RF control signal originates from the AWG software, which operates each AOM individually in this manner. The control box houses the circuit described in Fig. B.3, which provides a quasi-DC voltage to the mixer, and supplies an operating voltage to the voltage control oscillator (VCO). The frequency control signal supplied by the control box is converted by the VCO into an oscillating voltage, which in turn is fed into the mixer. The mixer takes the VCO output and appropriately attenuates

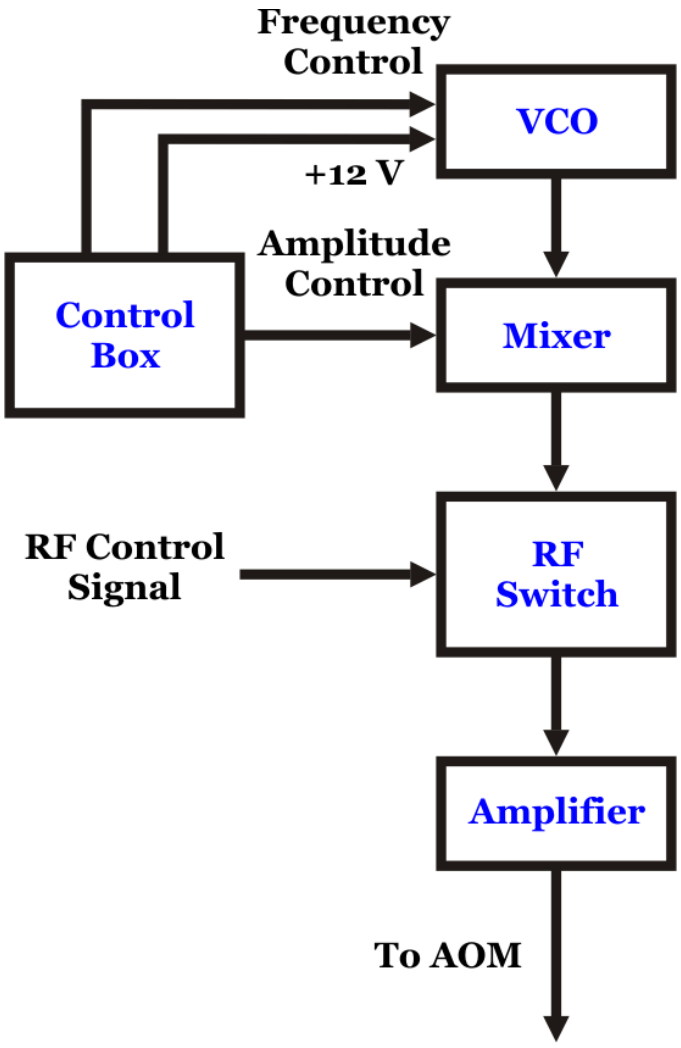

Figure B.3: Experimental setup for AOM control.

it depending on the value of the quasi-DC amplitude signal received from the control box. In this manner, the amplitude and frequency fed into the AOM are independently set by the controller box. The RF switch allows one to rapidly turn the signal on and off, while the amplifier places the signal in the proper operating range for the AOM.

Appendix C

Acquisition Program

XSYS uses a set of files for data acquisition. A series of header files control the interface with the CAMAC hardware, while the main event language (EVL) file delineates how data are to be handled after receiving the appropriate data streams. This appendix describes each of the relevant files necessary for data acquisition, and a copy of each file is also included.

C.1 XSYS Data Acquisition Program

The data taken for this thesis was exclusively written to the hard disk, DEPAOLA4. The disk structure is radically different than previous hard disks used by the MOTRIMS research group, and so a brief overview of how the files are organized is presented here. The data files themselves have been archived onto CD-ROM discs as well.

C.1.1 Disk Structure

DEPAOLA4 contains several subdirectories, following the historical convention of creating a new directory when the main .EVL file is altered significantly. In this manner, a complete set of acquisition files should remain intact (either on the DEPAOLA4 disk itself, or archived on CD-ROM) and any of the old data sets can be replayed with the appropriate acquisition files. The STIRAP3 subdirectory was used almost exclusively for the data presented in this dissertation.

The acquisition files are broken down and stored in various subdirectories, as shown

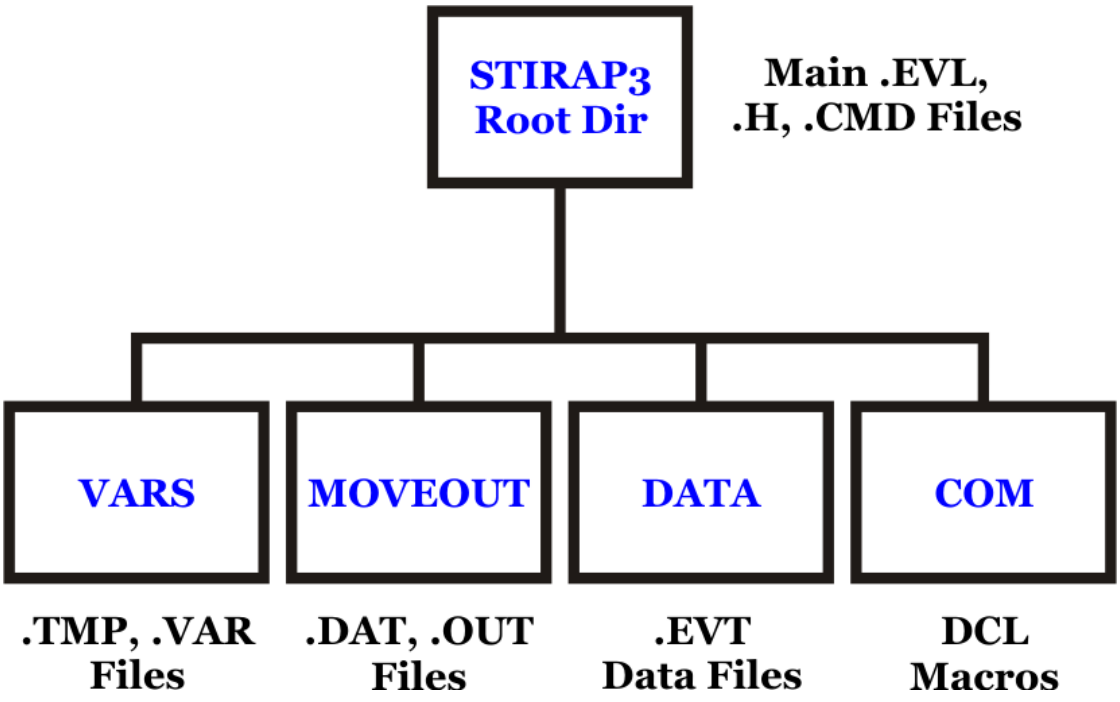

Figure C.1: Disk structure for the DEPAOLA4 hard disk.

in Fig. C.1. The actual event (EVT) data files are stored in the [.data] directory. Cuts, projections, or other data stored to disk using the XSYS `MOVE` command are located in the [.moveout] directory. All macros are located in the [.com] directory, with the exception of those that are run directly from the command line. The [.vars] directory contains temporary files and stored information regarding the state of variables stored during a given run.

C.1.2 Main Acquisition Program

The main data acquisition event language file is entitled STIRAP3.EVL. The file's main function is to collect data from a single TDC and three ADC CAMAC electronics modules. The signals are reconstructed as projectile and recoil events, and are plotted in 2D memory areas as they would appear on the 2D position-sensitive detectors (PSD) used for collecting data. Also, the time-of-flight (TOF) difference recorded by the TDC is plotted, showing a real-time Q-Value spectra as data are collected. The 2D TAC_1 and TAC_2 signals, showing the correlation between charge-transfer events and the laser period are also plotted.

C.1.3 STIRAP3.EVL

The main event language file controls the processing of the signals acquired by the electronics. The file used for data shown here is broken down into 5 parts dealing with projectile detector signals, recoil detector signals, TAC signals, TDC signals, and a separate event structure for incrementing counters and timers (labeled as event 50).

```
! +-------------------------------------------------------------+
! | STIRAP3.EVL                                                 |
! | Last modified on 06/22/04 by hcamp                          |
! +-------------------------------------------------------------+
!
! +-------------------------------------------------------------+
! |       EVL FILE FOR Projectile-Recoil coincidence studies    |
! |       Code for use with Ortec AD811 and TDC                 |
! |       Germanium WSA for Projectile                          |
! |       Resistive anode for Recoil ion                        |
! +-------------------------------------------------------------+
!
! +-------------------------------------------------------------+
! | The general layout of this file is as follows:              |
! |                                                             |
! | * Define Variables                                          |
! | * Begin Sort Process                                        |
! | * Check Special Condition: Event Buffer                     |
! | * Sort Time-Of-Flight                                       |
! | * Sort Recoil                                               |
! | * Sort Projectile                                           |
! | * Sort TAC-related signals                                  |
! +-------------------------------------------------------------+
```

```
                              ! +-----------------------------------+
OPTION TAPE                   ! | Event records which have no       |
                              ! | explicit EVAL EVENT statement     |
                              ! | declaration will automatically be |
                              ! | put into the output buffers as they|
                              ! | are encountered.  (See p. 24 of the|
                              ! | XSYS Manual.)                     |
                              ! +-----------------------------------+

! +-------------------------------------------------------------+
! | GENERAL PURPOSE PARAMETERS                                  |
! +-------------------------------------------------------------+

                              ! +-----------------------------------+
INTEGER        I1 = 1         ! | Integer 1                         |
INTEGER      I255 = 255       ! | Integer 255                       |
INTEGER      I511 = 511       ! | Integer 511                       |
INTEGER     I1023 = 1023      ! | Integer 1023                      |
INTEGER     I2047 = 2047      ! | Integer 2047                      |
INTEGER      Seed = 8723645   ! | Random number seed                |
INTEGER      INEG = -1        ! | Negative Integer -1               |
INTEGER     USEBF = 0         ! | Buffer Control: 1 = On, 0 = Off   |
INTEGER   STARTBF = 0         ! | Start Buffer Number  (when USEBF=1)|
INTEGER    STOPBF = 0         ! | Stop Buffer Number   (when USEBF=1)|
INTEGER    CURRBF = 0         ! | Current Buffer Number(when USEBF=1)|
INTEGER   STAC1D2             ! | STAC1 divided by 2                |
INTEGER   STAC2D2             ! | STAC2 divided by 2                |
   REAL  dChannel             ! |                                   |
   REAL        R2 = 2.0       ! | Real number 2.0                   |
   REAL      temp             ! | Temporary variable                |
                              ! +-----------------------------------+
```

```
! +-------------------------------------------------------------+
! | Recoil position parameters                                  |
! +-------------------------------------------------------------+

                          ! +-----------------------------------+
   REAL   RX1sig          ! | Recoil X1 signal                  |
   REAL   RX2sig          ! | Recoil X2 signal                  |
   REAL   RY1sig          ! | Recoil Y1 signal                  |
   REAL   RY2sig          ! | Recoil Y2 signal                  |
   REAL    RSsig          ! | Recoil SUM X1 + X2 + Y1 + Y2      |
   REAL  RX1thre = 25     ! | Recoil X1 threshold               |
   REAL  RX2thre = 25     ! | Recoil X2 threshold               |
   REAL  RY1thre = 25     ! | Recoil Y1 threshold               |
   REAL  RY2thre = 25     ! | Recoil Y2 threshold               |
   REAL   RSthre = 150    ! | Recoil SUM theshold               |
   REAL RX1offse = 11.3   ! | Recoil X1 offset                  |
   REAL RX2offse = 9.4    ! | Recoil X2 offset                  |
   REAL RY1offse = 11.8   ! | Recoil Y1 offset                  |
   REAL RY2offse = 10.5   ! | Recoil Y2 offset                  |
   REAL  RX1gain = 0.9961 ! | Recoil X1 gain                    |
   REAL  RX2gain = 0.9930 ! | Recoil X2 gain                    |
   REAL  RY1gain = 0.9582 ! | Recoil Y1 gain                    |
   REAL  RY2gain = 0.9987 ! | Recoil Y2 gain                    |
   REAL   RXipos          ! | Recoil X intermediate position    |
   REAL   RYipos          ! | Recoil Y intermediate position    |
   REAL  RcosAng = 1.0    ! | Recoil PSD rotation angle cosine  |
   REAL  RsinAng = 0.0    ! | Recoil PSD rotation angle sine    |
INTEGER    RXpos          ! | Recoil X position                 |
INTEGER    RYpos          ! | Recoil Y position                 |
INTEGER   RXpos0 = 290    ! | Recoil X initial position         |
INTEGER   RYpos0 = 230    ! | Recoil Y initial position         |
INTEGER    RXmin = 300    ! | Recoil x minimum range gate       |
INTEGER    RXmax = 360    ! | Recoil x maximum range gate       |
INTEGER    RYmin = 280    ! | Recoil y minimum range gate       |
INTEGER    RYmax = 350    ! | Recoil y maximum range gate       |
INTEGER    DXmin =   0    ! | D Fraction x minimum range gate   |
INTEGER    DXmax = 255    ! | D Fraction x maximum range gate   |
INTEGER    DYmin =   0    ! | D Fraction y minimum range gate   |
INTEGER    DYmax = 511    ! | D Fraction y maximum range gate   |
                          ! +-----------------------------------+
```

```
! +-------------------------------------------------------------+
! | Projectile position parameters                              |
! +-------------------------------------------------------------+

                          ! +-----------------------------------+
   REAL    PXsig          ! | Projectile X signal               |
   REAL    PYsig          ! | Projectile Y signal               |
   REAL    PRsig          ! | Projectile R signal               |
   REAL    PSsig          ! | Projectile SUM X + Y + R          |
   REAL   PXthre = 50     ! | Projectile X threshold            |
   REAL   PYthre = 50     ! | Projectile Y threshold            |
   REAL   PRthre = 50     ! | Projectile R threshold            |
   REAL   PSthre = 140    ! | Projectile SUM threshold          |
   REAL  PXoffse = 0      ! | Projectile X offset               |
   REAL  PYoffse = 0      ! | Projectile Y offset               |
   REAL  PRoffse = 0      ! | Projectile R offset               |
   REAL   PXgain = 1.0    ! | Projectile X gain                 |
   REAL   PYgain = 1.0    ! | Projectile X gain                 |
   REAL   PRgain = 1.0    ! | Projectile X gain                 |
   REAL    PRamp = 2.55   ! | Projectile R factor (1.46*1.75)   |
   REAL PDetScal = 5      ! | Scaling  for Proj. Detector size  |
   REAL   PSHIFT = 68.    ! | Shift of Proj. detector image     |
   REAL   PXipos          ! | Projectile x intermediate position |
   REAL   PYipos          ! | Projectile y intermediate position |
   REAL  PcosAng = 1.0    ! | Projectile PSD rotation angle cose |
   REAL  PsinAng = 0.0    ! | Projectile PSD rotation angle sine |
INTEGER    PXpos          ! | Projectile x position             |
INTEGER    PYpos          ! | Projectile y position             |
INTEGER    PXmin = 140    ! | Projectile x minimum range gate   |
INTEGER    PXmax = 340    ! | Projectile x maximum range gate   |
INTEGER    PYmin = 260    ! | Projectile y minimum range gate   |
INTEGER    PYmax = 400    ! | Projectile y maximum range gate   |
                          ! +-----------------------------------+
```

```
! +-------------------------------------------------------------+
! | Time-Of-Flight parameters                                   |
! +-------------------------------------------------------------+

                          ! +-----------------------------------+
INTEGER    TIsig          ! | TDC signal                        |
INTEGER     TNUM          ! | Number of multihits               |
   REAL     Tsig          ! | TDC signal                        |
   REAL    Tmin1 = 75000.0 ! | TSpec1 Time window minimum        |
   REAL    Tmax1 = 85000.0 ! | TSpec1 Time window maximum        |
   REAL    Tmin2 = 7000.0  ! | Time window min for refine sorting |
   REAL    Tmax2 = 7255.0  ! | Time window max for refine sorting |
   REAL  TRXfact = 0.21    ! | Time Xpos stretch factor          |
   REAL  TRYfact = -0.046  ! | Time Ypos stretch factor          |
   REAL   Tdrift = 0       ! | Time shift to correct drift       |
                          ! +-----------------------------------+

! +-------------------------------------------------------------+
! | TAC parameters                                              |
! +-------------------------------------------------------------+

                          ! +-----------------------------------+
INTEGER   Tacmin = 100     ! | Minimum TAC value allowed         |
INTEGER   Tacmax = 200     ! | Maximum TAC value allowed         |
                          ! +-----------------------------------+

! +-------------------------------------------------------------+
! | PeakSum & Gate parameters                                   |
! +-------------------------------------------------------------+

                          ! +-----------------------------------+
INTEGER    GType = 0       ! | Gate Type (0=TOFFULL, 1=TOFPEAK)  |
INTEGER     GMin = 130     ! | Peak Gate Minimum Value           |
INTEGER     GMax = 135     ! | Peak Gate Maximum Value           |
                          ! |                                   |
INTEGER   MAXCNT = -1      ! | Max Counts in Peak (or Full) Gate. |
                          ! | When this number is reached, the  |
                          ! | run will automatically be         |
                          ! | terminated.  If this number is <=0 |
                          ! | then the run will continue        |
                          ! | indefinitely.                     |
                          ! |                                   |
```

```
INTEGER  MAXTIME = -1      ! | Maximum time to run.  When this    |
                           ! | number is reached, the run will    |
                           ! | automatically be terminated.  If   |
                           ! | this number is <=0 then the run    |
                           ! | will continue indefinitely.  The   |
                           ! | run time is checked in 10 second   |
                           ! | intervals.                         |
                           ! |                                    |
INTEGER  PEAKSUM           ! | Sum of all counts within Peak (or  |
                           ! | Full) gate.  When PEAKSUM reaches  |
                           ! | MAXCNT, the run will automatically |
                           ! | be terminated.                     |
                           ! |                                    |
GATE TOFPEAK 5 1           ! | Gate 1 is the small peak gate set  |
                           ! | around the s-p peak.               |
                           ! |                                    |
GATE TOFFULL 5 2           ! | Gate 2 is the full range of        |
                           ! | channels in the TSpec2 window      |
                           ! | (1-255).                           |
                           ! +------------------------------------+

! +-------------------------------------------------------------+
! | Count Rate parameters                                       |
! +-------------------------------------------------------------+

                           ! +------------------------------------+
INTEGER     ISum = 0       ! | Sum of Counts in 10s time interval.|
INTEGER    ILAST = 0       ! | Prior sum in 10s time interval.    |
INTEGER     Chan = 0       ! | Current channel # of CNTRATE.      |
INTEGER   MOTVal = 0       ! | Note this can be used to determine |
INTEGER   IONVal = 0       ! | time elapsed during a given run    |
INTEGER     DVal = 0       ! | (within 10 sec.)                   |
                           ! +------------------------------------+
```

```
! +-------------------------------------------------------------+
! |                          SORTING                            |
! +-------------------------------------------------------------+

                          ! +-----------------------------------+
OPTION ALLSPEC            ! | Causes all data areas belonging to |
                          ! | the SPEC and GATE classes to be    |
                          ! | included in the compiler symbol    |
                          ! | table.  In other words, this       |
                          ! | defines the spectra and gate       |
                          ! | symbols so the .evl file knows what|
                          ! | they are.                          |
                          ! +-----------------------------------+

                          ! +-----------------------------------+
                          ! | Note that the real data are 11 bits|
                          ! | long.  Here, the 12th bit is       |
                          ! | included for overflow detection.   |
                          ! +-----------------------------------+

                          ! +-----------------------------------+
                          ! | Format defines a format called     |
                          ! | <NAME> which is used to unpack an  |
                          ! | event data word from the data      |
                          ! | buffer.  In other words, this is   |
                          ! | the interface to assign a          |
                          ! | particular part of the data buffer |
                          ! | to a particular format type.       |
                          ! +-------------+                      |
FORMAT ST1       1 24 1   $LONG $SIGNED ! | TDC Timing Signal    |
FORMAT STAC2     3 12 3   ! +-------------+ TAC2                 |
FORMAT SFLUO     4 12 1   ! | Flourescence (not currently used)  |
FORMAT SDETUN    5 12 1   ! | Detuning (not currently used)      |
FORMAT STAC1     6 12 3   ! | TAC1                               |
FORMAT SRX1      7 12 1   ! | Recoil Det. (X1 - Top North)       |
FORMAT SRX2      8 12 1   ! | Recoil Det. (X2 - Bottom South)    |
FORMAT SRY1      9 12 1   ! | Recoil Det. (Y1 - Bottom North)    |
FORMAT SRY2     10 12 1   ! | Recoil Det. (Y2 - Top South)       |
```

```
FORMAT MOTSIG   11 12 1    ! | MOT Signal                          |
FORMAT IONSIG   12 12 1    ! | Ion Beam Signal                      |
FORMAT SPY      13 12 1    ! | Projectile Detector (Wedge)          |
FORMAT SPX      14 12 1    ! | Projectile Detector (Strip)          |
FORMAT SPR      15 12 1    ! | Projectile Detector (Remainder)      |
                           ! +--------------------------------------+
! +-------------------------------------------------------------+
! |                      <<< Begin Event >>>                    |
! +-------------------------------------------------------------+

                           ! +--------------------------------------+
EVENT BEGIN                ! | Initializes beginning of an event.   |
TAPE                       ! | Starts recording to disk.            |
                           ! +--------------------------------------+

                           ! +--------------------------------------+
SET  PEAKSUM = 0           ! | Clear the  PEAKSUM variable          |
SET    ILAST = 0           ! | Clear the    ILAST variable          |
SET     ISUM = 0           ! | Clear the     ISUM variable          |
SET     CHAN = 0           ! | Clear the     CHAN variable          |
                           ! +--------------------------------------+

! +-------------------------------------------------------------+
! |                Special Condition: Event Buffer              |
! +-------------------------------------------------------------+

                           ! +--------------------------------------+
                           ! | The following block of code checks   |
                           ! | to see if USEBF is enabled.  If      |
                           ! | USEBF is 1, then the run will        |
                           ! | ignore any buffers that lie outside  |
                           ! | the STARTBF and STOPBF range.        |
                           ! | This allows one to reread only       |
                           ! | certain portions of an .evt file.    |
                           ! +--------------------------------------+
```

```
EVENT BUFFER

IF USEBF = 1

   INC CURRBF

   IF CURRBF < STARTBF

      EXIT BUFFER

   ELSEIF CURRBF >= STOPBF

      HALT

   ENDIF

ENDIF

! +-------------------------------------------------------------+
! |          Data Event 2 (Main Data event) takes place         |
! +-------------------------------------------------------------+

                            ! +-----------------------------------+
EVENT 2                     ! | This is the incoming data event.  |
                            ! +-----------------------------------+

! +-------------------------------------------------------------+
! |                  Dealing with Time of flight                |
! +-------------------------------------------------------------+

                            ! +-----------------------------------+
RAN (Seed)                  ! | Randomize to prevent beats        |
STA dChannel                ! |                                   |
GET ST1                     ! | Get TDC timing signal             |
FLOAT                       ! |                                   |
ADD dChannel                ! |                                   |
STA Tsig                    ! | Store as real variable            |
TAPE                        ! |                                   |
IF LE Tmin1 EXIT            ! | Exit if TOF is too small          |
IF GT Tmax1 EXIT            ! | EXIT if TOF is too large          |
                            ! |                                   |
```

```
LDA Tsig                  ! |                                      |
SUB Tmin1                 ! |                                      |
STA Tsig                  ! |                                      |
!IF GE 65535.0 exit       ! |                                      |
IF GE 16384.0 exit        ! |                                      |
FIX                       ! |                                      |
                          ! |                                      |
TINC Tspec                ! | Increment full Time spectrum         |
                          ! +--------------------------------------+

! +-------------------------------------------------------------------+
! |                        Recoil Position                            |
! +-------------------------------------------------------------------+

                          ! +--------------------------------------+
RAN (Seed)                ! | Randomize to prevent beats           |
STA dChannel              ! |                                      |
GET SRX1                  ! | Get X1 Recoil PSD signal             |
FLOAT                     ! |                                      |
ADD dChannel              ! |                                      |
SUB RX1offse              ! |                                      |
MUL RX1gain               ! |                                      |
IF GE I2047 EXIT          ! | Exit if it is an overflow            |
IF LE RX1thre EXIT        ! | Exit if less than lower threshold    |
STA RX1sig                ! | Store as real variable               |
FIX                       ! |                                      |
                          ! |                                      |
TINC RX1spec              ! | Increment RX1 spectrum               |
                          ! |                                      |
RAN (Seed)                ! | Randomize to prevent beats           |
STA dChannel              ! |                                      |
GET SRX2                  ! | Get X2 Recoil PSD signal             |
FLOAT                     ! |                                      |
ADD dChannel              ! |                                      |
SUB RX2offse              ! |                                      |
MUL RX2gain               ! |                                      |
IF GE I2047 EXIT          ! | Exit if it is an overflow            |
IF LE RX2thre EXIT        ! | Exit if less than lower threshold    |
STA RX2sig                ! | Store as real variable               |
FIX                       ! |                                      |
                          ! |                                      |
```

```
TINC RX2spec              ! | Increment RX2 spectrum             |
                          ! |                                    |
RAN (Seed)                ! | Randomize to prevent beats         |
STA dChannel              ! |                                    |
GET SRY1                  ! | Get Y1 Recoil PSD signal           |
FLOAT                     ! |                                    |
ADD dChannel              ! |                                    |
SUB RY1offse              ! |                                    |
MUL RY1gain               ! |                                    |
IF GE I2047 EXIT          ! | Exit if it is an overflow          |
IF LE RY1thre EXIT        ! | Exit if less than lower threshold  |
STA RY1sig                ! | Store as real variable             |
FIX                       ! |                                    |
                          ! |                                    |
TINC RY1spec              ! | Increment RY1 spectrum             |
                          ! |                                    |
RAN (Seed)                ! | Randomize to prevent beats         |
STA dChannel              ! |                                    |
GET SRY2                  ! | Get Y2 Recoil PSD signal           |
FLOAT                     ! |                                    |
ADD dChannel              ! |                                    |
SUB RY2offse              ! |                                    |
MUL RY2gain               ! |                                    |
IF GE I2047 EXIT          ! | Exit if it is an overflow          |
IF LE RY2thre EXIT        ! | Exit if less than lower threshold  |
STA RY2sig                ! | Store as real variable             |
FIX                       ! |                                    |
                          ! |                                    |
TINC RY2spec              ! | Increment  RY2 spectrum            |
                          ! |                                    |
LDA RX1sig                ! |                                    |
ADD RX2sig                ! |                                    |
ADD RY1sig                ! |                                    |
ADD RY2sig                ! |                                    |
IF LE RSthre EXIT         ! | Exit if less than lower threshold  |
STA RSsig                 ! | Recoil PSD charge signal           |
FIX                       ! |                                    |
                          ! |                                    |
TINC RSspec               ! | Increment RS spectrum              |
                          ! +------------------------------------+
```

```
                        ! +-----------------------------------+
                        ! | Position Calculation:             |
                        ! +-----------------------------------+

                        ! +-----------------------------------+
IF RSsig EQ 0. EXIT     ! | Prevent devide by zero            |
                        ! |                                   |
LDA RY2sig              ! |                                   |
ADD RX1sig              ! |                                   |
DIV RSsig               ! |                                   |
MUL I511                ! |                                   |
SUB I255                ! |                                   |
STA RYipos              ! | Store for 11 bit position later   |
                        ! |                                   |
LDA RY1sig              ! |                                   |
ADD RX1sig              ! |                                   |
DIV RSsig               ! |                                   |
MUL I511                ! |                                   |
SUB I255                ! |                                   |
STA RXipos              ! | Store for 11 bit position later   |
                        ! +-----------------------------------+

                        ! +-----------------------------------+
                        ! | Rotation Calculation:             |
                        ! +-----------------------------------+

                        ! +-----------------------------------+
MUL Rsinang             ! |                                   |
STA temp                ! |                                   |
LDA RXipos              ! |                                   |
MUL Rcosang             ! |                                   |
SUB temp                ! |                                   |
ADD I255                ! |                                   |
IF LT RXmin EXIT        ! | Exit if < lower X recoil gate     |
IF GT RXmax EXIT        ! | Exit if > upper X recoil gate     |
STA RXpos               ! | Recoil X coordinate               |
                        ! |                                   |
LDA RXipos              ! |                                   |
MUL Rsinang             ! |                                   |
STA temp                ! |                                   |
```

```
LDA RYipos                 ! |                                      |
MUL Rcosang                ! |                                      |
ADD temp                   ! |                                      |
ADD I255                   ! |                                      |
IF LT RYmin EXIT           ! | Exit if < lower Y recoil gate        |
IF GT RYmax EXIT           ! | Exit if > upper Y recoil gate        |
STA RYpos                  ! | Recoil Y coordinate                  |
                           ! |                                      |
TINC RXpos RYpos RXYspec   ! | Increment Recoil 2D position spec.   |
                           ! +--------------------------------------+

! +-------------------------------------------------------------------+
! |                      Projectile Position                          |
! +-------------------------------------------------------------------+

                           ! +--------------------------------------+
RAN (Seed)                 ! | Randomize to prevent beats           |
STA dChannel               ! |                                      |
GET SPX                    ! | Get X Projectile PSD signal          |
FLOAT                      ! |                                      |
ADD dChannel               ! |                                      |
SUB PXoffse                ! |                                      |
MUL PXgain                 ! |                                      |
IF GE I2047 EXIT           ! | Exit if it is an overflow            |
IF LE PXthre EXIT          ! | Exit if less than lower threshold    |
STA PXsig                  ! | Store as real variable               |
FIX                        ! |                                      |
                           ! |                                      |
TINC PXspec                ! | Increment PX spectrum                |
                           ! |                                      |
RAN (Seed)                 ! | Randomize to prevent beats           |
STA dChannel               ! |                                      |
GET SPY                    ! | Get Y Projectile PSD signal          |
FLOAT                      ! |                                      |
ADD dChannel               ! |                                      |
SUB PYoffse                ! |                                      |
MUL PYgain                 ! |                                      |
IF GE I2047 EXIT           ! | Exit if it is an overflow            |
IF LE PYthre EXIT          ! | Exit if less than lower threshold    |
STA PYsig                  ! | Store as real variable               |
FIX                        ! |                                      |
                           ! |                                      |
```

```
TINC PYspec               ! | Increment PY spectrum              |
                          ! |                                     |
RAN (Seed)                ! | Randomize to prevent beats          |
STA dChannel              ! |                                     |
GET SPR                   ! | Get R Projectile PSD signal         |
FLOAT                     ! |                                     |
ADD dChannel              ! |                                     |
SUB PRoffse               ! |                                     |
MUL PRgain                ! |                                     |
IF GE I2047 EXIT          ! | Exit if it is an overflow           |
IF LE PRthre EXIT         ! | Exit if less than lower threshold   |
STA PRsig                 ! | Store as real variable              |
FIX                       ! |                                     |
                          ! |                                     |
TINC PRspec               ! | Increment PR spectrum               |
                          ! |                                     |
LDA PRsig                 ! |                                     |
MUL PRamp                 ! |                                     |
ADD PXsig                 ! |                                     |
ADD PYsig                 ! |                                     |
IF LE PSthre EXIT         ! | Exit if less than lower threshold   |
STA PSsig                 ! | Recoil PSD charge signal            |
FIX                       ! |                                     |
                          ! |                                     |
TINC PSspec               ! | Increment PS spectrum               |
                          ! +-------------------------------------+

                          ! +-------------------------------------+
                          ! | Position Calculation:               |
                          ! +-------------------------------------+

                          ! +-------------------------------------+
                          ! |                                     |
IF PSsig EQ 0. EXIT       ! | prevent devide by zero              |
                          ! |                                     |
LDA PXsig                 ! |                                     |
DIV PSsig                 ! |                                     |
MUL I511                  ! |                                     |
SUB PSHIFT                ! | Shift detector image                |
MUL PDetScal              ! | Detector scaling factor             |
```

```
SUB I255                  ! |                                      |
STA PXipos                ! | Store for 11 bit position later      |
                          ! |                                      |
LDA PYsig                 ! |                                      |
DIV PSsig                 ! |                                      |
MUL I511                  ! |                                      |
SUB PSHIFT                ! | Shift detector image                 |
MUL PDetScal              ! | Detector scaling factor              |
SUB I255                  ! |                                      |
STA PYipos                ! | Store for 11 bit position later      |
                          ! +--------------------------------------+

                          ! +--------------------------------------+
                          ! | Rotation Calculation:                |
                          ! +--------------------------------------+

                          ! +--------------------------------------+
MUL Psinang               ! |                                      |
STA temp                  ! |                                      |
LDA PXipos                ! |                                      |
MUL Pcosang               ! |                                      |
SUB temp                  ! |                                      |
ADD I255                  ! |                                      |
STA PXpos                 ! | Projectile X coordinate              |
                          ! |                                      |
LDA PXipos                ! |                                      |
MUL Psinang               ! |                                      |
STA temp                  ! |                                      |
LDA PYipos                ! |                                      |
MUL Pcosang               ! |                                      |
ADD temp                  ! |                                      |
ADD I255                  ! |                                      |
STA PYpos                 ! | Projectile Y coordinate              |
                          ! |                                      |
TINC PXpos PYpos PXYspec  ! | Increment Proj. 2D position spec.    |
                          ! |                                      |
                          ! +--------------------------------------+
```

```
! +-------------------------------------------------------------+
! |                          TAC Positions                      |
! +-------------------------------------------------------------+

                           ! +-----------------------------------+
LDA Tsig                   ! |                                   |
IF GT Tmax2 EXIT           ! | Exit if not in time window        |
IF LE Tmin2 EXIT           ! | Exit if not in time window        |
SUB Tmin2                  ! |                                   |
SUB Tdrift                 ! |                                   |
STA Tsig                   ! |                                   |
                           ! |                                   |
TINC PXpos RXpos PXRXspec  ! | Increment Recoil/proj.X pos. spec.|
                           ! |                                   |
TINC PYpos RYpos PYRYspec  ! | Increment Recoil/proj.Y pos. spec.|
                           ! |                                   |
TINC Tsig RXpos TRXspec    ! | Increment Time/Rec.X pos. spec.   |
                           ! |                                   |
LDA RXpos                  ! |                                   |
SUB RXpos0                 ! |                                   |
MUL TRXfact                ! |                                   |
ADD Tsig                   ! |                                   |
STA Tsig                   ! |                                   |
                           ! |                                   |
LDA RYpos                  ! |                                   |
SUB RYpos0                 ! |                                   |
MUL TRYfact                ! |                                   |
ADD Tsig                   ! |                                   |
STA Tsig                   ! |                                   |
                           ! |                                   |
TINC Tsig RXpos TRXspec2   ! | Increment Time/Rec.X pos. spec.   |
                           ! |                                   |
TINC Tsig RYpos TRYspec    ! | Increment Time/Rec.Y pos. spec.   |
                           ! |                                   |
TINC Tsig Tspec2           ! | Increment corrected Time spectrum |
TINC Tsig Tspec3           ! | in both TSpec2 and TSpec3         |
                           ! |                                   |
                           ! +-----------------------------------+

                           ! +-----------------------------------+
IF GTYPE = 1 THEN          ! | Check Gate Type (Full=0, Peak=1)  |
                           ! +-----------------------------------+
```

```
                          ! +-----------------------------------+
   IF Tsig TOFPEAK THEN   ! | Check to see if event falls in the |
      INC PEAKSUM         ! | gate for TOFFULL (GMin - GMax).    |
      INC ISum            ! | If it's a valid event, increment   |
   ENDIF                  ! | the PEAKSUM and ISum counters.     |
                          ! +-----------------------------------+
ELSE
                          ! +-----------------------------------+
   IF Tsig TOFFULL THEN   ! | Check to see if event falls in the |
      INC PEAKSUM         ! | TSpec2 window (channels 1 - 255).  |
      INC ISum            ! | If it's a valid event, increment   |
   ENDIF                  ! | the PEAKSUM and ISum counters.     |
                          ! +-----------------------------------+
ENDIF
                          ! +-----------------------------------+
TINC STAC1 TAC1spec       ! | Increment TAC spectrum             |
                          ! |                                    |
TINC Tsig STAC1 TAC12d    ! | Increment Time/TAC spectrum        |
                          ! |                                    |
                          ! |                                    |
TINC STAC2 TAC2spec       ! | Increment TAC spectrum             |
                          ! |                                    |
TINC Tsig STAC2 TAC22d    ! | Increment Time/TAC spectrum        |
                          ! |                                    |
IF Tsig GE DXMin          ! | Check if TOF falls in DXMin and    |
   IF Tsig LE DXMax       ! | DXMAX variables for d gate events. |
      IF STAC2 GE DYMin   ! | Check if TAC2 falls in DYMin and   |
         IF STAC2 LE DYMax ! | DYMax variables for d gate events. |
            INC DVal      ! | If so, increment DVal              |
         ENDIF            ! |                                    |
      ENDIF               ! |                                    |
   ENDIF                  ! |                                    |
ENDIF                     ! |                                    |
                          ! |                                    |
GET STAC1                 ! |                                    |
DIV 2                     ! |                                    |
STA STAC1D2               ! |                                    |
                          ! |                                    |
TINC STAC1 TAC1smsp       ! | Increment TAC spectrum             |
                          ! |                                    |
```

```
TINC Tsig STAC1D2 TAC12dsm ! | Increment Time/TAC spectrum        |
                           ! |                                    |
TINC STAC2 TAC2smsp        ! | Increment TAC spectrum             |
                           ! |                                    |
GET STAC2                  ! |                                    |
DIV 2                      ! |                                    |
STA STAC2D2                ! |                                    |
                           ! |                                    |
TINC Tsig STAC2D2 TAC22dsm ! | Increment Time/TAC spectrum        |
                           ! |                                    |
GET SDETUN                 ! |                                    |
FLOAT                      ! |                                    |
DIV 8.0                    ! |                                    |
FIX                        ! |                                    |
STA SDETUN                 ! |                                    |
                           ! |                                    |
TINC SDETUN DETUNspe       ! | Increment detuning  spectrum       |
                           ! |                                    |
Tinc Tsig SDETUN TDETUNsp  ! | Increment Time/detuning spectrum   |
                           ! |                                    |
GET SFLUO                  ! |                                    |
FLOAT                      ! |                                    |
DIV 8.0                    ! |                                    |
FIX                        ! |                                    |
STA SFLUO                  ! |                                    |
                           ! |                                    |
TINC SFLUO FLUOspec        ! | Increment fluorescence spectrum    |
                           ! |                                    |
Tinc SDETUN SFLUO DFLUOspe ! | Increment Time/detuning spectrum   |
                           ! |                                    |
TINC RXpos RYpos RXYspec2  ! | Increment Recoil X Y position spec.|
                           ! |                                    |
TINC RXpos RXspec          ! | Increment X pos                    |
                           ! |                                    |
TINC RYpos RYspec          ! | Increment Y pos                    |
                           ! +------------------------------------+

                           ! +------------------------------------+
GET MOTSig                 ! | Get the MOTSig value and store it  |
STA MOTVal                 ! | in a variable for later use.       |
GET IONSig                 ! | Get the IONSig value and store it  |
STA IONVal                 ! | in a variable for later use.       |
                           ! +------------------------------------+
```

```
! +-------------------------------------------------------------+
! |                  Data Event 50 (Scalar event) takes place   |
! +-------------------------------------------------------------+

                           ! +-----------------------------------+
EVENT 50                   ! | This is the incoming scalar event |
                           ! +-----------------------------------+

                           ! +-----------------------------------+
TAPE                       ! | Starts recording to disk          |
                           ! +-----------------------------------+

                           ! +-----------------------------------+
SET CNTRATE(Chan) = ISum   ! | Update CNTRATE display w/ new rate |
!SET MOTSpec(Chan) = MOTVal! | Update MOTSpec display w/ new rate |
SET MOTSpec(Chan) = DVal   ! | Update MOTSpec display w/ new rate |
SET IONSpec(Chan) = IONVal ! | Update IONSpec display w/ new rate |
Set ILAST = ISum           ! | Reset ILAST for next 10 sec        |
Set  ISum = 0              ! | Reset ISum for next 10 sec         |
Set  DVal = 0              ! | Reset DVal for next 10 sec         |
INC Chan                   ! | Increment Chan  for next 10 sec    |
                           ! +-----------------------------------+

                           ! +-----------------------------------+
END                        ! | End of the .evl file              |
                           ! +-----------------------------------+
```

C.2 Header and Other Acquisition Files

There are four header files used to correctly interface XSYS with the CAMAC electronics. They are shown here, along with a brief description of their function:

- **USERBEGIN.H**

  Inital commands executed at the start of data acquisition

- **USERCONFIG.H**

  General setup for CAMAC interface (define LAM, *etc.*)

- **USEREVENTS.H**

  Defines the handling of events from the CAMAC hardware

- **USERSCALERS.H**

  Simple time counter to trigger event block 50

Two additional XSYS files are necessary for data acquisition. These two files together control the allocation and presentation of memory areas where the incoming events are being stored.

- **XDISP.CMD**

  Controls macros and other plotting presets for display

- **STIRAP3.COM**

  Initial memory allocation for data arrays

C.2.1 USERBEGIN.H

Three commands are initiated when data acquisition begins. First, the TDC is reset prior to reading any data. Second, the look-at-me (LAM) is enabled for the Recoil ADC. Finally, the inhibit is disabled, although it is unclear whether this is necessary.

```
/* $Id: userbegin.h,v 1.1 2000/12/03 19:48:49 kdc Exp kdc $      */
/* $Log: userbegin.h,v $                                         */
/* Revision 1.1  2000/12/03  19:48:49  kdc                       */
/* Initial revision                                              */
/*                                                               */
/* DEC/CMS REPLACEMENT HISTORY, Element USERBEGIN.H              */
/* 23-SEP-1994 15:25:57 YODER "DAQ User Begin run include code"  */
/* DEC/CMS REPLACEMENT HISTORY, Element USERBEGIN.H              */
/* userbegin.h                                                   */
/*                                                               */
/* +-------------------------------------------------------+ */
/* | User initialization of modules at beginning of run.   | */
/* | This is called before data flow is enabled.           | */
/* | A BiRa2206 is automatically initialized before this code. | */
/* | To abort begin run, execute "return -1;" here.        | */
/* +-------------------------------------------------------+ */
/*                                                               */
CNAFX (1, 12,  0,  9);          /* reset LeCroy 4208 TDC        */
CNAFX (1, 17, 12, 26);          /* set enable LAM for AD811 ADC */
CNAFX (1, 30,  9, 24);          /* remove inhibit just in case  */
```

C.2.2 USERCONFIG.H

This file is included in the standard user configuration function. It is executed once after the front end processor is started, and whenever an "init config" command is issued. This code is executed before any CAMAC crates are initialized. The purpose is to specify which crates are required and the location of the trigger module. No CAMAC operations should be performed here. Configuration can be aborted by executing "return -1;" here. Initialization will be done after successful configuration.

```
/*  $Id: userconfig.h,v 1.2 1996/02/27 19:35:36 kdc Exp kdc $    */
/* $Log: userconfig.h,v $                                        */
/*                                                               */
/* Revision 1.3  2004/02/10  17:45:20  hcamp                     */
/* Changed st0tik = 1000                                         */
/*                                                               */
/* Revision 1.2  1996/02/27  19:35:36  kdc                       */
/* Separate user version & title.                                */
/*                                                               */
/* Revision 1.1  1995/09/28  19:48:50  kdc                       */
/* Initial revision                                              */
/*                                                               */
/* DEC/CMS REPLACEMENT HISTORY, Element USERCONFIG.H             */
/* 23-SEP-1994 15:25:59 YODER "DAQ User Config. include code"    */
/* DEC/CMS REPLACEMENT HISTORY, Element USERCONFIG.H             */
/* userconfig.h                                                  */
/*                                                               */
/* +-----------------------------------------------------------+ */
/* | Static user parameters                                    | */
/* | User config (userconfig.h) should set these as follows:   | */
/* |  udb.usrverstr_p = "version"; * user prog. version string | */
/* |  udb.usrtitstr_p = "title";   * initial run title string  | */
/* |  udb.maxevtsiz = size;        * if size > default(bytes)  | */
/* |  udb.btbmasks[b] |= (1<<c);   * each required online crate| */
/* |  udb.trigbranch = b;          * branch for lam trigger    | */
```

```
/* |  udb.trigcrate = c;          * crate for lam trigger     | */
/* |  udb.trigslot = n;           * slot for trigger module   | */
/* |  udb.triglamn = n;           * slot for trigger lam      | */
/* |  udb.trigmask |= (1<<m);     * for each BiRa trig channel| */
/* |  udb.st0tik = n;             * ticks for scaler timer    | */
/* |  udb.st1tik = n;             * ticks for slave timer 1   | */
/* |  udb.st2tik = n;             * ticks for slave timer 2   | */
/* |  udb.st3tik = n;             * ticks for slave timer 3   | */
/* | For BiRa2206, set trigbranch, trigcrate, trigslot,      | */
/* | trigmask.  If LAM is jumpered to different slot, set    | */
/* | triglamn = LAM slot.  For other trigger module, set     | */
/* | trigmask=0; such trigger module must return Q=? for     | */
/* | A(0)F(8), and clear lam for A(0)F(10).  For no trigger  | */
/* | module, set trigslot=trigmask=0 and set trigcrate to an | */
/* | online crate whose controller returns Q=1 for           | */
/* | N(30)A(0)F(0).                                          | */
/* +---------------------------------------------------------+ */
/* +---------------------------------------------------------+ */
/* | Set parameters                                          | */
/* +---------------------------------------------------------+ */
udb.usrverstr_p = "<Version 1>";
udb.usrtitstr_p = "<2D PSD Ortec AD811 ADC>";
udb.maxevtsiz = 0;                          /* uses default if 0*/
udb.btbmasks[0] = (1<<1);
udb.lamcrmask = (1<<1);        /*LAM trig. crate mask (BD crates)*/
udb.lamslmask = (1<<16); /*LAM trig. glr mask (BD slots)(slot#-1)*/
udb.trigbranch = 0;
udb.trigcrate = 1;
udb.trigslot = 0;
udb.triglamn = 0;                  /*can be 0 if same as trigslot*/
udb.trigmask = 0;
udb.st0tik = 1000;                   /*Check Scalars every 10 sec*/
udb.st1tik = 0;
udb.st2tik = 0;
udb.st3tik = 0;
```

C.2.3 USEREVENTS.H

Data are read from the hardware indicated within this file each time a look-at-me (LAM) event is generated. For the MOTRIMS acquisition system, the LAM is generated when the Recoil ADC is strobed.

```
/* +-------------------------------------------------------------+ */
/* |  $Id: userevents.h,v 1.1 2000/12/03 19:48:50 kdc Exp kdc $  */
/* | $Log: userevents.h,v $                                      */
/* | Revision 1.1  2000/12/03  19:48:50  kdc                     */
/* | Initial revision                                            */
/* |  DEC/CMS REPLACEMENT HISTORY, Element USEREVENTS.H          */
/* | 23-SEP-1994 15:26:05 YODER "DAQ User Event read code"       */
/* | DEC/CMS REPLACEMENT HISTORY, Element USEREVENTS.H           */
/* +-------------------------------------------------------------+ */
/* userevents.h                                                  */
/* +-------------------------------------------------------------+ */
/* | Get event data record.                                    | */
/* | Reads data from CAMAC and fills event record with id      | */
/* | and data.                                                 | */
/* | Event id is preset to trigchan if using BiRa2206.         | */
/* | BiRa2206 is handled outside of this code.                 | */
/* +-------------------------------------------------------------+ */
  ev_p->hdr.evtid += 2;   /* Indicates Event ID #2 will fire     */
READsL(0, 1, 12, 0, 0);   /* 24 bit 4208 TDC read                */
 READW(0, 1, 19, 6, 2);   /* Read & clear AD811 recoil - TAC2    */
 READW(0, 1, 17, 1, 2);   /* Read & clear AD811 recoil - Unused  */
 READW(0, 1, 17, 2, 2);   /* Read & clear AD811 recoil - Unused  */
 READW(0, 1, 19, 7, 2);   /* Read & clear AD811 recoil - TAC1    */
 READW(0, 1, 17, 4, 2);   /* Read & clear AD811 recoil - (Top N) */
 READW(0, 1, 17, 5, 2);   /* Read & clear AD811 recoil - (Bot S) */
 READW(0, 1, 17, 6, 2);   /* Read & clear AD811 recoil - (Bot N) */
 READW(0, 1, 17, 7, 2);   /* Read & clear AD811 recoil - (Top S) */
 READW(0, 1, 18, 3, 2);   /* Read & clear AD811 MOT Signal       */
 READW(0, 1, 18, 4, 2);   /* Read & clear AD811 Ion Beam Signal  */
 READW(0, 1, 18, 5, 2);   /* Read & clear AD811 projectile Z     */
 READW(0, 1, 18, 6, 2);   /* Read & clear AD811 projectile W     */
 READW(0, 1, 18, 7, 2);   /* Read & clear AD811 projectile S     */
 CNAFX(1, 12, 0, 9);      /* Clear 4208 TDC                      */
```

C.2.4 USERSCALERS.H

This file is used as a trigger to activate event block 50 in the stirap.evl code. Every ten seconds, a single channel is read (the value of which is unimportant), and event 50 is executed. This file is essentially used as a timer, nothing more. This allows us to make time measurements asynchronous with the incoming events, thus updating things such as the count rate strip chart, *etc.*

```
/* +-----------------------------------------------------------+ */
/* | $Id: userscalers.h,v 1.0 2004/02/10 18:36:25 hcamp        | */
/* | This user file reads all scalars.                         | */
/* +-----------------------------------------------------------+ */
/*                                                               */
/* READsL(0, 1, 12, 2, 0);  */  /* 24 bit 4208 TDC read         */
/*                                                               */
/* +-----------------------------------------------------------+ */
/* | NOTE:  It doesn't matter which channel is read -          | */
/* | userscalars.h is only used to trigger the event 50 block  | */
/* | in the stirap2.evl file.                                  | */
/* +-----------------------------------------------------------+ */
```

C.2.5 XDISP.CMD

The memory allocation file, XDISP.CMD is included for completeness. Each macro is defined by a leading number, indicating the view in the VMS 'display' program. Macros 1–15 call some combination of the defined windows listed from 1001–1022. Each window begins with the allocated memory area, followed by a set of coordinates indicating where it should be printed on the screen, and what size to make the window. For example, "`10v4,2,0,6n2j`" will print memory area #10 (the X_1 coordinate from the recoil detector) in a position (0,6) with a size (4,2).

```
   1: *r 1001@p 1021@pp
   2: *r 1002@p 1022@pp
   3: *r 1003@pp
   4: *r 1004@pp
   5: *r 1005@pp
   6: *r 1006@pp
   7: *r 1007@pp
   8: *r 1008@pp
   9: *rp 1009@p
  10: *r 1010@pp
  11: 3@
  12: *rp 1011@p 1012@p 1013@p
  13: *r 1011@pp
  14: *r 1012@pp
  15: *r 1013@pp
1001: 10v4,2,0,6n2j 11v4,2,0,4n2j 12v4,2,4,6n2j 13v4,2,4,4n2j
      14v4,2,4,2n2j
1002: 20v4,2,0,6n2j 21v4,2,0,4n2j 22v4,2,4,6n2j 23v4,2,4,4n2j
      25v4,4,0,0n2j
1003:  1v4,4,0,4n2j  5v4,4,4,4n2j 35v4,4,0,0n2j 50v4,4,4,0n2j
1004: 16v4,4,0,4n2j  5v4,4,4,4n2j 76v8,2,0,0n2j 77v4,2,0,2n2j
      78v4,2,4,2n2j
1005: 41v4,4,0,4n2j 42v4,4,4,4n2j 40v4,4,0,0n2j 60v4,4,4,0n2j
1006: 51v4,4,0,4n2j 52v4,4,4,4n2j 53v4,4,0,0n2j 54v4,4,4,0n2j
1007:  5v6,2,0,6n2j 55v6,6,0,0n2j
1008: 45v4,4,0,0n2j 46v4,4,4,0n2j
```

```
1009: 65v4,4,0,0n2j 72v4,4,0,4n2j 74v4,4,4,4n2j 70v4,4,4,0n2j
1010:  9v4,4,0,4n2j 68v4,4,0,0n2j 69v4,4,4,0n2j
1011: 77v2,2,2,6n2j 78v2,2,4,6n2j 76v2,2,6,6n2j 65v3,4,2,2n2j
      70v3,4,5,2n2j
1012: 51v2,2,0,6n2j 52v2,2,0,4n2j 53v2,2,0,2n2j 54v2,2,0,0n2j
1013:  5v2,2,2,0n2j 15v2,2,4,0n2j 25v2,2,6,0n2j
1021: 15v4,4,0,0n2j
1022: 45v4,4,4,0n2j
```

C.2.6 STIRAP3.COM

At the beginning of a new XSYS acquisition session, it is necessary to allocate memory space for 1D and 2D spectra that will be stored. This is done in a small DCL file with the same name as the acquisition .EVL file (in this case, STIRAP3). The file shows each memory area, along with the allocated number of blocks to be associated with the area. Two arguments after the area name indicate a 2D data area is defined.

```
$! STIRAP3.COM
$!
$! .COM file for STIRAP-related
$! experiments using the MOTRIMS
$! experimental apparatus
$!
$ DMEM ALL GLOBAL FILE
$ AMEM NEW 22000 PAGES*
$!
$ AMEM  1    Tspec 16384
$ AMEM  2    TspecCN   256
$ AMEM  3    TspecCF   256
$ AMEM  4    TspecDF   256
$ AMEM  5    Tspec2    256
$ AMEM  6    TAC1spec  512
$ AMEM  7    TNUMspec  4
$ AMEM  8    DETUNspe   256
$ AMEM  9    TDETUNsp  256 256
$ AMEM  10   RX1spec   2048
$ AMEM  11   RX2spec   2048
$ AMEM  12   RY1spec   2048
$ AMEM  13   RY2spec   2048
$ AMEM  14   RSspec    8192
$ AMEM  15   RXYspec   512 512
$ AMEM  16   Tspec3    256
$ AMEM  20   PXspec    2048
$ AMEM  21   PYspec    2048
$ AMEM  22   PRspec    2048
```

```
$ AMEM  23   PSspec    8192
$ AMEM  25   PXYspec   512 512
$ AMEM  30   TRXspec   256 512
$ AMEM  35   TRYspec   256 512
$ AMEM  40   RXYspec2  512 512
$ AMEM  41   RXspec    512
$ AMEM  42   RYspec    512
$ AMEM  45   PXRXspec  512 512
$ AMEM  46   PYRYspec  512 512
$ AMEM  50   TRXspec2  256 512
$ AMEM  51   Proj1 512
$ AMEM  52   Proj2 512
$ AMEM  53   Proj3 512
$ AMEM  54   Proj4 512
$ AMEM  55   TVtspec   256 256
$ AMEM  60   VXVYspec  512 512
$ AMEM  65   TAC12d    256 512
$ AMEM  66   FLUOspec   256
$ AMEM  67   DFLUOspe  256 256
$ AMEM  68   TDetunON  256 256
$ AMEM  69   TDetunOF  256 256
$ AMEM  70   TAC22d    256 512
$ AMEM  71   TAC2spec  512
$ AMEM  72   TAC12dsm  256 256
$ AMEM  73   TAC1smsp  256
$ AMEM  74   TAC22dsm  256 256
$ AMEM  75   TAC2smsp  256
$ AMEM  76   CNTRate   8192
$ AMEM  77   MOTSpec    8192
$ AMEM  78   IONSpec    8192
$!
$ GATE NEW
$ GATE 5 2
$ GATE 51 4
$ GATE 16 1
$!
$ CLEAR FLAGS
$ CLEAR ALL
$ EXIT
```

C.3 Additional XSYS Macros

In order to facilitate data acquisition, several DCL programs were written. They are listed here in alphabetical order with a brief description as to their function:

- **CLEAR.COM**

  Wipes all temporary files from disk and closes open data files

- **CROSSDIFF.COM**

  Takes on-the-fly differences between two data areas

- **DISKSPACE.COM**

  Returns free disk space for any mounted disks

- **DOPRINT.COM**

  Prints all relevant variable information at the end of a run

- **PRINTVARS.COM**

  Prints current variables to a dated file

- **READVARS.COM**

  Reads previously stored variables and their values

- **SETGATE.COM**

  Defines optional Projectile and Recoil 2D Detector gates

- **SHOWVARS.COM**

  Lists current variables and their values

- **STOPDRIFT.COM**

  Monitors and controls ion beam drift during acquisition

- **STOREVARS.COM**

  Stores current variables and their values to disk

- **VARS.COM**

  Simple interface for reading and setting current variables

C.3.1 CLEAR.COM

During data acquisition, several autonomous macros can be in operation, including drift suppression and low count rate monitoring. When these macros are halted, they sometimes leave behind opened or uncleared temporary files. This simple program closes all opened data files and clears any temporary files that may have been in use.

```
$! +-------------------------------------------------------------+
$! |      Filename: clear.com                                    |
$! | Last modified: 09/21/2004 by hcamp                          |
$! |                                                             |
$! |   Description: clear simply wipes all temporary files from  |
$! |                the disk and closes any open files that may  |
$! |                have been accidentally left opened in the    |
$! |                event of a .com file crashing (which never   |
$! |                happens...)                                  |
$! |                                                             |
$! |         Usage: clear                                        |
$! |                                                             |
$! |                Specifically, all .tmp files will be deleted |
$! |                from the data:[stirap4.vars] directory, and  |
$! |                a list of all files opened during the use of |
$! |                the macros found in data:[stirap4.com] will  |
$! |                be closed.                                   |
$! +-------------------------------------------------------------+
$!
$ TempFiles = F$SEARCH("data:[stirap3.vars]*.tmp")
$ IF TempFiles .NES. "" THEN DELETE data:[stirap3.vars]*.tmp;*
$!
$! +-------------------------------------------------------------+
$! | Files opened in DOPRINT.COM                                 |
$! +-------------------------------------------------------------+
$   CLOSE/NOLOG instat
$   CLOSE/NOLOG headerfile
$! +-------------------------------------------------------------+
$! | Files opened in PRINTVARS.COM                               |
$! +-------------------------------------------------------------+
$   CLOSE/NOLOG invars
$   EXIT
```

C.3.2 CROSSDIFF.COM

Sometimes it is useful to take the difference of two projections while in XSYS. For example, when measuring cross sections, it is useful to check whether enough statistics have been accumulated by taking the difference between two projections corresponding to Stokes laser on, and Stokes laser off. This program performs such operations on two data areas in XSYS. Note that proper analysis should be done in Origin, using Origin scripts shown in Sec. E.

```
$! +-------------------------------------------------------------+
$! |      Filename: crossdiff.com                                |
$! | Last modified: 10/12/04 by hcamp                            |
$! |                                                             |
$! |   Description: Simple file designed to take the difference  |
$! |                of L2 On vs. L2 Off data on-the-fly.  REAL   |
$! |                ANALYSIS SHOULD BE DONE EXTERNALLY (VIA      |
$! |                ORIGIN)!                                     |
$! +-------------------------------------------------------------+
$!
$!
$ on error then continue
$!
$ LOOP:
$!
$ WAIT 00:00:01
$!
$ PROJ 65 51 DO 42 87
$ PROJ 65 52 DO 442 487
$ PROJ 65 54 DO 303 348
$ ADD 52 -51 53
$!
$ GOTO LOOP
$ exit
```

C.3.3 DISKSPACE.COM

Often with long data runs, it is valuable to monitor the amount of free disk space remaining before errors will occur and data may potentially be lost. Since VMS doesn't have a convenient method of doing such, this short program parses through the disk information and returns only the remaining free space on any available mounted disks (usually `data` and `data2`).

Note that "_" at the end of a line indicates a wrapped entry for printing purposes. This character should be removed, and the two lines concatenated if the program is to be used.

```
$! +-------------------------------------------------------------+
$! |      Filename: diskspace.com                                |
$! | Last modified: 09/09/2004 by hcamp                          |
$! |                                                             |
$! |   Description: Diskspace returns the free diskspace for     |
$! |                mounted drives.  Note that disk2 will attempt |
$! |                to mount prior to checking the empty disk    |
$! |                space.                                       |
$! +-------------------------------------------------------------+
$!
$ on error then continue
$!
$    WRITE sys$output "+---------------------------------------+"
$!
$    mount1 = f$getdvi("stark$dka000:","mnt")
$    mount2 = f$getdvi("stark$dka200:","mnt")
$!
$    IF mount1 .EQS. "TRUE"
$       THEN
$       disk1 = f$getdvi("stark$dka000:","volnam")
$       free1 = f$integer(f$getdvi("stark$dka000:",_
                "freeblocks")) * 100
$        max1 = f$integer(f$getdvi("stark$dka000:","maxblock"))
$       perc1 = free1 / max1
$       free1 = free1 / 100
$       WRITE sys$output " Disk #1 (DKA000): ''disk1'_
              ('perc1'% Free) 'free1'/'max1'"
$    ENDIF
$!
```

```
$    IF mount2 .EQS. "TRUE"
$       THEN
$       disk2 = f$getdvi("stark$dka200:","volnam")
$       free2 = f$integer(f$getdvi("stark$dka200:",_
                "freeblocks")) * 100
$        max2 = f$integer(f$getdvi("stark$dka200:","maxblock"))
$       perc2 = free2 / max2
$       free2 = free2 / 100
$       WRITE sys$output " Disk #2 (DKA200): ''disk2'_
              (''perc2'% Free) ''free2''/''max2'"
$    ENDIF
$!
$    WRITE sys$output "+----------------------------------------+"
$ EXIT
```

C.3.4 DOPRINT.COM

When a run has been completed, this macro will print out a concise page, showing all variable settings used, and the corresponding run information. Also, a graphics page is printed, showing all important data areas. (This typically consists of view "12@" from `DISPLAY`.)

Note that "_" at the end of a line indicates a wrapped entry for printing purposes. This character should be removed, and the two lines concatenated if the program is to be used.

```
$! +-------------------------------------------------------------+
$! |      Filename: doprint.com                                  |
$! |  Last Updated: 09/22/2004 by hcamp and mltrachy             |
$! |                                                             |
$! |   Description: When a run completes, the variables          |
$! |                and current stats corresponding to the       |
$! |                given run can be printed, along with a       |
$! |                print of view @12 (most important data       |
$! |                areas).                                      |
$! |                                                             |
$! |         Usage: doprint                                      |
$! |                                                             |
$! +-------------------------------------------------------------+
$!
$! +-------------------------------------------------------------+
$! | Clean up any old/existing files and open declarations       |
$! +-------------------------------------------------------------+
$!
$ @data:[stirap3.com]clear
$!
$! +-------------------------------------------------------------+
$! | Create and parse the stat time file to get  the run #       |
$! +-------------------------------------------------------------+
$!
$ DEFINE SYS$OUTPUT data:[stirap3.vars]stattime.tmp
$ STAT TIME 1
$ DEFINE SYS$OUTPUT SYS$COMMAND
$!
$ OPEN/READ instat data:[stirap3.vars]stattime.tmp
```

```
$ READ instat linein
$ READ instat linein
$ READ instat linein
$    len = F$LOCATE("#", linein)
$ linein = F$EXTRACT(len, F$LENGTH(linein), linein)
$    len = F$LOCATE("=", linein)
$ runnum = F$EXTRACT(len + 8, F$LENGTH(linein), linein)
$ runnum = F$EDIT(runnum,"TRIM")
$ CLOSE instat
$!
$! +-------------------------------------------------------------+
$! | Run 'storevars' to save current run variables to file       |
$! +-------------------------------------------------------------+
$!
$ @VARS STORE 'runnum'
$!
$ OPEN/WRITE headerfile data:[stirap3.vars]headerfile.tmp
$ timein = F$TIME()
$ WRITE headerfile "       "+ timein + "           Run: ''runnum'"
$ WRITE headerfile " "
$ WRITE headerfile " "
$ WRITE headerfile " "
$ WRITE headerfile " "
$ WRITE headerfile " "
$ WRITE headerfile " "
$ CLOSE headerfile
$!
$! +-------------------------------------------------------------+
$! | 'Pretty-print' to file and print the final results           |
$! +-------------------------------------------------------------+
$!
$ @VARS PRINT
$ OPEN/APPEND instat data:[stirap3.vars]printout.tmp
$ WRITE instat " "
$ WRITE instat " "
$ WRITE instat " "
$ WRITE instat " "
$ WRITE instat " "
$ WRITE instat " "
$ CLOSE instat
$!
```

```
$ copy data:[stirap3.vars]headerfile.tmp,_
  data:[stirap3.vars]printout.tmp, data:[stirap3.vars]stattime.tmp_
  data:[stirap3.vars]run_'runnum'.prn
$!
$ print/queue=hpsquare data:[stirap3.vars]run_'runnum'.prn
$ disp 6@ 12‘ 7pq
$ print/queue=hpsquare xdisp.eps
$ @data:[stirap3.com]clear
$ EXIT
```

C.3.5 PRINTVARS.COM

This macro is used in the DOPRINT.COM program to print all relevant variables and their values to a file. It is useful to have such a file on disk when data are rerun in the future. Thus, one can review what settings were present at the time of the run.

```
$! +-------------------------------------------------------------+
$! |      Filename: printvars.com                                |
$! | Last modified: 09/22/2004 by hcamp                          |
$! |                                                             |
$! |   Description: Printvars prints all relevant variable values |
$! |                to the screen.  It's done in a way useful for |
$! |                printing hard copies using the 'doprint'     |
$! |                macro.                                       |
$! |                                                             |
$! |         Usage: printvars                                    |
$! |                                                             |
$! |                All printvars output is stored in a temporary |
$! |                file named 'printout.tmp', used in the       |
$! |                doprint.com script.  Printvars will *only*   |
$! |                print the current variables (hence, there is  |
$! |                no option to supply a run #).                |
$! +-------------------------------------------------------------+
$!
$ WRITE SYS$OUTPUT " "
$!
$ define sys$output data:[stirap3.vars]printvars.tmp
$ evop var TMIN1
$ evop var TMAX1
$ evop var TMIN2
$ evop var TMAX2
$ evop var TRXFACT
$ evop var TRYFACT
$ evop var RXPOS0
$ evop var RYPOS0
$ evop var TDRIFT
$ evop var MAXCNT
$!
```

```
$ define sys$output sys$command
$!
$ open/read INVARS data:[stirap3.vars]printvars.tmp
$ GOSUB GETVAR
$       TMIN1 = GOTVAR
$ GOSUB GETVAR
$       TMAX1 = GOTVAR
$ GOSUB GETVAR
$       TMIN2 = GOTVAR
$ GOSUB GETVAR
$       TMAX2 = GOTVAR
$ GOSUB GETVAR
$     TRXFACT = GOTVAR
$ GOSUB GETVAR
$     TRYFACT = GOTVAR
$ GOSUB GETVAR
$      RXPOS0 = GOTVAR
$ GOSUB GETVAR
$      RYPOS0 = GOTVAR
$ GOSUB GETVAR
$      TDRIFT = GOTVAR
$ GOSUB GETVAR
$      MAXCNT = GOTVAR
$!
$ close INVARS
$!
$ DEFINE SYS$OUTPUT data:[stirap3.vars]printout.tmp
$ WRITE SYS$OUTPUT "-----------------------------------------------"
$ WRITE SYS$OUTPUT "  TMIN1: " + TMIN1 + "   TMIN2: " + TMIN2
$ WRITE SYS$OUTPUT "  TMAX1: " + TMAX1 + "   TMAX2: " + TMAX2
$ WRITE SYS$OUTPUT "-----------------------------------------------"
$ WRITE SYS$OUTPUT "TRXFACT: " + TRXFACT + "    RXPOS0: " + RXPOS0
$ WRITE SYS$OUTPUT "TRYFACT: " + TRYFACT + "    RYPOS0: " + RYPOS0
$ WRITE SYS$OUTPUT "-----------------------------------------------"
$ WRITE SYS$OUTPUT " TDRIFT: " + TDrift + "     MAXCNT: " + MAXCNT
$ WRITE SYS$OUTPUT "-----------------------------------------------"
$ GATE EXAM  5 1
$ GATE EXAM  5 2
$ GATE EXAM 16 1
$ WRITE SYS$OUTPUT "-----------------------------------------------"
$!
```

```
$ define sys$output sys$command
$ delete data:[stirap3.vars]printvars.tmp;*
$ EXIT
$!
$ GETVAR:
$    read INVARS TMPVAR
$      LEN=f$locate("=", TMPVAR)+1
$   GOTVAR=f$extract(LEN, f$length(TMPVAR), TMPVAR)
$ RETURN
$!
```

C.3.6 READVARS.COM

After each run, `DOPRINT.COM` is typically executed. For runs where this is the case, the variable settings used during that run can be reloaded at a later time using this macro. If `READVARS.COM` is called without an argument, the last generically saved variable settings will be recalled.

```
$! +-------------------------------------------------------------+
$! |      Filename: readvars.com                                 |
$! | Last modified: 09/22/2004 by hcamp                          |
$! |                                                             |
$! |   Description: Readvars sets all relevant variables to      |
$! |                stored parameters, either from the current   |
$! |                run, or a supplied run number indicating a   |
$! |                previous run, assuming the variables were    |
$! |                previously stored using "storevars".         |
$! |                                                             |
$! |         Usage: readvars [run #]                             |
$! |                                                             |
$! |                The run# is optional.  Supplying it will read |
$! |                the variable settings used for a given run,  |
$! |                assuming that 'storevars' was used to store  |
$! |                the parameters at that time.                 |
$! |                                                             |
$! |                Without supplying a run#, all current values |
$! |                will be overwritten with stored values       |
$! |                in the 'default' storage file.  WARNING:     |
$! |                The varible values stored in the 'default'   |
$! |                storage file *might not* correspond to the   |
$! |                current run!  See "storevars.com" for further |
$! |                information.                                 |
$! +-------------------------------------------------------------+
$!
$ ON ERROR THEN CONTINUE
$!
```

```
$ IF p1 .EQS. ""
$    THEN
$       OPEN/READ READIN data:[stirap3.vars]temporary.var
$    ELSE
$       OPEN/READ READIN data:[stirap3.vars]run_'p1'.var
$ ENDIF
$!
$ READ READIN TEMPLINE
$ NEWVAR=F$EXTRACT(21,13,TEMPLINE)
$ EVOP VAR TMIN1='NEWVAR
$!
$ READ READIN TEMPLINE
$ NEWVAR=F$EXTRACT(21,13,TEMPLINE)
$ EVOP VAR TMAX1='NEWVAR
$!
$ READ READIN TEMPLINE
$ NEWVAR=F$EXTRACT(21,13,TEMPLINE)
$ EVOP VAR TMIN2='NEWVAR
$!
$ READ READIN TEMPLINE
$ NEWVAR=F$EXTRACT(21,13,TEMPLINE)
$ EVOP VAR TMAX2='NEWVAR
$!
$ READ READIN TEMPLINE
$ NEWVAR=F$EXTRACT(21,13,TEMPLINE)
$ EVOP VAR TRXFACT='NEWVAR
$!
$ READ READIN TEMPLINE
$ NEWVAR=F$EXTRACT(21,13,TEMPLINE)
$ EVOP VAR TRYFACT='NEWVAR
$!
$ READ READIN TEMPLINE
$ NEWVAR=F$EXTRACT(21,13,TEMPLINE)
$ EVOP VAR RXPOS0='NEWVAR
$!
$ READ READIN TEMPLINE
$ NEWVAR=F$EXTRACT(21,13,TEMPLINE)
$ EVOP VAR RYPOS0='NEWVAR
$!
$ READ READIN TEMPLINE
$ NEWVAR=F$EXTRACT(21,13,TEMPLINE)
$ EVOP VAR TDRIFT='NEWVAR
$!
```

```
$ READ READIN TEMPLINE
$ NEWVAR=F$EXTRACT(21,13,TEMPLINE)
$ EVOP VAR MAXCNT='NEWVAR
$!
$ READ READIN TEMPLINE
$ MINGATE = F$EXTRACT(29,6,TEMPLINE)
$ MAXGATE = F$EXTRACT(37,6,TEMPLINE)
$ GATE SET 5 1 'MINGATE 'MAXGATE
$!
$!
$ READ READIN TEMPLINE
$ MINGATE = F$EXTRACT(29,6,TEMPLINE)
$ MAXGATE = F$EXTRACT(37,6,TEMPLINE)
$ GATE SET 5 2 'MINGATE 'MAXGATE
$!
$ READ READIN TEMPLINE
$ MINGATE = F$EXTRACT(29,6,TEMPLINE)
$ MAXGATE = F$EXTRACT(37,6,TEMPLINE)
$ GATE SET 16 1 'MINGATE 'MAXGATE
$!
$ CLOSE READIN
$ WRITE SYS$OUTPUT "----------------------------------------"
$ WRITE SYS$OUTPUT "Variables set."
$ WRITE SYS$OUTPUT "----------------------------------------"
$exit
```

C.3.7 SETGATE.COM

In order to expedite the multi-variable process of setting and clearing software gates on both the projectile and recoil detectors, this short program was written.

Note that "_" at the end of a line indicates a wrapped entry for printing purposes. This character should be removed, and the two lines concatenated if the program is to be used.

```
$!   +-------------------------------------------------+
$!   |                                                 |
$!   |  Created:  07/01/2004                           |
$!   |   Author:  MLTrachy                             |
$!   | CoAuthor:  HCamp                                |
$!   |                                                 |
$!   | Usage: 'setgate <d,r> min_x max_x min_y max_y'  |
$!   |                                                 |
$!   | Description: Sets either the recoil detector gate |
$!   |              or the d event gate to the rectangle |
$!   |              described by min_x, max_x, min_y and |
$!   |              max_y                              |
$!   |                                                 |
$!   +-------------------------------------------------+
$!
$ WRITE SYS$OUTPUT P1
$ If p1 .EQS. "D"
$   THEN
$    evop var dxmin='p2'
$    evop var dxmax='p3'
$    evop var dymin='p4'
$    evop var dymax='p5'
$   ELSE If p1 .EQS. "R"
$     THEN
$      evop var rxmin='p2'
$      evop var rxmax='p3'
$      evop var rymin='p4'
$      evop var rymax='p5'
$     ELSE If p1 .EQS. "P"
```

```
$       THEN
$        evop var pxmin='p2'
$        evop var pxmax='p3'
$        evop var pymin='p4'
$        evop var pymax='p5'
$       ELSE
$      WRITE SYS$OUTPUT "Usage: setgate <d,r> min_x max_x _
       min_y max_y"
$   ENDIF
$ ENDIF
```

C.3.8 SHOWVARS.COM

Variables stored using the `DOPRINT.COM` macro can be redisplayed at a later point in time. If no argument is supplied, the current variables and their values are displayed.

```
$! +-------------------------------------------------------------+
$! |      Filename: showvars.com                                 |
$! | Last modified: 09/21/2004 by hcamp                          |
$! |                                                             |
$! |   Description: Showvars prints all relevant variable values |
$! |                to the screen.                               |
$! |                                                             |
$! |         Usage: showvars [run #]                             |
$! |                                                             |
$! |                The run# is optional.  Supplying it will show |
$! |                the variable settings used for a given run,  |
$! |                assuming that 'storevars' was used to store  |
$! |                the parameters at that time.                 |
$! |                                                             |
$! |                Without supplying a run#, all current values |
$! |                will be displayed.                           |
$! +-------------------------------------------------------------+
$!
$ IF p1 .NES. ""
$    THEN
$    TYPE data:[stirap3.vars]run_'p1'.var
$    EXIT
$ ENDIF
```

```
$ evop var TMIN1
$ evop var TMAX1
$ evop var TMIN2
$ evop var TMAX2
$ evop var TRXFACT
$ evop var TRYFACT
$ evop var RXPOS0
$ evop var RYPOS0
$ evop var TDRIFT
$ evop var MAXCNT
$!
$ gate exam  5 1
$ gate exam  5 2
$ gate exam 16 1
$!
$ EXIT
```

C.3.9 STOPDRIFT.COM

During each run, the ion gun energy will wander slightly, causing a drifting effect on the measured time-of-flight. In order to minimize such an effect, this program monitors a selected peak (defined by a software gate) and shifts the incoming TDC spectrum appropriately to compensate for any such ion beam drift.

Note that "_" at the end of a line indicates a wrapped entry for printing purposes. This character should be removed, and the two lines concatenated if the program is to be used.

```
$! +-------------------------------------------------------------+
$! |      Filename: stopdrift.com                                |
$! | Last modified: 02/08/04 by hcamp                            |
$! |                                                             |
$! |   Description: This file repeatedly scans TSpec3 and measures|
$! |                the centroid position of the sp-peak.  When   |
$! |                the centroid drifts, the program automatically|
$! |                adjusts TDRIFT to match, thus keeping the data|
$! |                in the proper place.                         |
$! +-------------------------------------------------------------+
$!
$ on error then continue
$!
$ close infile
$  STARTPOS = 0
$ FIRSTTIME = 1
$!
$ STARTUP:
$!
$ CENTRPOS = 0
$ NEWDRIFT = 0
$!
```

```
$! +-------------------------------------------------------------+
$! | The file var_gatelock.out hold the information about how many|
$! | counts are in the current rendition of spectra #16 (Tspec3), |
$! | gate #1.  This allows us to constantly refresh the driftlock |
$! | to obtain more accurate values than we would by using spectra|
$! | #5 (Tspec2) as counts build up longer and longer.  When too  |
$! | many (>5000) counts fill spectra #16, it is cleared and the  |
$! | process begins again.                                        |
$! +-------------------------------------------------------------+
$!
$    define sys$output var_gatelock.out
$    sum 16 0 1
$    purge var_gatelock.out
$    rename var_gatelock.out;* var_gatelock.out;1
$!
$! +-------------------------------------------------------------+
$! | The file var_tdrift.out holds the current tdrift value       |
$! | obtained from 'evop var tdrift'.  This is crucial in setting |
$! | the proper shift once a drift is detected.                   |
$! +-------------------------------------------------------------+
$!
$    define sys$output var_tdrift.out
$    evop var tdrift
$    purge var_tdrift.out
$    rename var_tdrift.out;* var_tdrift.out;1
$!
$! +-------------------------------------------------------------+
$! | The file var_maxcnt.out holds the current tdrift value       |
$! | obtained from 'evop var maxcnt'.  This is crucial in stopping|
$! | the run when MAXCNT is reached.                              |
$! +-------------------------------------------------------------+
$!
$    define sys$output var_maxcnt.out
$    evop var maxcnt
$    purge var_maxcnt.out
$    rename var_maxcnt.out;* var_maxcnt.out;1
$!
$! +-------------------------------------------------------------+
$! | The file var_totalsum.out holds the current total sum value  |
$! | obtained from spectra #5 (Tspec2) gate #1.  This value is    |
$! | checked against MAXCNT to see if the run should be halted.   |
$! +-------------------------------------------------------------+
$!
```

```
$    define sys$output var_totalsum.out
$    sum 5 0 1
$    purge var_totalsum.out
$    rename var_totalsum.out;* var_totalsum.out;1
$!
$! +-------------------------------------------------------------+
$! | The file MOTrack.out holds two variables - ilast and chan,  |
$! | used EXTERNALLY (on either Anderson, or Hawking) to monitor |
$! | the count rate and alert the users when it dips below a     |
$! | threshold value. No further action is taken to MOTrack.out  |
$! | here in XSYS.                                               |
$! +-------------------------------------------------------------+
$!
$    define sys$output MOTrack.out
$    evop var ilast
$    evop var chan
$    purge MOTrack.out
$    rename MOTrack.out;* MOTrack.out;1
$    define sys$output sys$command
$!
$! +-------------------------------------------------------------+
$! | Now, we reread some of the values we've just stored, and    |
$! | place them in local variables.                              |
$! +-------------------------------------------------------------+
$!
$    open/read infile var_tdrift.out
$    read infile stvar
$    close infile
$!
$    len=f$locate("=", stvar)+1
$    len2=f$locate(".", stvar)
$    newvar=f$extract(len,len2-len,stvar)
$    tdrift=f$integer(newvar)
$!
$    open/read infile var_maxcnt.out
$    read infile stvar
$    close infile
$!
$    len=f$locate("=", stvar)+1
$    len2=f$locate(".", stvar)
$    newvar=f$extract(len,len2-len,stvar)
$    maxcnt=f$integer(newvar)
$!
```

```
$    open/read infile var_totalsum.out
$    read infile CENTERPOS
$    read infile CENTERPOS
$    read infile CENTERPOS
$    read infile CENTERPOS
$    read infile CENTERPOS
$    close infile
$!
$    len=f$locate(".", CENTERPOS)-21
$    TOTALSUM = f$extract(21,len,CENTERPOS)
$    TOTALSUM = f$integer(TOTALSUM)
$!
$    open/read infile var_gatelock.out
$    read infile CENTERPOS
$    read infile CENTERPOS
$    read infile CENTERPOS
$    read infile CENTERPOS
$    read infile CENTERPOS
$    close infile
$!
$    If 'MAXCNT .GT. 0
$       THEN
$       IF 'TOTALSUM .GE. 'MAXCNT
$          THEN
$          WRITE SYS$OUTPUT "Total counts reached: ''TOTALSUM' / _
                 ''MAXCNT'  Halting..."
$          vrctl -c halt
$          halt
$          EXIT
$       ENDIF
$    ENDIF
$!
$    len=f$locate(".", CENTERPOS)-21
$    PEAKSUM = f$extract(21,len,CENTERPOS)
$    PEAKSUM = f$integer(PEAKSUM)
$    IF PEAKSUM .LT. 250
$       THEN
$       WRITE SYS$OUTPUT "Not enough counts in sp peak (only _
              ''PEAKSUM'):  Waiting..."
$       GOTO STARTUP
$    ENDIF
$!
```

```
$    ACTUAL=f$extract(52,6,CENTERPOS)
$    NEWPOS=f$extract(51,4,CENTERPOS)
$    NEWPOS=f$integer(NEWPOS)
$!
$ IF FIRSTTIME .EQ. 1
$    THEN
$    STARTPOS = NEWPOS
$    FIRSTTIME = 0
$ ENDIF
$!
$    WRITE SYS$OUTPUT "----------------------------------------"
$    NOW = f$time()
$    PERC = TOTALSUM * 100 / MAXCNT
$    IF 'MAXCNT .GT. 0
$    THEN
$    WRITE SYS$OUTPUT "Start: ''STARTPOS' Now: ''ACTUAL' TDrift: _
          ''TDRIFT' Counts: ''PEAKSUM' Total: _
          ''TOTALSUM'/''MAXCNT' (''PERC'%)..."
$    ELSE
$    WRITE SYS$OUTPUT "Start: ''STARTPOS' Now: ''ACTUAL' TDrift: _
          ''TDRIFT' Counts: ''PEAKSUM' Total: _
          ''TOTALSUM'/''MAXCNT'..."
$    ENDIF
$!
$ LOOP:
$!
$    WAIT 00:00:01
$!
$    define sys$output var_gatelock.out
$    sum 16 0 1
$    purge var_gatelock.out
$    rename var_gatelock.out;* var_gatelock.out;1
$!
$    define sys$output var_tdrift.out
$    evop var tdrift
$    purge var_tdrift.out
$    rename var_tdrift.out;* var_tdrift.out;1
$!
$    define sys$output var_maxcnt.out
$    evop var maxcnt
$    purge var_maxcnt.out
$    rename var_maxcnt.out;* var_maxcnt.out;1
$!
```

```
$    define sys$output var_totalsum.out
$    sum 5 0 1
$    purge var_totalsum.out
$    rename var_totalsum.out;* var_totalsum.out;1
$!
$    define sys$output MOTrack.out
$    evop var ilast
$    evop var chan
$    purge MOTrack.out
$    rename MOTrack.out;* MOTrack.out;1
$!
$    define sys$output sys$command
$!
$! +-------------------------------------------------------------+
$! | Now, we reread some of the values we've just stored, and    |
$! | place them in local variables.                              |
$! +-------------------------------------------------------------+
$!
$    open/read infile var_tdrift.out
$    read infile stvar
$    close infile
$!
$    len=f$locate("=", stvar)+1
$    len2=f$locate(".", stvar)
$    newvar=f$extract(len,len2-len,stvar)
$    tdrift=f$integer(newvar)
$!
$    open/read infile var_maxcnt.out
$    read infile stvar
$    close infile
$!
$    len=f$locate("=", stvar)+1
$    len2=f$locate(".", stvar)
$    newvar=f$extract(len,len2-len,stvar)
$    maxcnt=f$integer(newvar)
$!
```

```
$    open/read infile var_totalsum.out
$    read infile CENTERPOS
$    read infile CENTERPOS
$    read infile CENTERPOS
$    read infile CENTERPOS
$    read infile CENTERPOS
$    close infile
$!
$    len=f$locate(".", CENTERPOS)-21
$    TOTALSUM = f$extract(21,len,CENTERPOS)
$    TOTALSUM = f$integer(TOTALSUM)
$!
$    open/read infile var_gatelock.out
$    read infile CENTERPOS
$    read infile CENTERPOS
$    read infile CENTERPOS
$    read infile CENTERPOS
$    read infile CENTERPOS
$    close infile
$!
$    len=f$locate(".", CENTERPOS)-21
$    PEAKSUM = f$extract(21,len,CENTERPOS)
$    PEAKSUM = f$integer(PEAKSUM)
$    IF PEAKSUM .GT. 5000
$       THEN
$       WRITE SYS$OUTPUT "Too many counts in sp peak _
            (~ ''PEAKSUM'):  Clearing..."
$       cle 16
$       GOTO STARTUP
$    ENDIF
$    ACTUAL=f$extract(52,6,CENTERPOS)
$    ACTUAL=f$extract(52,6,CENTERPOS)
$    NEWPOS=f$extract(51,4,CENTERPOS)
$    NEWPOS=f$integer(NEWPOS)
$    NEWDRIFT = NEWPOS - STARTPOS + 'TDRIFT
$    NOW = f$time()
$    PERC = TOTALSUM * 100 / MAXCNT
$    IF 'MAXCNT .GT. 0
$    THEN
$    WRITE SYS$OUTPUT "Start: ''STARTPOS' Now: ''ACTUAL' TDrift: _
          ''TDRIFT' Counts: ''PEAKSUM' Total: _
          ''TOTALSUM'/''MAXCNT' (''PERC'%)..."
```

```
$    ELSE
$    WRITE SYS$OUTPUT "Start: ''STARTPOS' Now: ''ACTUAL' TDrift: _
           ''TDRIFT' Counts: ''PEAKSUM' Total: _
           ''TOTALSUM'/''MAXCNT'..."
$    ENDIF
$!
$    IF 'NEWPOS .NE. 'STARTPOS
$       THEN
$       EVOP VAR TDRIFT = 'NEWDRIFT
$       cle 16
$       GOTO STARTUP
$    ENDIF
$!
$    If 'MAXCNT .GT. 0
$       THEN
$       IF 'TOTALSUM .GE. 'MAXCNT
$          THEN
$          WRITE SYS$OUTPUT "Total counts reached: _
                 ''TOTALSUM' / ''MAXCNT'  Halting..."
$          vrctl -c halt
$          halt
$          EXIT
$       ENDIF
$    ENDIF
$!
$ GOTO LOOP
$ exit
```

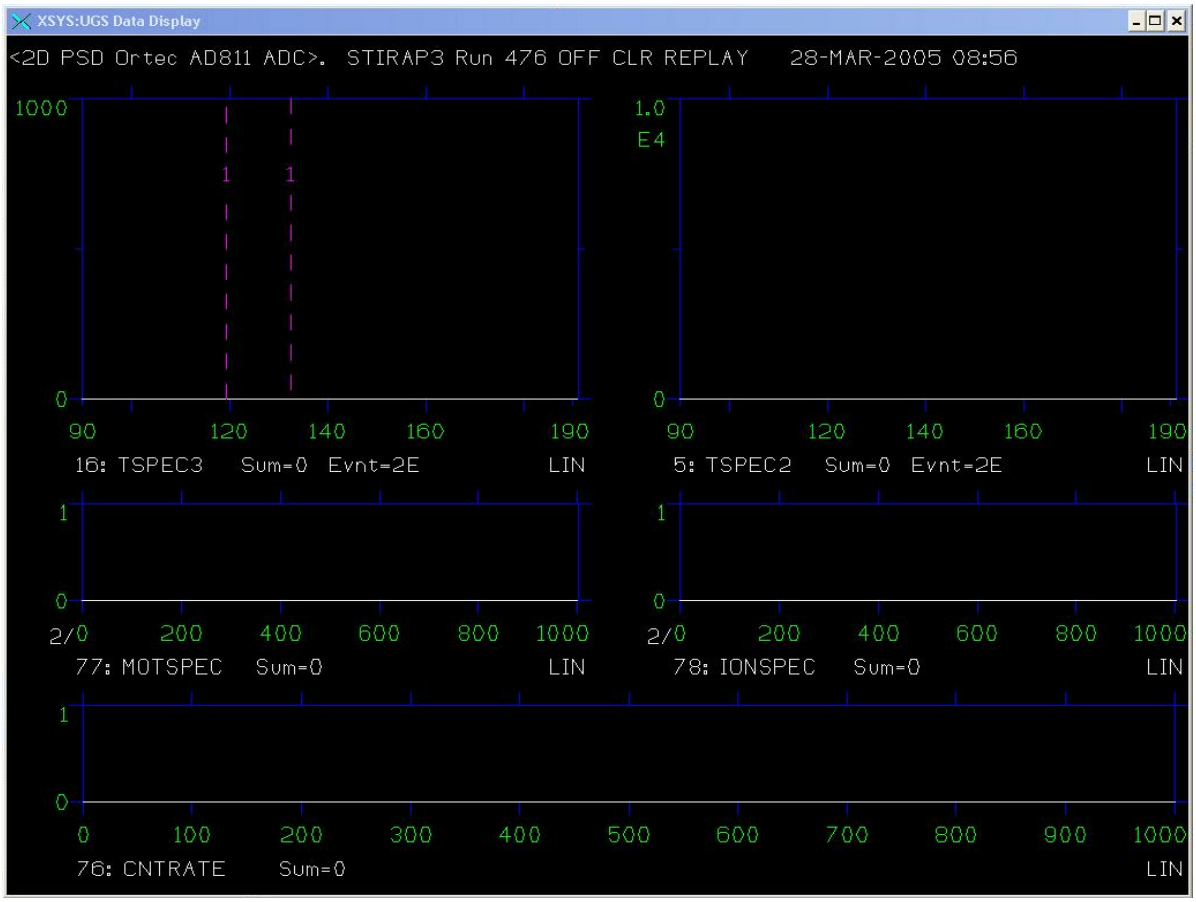

Figure C.2: The STOPDRIFT.COM macro in action. Click the image to view the film, or click here to launch an external media player to see the film at full scale.

A short film clip of the STOPDRIFT.COM macro in action is shown in Fig. C.2. The left-hand data area (labeled 16: 'TSPEC3') contains two vertical purple bars representing the position of window used to measure the peak for drift purposes. As this peak reaches continues to grow, smaller drift rates will be less noticeable, so every few minutes this data area is cleared, and the peak drift detection starts fresh.

C.3.10 STOREVARS.COM

This program allows one to store the current variables and their settings to disk. They can be reviewed later using the `SHOWVARS.COM` macro, or reloaded using the `SETVARS.COM` macro.

```
$! +-------------------------------------------------------------+
$! |      Filename: storevars.com                                |
$! | Last modified: 09/21/04 by hcamp                            |
$! |                                                             |
$! |   Description: This file stores relevant variables after    |
$! |                they've been set via 'evop var [varname]' (or |
$! |                initial setup from the .EVL file).  These same|
$! |                vars can be reloaded using the 'readvars.com' |
$! |                file.                                        |
$! |                                                             |
$! |         Usage: storevars [run #]                            |
$! |                                                             |
$! |                The run# is optional.  Supplying it will store|
$! |                the current active variables into a file     |
$! |                associated with the indicated run#.          |
$! |                                                             |
$! |                Without supplying run#, all current values   |
$! |                will be written to a 'default' storage file.  |
$! |                This allows one to read/write variables while |
$! |                testing parameters, and avoid accumulating   |
$! |                many variable storage files for a given run.  |
$! +-------------------------------------------------------------+
$!
$ ON ERROR THEN CONTINUE
$!
$ IF p1 .EQS. ""
$    THEN
$    DEFINE SYS$OUTPUT data:[stirap3.vars]temporary.var
$    ELSE
$    DEFINE SYS$OUTPUT data:[stirap3.vars]run_'p1'.var
$ ENDIF
$!
```

```
$ @data:[stirap3.com]showvars
$ DEFINE SYS$OUTPUT SYS$COMMAND
$ WRITE SYS$OUTPUT "+-----------------------------------------+"
$ WRITE SYS$OUTPUT "  The following variables have been stored"
$ IF p1 .EQS. ""
$    THEN
$    WRITE SYS$OUTPUT "  in the file temporary.var:"
$    ELSE
$    WRITE SYS$OUTPUT "  in the file run_''p1'.var:"
$ ENDIF
$ WRITE SYS$OUTPUT "+-----------------------------------------+"
$ IF p1 .EQS. ""
$    THEN
$       @data:[stirap3.com]showvars
$    ELSE
$       @data:[stirap3.com]showvars 'p1'
$ ENDIF
$ EXIT
```

C.3.11 VARS.COM

This is simply a "wrapper" program, creating a simple interface to control variables while taking (or rereading) data. Variable values can be stored for a particular data session, or recalled from previous sessions. Generic storage allows one to temporarily store or retrieve current variable settings without needing to associate such settings with a given run number. This is useful for tasks such as debugging or alignment.

Note that "_" at the end of a line indicates a wrapped entry for printing purposes. This character should be removed, and the two lines concatenated if the program is to be used.

```
$! +-------------------------------------------------------------+
$! |      Filename: vars.com                                     |
$! | Last modified: 09/21/2004 by hcamp                          |
$! |                                                             |
$! |   Description: Vars is a 'switchboard' to redirect requests |
$! |                to a variety of destinations.                |
$! |                                                             |
$! |         Usage: vars <type> [run #]                          |
$! |                                                             |
$! |                The <type> argument is not optional.  Valid  |
$! |                types include -                              |
$! |                                                             |
$! |                 READ - Resets variables to previous values  |
$! |                STORE - Saves current variables to disk      |
$! |                 SHOW - Lists the current variable values    |
$! |                PRINT - Prints the current variable values   |
$! |                                                             |
$! |                SHOW will output raw variable values to the  |
$! |                screen, while PRINT will 'pretty-print' the  |
$! |                variables for printing (used in the lab      |
$! |                notebook).                                   |
$! |                                                             |
$! |                Run# is optional.  Without it, the 'default' |
$! |                variables currently stored to file will be   |
$! |                used.  Otherwise, the indicated run # can be |
$! |                used for reading/writing.                    |
$! |                                                             |
```

```
$! |                Alternately, vars can be used to set/display  |
$! |                gate parameters:                               |
$! |                                                               |
$! |                vars GATE [<min_x> <max_x> <min_y> <max_y>]    |
$! |                                                               |
$! |                    GATE - Sets various gate parameters        |
$! |                                                               |
$! |                 <min_x> - Minimum X Gate Channel              |
$! |                 <max_x> - Minimum X Gate Channel              |
$! |                 <min_y> - Maximum Y Gate Channel              |
$! |                 <max_y> - Maximum Y Gate Channel              |
$! |                                                               |
$! |                Min/Max values are optional.  Without them,    |
$! |                (supplying only 'vars GATE') vars will return |
$! |                a list of current gate settings.               |
$! +---------------------------------------------------------------+
$!
$  p1 = F$EDIT(p1,"UPCASE")
$  IF p1 .EQS. "READ"
$     THEN
$        @data:[stirap3.com]readvars 'p2'
$     EXIT
$  ENDIF
$  IF p1 .EQS. "STORE"
$     THEN
$        @data:[stirap3.com]storevars 'p2'
$     EXIT
$  ENDIF
$  IF p1 .EQS. "SHOW"
$     THEN
$        @data:[stirap3.com]showvars 'p2'
$     EXIT
$  ENDIF
$  IF p1 .EQS. "PRINT"
$     THEN
$        @data:[stirap3.com]printvars 'p2'
$     EXIT
$  ENDIF
```

```
$  IF p1 .EQS. "GATE"
$     THEN
$        @data:[stirap3.com]setgate 'p2' 'p3' 'p4' 'p5' 'p6'
$     EXIT
$  ENDIF
$!
$! +-------------------------------------------------------------+
$! | If nothing else matches, print some instructions:           |
$! +-------------------------------------------------------------+
$!
$  WRITE SYS$OUTPUT " "
$  WRITE SYS$OUTPUT "Usage: "
$  WRITE SYS$OUTPUT " "
$  WRITE SYS$OUTPUT "  vars <type> [run #]"
$  WRITE SYS$OUTPUT " "
$  WRITE SYS$OUTPUT "  <type> - READ, STORE, SHOW, PRINT"
$  WRITE SYS$OUTPUT "  [run #] is optional"
$  WRITE SYS$OUTPUT " "
$  WRITE SYS$OUTPUT " Alternately..."
$  WRITE SYS$OUTPUT " "
$  WRITE SYS$OUTPUT "  vars GATE <d, r, p> <min_x> <max_x> _
         <min_y> <max_y>"
$  WRITE SYS$OUTPUT " "
$  WRITE SYS$OUTPUT "  <d, r, p> - D-state, Recoil, Projectile _
         Gates"
$  WRITE SYS$OUTPUT "    <min_x> - Minimum X Gate Channel"
$  WRITE SYS$OUTPUT "    <max_x> - Maximum X Gate Channel"
$  WRITE SYS$OUTPUT "    <min_y> - Minimum Y Gate Channel"
$  WRITE SYS$OUTPUT "    <max_y> - Maximum Y Gate Channel"
$  WRITE SYS$OUTPUT " "
$!
```

Appendix D

Theoretical Simulation Program

The Mathematica software was used to numerically solve the system of coupled, first-order differential equations making up the 3-level system of interest. Two relevant codes are presented in this appendix, followed by a brief discussion regarding how Mathematica handles accuracy and precision:

- **3LEVEL.NB**

  Contains the engine necessary for setting up the density matrix, and numerically solving for the population dynamics. The machinery contained in this code allows for a variety of diagnostic output, including monitoring maximum population in each state ($5s$, $5p$, and $4d$) as well as off-diagonal matrix element evolution. There are an assortment of default graphs created, or the actual numerical interpolation function can be used directly for specialized use of the solutions.

- **RANDOMSPACE.NB**

  This is actually used in conjunction with `3LEVEL.NB`, and randomly steps through the 7-dimensional parameter space, as defined by the boundary conditions at the beginning of this code. It can be run as a separate notebook, or can be pasted into the `3LEVEL.NB` file and run directly following the compilation of the initial code.

D.1 3LEVEL.NB

This Mathematica code cannot be directly placed inside a notebook and run accordingly. The `3LEVEL.NB` file itself is broken into a hierarchy of Mathematica notebook cells, some of which are explicitly set as comment text (and thus are not evaluated at runtime). The code is presented here for archival purposes, but the actual `3LEVEL.NB` file should be consulted. Commented text is shown in brown.

```
    Filename: 3Level.nb
        Date: 04/27/2004
      Author:  hcamp
        Desc: 3Level is a simulation of coherent excitation
              populations using the STIRAP method of transfer
              from a ground state to an excited state.

(* Header Definitions *)
(* 3Level Program Settings *)

Four steps are conducted here:
(1) Turn off spelling errors that don't apply.  Mathematica
generates 'possible spelling errors' if two variable names
are too closely matched.  (2) Turn off "ssym" warnings.
When we clear our variables, they may not be declared yet
(that's okay).  So, we can ignore this message.  (3) Load
the Graphics3D package.  This allows us to use extended
plotting routines.  (4) Set the directory properly.


Off[General::"spell1"];
Off[General::"spell"];
Off[Clear::"ssym"];
<<Graphics`Graphics3D`

If[
Directory[]=="N:/Apps/Mathematica/4.1",
SetDirectory["y:/theory/stirap/3level"];,
SetDirectory["c:/files/mathematica"];
];

(* Program Setup Routines *)

At present, only a single setup variable exists.  This variable,
SpEm, controls whether Spontaneous Emission is ON or OFF in the
calculations.  If SpEm = 1, then spontaneous emission is on.
Otherwise, SpEm should be set to 0.  Note, however, that in reality
```

the spontaneous emission is not actually turned off. The decay rates are simply made to be very large so that on our short timescale (1000 ns) the decay does not seem to be present. This was done to avoid problems of dividing by zero.

```
SpEm = 1;
```

(* General Density Matrix Manipulations *)

This section sets up the density matrix, and prepares the equations for solving them in Mathematica. The steps are as follows:

(1) Clear any existing variable definitions. This is necessary since we routinely rerun the code during an active session, and don't want lingering values to screw things up. (2) Set up the Hamiltonian H, and the Density Matrix 'R' (Rho). (3) Define impossible Einstein A coefficients to be 0. It should be pretty obvious that we cannot have population transfer from, say, level 3 to level 1. Hence, $A_{3,1}$ should be 0, etc. (4) Find Hρ, Γ, ρ, and ρ'. The time-derivative of the density matrix is now ready for use. (5) Set up our six equations to be solved. These will be $\rho_{1,1}$, $\rho_{1,2}$, $\rho_{1,3}$, $\rho_{2,2}$, $\rho_{2,3}$, and $\rho_{3,3}$. The other matrix elements are complex conjugates of the others. (6) Replace all $\rho_{2,1}$, $\rho_{3,1}$ and $\rho_{3,2}$ terms with their complex conjugates. This way, we have six equations and only six unknowns. (7) Set the initial conditions of the system. Population in all elements except the ground state $\rho_{1,1}$ should be zero. $\rho_{1,1}$ = 1 since all the population is in the $5s$ state to begin with.

The remainder of the program is mostly setting up functions and plots. All the mathematics have been performed here. The only crucial step remaining is to numerically solve these six equations for their six unknowns.

```
Clear[H, R, Hρ, Γρ, ρ̇, ρ11a, ρ12a, ρ12b, ρ13a, ρ13b, ρ22a, ρ23a, ρ23b,
 eq1, eq2, eq3, eq4, eq5, eq6, eq7, eq8, Ω1, Ω2, Δ1, Δ2, ρ1,1, ρ1,2,
 ρ1,3, ρ2,1, ρ2,2, ρ2,3, ρ3,1, ρ3,2, ρ3,3];
```

$$\mathrm{H} := \begin{pmatrix} 0 & \Omega_1[t] & 0 \\ \Omega_1[t] & 2\Delta_1 & \Omega_2[t] \\ 0 & \Omega_2[t] & 2\Delta_2 \end{pmatrix};$$

$$\mathrm{R} := \begin{pmatrix} \rho_{1,1}[t] & \rho_{1,2}[t] & \rho_{1,3}[t] \\ \rho_{2,1}[t] & \rho_{2,2}[t] & \rho_{2,3}[t] \\ \rho_{3,1}[t] & \rho_{3,2}[t] & \rho_{3,3}[t] \end{pmatrix};$$

```
A1,1:=0;
A1,2:=0;
A1,3:=0;
A2,2:=0;
A2,3:=0;
A3,1:=0;
A3,3:=0;

Hρ := H.R - R.H //Simplify;

Γρ := Table[ρi,j[t] ∑k=1^3 1/2(Ai,k+Aj,k)−KroneckerDelta[i, j]
 ∑k=1^3 ρk,k[t] Ak,i, i,3, j,3] //Simplify;

ρ' := −i/2 Hρ − ρH //Simplify;

eq1 = ρ1,1'[t] == ρ'[[1, 1]] + 1/2 ρ3,3[t]^2 PLossdd;
eq2 = ρ1,2'[t] == ρ'[[1, 2]] + 1/4 ρ3,3[t]^2 PLossdd;
eq3 = ρ1,3'[t] == ρ'[[1, 3]] + 1/4 ρ3,3[t]^2 PLossdd;
eq4 = ρ2,2'[t] == ρ'[[2, 2]] - 1/2 ρ2,2[t] ρ3,3[t] ALosspd;
eq5 = ρ2,3'[t] == ρ'[[2, 3]];
eq6 = ρ3,3'[t] == ρ'[[3, 3]] - ρ2,2[t] ρ3,3[t] ALosspd
 −ρ3,3[t]^2 ALossdd − ρ3,3[t]^2 PLossdd;

eq1 = ReplaceAll[eq1, {ρ2,1[t] → Conjugate[ρ1,2[t]], ρ3,1[t]
 → Conjugate[ρ1,3[t]], ρ3,2[t]→Conjugate[ρ2,3[t]]}];

eq2 = ReplaceAll[eq1, {ρ2,1[t] → Conjugate[ρ1,2[t]], ρ3,1[t]
 → Conjugate[ρ1,3[t]], ρ3,2[t]→Conjugate[ρ2,3[t]]}];

eq3 = ReplaceAll[eq1, {ρ2,1[t] → Conjugate[ρ1,2[t]], ρ3,1[t]
 → Conjugate[ρ1,3[t]], ρ3,2[t]→Conjugate[ρ2,3[t]]}];

eq4 = ReplaceAll[eq1, {ρ2,1[t] → Conjugate[ρ1,2[t]], ρ3,1[t]
 → Conjugate[ρ1,3[t]], ρ3,2[t]→Conjugate[ρ2,3[t]]}];

eq5 = ReplaceAll[eq1, {ρ2,1[t] → Conjugate[ρ1,2[t]], ρ3,1[t]
 → Conjugate[ρ1,3[t]], ρ3,2[t]→Conjugate[ρ2,3[t]]}];

eq6 = ReplaceAll[eq1, {ρ2,1[t] → Conjugate[ρ1,2[t]], ρ3,1[t]
 → Conjugate[ρ1,3[t]], ρ3,2[t]→Conjugate[ρ2,3[t]]}];
```

ic1 = $\rho_{1,1}$[0] == 1;
ic2 = $\rho_{1,2}$[0] == 0;
ic3 = $\rho_{1,3}$[0] == 0;
ic4 = $\rho_{2,2}$[0] == 0;
ic5 = $\rho_{2,3}$[0] == 0;
ic6 = $\rho_{3,3}$[0] == 0;

(* Functions *)

We have defined a number of useful functions for analyzing the results of our density matrix. They are:

(1) SolveEqns: This is the 'brain' of the entire program. SolveEqns numerically solves the system of six equations and their six unknowns. The result is a function called 'nsoln' which contains the numerical solutions for each of our six terms ($\rho_{1,1}$[t], $\rho_{1,2}$[t], $\rho_{1,3}$[t], $\rho_{2,2}$[t], $\rho_{2,3}$[t] and $\rho_{3,3}$[t]). The diagonal terms ($\rho_{1,1}$[t], $\rho_{2,2}$[t], and $\rho_{3,3}$[t]) describe the fractional population for each level ($5s$, $5p$ and $4d$), while the off-diagonal terms represent the decoherence effects coupling the states together. We make heavy use of the 'nsoln' results throughout the rest of the program.

(2) Pop[L, t]: A simple function requiring a population level (s, p, or d) and a corresponding time argument, and returns the fractional population in the level at that time. For example, Pop["s", 300] will return the fractional population in the ground state at time t=300ns. Level arguments are "s", "p" and "d".

(3) θ[t]: This returns the mixing angle for the system at time t.

(4) ADC: The ADC function stands for 'ADiabatic Condition". For each run, you can check to get a rough idea what the adiabatic condition is based on this simple calculation. The result depends on whether the maximum Ω_1 is larger or smaller than the maximum Ω_2.

(5) CalcStats: In order to make analysis of a given set of parameters easier, we simply perform some simple measurements of statistics and bunch them together using this function. Specifically, CalcStats measures the following: Maximum s, p, and d percentages along with corresponding time of occurrence, maximum Ω_1, Ω_2 and θ values along with corresponding time of occurrence, areas beneath $5s$, $5p$ and $4d$ curves, and the ADC condition.

(6) ShowStats: This function simply prints the values that were calculated using CalcStats in some 'pretty' format.

(7) VecPop2D: This sets up a table of values containing the populations of $5p$ and $4d$ levels for discreet time values. This table will be used later in plotting a 2D representation of the population transfer.

(8) VecPop3D: Similar to VecPop2D, this sets up a table of values containing the populations of all three populations ($5s$, $5p$ and $4d$). This table will be used later in plotting a 3D representation of the population transfer.

(9) DoItAll: This is a combined set of several functions placed in a single wrapper for ease-of-use. The DoItAll function will run 'SolveEqns' and 'CalcStats', then create a graphics array to display the generated statistics and plots.

Ω_1 and Ω_1 are both in units of GHz for the following SolveEqns Function:

```
SolveEqns := { nsoln = NDSolve[ { eq1, eq2, eq3, eq4, eq5, eq6,
 ic1, ic2, ic3, ic4, ic5, ic6 }, {ρ1,1, ρ1,2, ρ1,3, ρ2,2, ρ2,3, ρ3,3 },
 {t, TMin, TMax}, MaxSteps -> 100000 ]; };

Pop := Module[ {L, t}, Switch[ L, s, ρ1,1[t] /.  nosln[[1]], p,
 ρ2,2[t] /.  nosln[[1]], d, ρ3,3[t] /.  nosln[[1]], _, "Error" ] ] ];

CalcStats := Module[ {TestS, TestP, TestD, Testθ, Testϕ, TestΩ1,
 TestΩ2, t},

Off[General::"spell1"];

TestS = TestP = TestD = Testθ = TestΩ1 = TestΩ2 = 0;

TotAreaS = TotAreaP = TotAreaD = 0;

MaxS = MaxP = MaxD = Maxθ = Maxϕ = MaxΩ1 = MaxΩ2 = 0;

MaxSPos = MaxPPos = MaxDPos = MaxθPos = MaxϕPos = MaxΩPos1 =
 MaxΩPos2 = 0;

For[
 t = TMin,
 t < TMax,
 TestS = Re[ρ1,1[t] /.  nsoln[[1]]];
 TestP = Re[ρ2,2[t] /.  nsoln[[1]]];
 TestD = Re[ρ3,3[t] /.  nsoln[[1]]];
```

```
 TotAreaS = TotAreaS + TestS;
 TotAreaP = TotAreaP + TestP;
 TotAreaD = TotAreaD + TestD;

 Testθ = θ'[Evaluate[t]];
 Testϕ = ϕ'[Evaluate[t]];
 TestΩ1 = Ω1'[Evaluate[t]];
 TestΩ2 = Ω2'[Evaluate[t]];

 If[TestS > MaxS, MaxS = TestS; MaxSPos = t];
 If[TestP > MaxP, MaxP = TestP; MaxPPos = t];
 If[TestD > MaxD, MaxD = TestD; MaxDPos = t];
 If[Testθ > Maxθ, Maxθ = Testθ; MaxθPos = t];
 If[Testϕ > Maxϕ, Maxϕ = Testϕ; MaxϕPos = t];
 If[TestΩ1 > MaxΩ1, MaxΩ1 = TestΩ1; MaxΩPos1 = t];
 If[TestΩ2 > MaxΩ2, MaxΩ2 = TestΩ2; MaxΩPos2 = t];

 t++
]:

Stat1a := StringJoin["Spontaneous Decay:  ", If[SpEm > 0, "ON",
 "OFF"]];

Stat1b := "";

Stat2a := StringJoin["Maximum %s:  ", ToString[MaxS], " (",
 ToString[MaxSPos], " ns)"];

Stat2b := StringJoin["%s Area:  ", ToString[TotAreaS]];

Stat3a := StringJoin["Maximum %p:  ", ToString[MaxP], " (",
 ToString[MaxPPos], " ns)"];

Stat3b := StringJoin["%p Area:  ", ToString[TotAreaP]];

Stat4a := StringJoin["Maximum %d:  ", ToString[MaxD], " (",
 ToString[MaxDPos], " ns)"];

Stat4b := StringJoin["%d Area:  ", ToString[TotAreaD]];

Stat5a := StringJoin["Maximum Ω1:  ", ToString[MaxΩ1], " GHz (",
 ToString[MaxΩPos1], " ns)"];

Stat5b := StringJoin["Maximum θ':  ", ToString[Maxθ], " (",
 ToString[MaxθPos], " ns)"];
```

```
Stat6a := StringJoin["Maximum Ω2:  ", ToString[MaxΩ2], " GHz (",
 ToString[MaxΩPos2], " ns)"];

Stat6b := StringJoin["Maximum ϕ':  ", ToString[Maxϕ], " (",
 ToString[MaxϕPos], " ns)"];

];

ShowStats := {
 CalcStats;
 Print[" Spontaneous Decay:  ", If[SpEm > 0, "ON", "OFF"]];
 Print[" Maximum %s:  ", MaxS, " (", MaxSPos, " ns) Area:  ",
 TotAreaS];
 Print[" Maximum %p:  ", MaxP, " (", MaxPPos, " ns) Area:  ",
 TotAreaP];
 Print[" Maximum %d:  ", MaxD, " (", MaxDPos, " ns) Area:  ",
 TotAreaD];
 Print[" Maximum θ:  ", Maxθ, " (", Maxθ Pos, " ns) Area:  ",
 TotAreaθ];
 Print[" Maximum ϕ:  ", Maxϕ, " (", Maxϕ Pos, " ns) Area:  ",
 TotAreaϕ];
 Print[" Maximum Ω1:  ", MaxΩ1, " (", MaxΩ1Pos, " ns) Area:  ",
 TotAreaΩ1];
 Print[" Maximum Ω2:  ", MaxΩ2, " (", MaxΩ2Pos, " ns) Area:  ",
 TotAreaΩ2];
};

VecPop2D := Module[ {t}, Table[ { Pop[p, t], Pop[d, t] },
 {t, TMin, TMax} ] ];

VecPop3D := Module[ {t}, Table[ { Pop[s, t], Pop[p, t] },
Pop[d, t] } {t, TMin, TMax, 1} ] ];

DoItAll := {
 SolveEqns;
 PlotIt;
 CalcStats;
 Show[GraphicsArray[{{StatPlot_a, StatPlot_b}, {RabiPulse,
 TimeEvolution}, {MixingRate, RabiTimeCombo}, {Vec2D, Vec3D},
 {Adiab}}]];
};

(* Plots *)

There are several different ways to plot the data.  Here are the
existing plot functions:
```

(1) RabiPulse: 2D graph showing Ω_1 and Ω_2 as a function of time.

(2) TimeEvolution: 2D graph showing populations s, p and d as a function of time.

(3) RabiTimeCombo: 2D graph combining (1) and (2) - s, p, d, Ω_1 and Ω_2 are all shows versus time.

(4) MixingRate: 2D graph showing the rate of change of the mixing angle as a function of time.

(5) Vec2D: 2D graph showing the p and d population dynamics.

(6) Vec3D: 3D graph showing the s, p and d population dynamics.

(7) StatPlot$_a$: GraphicsArray (part 1) showing relevant data from CalcStats.

(8) StatPlot$_b$: GraphicsArray (part 2) showing relevant data from CalcStats.

```
RabPulse :=
 Plot[ {Ω1[t], Ω2[t]}, {t, TMin, TMax}, Frame → True, FrameLabel
 → { "Time (ns)", "Frequency (GHz)", "Rabi Frequency vs.  Time",
 "" }, PlotRange → {-0.001, Max[α1, α2] * 1.1}, PlotStyle
 → { RGBColor[0.0, 0.75, 0.75], Thickness[0.01]}, {RGBColor[0.75,
 0, 0.75], Thickness[0.01]}}, DisplayFunction → Identity
];

TimeEvolution :=
 Plot[ {ρ1,1[t] /.  nsoln[[1]], ρ2,2[t] /.  nsoln[[1]], ρ3,3[t] /.
 nsoln[[1]]}, {t, TMin, TMax}, Frame → True, FrameLabel → {
 "Time (ns)", "Population", "Population vs.  Time", " " },
 PlotRange → {-0.001,1}, PlotStyle → {{RGBColor[1,0,0],
 Thickness[0.01]}, {RGBColor[0,1,0], Thickness[0.01]{,
 {RGBColor[0,0,1], Thickness[0.01]}}, DisplayFunction → Identity
];

RabiTimeCombo :=
 Plot[ {ρ1,1[t] /.  nsoln[[1]], ρ2,2[t] /.  nsoln[[1]], ρ3,3[t] /.
 nsoln[[1]], Ω1[t]/(MaxΩ1 + MaxΩ1/10), Ω2[t]/(MaxΩ1 + MaxΩ1/10)},
 {t, TMin, TMax}, Frame → True, FrameLabel → { "Time (ns)",
 "Population", "Rabi Frequency and Population vs.  Time", "Rabi
 Frequency (GHz)" }, PlotRange → {-0.0001, 1}, FrameTicks → {
 Automatic, Range[0,1,0.2], Automatic, {{0.0, "0.000"}, {0.2,
 NumberForm[0.2 (MaxΩ1 + MaxΩ1/10), {100,4}]}, {0.4,
 NumberForm[0.4 (MaxΩ1 + MaxΩ1/10), {100,4}]}, {0.6,
```

```
 NumberForm[0.6 (MaxΩ1 + MaxΩ1/10), {100,4}]}, {0.8,
 NumberForm[0.8 (MaxΩ1 + MaxΩ1/10), {100,4}]}, {0.9999,
 NumberForm[1.0 (MaxΩ1 + MaxΩ1/10), {100,4}]}}}, PlotStyle → {{
 RGBColor[1,0,0], Thickness[0.01]}, {RGBColor[0,1,0],
 Thickness[0.01]}, {RGBColor[0,0,1], Thickness[0.01]},
 {RGBColor[0,0.75,0.75], Thickness[0.005], Dashing[{0.01, 0.01}]},
 {RGBColor[0.75, 0, 0.75], Thickness[0.005], Dashing[{0.01, 0.01}]
 }}, DisplayFunction → Identity
];


MixingRate :=
 Plot[ {θ'[t], Abs[φ'[t]]}, {t, TMin, TMax}, Frame → True,
 FrameLabel → {"Time (ns)", "dθ, dφ", "Rate of Change of Mixing
 Angle vs.  Time", ""}, PlotRange → {-0.0001, Maxθ + Maxθ/10},
 PlotStyle → {{ RGBColor[1,0.75,0], Thickness[0.01]},
 {RGBColor[1,0.25,0], Thickness[0.01]}}, DisplayFunction →
 Identity
];


Adiab :=
Plot[ {Abs[φ'[t]], θ'[t]cosφ[t], θ'[t]sinφ[t], Abs[λ_[t]]}, {t,
 TMin, TMax}, PlotRange → {0, Abs[λ_[(TMax-TMin)/2]] +
 Abs[λ_[(TMax-TMin)/2]] * 0.1}, PlotStyle → {{RGBColor[1,0,0],
 Thickness[0.01]}, {RGBColor[0,0.66,0], Thickness[0.01]},
 {RGBColor[0,0,1], Thickness[0.01]}, {RGBColor[1,0.75,0],
 Thickness[0.01]}}, Frame → True, FrameLabel → {"Time (ns)",
 "Arbitrary Units"}, PlotLabel → "Adiabatic Check",
 DisplayFunction → Identity
];


Vec2D :=
 ListPlot[ VecPop2D, AxesLabel → {p, d}, PlotStyle → {
 RGBColor[0.66,0,0.75], Thickness[0.01]}, Frame → True,
 FrameLabel → { "← S-Pop P-Pop →", "← S-Pop D-Pop →", "2D
 Vector Representation of Population", "" }, PlotRange → {{-0.001,
 1}, {-0.001, 1}}, DisplayFunction → Identity
];


Vec3D :=
 ScatterPlot3D[ VecPop3D, PlotRange → {{0, 1}, {0,1}, {0,1}},
 ViewPoint → {-0.75, -2, 0.75}, ViewCenter → {0.5, 0.5, 0.5},
 BoxStyle → Dashing[{0.02, 0.02}], Axes → True, AxesStyle →
 Thickness[0.01], AxesEdge → {{-1, -1}, {-1, -1}, {1, -1}},
 AxesLabel → {"S-Pop", "P-Pop", "D-Pop"}, PlotStyle →
 {RGBColor[0.66,0,0.75], Thickness[0.01]}, PlotLabel → "3D Vector
 Representation of Population", Displayfunction → Identity
];
```

```
StatPlot_a :=
 Plot[ x, {x, 0, 1}, Axes → False, PlotStyle → RGBColor[1,1,1],
 Prolog → { Text[Stat1a, {0.5, 0.90}], Text[Stat2a, {0.5, 0.75}],
 Text[Stat3a, {0.5, 0.60}], Text[Stat4a, {0.5, 0.45}], Text[Stat5a,
 {0.5, 0.30}], Text[Stat6a, {0.5, 0.15}]}, DisplayFunction →
 Identity
];

StatPlot_b :=
 Plot[ x, {x, 0, 1}, Axes → False, PlotStyle → RGBColor[1,1,1],
 Prolog → { Text[Stat1b, {0.5, 0.90}], Text[Stat2b, {0.5, 0.75}],
 Text[Stat3b, {0.5, 0.60}], Text[Stat4b, {0.5, 0.45}], Text[Stat5b,
 {0.5, 0.30}], Text[Stat6b, {0.5, 0.15}]}, DisplayFunction →
 Identity
];

(* Variable Definitions *)
```

The Variable Definitions section is broken into two parts: Fixed Parameters and STIRAP Parameters. The first section contains values that will not be changing on a constant basis. These values set up the specific system we are looking at (namely, ^{87}Rb ladder excitation from $5s$ to $5p$ to $4d$). The second section contains values that are most likely to change, depending on the experimental conditions we wish to examine.

```
(* Fixed Parameters *)
```

This section contains values relevant to our specific system of study (namely, ^{87}Rb ladder excitation from $5s$ to $5p$ to $4d$). Note that this where we examine the SpEm variable set back at the beginning to check whether we should use spontaneous emission or not. If SpEm is > 0, normal decay rates for d-p and p-s will be used. Otherwise, extended decay rates are used (four orders of magnitude larger).

Also note this is where we define the shape of Ω_1 and Ω_2. There is no reason the Rabi frequencies must be Gaussian. This was chosen as a convienence -- experimentally the pulse shapes are very similar. In fact, code was originally implemented allowing us to record actual pulse shapes and implement them here, although this proved unnecessary. The pulse shapes were close enough not to make any difference.

Lastly, the variable I_{match} allows one to match the Rabi frequencies Ω_1 and Ω_2 by setting I_1 and then selecting I_2 to be equal to I_{match}. This is very useful since, experimentally, it is much easier to measure intensities than Rabi frequencies, but in reality we care

about what the Rabi frequencies are doing more than we care what the intensities are.

The following units apply to the variables below: $\Omega_{1,2}$ [GHz], $\lambda_{1,2}$ [nm], $A_{2,1/3,2}$ [GHz], $\alpha_{1,2}$ [GHz]

Ω_1[t_] = α_1 Exp[-2.773 (t + 1/2 Delay - TOffset)^2/PulseWidth$_1^2$];

Ω_2[t_] = α_1 Exp[-2.773 (t - 1/2 Delay - TOffset)^2/PulseWidth$_2^2$];

Ω[t_] = $\sqrt{\Omega_1[t]^2 + \Omega_2[t]^2}$;

Ω_p[t_] = $\sqrt{\Omega[t]^2 + \Delta_1^2}$;

θ[t_] = ArcTan[Ω_1[t]/Ω_2[t]];

cosϕ[ϕ_] = $\sqrt{1/2(1 + \Delta_1/\Omega_p[t])}$;

sinϕ[ϕ_] = $\sqrt{1/2(1 - \Delta_1/\Omega_p[t])}$;

ϕ[t_] = 1/2 ArcTan[Ω[t] / Δ_1];

λ_-[t_] = 1/2 (Δ_1 - Ω_p[t]);

λ_+[t_] = 1/2 (Δ_1 + Ω_p[t]);

λ_1 = 780;

λ_2 = 1529;

$A_{2,1}$ = If[SpEn > 0, 3.76 10^{-2}, 3.76 10^{-6}];

$A_{3,2}$ = If[SpEn > 0, 1.19 10^{-2}, 1.19 10^{-6}];

ALoss$_{pd}$ = 1.19 10^{-32}];

ALoss$_{dd}$ = 1.19 10^{-32}];

PLoss$_{dd}$ = 1.19 10^{-32}];

α_1 = If[SpEm > 0, $\frac{1.5510^{-7}}{1000}\sqrt{\lambda_1^3 I_1(10^9 A_{2,1})}$, $\frac{1.5510^{-5}}{1000}\sqrt{\lambda_1^3 I_1(10^9 A_{2,1})}$];

α_2 = If[SpEm > 0, $\frac{1.5510^{-7}}{1000}\sqrt{\lambda_2^3 I_2(10^9 A_{2,1})}$, $\frac{1.5510^{-5}}{1000}\sqrt{\lambda_2^3 I_2(10^9 A_{2,1})}$];

```
I_Match = I_1 λ_1^3 A_2,1 / λ_2^3 A_3,2;

(* STIRAP Parameters *)

These are the important variables controlling STIRAP. Each can be
independently changed.  It is important to note, however, that if we
desire to change other values besides the ones listed here (say, for
example, α_1), it will most likely be necessary to quit the
Mathematica kernel and recompile the equations above.  Otherwise,
the changes will not take place.  The variables below, however, are
not compiled anywhere, and can be changed without needing to
recompile our equations.

The following units apply to the variables below:  I_1,2 [mw/cm^2],
PulseWidth_1,2 [ns], Delay [ns], Δ_1,2 [GRad], TMin/TMax,TOffset [ns]

(* Note:  Δ_1 contains a value like "53".  This is the measured
LINEAR frequency (GHz), but Δ_1,2 are ANGULAR frequencies (GRad)
-- hence the explicit 2π *)

I_1 = 55 * 776 * 0.148153;
I_2 = I_Match;

PulseWidth_1 = 50;
PulseWidth_2 = 50;

Delay = -50;

Δ_1 = 53/1000 2π;
Δ_2 = 0/1000 2π;

TMin = 0;
TMax = 800;
TOffset = 400;

(***************************)
(* STIRAP Execution Routine *)
(***************************)

DoItAll;
```

D.2 RANDOMSPACE.NB

Trying to efficiently measure the 7-dimensional parameter space is not easy. Brute force calculations covering each parameter in a comprehensive manner are far too time-consuming to be feasable for the interative code presented in the previous section. Thus, in order to make the search through the 7-dimensional parameter space reasonable, a short program was written that randomly selects values for each of the seven desired variables. Range limits are placed on each variable, based on experimental feasability, and the program was iteratively run several hundred thousand times. The program is shown here.

```
(* Random Parameter Search *)

Module[
 {Count, t},
 SeedRandom[58732];
 For[
 Count = 0,
 Count < 200000,
 {
 I1 = Random[Integer, {5,25000}];
 I2 = Random[Integer, {5,25000}];
 Δ1 = Random[Real, {-150/1000 * 2π, 150/1000 * 2π}];
 Δ2 = Random[Real, {-150/1000 * 2π, 150/1000 * 2π}];
 Delay = Random[Integer,{-150, 150}];
 PulseWidth1 = Random[Integer,{-50, 50}];
 PulseWidth2 = Random[Integer,{-50, 50}];

 SolveEqns;
 PlotIt;
 CalcStats;

 popS := ρ1,1[MaxDPos] /.  nsoln[[1]];
 popP := ρ2,2[MaxDPos] /.  nsoln[[1]];
 popD := MaxD;

 NotebookLocate["FrameOut"];
 NotebookWrite[
 SelectedNotebook[],
 Cell[BoxData[RowBox[{"#", Count, ":  Time:  ", MaxDPos, " %s:  ", popS,
 " %p:  ", popP, " %d:  ", popD}]], "Input", CellTags → "FrameOut"]];
 If[
 Count == 0,
 {
 OutText1 := StringJoin[
 " ", "Run#",
```

```
" ", "Time",
" ", "I1",
" ", "I2",
" "
];
OutText2 := StringJoin[
" ", "Omega1",
" ", "Omega2",
" "
];
OutText3 := StringJoin[
" ", "Delta1",
" ", "Delta2",
" ", "Delay",
" "
];
OutText4 := StringJoin[
" ", "Width1",
" ", "Width2",
" ", "%s",
" "
];
OutText5 := StringJoin[
" ", "%p",
" ", "%d",
" "
];
},
{
OutText1 := StringJoin[
" ", ToString[Count],
" ", ToString[MaxDPos],
" ", ToString[FortranForm[I_1]],
" ", ToString[FortranForm[I_2]],
" "
];
OutText2 := StringJoin[
" ", ToString[FortranForm[Ω_1[MaxDPos]]],
" ", ToString[FortranForm[Ω_2[MaxDPos]]],
" "
];
OutText3 := StringJoin[
" ", ToString[FortranForm[Δ_1]],
" ", ToString[FortranForm[Δ_2]],
" ", ToString[FortranForm[Delay]],
" "
];
OutText4 := StringJoin[
" ", ToString[FortranForm[PulseWidth_1]],
" ", ToString[FortranForm[PulseWidth_2]],
" ", ToString[FortranForm[popS]],
```

```
 " "
 ];
 OutText5 := StringJoin[
 " ", ToString[FortranForm[popP]],
 " ", ToString[FortranForm[popD]],
 " "
 ];
 }
 ];

 OutText1 >>> "Y:/theory/stirap/3level/3levelout1.txt";
 OutText2 >>> "Y:/theory/stirap/3level/3levelout2.txt";
 OutText3 >>> "Y:/theory/stirap/3level/3levelout3.txt";
 OutText4 >>> "Y:/theory/stirap/3level/3levelout4.txt";
 OutText5 >>> "Y:/theory/stirap/3level/3levelout5.txt";
 }
 Count++
 ];
 NotebookLocate["FrameOut"];
 NotebookWrite[
 SelectedNotebook[],
 Cell[BoxData[RowBox[{"Finished."}]], "Input", CellTags → "FrameOut"]];
]
```

FrameOut

Finished.

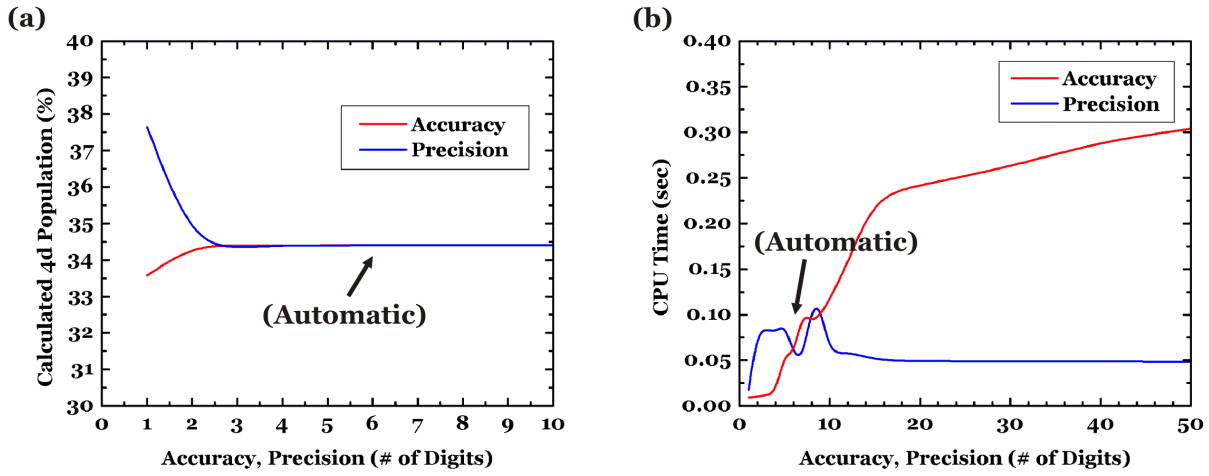

Figure D.1: **(a)** Calculated $4d$ fraction and **(b)** CPU time for solving the differential equations, both plotted *versus* accuracy and precision. The label "Automatic" represents Mathematica's default settings.

D.2.1 Accuracy and Precision

The Mathematica function, NDSolve, is the "workhorse" of the code generating the numerical solution to the coupled, first-order differential equations expressed in Sec. A.3.3. Mathematica allows one to select the number of digits of accuracy and precision for the NDSolve function, along with a specific method of solving the differential equations. Figure D.1a shows a typical[127] calculated value for the fractional $4d$ population as a function of the selected number of digits of accuracy and precision. As is evident, any selection above five yields the same solution, regardless of the increase in accuracy or precision. The label "Automatic" represents the default value chosen by Mathematica if no specific digit of accuracy or precision is indicated. As is evident, the automatic selection of accuracy and precision is sufficient to ensure the correct solution.

An increase in required accuracy should take more time to calculate. Figure D.1b shows the average CPU time involved in solving the coupled, first-order differential equations as a function of the number of digits of accuracy and precision selected. Precision does not seem to affect the time involved in the calculations past about ten digits, while increasing the number of digits of accuracy continues to increase the average time per calculation.

Again, the automatic default selected by Mathematica when no accuracy or precision value is supplied is reasonable and does not require an unnecessary amount of CPU time. Thus, the default values for both accuracy and precision for the calculations presented in this dissertation were set for the Mathematica limits.

Four options are allowed for selecting the computational method used by Mathematica in solving coupled sets of equations. The methods include Adams, Gear, RungeKutta, and "Automatic", where the automatic (or default) chosen by Mathematica involves "switching between a nonstiff Adams method and a stiff Gear method." Details on such methods and their implementation within Mathematica can be viewed in the Mathematica documentation.

A simple test was performed to see which of the options provided the most accurate solution, and to see which was computationally the most expensive. Figure D.2 shows a bar chart representing each method and the corresponding average CPU time used to calculate the fractional $4d$ population. The points on the graph correspond to the actual computed value of $4d$ population. The automatic method used by Mathematica provides the shortest computation time, and the calculated fractional $4d$ values agree to within 0.01%. Thus, again, the default method was selected for the simulations presented within this dissertation.

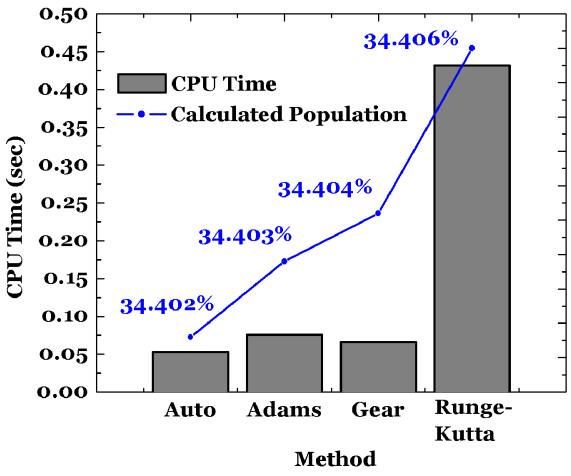

Figure D.2: CPU time for solving the differential equations shown *versus* accuracy and precision.

Appendix E

Cross Section Scripts

The following scripts were used with Origin to iteratively determine the $\frac{4d-5d}{5s-3s}$ cross section ratio in ^{87}Rb (Stokes Laser On/Off Scripts), and to calculate the population dynamics for a given data set (Population Dynamics Script). The actual Origin notebook `STIRAP3_ANALYSIS_IMPROVED.OPJ` should be accessed if this code is intended to be run. The scripts included here are:

- **Stokes "Laser Off" Script**

  Iteratively fits a series of Q-value spectra where no Stokes laser is present. The peak positions are defined in the script (and thus can be altered), and the background peak position, width, and area are randomly generated, within indicated ranges.

- **Stokes "Laser On" Script**

  Iteratively fits a series of Q-value spectra where the Stokes laser is present. The same positions as for the "Laser Off" script are set, along with the new peaks present due to the Stokes influence. This script works the same as the "Laser Off" script.

- **Population Dynamics Script**

  Automatically generates a population dynamics notebook, based on the input names of two data files (TAC_1 and TAC_2).

E.1 Stokes "Laser Off" Script

The multifit parameter script allows one to run several hundred fits for a given data set automatically. The first few fits are checked by hand to make sure the range of background values allowed is appropriate, and then the program can be run as many times as one would like. The fit values, along with error bars, for each peak are stored in the indicated columns after each iteration.

```
/*
// +-------------------------------------------------------------+
// |    Filename: multifit_l2OFF.ogs                             |
// |        Date: 10/25/04                                       |
// |  Created by: hcamp                                          |
// |                                                             |
// | Description: Attempt at multifit scripting.  The purpose of |
// |              this script is to allow one to perform several |
// |              hundred fits with different background settings.|
// +-------------------------------------------------------------+
*/
[Start]                 // Section Name

i = 0;                      // a simple counter.

runnum = 436;               // Current (Last) evaluated
                        // run #.  This is simply
                        // a way to keep track of
                        // which run the parameters
                        // were referring to.  This
                        // value is not used in the
                        // fitting code anywhere.

// +-----------------------------------------------------+
// | Open and clear the Script window                    |
// +-----------------------------------------------------+

//Type -o new;
//Type -a;

if(exist(nlsf1)==2) win -cd nlsf1;  // Delete old nlsf fit run (1)
if(exist(nlsf2)==2) win -cd nlsf2;  // Delete old nlsf fit run (2)
if(exist(nlsf3)==2) win -cd nlsf3;  // Delete old nlsf fit run (3)
```

```
// +-------------------------------------------------------------+
// | L2 Off Parameters                                           |
// +-------------------------------------------------------------+

yL2Off = 145;                   // y0 (background) for L2 Off

xSPOff = 117;               // --+
wSPOff = 3;                 //   +-- SP Peak
aSPOff = 10000;             // --+

xPSOff = 123;               // --+
wPSOff = 4;                 //   +-- PS Peak
aPSOff = 1000;              // --+

xPPOff = 137;               // --+
wPPOff = 3;                 //   +-- PP Peak
aPPOff = 4000;              // --+

xSSOff = 143;               // --+
wSSOff = 3;                 //   +-- SS Peak
aSSOff = 4000;              // --+


// +-------------------------------------------------------------+
// | Fit Routine                                                 |
// +-------------------------------------------------------------+

nlsf.init();                // initialize fit session
nlsf.func$      = "Gauss";      // function to be used for fit
nlsf.Y$         = dRawData_c;       // define data to be fitted
nlsf.X$         = dRawData_a;       // X data to be used
nlsf.xPoints    = 2048;         // Plot Points
nlsf.numReplica = 3;            // 4 peaks

nlsf.p1  = yL2Off + 20*rnd()-10;        // initialize y0

nlsf.p2  = xSPOff;          // initial center of SP
nlsf.p3  = wSPOff;          // initial width of SP
nlsf.p4  = aSPOff;          // initial area of SP

nlsf.p5  = xPSOff;          // initial center of PS
nlsf.p6  = wPSOff + 1*rnd();            // initial width of PS
nlsf.p7  = aPSOff;          // initial area of PS
```

```
nlsf.p8  = xPPOff;          // initial center of PP
nlsf.p9  = wPPOff;          // initial width of PP
nlsf.p10 = aPPOff;          // initial area of PP

nlsf.p11 = xSSOff;          // initial center of SS
nlsf.p12 = wSSOff;          // initial width of SS
nlsf.p13 = aSSOff;          // initial area of SS

// +-------------------------------------------------------------+
// | Fixed Parameters:  0 = do not vary,  1 = vary               |
// +-------------------------------------------------------------+

nlsf.v1  = 0;
nlsf.v6  = 0;

nlsf.iterate(200);              // perform 200 iterations
nlsf.iterate(200);              // perform 200 iterations
nlsf.iterate(200);              // perform 200 iterations

// +-------------------------------------------------------------+
// | Copy Parameters into worksheet                              |
// +-------------------------------------------------------------+

j = 1;
for (j=1; j<50; j++){if ($(dFitOff_b[j]) == "--") break};

dFitOff!cell(j, 2)=nlsf.p1;     // y0 Value
dFitOff!cell(j, 3)=nlsf.p10;        // PP Area
dFitOff!cell(j, 4)=nlsf.e10;        // PP Error
dFitOff!cell(j, 5)=nlsf.p13;        // SS Area
dFitOff!cell(j, 6)=nlsf.e13;        // SS Error

// +-------------------------------------------------------------+
// | Final Fit Cleanup                                           |
// +-------------------------------------------------------------+

//nlsf.unInit();                // Uninitialize array stack
//nlsf.cleanUpFitData();            // Destroy Variable allocation
nlsf.end();                 // Finish and end the session
```

```
Layer1.x.from =   50;
Layer1.x.to   =  180;
Layer1.y.from =    0;
Layer1.y.to   = 2000;
set drawdata_c -c 4;
set nlsf1_B -w 1000;
set nlsf1_Pk1dRawDC -w 1000;
set nlsf1_Pk2dRawDC -w 1000;
set nlsf1_Pk3dRawDC -w 1000;
set nlsf1_Pk4dRawDC -w 1000;
```

E.2 Stokes "Laser On" Script

The multifit parameter script allows one to run several hundred fits for a given data set automatically. The first few fits are checked by hand to make sure the range of background values allowed is appropriate, and then the program can be run as many times as one would like. The fit values, along with error bars, for each peak are stored in the indicated columns after each iteration.

```
/*
// +-------------------------------------------------------------+
// |    Filename: multifit_l2ON.ogs                              |
// |        Date: 10/25/04                                       |
// |  Created by: hcamp                                          |
// |                                                             |
// | Description: Attempt at multifit scripting.  The purpose of |
// |              this script is to allow one to perform several |
// |              hundred fits with different background settings.|
// +-------------------------------------------------------------+
*/

[Start]                 // Section Name


i = 0;                      // a simple counter.

runnum = 436;               // Current (Last) evaluated
                        // run #.  This is simply
                        // a way to keep track of
                        // which run the parameters
                        // were referring to.  This
                        // value is not used in the
                        // fitting code anywhere.

if(exist(nlsf1)==2) win -cd nlsf1;  // Delete old nlsf fit (1)
if(exist(nlsf2)==2) win -cd nlsf2;  // Delete old nlsf fit (2)
if(exist(nlsf3)==2) win -cd nlsf3;  // Delete old nlsf fit (3)
```

```
// +-------------------------------------------------------------+
// | L2 On Parameters                                            |
// +-------------------------------------------------------------+

yL2On = 145;                // (01) y0 (background) L2 On
xSPOn = 116.903;            // (02) --+
wSPOn = 4;                  // (03)   +-- SP Peak
aSPOn = 10000;              // (04) --+

xPSOn = 123.171;            // (05) --+
wPSOn = 4.5;                // (06)   +-- PS Peak
aPSOn = 4000;               // (07) --+

xDDOn = 127.603;            // (08) --+
wDDOn = 5;                  // (09)   +-- DD Peak
aDDOn = 4000;               // (10) --+

xDSOn = 133.391;            // (11) --+
wDSOn = 5;                  // (12)   +-- DS Peak
aDSOn = 4000;               // (13) --+

xPPOn = 136.858;            // (14) --+
wPPOn = 5;                  // (15)   +-- PP Peak
aPPOn = 4000;               // (16) --+

xSSOn = 143.321;            // (17) --+
wSSOn = 5;                  // (18)   +-- SS Peak
aSSOn = 4000;               // (19) --+

xDPOn = 148;                // (20) --+
wDPOn = 3.5;                // (21)   +-- DP Peak
aDPOn = 300;                // (22) --+

xBKOn = 125;                // (23) --+
wBKOn = 50;                 // (24)   +-- BK Gaussian Peak
aBKOn = 4000;               // (25) --+
```

```
// +-------------------------------------------------------------+
// | Fit Routine                                                 |
// +-------------------------------------------------------------+

nlsf.init();                // initialize fit session
nlsf.func$      = "Gauss";      // function to be used for fit
nlsf.Y$         = dRawData_b;       // define data to be fitted
nlsf.X$         = dRawData_a;       // X data to be used
nlsf.xPoints    = 2048;         // Plot Points
nlsf.numReplica = 7;            // 8 peaks
nlsf.p1  = yL2On + 10*rnd() - 5;    // initialize y0

nlsf.p2  = xSPOn;               // initial center of SP
nlsf.p3  = wSPOn;               // initial width of SP
nlsf.p4  = aSPOn;               // initial area of SP

nlsf.p5  = xPSOn;               // initial center of PS
nlsf.p6  = wPSOn;               // initial width of PS
nlsf.p7  = aPSOn;               // initial area of PS

nlsf.p8  = xDDOn;               // initial center of DD
nlsf.p9  = wDDOn;               // initial width of DD
nlsf.p10 = aDDOn;               // initial area of DD

nlsf.p11 = xDSOn;               // initial center of DS
nlsf.p12 = wDSOn;               // initial width of DS
nlsf.p13 = aDSOn;               // initial area of DS

nlsf.p14 = xPPOn;               // initial center of PP
nlsf.p15 = wPPOn;               // initial width of PP
nlsf.p16 = aPPOn;               // initial area of PP

nlsf.p17 = xSSOn;               // initial center of SS
nlsf.p18 = wSSOn;               // initial width of SS
nlsf.p19 = aSSOn;               // initial area of SS

nlsf.p20 = xDPOn;               // initial center of DP
nlsf.p21 = wDPOn;               // initial width of DP
nlsf.p22 = aDPOn;               // initial area of DP

nlsf.p23 = xBKOn +  10*rnd() - 5;   // initial center of BK
nlsf.p24 = wBKOn +  80*rnd() - 25;  // initial width of BK
nlsf.p25 = aBKOn + 700*rnd() - 250; // initial area of BK
```

```
// +-------------------------------------------------------------+
// | Fixed Parameters:  0 = do not vary,  1 = vary               |
// +-------------------------------------------------------------+

nlsf.v1  = 0;
nlsf.v2  = 0;
nlsf.v5  = 0;
nlsf.v6  = 0;
nlsf.v8  = 0;
nlsf.v11 = 0;
nlsf.v14 = 0;
nlsf.v17 = 0;
nlsf.v20 = 0;
nlsf.v21 = 0;
nlsf.v22 = 0;
nlsf.v23 = 0;
nlsf.v24 = 0;
nlsf.v25 = 0;

nlsf.iterate(100);              // perform 200 iterations
nlsf.iterate(100);              // perform 200 iterations
nlsf.iterate(100);              // perform 200 iterations

// +-------------------------------------------------------------+
// | Copy Parameters into worksheet                              |
// +-------------------------------------------------------------+

j = 1;

loop(j,1,100)
{
    if ($(dFitOn_b[j]) == "--") break;
}

dFitOn!cell(j, 2)=nlsf.p1;      // y0 Value
dFitOn!cell(j, 3)=nlsf.p10;     // DD Area
dFitOn!cell(j, 4)=nlsf.e10;     // DD Error
dFitOn!cell(j, 5)=nlsf.p16;     // PP Area
dFitOn!cell(j, 6)=nlsf.e16;     // PP Error
dFitOn!cell(j, 7)=nlsf.p19;     // SS Area
dFitOn!cell(j, 8)=nlsf.e19;     // SS Error
```

```
// +-------------------------------------------------------------+
// | Final Fit Cleanup                                           |
// +-------------------------------------------------------------+

//nlsf.unInit();                // Uninitialize array stack
//nlsf.cleanUpFitData();            // Destroy Variable allocation
nlsf.end();                 // Finish and end the session

Layer1.x.from =   50;
Layer1.x.to   =  180;
Layer1.y.from =    0;
Layer1.y.to   = 1000;
set drawdata_b -c 4;
set nlsf1_B -w 1000;
set nlsf1_Pk1dRawDC -w 1000;
set nlsf1_Pk2dRawDC -w 1000;
set nlsf1_Pk3dRawDC -w 1000;
set nlsf1_Pk4dRawDC -w 1000;
```

E.3 Population Dynamics Script

The main population dynamics script takes two raw files directly after the XSYS `MOVEOUT` command and parses them appropriately to calculate the excited-state fractions for each of the three energy levels $5s$, $5p$, and $4d$. The Q-value channels indicative of the proper capture channels must be defined prior to running the script. The rest is automatic.

```
// ************************************************************
// *                                                          *
// *     Filename: test_script_01.ogs                         *
// * Last updated: 02/20/04 by hcamp                          *
// *                                                          *
// ************************************************************

[Main]

Type -o new
Type -a

Window -a Params

%A=%(Params,1,1)
%B=%(Params,1,2)

//
// TAC1 Import
//

Window -n data RawTAC1
Open -w %A
Window -r %H RawTAC1

Worksheet -t 1 4
Worksheet -t 2 1
Worksheet -t 3 6

wks.colsel(3,1)
run.section(Wks,ConvReg)
Window -r %H MatTAC1
```

```
Worksheet -p 226 y:\miscellaneous\origin_templates\tac1.otp
Window -r %H TAC1

//
// TAC2 Import
//

Window -n data RawTAC2
Open -w %B
Window -r %H RawTAC2

mark -d RawTAC2_a -b 65281 -e 65550

i=0
loop(i, 1025, 1279){%(RawTAC2,3,i)=0}
loop(i,65281,65536){%(RawTAC2,3,i)=0}

Worksheet -t 1 4
Worksheet -t 2 1
Worksheet -t 3 6

wks.colsel(1,0)
wks.colsel(2,0)
wks.colsel(3,1)

run.section(Wks,ConvReg)
Window -r %H MatTAC2

Worksheet -p 226 y:\miscellaneous\origin_templates\tac2.otp
Window -r %H TAC2

mat.matname$="MatTAC2"
win -t TAC2cuts
mat.wksname$=%H
mat.m2w()
Window -r %H TAC2cuts

mat.matname$="MatTAC1"
win -t TAC1cuts
mat.wksname$=%H
mat.m2w()
Window -r %H TAC1cuts
```

```
pp=0
l=%(Params,5,1)
u=%(Params,5,2)
tmin=%(Params,7,1)
tmax=%(Params,7,2)
td=%(Params,2,1)
loop(i,tmin,tmax){loop(j,l-(144-td),u-(144-td)){pp=pp+
 %(TAC1cuts,j,i)}}

ss=0
l=%(Params,6,1)
u=%(Params,6,2)
tmin=%(Params,7,1)
tmax=%(Params,7,2)
td=%(Params,2,1)
loop(i,tmin,tmax){loop(j,l-(144-td),u-(144-td)){ss=ss+
 %(TAC1cuts,j,i)}}

n=(pp/11.29+ss)/(tmax-tmin+1)
npr=n/5

pp=
ss=
n=
npr=

Window -t FinalData
Window -r %H FinalData
Worksheet -a 5
%H!wks.col1.label$=Raw dd
%H!wks.col2.label$=RCombo
%H!wks.col3.label$=Raw pp
%H!wks.col4.label$=Raw ss
%H!wks.col5.label$=%%dd
%H!wks.col6.label$=%%pp
%H!wks.col7.label$=%%ss
Worksheet -t 1 1
%H!wks.labels()
```

```
d=0
l=%(Params,3,1)
u=%(Params,3,2)
td=%(Params,2,1)
loop(i,1,255){loop(j,l-(144-td),u-(144-td)){d=d+
 %(TAC2cuts,j,i)};
 %(FinalData,1,i)=d; d=0}

c=0
l=%(Params,4,1)
u=%(Params,4,2)
td=%(Params,2,1)
loop(i,1,255){loop(j,l-(144-td),u-(144-td)){c=c+%(TAC2cuts,j,i)};
 %(FinalData,2,i)=c; c=0}

s=0
l=%(Params,6,1)
u=%(Params,6,2)
td=%(Params,2,1)
loop(i,1,255){loop(j,l-(144-td),u-(144-td)){s=s+%(TAC2cuts,j,i)};
 %(FinalData,4,i)=s; s=0}

%(FinalData,@DN,3)=%(FinalData,@DN,2)-%(FinalData,@DN,1)/2.12
//loop(i,1,255){%(FinalData,3,i)=%(FinalData,2,i)-
 %(FinalData,1,i)/2.12}

%(FinalData,@DN,5)=%(FinalData,@DN,1)/8.7/npr
%(FinalData,@DN,6)=%(FinalData,@DN,3)/11.29/npr
%(FinalData,@DN,7)=1-(%(FinalData,@DN,5)+%(FinalData,@DN,6))

wks.colsel(5,1)

Worksheet -p 201 y:\miscellaneous\origin_templates\pop_vs_time.otp
Window -r %H gPop
graph finaldata_f
graph finaldata_g
window -i RawTAC1
window -i RawTAC2
window -i TAC1
window -i TAC2
window -i MatTAC1
window -i MatTAC2
window -i Tac1cuts
window -i Tac2cuts
```

Appendix F

Personal Publication List

The following is a list of publications that H. Camp has co-authored during his career at Kansas State University.

X. Fléchard, H. Nguyen, R. Brédy, S. R. Lundeen, M. Stauffer, **H. A. Camp**, C. W. Fehrenbach, and B. D. DePaola, State Selective Charge Transfer Cross Sections for Na^+ with Excited Rubidium: A Unique Diagnostic of the Population Dynamics of a Magneto-Optical Trap. Phys. Rev. Lett. **91**, 243005 (2003).

R. Brédy, H. Nguyen, **H. A. Camp**, X. Fléchard, and B. D. DePaola, MOTRIMS as a generalized probe of AMO processes. Nucl. Instrum. Methods Phys. Res., Sect. B **205**, 191 (2003).

H. Nguyen, X. Fléchard, R. Brédy, **H. A. Camp**, and B. D. DePaola, Recoil ion momentum spectroscopy using magneto-optically trapped atoms. Rev. Sci. Instrum. **75** 2638 (2004).

R. Brédy, **H. A. Camp**, H. Nguyen, T. Awata, B. Shan, Z. Chang, and B. D. DePaola, Three-dimensional spatial imaging in multiphoton ionization rate measurements. J. Opt. Soc. Am. B **21**, 2221 (2004).

H. Nguyen, R. Brédy, **H. A. Camp**, T. Awata, and B. D. DePaola, Differential charge-transfer cross sections for systems with energetically degenerate or near-degenerate channels. Phys. Rev. A **70**, 032704 (2004).

H. A. Camp, M. H. Shah, M. L. Trachy, O. L. Weaver, and B. D. DePaola, Numerical exploration of coherent excitation in three-level systems. Phys. Rev. A **71**, 053401 (2005).

H. Nguyen, R. Brédy, T. G. Lee, **H. A. Camp**, T. Awata, and B. D. DePaola, Counterintuitive Entropy Lowering in Ion Atom collision. Phys. Rev. A (2005, accepted).

M. A. Gearba, **H. A. Camp**, M. H. Shah, M. L. Trachy, H. Nguyen, R. Brédy, X. Fléchard, and B. D. DePaola, Measurement of excited state fraction in a MOT *versus* laser detuning. Opt. Lett. (2005, submitted).

M. H. Shah, **H. A. Camp**, M. L. Trachy, X. Fléchard, M. A. Gearba, H. Nguyen, R. Brédy, and B. D. DePaola, Excited state relative cross section in Rb. Phys. Rev. A (2005, submitted).

H. A. Camp, M. H. Shah, M. L. Trachy, H. Nguyen, M. A. Gearba, R. Brédy, X. Fléchard, and B. D. DePaola, Time evolution of coherent excitation. Science (2005, submitted).