INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6” x 9” black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700  800/521-0600
RICE UNIVERSITY

DETERMINATION OF TRANSIENT NEGATIVE ION LIFETIMES USING RYDBERG ATOMS

by

RICHARD A. POPPLE

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE DOCTOR OF PHILOSOPHY

APPROVED, THESIS COMMITTEE

F. B. Dunning, Chair
Professor of Physics

G. K. Walters
Professor of Physics

P. R. Brooks
Professor of Chemistry

Houston, Texas
February, 1996
ABSTRACT

Determination of Transient Negative Ion Lifetimes Using Rydberg Atoms

by

Richard A. Popple

Potassium Rydberg atoms are used to investigate the lifetimes, on a picosecond timescale, of transient negative ions formed during dissociative electron attachment to CCl₄ and CF₃I. The transient negative ions are formed during Rydberg electron transfer reactions of the type

\[ \text{K}(np) + AB \rightarrow K^+ + AB^{-*} \rightarrow K^+ + A^- + B \]

For low values of principal quantum number \( n \) (\( n \leq 20 \)) the electrostatic interaction between the \( K^+ - AB^{-*} \) and \( K^+ - A^- \) ion pairs perturbs the ion trajectories. As a consequence, the final angular and velocity distributions of the \( K^+ - A^- \) ions are dependent on the lifetime of the \( AB^{-*} \) intermediate ion. The \( AB^{-*} \) lifetime is determined by comparing product ion distributions, determined using position sensitive detection techniques, with the results of a Monte Carlo simulation that models the kinematics of the reaction. The data show that the lifetime of the excited CF₃I⁻* intermediate formed by Rydberg electron capture is short, \( \leq 2 \) ps, and that the lifetime of the CCl₄⁻* intermediate is \( 7.5 \pm 2.5 \) ps.
Acknowledgments

Many people have participated in this work, and it is my pleasure to acknowledge their contributions. My advisor, Professor Barry Dunning, gave me the opportunity to work with him in his laboratory. He gave me the freedom to make my own mistakes and, in doing so, gave me the opportunity to create my own successes. I have learned a lot here and am thankful to him. I am also grateful to Dr. Ken Smith for the occasional discussion about experimental difficulties.

The help of fellow students Mark Frey and Carla Finch was invaluable. Carla came to the lab eager to make a contribution. Without her, this work would have been much more difficult. Discussions with Mark were very useful. His breadth of knowledge and physical insight contributed to the solution of many problems. I thank both of them for their help.

No life is complete without friends, and I have had the good fortune of many. The friendship of Dean and Kristen Sieglaff, Nicholas and Mary Ellen Ritchie, and Heather Ensminger supported me over the past years. I am grateful to all of them.

My family has always been a source of strength. I would like to thank my brothers, David and Jeff, whose love has supported me throughout my education. I am deeply grateful to my parents, who laid the foundation that this work is built upon. Thanks, Mom and Dad.

Most importantly, my deepest thanks go to my wife. This work took many years to complete, and her patience and love supported me through all of them. Thanks Nancy.

Finally, without money, this research would have been nothing more than a wishful thought. Financial support was provided by the National Science Foundation and the Robert A. Welch Foundation.
# Table of Contents

Chapter 1: Introduction ............................................................... 1

Chapter 2: Experimental Apparatus .............................................. 5
  2.1 Vacuum System ............................................................. 5
  2.2 Alkali Source ............................................................... 5
  2.3 Laser ............................................................................ 7
  2.4 Dual Mass Spectrometer .................................................. 10
  2.5 Electronics ..................................................................... 11

Chapter 3: Experimental Techniques ........................................... 13
  3.1 Velocity Selection .......................................................... 13
  3.2 Product Ion Position Distributions .................................... 18
  3.3 Time-of-Flight Mass Spectrometry .................................... 18

Chapter 4: Monte Carlo Simulation .............................................. 28
  4.1 Monte Carlo Method ....................................................... 28
  4.2 Selection of Initial Conditions ........................................ 29
  4.3 Dissociation .................................................................... 31
  4.4 Post-Dissociation ............................................................ 32
  4.5 Classical Orbital Mechanics .......................................... 32

Chapter 5: Data Analysis ............................................................ 34
  5.1 Radial Distribution of Position Data .................................. 34
  5.2 Angular Distribution of Position Data .............................. 39
  5.3 Instrumental Width ......................................................... 44
  5.4 Decay Energetics ............................................................. 44
  5.5 Lifetime ........................................................................... 46
Chapter 6: Results and Discussion ................................................................. 48
  6.1 CF₃I ...................................................................................... 48
  6.2 CCl₄ ................................................................................. 56
Chapter 7: Future Work ................................................................. 67
  7.1 Transient Negative Ions .................................................... 67
  7.2 Exotic Negative Ions ............................................................ 69
Chapter 8: Conclusion ................................................................. 71
Appendix A: Validity of the Free Electron Model for Low-Energy Electron Attachment .................................................... 72
Appendix B: Radial Distribution Calculation Algorithm in C ................................................................. 75
Appendix C: Angular Distribution Calculation Algorithm in C ................................................................. 87
References .................................................................................................. 99
Chapter 1
Introduction

Many molecules capture thermal energy electrons to form excited negative ions that may subsequently dissociate or undergo autodetachment leading to reaction sequences of the form

\[ \rightarrow AB + C^- \]
\[ e^- + ABC \rightarrow ABC^* \]  \hspace{1cm} (1.1)
\[ \rightarrow ABC + e^- \]

Such reactions are characterized by a rate constant (or cross section) for the attachment process, and by the lifetime and dissociation energetics of the transient negative ion. Rate constants and cross sections for such reactions have been determined using a number of different experimental approaches, which include a variety of afterglow and drift tube techniques.\(^1\)-\(^6\) Electron attachment has also been studied utilizing low-energy electrons produced by near threshold photoionization.\(^7\)-\(^9\) The translational energy release associated with dissociative attachment reactions has been measured using a time-of-flight technique. In this technique, the fragment ion time-of-flight peak shapes resulting from free electron capture were analyzed in a time-of-flight mass spectrometer.\(^10\),\(^11\)

Negative ions may also be formed in collisions between atoms in high Rydberg states and attaching targets.\(^12\),\(^13\) Table 1.1 summarizes several properties of Rydberg atoms that make them well suited for studying low energy electron attachment. Of particular importance are the large Rydberg atom size and the long electron orbital period. If the range of interaction between a neutral molecule and an ion or electron is much smaller than the electron-core separation and if the duration of the collision is much less than the typical electron orbital period, the electron and the core ion behave as independent scatterers and Rydberg atom collisions with neutral molecules will be dominated by either the electron-molecule interaction or the core-molecule interaction. The Rydberg electron transfer reaction
\[ \text{K}(np) + ABC \rightarrow \text{K}^+ + ABC^- \]
\[ \rightarrow \text{K}^+ + ABC + e^- \]

\((1.2)\)

can be described in terms of a binary electron-molecule interaction in which the electron is considered as a free electron with a velocity distribution characterized by the Rydberg state (Appendix A demonstrates the validity of this approach). This free-electron model provides the basis of studying low-energy free-electron attachment processes using Rydberg atoms. Rate constants for electron capture\(^{13}\) and the translational energy release associated with dissociative attachment reactions\(^{14,15}\) have been measured using Rydberg atoms in states of high principal quantum number \(n\).

**Table 1.1 Properties of Hydrogen Rydberg Atoms**

<table>
<thead>
<tr>
<th>Property</th>
<th>(n) dependence (atomic units)</th>
<th>(n = 1)</th>
<th>(n = 15)</th>
<th>(n = 50)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean radius</td>
<td>(\frac{3}{2}n^2)</td>
<td>8\times10^{-9}\text{ cm}</td>
<td>1.8\times10^{-6}\text{ cm}</td>
<td>2\times10^{-5}\text{ cm}</td>
</tr>
<tr>
<td>Orbital period</td>
<td>(2\pi n^3)</td>
<td>1.5\times10^{-16}\text{ s}</td>
<td>5\times10^{-13}\text{ s}</td>
<td>2\times10^{-11}\text{ s}</td>
</tr>
<tr>
<td>Binding energy</td>
<td>(\frac{1}{2n^2})</td>
<td>13.6\text{ eV}</td>
<td>60\text{ meV}</td>
<td>5.4\text{ meV}</td>
</tr>
</tbody>
</table>

At high-\(n\), experiment shows that the free-electron model provides an adequate description of the collision process. However, as \(n\) decreases, the observed rate constant for the production of charged products falls below that predicted by the free-electron model. This is not due to a breakdown of the free-electron model, but results from three additional effects that must be taken into account as \(n\), and thus the size of the Rydberg electron cloud, decreases. The first is the post-attachment interaction between the product ions. At low-\(n\), the ion pairs created by Rydberg electron capture are formed in sufficiently close proximity that a significant fraction of them possess insufficient kinetic
energy to overcome their mutual electrostatic interaction. These ion pairs do not separate and are not observed.\textsuperscript{16-18} The second effect is the increasing opacity of the Rydberg electron cloud. At high-$n$, the electron cloud is diffuse and the probability of a molecule attaching the electron as it passes through the cloud is small. As $n$ decreases, the cloud becomes denser and the probability of attachment during passage through the cloud approaches unity and the heavy particle collision cross section becomes limited by the geometric size of the Rydberg atom. Thus, the rate constant must decrease and falls below that predicted by the free-electron model alone.\textsuperscript{19} The magnitude of the effect of opacity is dependent on the rate constant for free-electron capture and on the relative collision velocity. While the rate constant decreases, calculations suggest that opacity does not affect the product ion velocity and angular distributions except at very low-$n$ ($n \leq 10$). The third effect is the possibility of energy transfer from the excited intermediate to the core ion at low-$n$. Ions that decay by autodetachment can be stabilized by collision with the core ion to produce a long-lived product\textsuperscript{20,21}, or energy transfer may result in separation of products that would otherwise be bound by their electrostatic interaction.\textsuperscript{22} However, this effect is observed only at very low-$n$ ($n \leq 15$) and occurs on a time scale that is long compared with the typical orbital period of the intermediate ion pair and so is not an important effect for the present work.\textsuperscript{21,22}

Information about the lifetime of the intermediate negative ion can be obtained by examining the angular and velocity distributions of product ions formed by Rydberg electron transfer.\textsuperscript{23} This work reports the development of a technique to determine the lifetime of transient negative ions which extends previous Rydberg atom work by using velocity selection and time-of-flight techniques to improve the kinematic specificity of the measurements.\textsuperscript{24} This thesis begins with a description of the apparatus and experimental techniques. A Monte Carlo simulation of the experiment based on the free-electron model and the methods of data analysis are discussed, followed by the results of
studies of low-energy electron attachment to iodotrifluoromethane (CF₃I) and carbon tetrachloride (CCl₄). The decay energetics are compared with the results of previous work and estimates of the transient negative ion lifetimes are presented. The results demonstrate that Rydberg atom techniques can illuminate decay energetics and allow determination of intermediate lifetimes on a picosecond timescale.
Chapter 2
Experimental Apparatus

The apparatus used in the present investigation is shown in Fig. 2.1. Potassium atoms contained in a collimated beam are excited to a selected np state using an intracavity doubled ring dye laser. The Rydberg atoms are allowed to interact with the target molecules for a defined period of time after which the charged collision products are extracted by a voltage pulse. The products then traverse time-of-flight mass spectrometers and are detected by position sensitive detectors.

2.1 Vacuum System

The vacuum system consists of two differentially pumped stainless steel chambers, the source chamber and the main chamber. The alkali beam is produced in the source chamber. Rydberg atoms are created, allowed to interact with target molecules, and the resulting products are detected in the main chamber.

The source chamber contains an effusive alkali source and is pumped by a 4" diffusion pump. It is separated from the main chamber by a 0.012" diameter aperture which collimates the atomic beam. The aperture is heated to prevent clogging by condensed alkali.

The main chamber is pumped by two 6" diffusion pumps. The laser enters and exits the chamber through quartz windows set at Brewster's angle. Target molecules are admitted to the chamber through a needle valve and a sintered plug. The valve and plug are heated to prevent condensation, which can cause pressure fluctuations. The pressure in the main chamber is monitored with a Bayard-Alpert type ionization gauge, which is calibrated against a capacitance manometer.

2.2 Alkali Source

Potassium is used in this research because its Rydberg states are accessible using commercial ring dye lasers and it has a simple electronic configuration, consisting of a
FIG. 2.1.  Schematic diagram of the apparatus
single 4s electron orbiting a closed, rare-gas-like core. An effusive beam of potassium is formed by a two-stage stainless steel oven located in the source chamber. Potassium metal is heated to increase the vapor pressure to \( \sim 0.5 \) Torr. Vapor then effuses out of the oven through a 0.02" diameter hole drilled in an endcap, which is maintained about 20° C above the body to prevent condensation. To obtain adequate beam densities, the temperature of the body is held at about 300° C. The beam is collimated by the 0.012" diameter aperture separating the source and main chambers. The aperture is located 3" beyond the oven nose, giving the beam a divergence of \( \sim 0.2° \). The intersection of the laser and potassium beams is 2.75" beyond the aperture, resulting in a beam width of \( \sim 0.5 \) mm. The beam density is monitored using a hot wire ionization detector in the main chamber. Typical beam density at the interaction region is \( \sim 10^9 \) cm\(^{-3}\).

### 2.3 Laser

Rydberg states are formed by exciting ground state potassium to a selected \( np \) state by single photon excitation. This requires light in the range 285-295 nm. The light is provided by an intracavity frequency doubled ring dye laser, a Coherent model CR699-21 with a model 7500-06 doubling package. In a ring dye laser, the output of a pump laser is focused on a flat stream of dye, which flouresces over a broad frequency range. Rhodamine 6G dye is used in this work, pumped by 9-10 W of 514 nm light provided by a Coherent Innova 400 Argon ion laser. The Rh6G tuning range is 560 - 640 nm. The circulating visible radiation is partially converted into UV by a potassium dihydrogen phosphate (KDP) crystal through second harmonic generation. Over the range of interest for this work, 285-295 nm, 3-5 mW can be obtained. Single frequency operation is accomplished by use of 3 tuning elements: a 3 plate birefringent filter with a passive bandwidth of 380 GHz, a thin etalon with 200 GHz free spectral range (FSR), and a thick etalon with 10 GHz FSR. Laser oscillation occurs in a ring resonator and thus is a traveling wave, which can propagate either clockwise or counterclockwise. One direction
is selected by an optical diode which consists of a glass element in a magnetic field and a polarization rotating element. Due to the Faraday effect, one direction of propagation suffers a net polarization rotation. The resulting losses at Brewster surfaces within the cavity are sufficient to insure unidirectional oscillation. The laser is actively stabilized by locking it to a temperature controlled Fabry-Perot etalon. Fast deviations are corrected by a piezoelectric translator (PZT) mounted mirror and long term drifts are corrected by rotating a galvo mounted quartz plate about Brewster's angle. The typical long term stability is 40 MHz / hour in the fundamental. By adjusting the lock point of the reference cavity, the laser may be scanned over a 30 GHz range.

The typical long term stability of the laser system, 40 MHz / hour in the fundamental, is not sufficient for the present work. To minimize the long term drift, the laser is locked to a frequency stabilized Helium-Neon laser using a second temperature controlled Fabry-Perot etalon. A schematic of the system is shown in Fig. 2.2. A fraction of the visible radiation emerging through the end mirror of the dye laser is superposed with the output of a frequency stabilized Helium-Neon laser using a dichroic beam splitter, which reflects at 633 nm and transmits below ~600 nm. The beams are directed through a sealed temperature-stabilized confocal Fabry-Perot etalon. The mirror spacing in the etalon is scanned at 50 Hz using a PZT. The etalon outputs two series of sharp transmission peaks, one associated with each input wavelength. The transmission peaks are independently monitored by use of a second dichroic filter and photodiodes. The locations of the etalon transmission peaks are determined by measuring the drive voltage applied to the PZT. If the dye laser frequency shifts, the position of its associated transmission peak will shift relative to the Helium-Neon laser transmission peak. Changes in peak separation are monitored and used to generate an error signal that restores the dye laser to the desired frequency. The stability of the Helium-Neon laser is very sensitive to retroreflections. Isolation of the Helium-Neon laser from the etalon is
FIG. 2.2: Schematic diagram of the CR699-21 laser control system
therefore important, because when the laser is properly aligned, reflections from the etalon are directed back along the beam line into the laser. To isolate the Helium-Neon laser from retroreflections, the beam is directed through a polarizer and then a quarter wave plate oriented to produce circular polarization. Any retroreflections return through the quarter wave plate and are linearly polarized orthogonal to the axis of the linear polarizer. The frequency stability is limited by the HeNe to $\pm 1 \text{ MHz} / 8 \text{ hours in the fundamental (f} \pm 2 \text{ MHz UV)}. \text{ By changing the lockpoint, the frequency can be scanned over a } 500 \text{ MHz range. The lockpoint can be externally controlled using a DC voltage in the range } -5 \text{ V to } +5 \text{ V.}

The laser wavelength is monitored using a scanning Michelson interferometer.$^{26}$ The system counts the fringes of the unknown wavelength and compares this number to the fringe count of a reference laser, in this case a polarization balanced HeNe.$^{27}$ Over the present wavelength range, the system provides a resolution of $\sim 1 \text{ GHz}$, or $\sim 0.1 \text{ Å}.

The continuous output of the laser is chopped into pulses by an acousto-optic modulator (AOM).$^{28,29}$ The AOM consists of a PZT transducer mounted on high purity fused quartz. The transducer generates a high frequency RF acoustic traveling wave inside the glass. The resulting periodic variation of the refractive index produces a diffraction grating which can deflect up to 90% of the light out of the zero order. To produce a pulse of light, the sound wave is pulsed to deflect the light into the first order. Typical pulse lengths in this work are $0.5 \mu\text{s} - 5 \mu\text{s}.$

2.4 Dual Mass Spectrometer

Excitation of the Rydberg atoms and their collisions with the target gas occur in a field-free interaction region located between two planar fine mesh grids. After collisions have been allowed to occur for a predetermined time interval, a pulsed electric field is used to accelerate charged collision products into time-of-flight mass spectrometers located above and below the interaction region, as shown in Fig. 2.1. Each time-of-flight
mass spectrometer consists of an initial acceleration region, a field free drift region, and a final region to accelerate the particles to the detection energy. Particles are detected at the exit of each mass spectrometer by a position-sensitive detector (PSD). A µ-metal shield surrounds the assembly, reducing magnetic fields to ~20 mG in the center of the interaction region.

2.5 Electronics

The PSD, a Surface Science Laboratories model 239G Imaging Particle Detector, consists of two microchannel plates (MCPs) stacked in a chevron configuration in front of a resistive anode. An incident particle produces a pulse of charge on the anode that is collected at electrodes at each of the corners. An electronic unit calculates the position of the incident particle from the ratio of the charge collected at each corner. The position in each direction is output as a 0-5 V analog level. A TTL logic level, called the strobe pulse, indicates that the electronic unit has position information available. The time of arrival of the particle is obtained from the current pulse recharging the rear MCP. This pulse is amplified by an EG&G Ortec model VT120B preamplifier, and is discriminated by a LeCroy model 821 Quad Discriminator.

Data acquisition and analysis is performed with a Motorola VME131 computer that is interfaced with the experiment through a CAMAC (Computer Automated Measurement And Control) bus. Particle arrival times are recorded with a Bi-Ra model 2205 Quad Event Timer (QET) that measures the time difference between up to 4 stop signals and a common start signal using a digital counter. The QET has a 10 ns resolution and a 12-bit (41 µs) range. Particle arrival positions are digitized by a 12-bit Jorway model 32 analog-to-digital converter (ADC) set up for a 0-5 V range. The spatial resolution of the PSD is 50-100 µm, so only 8 bits of the result is significant. The computer initiates the experiment and controls a programmable delay using a Bi-Ra model 3222 Output module which has 48 output lines. The Output module has a data
accepted input that is used to indicate when an experimental cycle is complete. The laser frequency can be controlled over a 500 MHz range by a -5V to +5V analog input. This is computer controlled by a Standard Engineering model E250 12-bit digital-to-analog converter (DAC).
Chapter 3
Experimental Techniques

The experimental apparatus used in the present work improves the kinematic specificity as compared to previous measurements made in this laboratory.\textsuperscript{15,18,23} Velocity selected Rydberg atoms are produced by Doppler tuning. Ion arrival position distributions are measured with a PSD, and time-of-flight (TOF) techniques are used to isolate those reactions in which both product ions initially travel in the xy plane.

3.1 Velocity Selection

To select the velocity of the Rydberg atom, the laser beam crosses the atomic beam at an angle slightly off of normal incidence. Because of the Doppler effect, the laser excites only ground state atoms within a narrow range of velocities. Consider a laser of frequency $f$ incident at angle $\theta$ off normal to the atomic beam. The frequency observed in the frame of an atom moving with velocity $v \ll c$ is given by

$$f' = f\left(1 + \frac{v}{c}\sin\Theta\right)$$  \hspace{1cm} (3.1)

Rearranging to obtain the laser frequency for which the atom observes the laser shifted to the resonance frequency $f_0$ gives

$$f = f_0\left(1 + \frac{v}{c}\sin\Theta\right)^{-1}$$  \hspace{1cm} (3.2)

Noting that $v \ll c$, Eq. 3.2 becomes

$$f = -\frac{f_0}{c}v\sin\Theta + f_0$$  \hspace{1cm} (3.3)

The angle $\Theta$ is determined by beam steering mirrors. The Rydberg atom velocity is measured by field ionizing the atoms and recording the arrival positions of the resulting ions for two different drift times. The ion displacement during the incremental drift time gives the Rydberg atom velocity. The angle $\Theta$ is then determined from the slope of a plot of laser frequency vs. velocity which will yield a straight line with slope
\[ \text{slope} = -\frac{f_0 \sin \Theta}{c} = -\frac{\sin \Theta}{\lambda} \]  

(3.4)

Figure 3.1 shows the dependence of laser frequency offset on the velocity of excited atoms at \( n = 55 \) for which \( \lambda = 285.9 \) nm. The slope is 0.152 MHz/m/s, corresponding to an angle \( \Theta \) of 2.5°.

The range of velocities selected is determined primarily by the atomic beam divergence. For an atomic beam divergence \( \delta \) and a laser divergence \( \eta \), the incidence angle \( \Theta \) varies by \( \pm (\eta + \delta) = \pm \phi \). The variation in \( \nu \), \( \Delta \nu \), is given by

\[ \Delta \nu = \frac{c f_0 (\csc \Theta)}{f} \]  

(3.5)

\[ \Delta (\csc \Theta) = |\csc(\Theta + \phi) - \csc(\Theta - \phi)| \]  

(3.6)

If \( \phi \ll \Theta \ll 1 \) then

\[ \Delta (\csc \Theta) = \frac{2 \phi}{\Theta^2} \]  

(3.7)

and

\[ \frac{\Delta \nu}{\nu} \approx \frac{2 \phi}{\Theta} \]  

(3.8)

The laser beam divergence is negligible compared to the atomic beam divergence, which is \( -0.2° \). For a typical intersection angle of 2.5°, \( \Delta \nu / \nu = 0.1 \).

The velocity range is also increased by the natural linewidth of the transition, the linewidth of the laser, and transit time broadening. An estimate of these effects may be obtained by assuming that the resonant frequency is broadened by \( \Delta f_0 \). The variation in \( \nu \), \( \Delta \nu \), is given by

\[ \Delta \nu = \frac{c \Delta f_0 \csc \Theta}{f} \]  

(3.9)

Observing that \( f = f_0 \),

\[ \Delta \nu = \frac{c \Delta f_0 \csc \Theta}{f_0} \]  

(3.10)
FIG 3.1. Frequency vs. Rydberg atom velocity for laser intersection angle $\theta = 2.5^\circ$
The instantaneous linewidth of the laser is a fraction of a Hz, but it has a 1 MHz jitter, which corresponds to $\Delta \nu = 15$ m/s. The natural linewidth of the transition is $< 500$ kHz, and so the corresponding $\Delta \nu < 7$ m/s. Transit time broadening is given by\cite{31}

$$\Delta f = \frac{\nu \sqrt{2 \ln 2}}{2 \pi \nu}$$

(3.11)

where $\nu$ is the laser beamwaist. So, for transit time broadening,

$$\frac{\Delta \nu}{\nu} = \frac{c \sqrt{2 \ln 2} \csc \Theta}{2 \pi \omega \theta_0}$$

(3.12)

The laser beamwaist is $\sim 50$ $\mu$m, which gives $\Delta \nu/\nu = 0.05$. These considerations demonstrate that for the angles used in this work, the primary contribution to the width of the velocity distribution is the atomic beam divergence. The experimentally measured width agrees with that predicted from the above considerations.

The choice of angle is governed by competing considerations. For large angle, the velocity spread due to beam divergence decreases. However, making the angle too large poses a problem. Each potassium p-state with principal quantum number $n$ comprises 4 states. These result from the hyperfine splitting of the ground state, which is 462 MHz, and from the fine-structure splitting of each state, which scales as $n^{-3}$. At low-$n$ ($n \lesssim 36$), the hyperfine splitting is the smaller of the two, but at larger values of $n$, the fine structure splitting is smaller. If the angle is too large, the Doppler widths of adjacent states overlap and atoms with different velocities are selected. To avoid this problem, the angle is chosen to be the maximum that will not result in overlap at the highest $n$ of interest. At $n = 55$, the fine structure splitting is 126 MHz, and the angle must be limited to less than $\sim 2.5^\circ$ to prevent excitation of more than one atom velocity. A typical spectrum is shown in Fig. 3.2. In addition, care must be taken when working in the range $30 < n < 40$ because the $F = 2, J = \frac{1}{2}$ state crosses the $F = 1, J = \frac{3}{2}$ state.
FIG. 3.2. State spacing at $n = 55$ for (a) normal incidence and (b) laser/atomic-beam intersection angle $\theta = 2.5^\circ$
3.2 Product Ion Position Distributions

A diagram of the experimental timing is shown in Fig. 3.3. To obtain arrival position distributions of product ions, velocity selected Rydberg atoms are produced by a short laser pulse, typically 0.5 - 1 μs duration, in the center of the interaction region in the presence of target gas. Collisions are allowed to occur for a predetermined time interval following the laser pulse, typically 0.5 - 1 μs, after which a pulsed electric field is applied to expel the product ions into the time-of-flight mass spectrometers (TOFMS). Ion flight times to the detector are typically 5-20 μs. The arrival time and position of each ion is recorded. The ion pairs are recorded in coincidence to insure that the ions detected result from the reaction of interest. The probability that an ion pair is formed following each laser pulse is low, ≤ 0.05, to ensure that the probability of false coincidences is small. The PSD strobe pulse, which is used to initiate the ADC, is gated so that only ions arriving in a specified time-of-flight range cause the ADC to record the position data.

3.3 Time-of-Flight Mass Spectrometry

The TOFMS used in this work comprises three types of regions: an extraction region, two drift regions, and two acceleration regions. The drift regions have no electric field, the acceleration regions contain a constant electric field, and the extraction region contains an electric field that increases linearly from zero to a constant value over some rise time. The field rises rapidly (~100 ns rise time) and so the extraction region can be approximated as a region of constant field. A schematic of the TOFMS is shown in Fig. 3.4.

Time-of-flight techniques can be used to identify the component of velocity of a product ion along the axis perpendicular to the plane of the PSD (the z-axis). Such techniques have been widely used to infer ion energy distributions.32,33 In the present work, TOF techniques are used to identify product ions that are traveling in the xy plane. Consider an ion of charge q and mass m. At the time the extraction pulse is applied, the
FIG. 3.3. Data acquisition timing diagram
FIG. 3.4. Schematic diagram of the time-of-flight mass spectrometer
ion is located at a distance $s$ from the interaction region grid and has a $z$-component of velocity $v_z$. The flight time of the ion is given by

$$
\begin{align*}
t(U_z, s) &= t_{IR}(U_z, s) + t_{ARI}(U_z, s) + t_{DR1}(U_z, s) + \\
&
+ t_{AR2}(U_z, s) + t_{DR2}(U_z, s)
\end{align*}
$$

(3.13)

where

$$
U_z = \frac{1}{2} m v_z^2
$$

(3.14)

$$
\begin{align*}
t_{IR}(U_z, s) &= \left( \frac{\sqrt{2m}}{qE_{IR}} \right) \left( \sqrt{qsE_{IR} + U_z} \pm \sqrt{U_z} \right)
\end{align*}
$$

(3.15)

$$
\begin{align*}
t_{ARI}(U_z, s) &= \left( \frac{\sqrt{2m}}{qE_{ARI}} \right) \left( \sqrt{qd_{ARI}E_{ARI} + qsE_{IR} + U_z} - \sqrt{qsE_{IR} + U_z} \right)
\end{align*}
$$

(3.16)

$$
\begin{align*}
t_{DR1}(U_z, s) &= \frac{\sqrt{2md_{DR1}}}{2 \sqrt{qd_{ARI}E_{ARI} + qsE_{IR} + U_z}}
\end{align*}
$$

(3.17)

$$
\begin{align*}
t_{AR2}(U_z, s) &= \left( \frac{\sqrt{2m}}{qE_{AR2}} \right) \left( \sqrt{qd_{AR2}E_{AR2} + qd_{ARI}E_{ARI} + qsE_{IR} + U_z} - \sqrt{qd_{ARI}E_{ARI} + qsE_{IR} + U_z} \right)
\end{align*}
$$

(3.18)

$$
\begin{align*}
t_{DR2}(U_z, s) &= \frac{\sqrt{2md_{DR2}}}{2 \sqrt{qd_{AR2}E_{AR2} + qd_{ARI}E_{ARI} + qsE_{IR} + U_z}}
\end{align*}
$$

(3.19)

The plus and minus signs correspond to initial ion velocity directed away from and toward the detector, respectively. In the present work, $qsE_{IR} \ll U_z$, and so the flight time is given by

$$
\begin{align*}
t(U_z, s) &= t_0(s) - \frac{m}{qE_{IR}} v_z
\end{align*}
$$

(3.20)

The flight time is linear with velocity and so the mean time-of-flight corresponds to zero $z$-velocity. Determination of the mean TOF therefore permits straightforward experimental identification of ions with velocities in the plane of the PSD (the $xy$ plane).

As an example, consider the Rydberg electron transfer reaction

$$
K(55p) + CF_3I \rightarrow K^+ + CF_3I^- \rightarrow K^+ + I^- + CF_3
$$

(3.21)
A typical time-of-flight spectrum for $\Gamma^-$ ions resulting from reaction 3.21 is shown in Fig. 3.5. As will be discussed later, the $\text{CF}_3\Gamma^{-\ast}$ intermediate dissociates with a narrow translational energy release distribution and the $\Gamma^-$ ions travel outward from the interaction volume in all directions with a narrow distribution of velocities. When this distribution is observed in two dimensions on the PSD, it results in an arrival position distribution that is broad and flat. However, if only those ions with velocity components in the $xy$ plane are selected, the resulting distribution is a ring. This is illustrated in Fig 3.6, which shows the $\Gamma^-$ position distributions obtained without arrival time gating and for only those $\Gamma^-$ ions with flight times in a narrow range centered on the mean flight time.

Application of this technique to the other negative ion of interest in this work, $\text{Cl}^-$, requires care because it has two isotopes, $^{35}\text{Cl}$ and $^{37}\text{Cl}$, which have abundances of 75.77% and 24.23% respectively. The TOFMS is unable to completely resolve the TOF peaks of these two species. A typical TOF spectrum for $\text{Cl}^-$ ions resulting from the electron transfer reaction

$$K(65p) + \text{CCl}_4 \rightarrow K^+ + \text{CCl}_4^{-\ast} \rightarrow K^+ + \text{Cl}^- + \text{CCl}_3 \tag{3.22}$$

is shown in Fig. 3.7. The incomplete separation of the TOF peaks causes two problems. First, the determination of the mean flight time for each isotope is not straightforward. Second, the mean arrival time of $^{35}\text{Cl}^-$ ions, which corresponds to zero velocity, will also correspond to $^{37}\text{Cl}^-$ ions with a significant velocity. To determine the mean arrival time of each isotope and to estimate the level of contamination, the TOF peak of each isotope is modeled as a gaussian. The observed TOF spectrum is then the sum of two gaussians having a common width and normalized to the isotopic abundances:

$$s(t) = \frac{1}{\sqrt{2\pi}\sigma} \left( 0.76e^{\frac{(t-\mu_1^c)}{2\sigma^2}} + 0.24e^{\frac{(t-\mu_2^c)}{2\sigma^2}} \right) \tag{3.23}$$
This function is fit to the observed TOF with the width and the mean flight times as fitting parameters. The result of such a fit is shown in Fig. 3.8. Because $^{35}\text{Cl}$ is the most abundant isotope, it is used in this work. From the fit in Fig. 3.8, it is apparent that $^{37}\text{Cl}^-$ ions contribute less than 1% of the total $\text{Cl}^-$ signal in the time range of interest to the present work.
FIG. 3.5.  Time-of-flight spectrum of $\Gamma^-$ ions formed in CF$_3$I/K(55p) collisions
FIG. 3.6. Arrival position distributions of $\Gamma^-$ ions formed in CF$_3$I/K(55p) collisions. The data are for (a) all $\Gamma^-$ ions and (b) $\Gamma^-$ ions traveling in the xy plane. The shading indicates the relative $\Gamma^-$ signal at different elements of the PSD; the light gray, mid-gray, and black areas correspond to 10-40%, 40-70%, and 70-100% of the peak signal, respectively.
FIG. 3.7. Time-of-flight spectrum of Cl$^-$ ions formed in CCl$_4$/K(65p) collisions
FIG. 3.8. Fit to time-of-flight spectrum of Cl\(^-\) ions formed in CCl\(_4\)/K(65p) collisions. Solid squares are data and the line is the best fit using Eq. 3.23.
Chapter 4
Monte Carlo Simulation

The experimentally determined ion arrival position distributions are analyzed using a Monte Carlo simulation based on the free electron model. The excited electron is considered as free except that its probability distribution is determined by its quantum state. The initial collision conditions are chosen at random from the appropriate distributions of these parameters and the probability of attachment at some point is taken to be proportional to the local electron probability density. Following electron capture, the motions of the product ion pairs are determined using classical orbital theory. Ion arrival position distributions are built up by considering the outcomes of many different collisions.

4.1 Monte Carlo Method

The Monte Carlo technique requires choosing the collision conditions at random from the appropriate distributions of these parameters. To choose a value, the distribution of the appropriate parameter is broken into \( N \) bins. Consider a normalized distribution of a parameter \( x, f(x) \). \( X_{i+1} \) is chosen such that

\[
\int_{X_i}^{X_{i+1}} f(x) \, dx = \frac{1}{N} \tag{4.1}
\]

where \( X_0 = 0 \). The mid-point of the bin is then taken to be representative of the bin. This produces a list of numbers, \( x_i, i = 0 \) to \( N-1 \), such that

\[
x_i = \frac{1}{2} \left( X_i + X_{i+1} \right) \tag{4.2}
\]

When a value is required, one of these \( N \) numbers is then chosen at random. For distributions that are symmetric about zero, such as the target gas velocity distribution, the values are calculated for the distribution of magnitudes. When a value is required, the magnitude is chosen from the list of values and the sign is chosen at random.
4.2 Selection of Initial Conditions

For any collision, each component of the target molecule velocity is chosen from the ideal gas velocity distribution. If the Rydberg atom velocity can be chosen one of two ways. If the Rydberg atom is not velocity selected, the velocity is chosen from the Maxwell-Boltzmann speed distribution. If the Rydberg atom is velocity selected, then the distribution is modeled as a gaussian centered on $v_0$ with a width $\sigma_v$:

$$f(v) = \frac{1}{\sqrt{2\pi\sigma_v^2}} e^{-\frac{(v-v_0)^2}{2\sigma_v^2}}$$ (4.3)

The formation position of the Rydberg atom is chosen from within a cylinder that models the intersection of the laser beam with the potassium beam. The position along the axis of the laser is chosen randomly along the width of the atomic beam. The position in the plane normal to the laser direction is chosen from the 2-dimensional gaussian intensity distribution of the laser with the appropriate beam waist.

Choosing the time of the electron capture event is a two-step procedure. For a Rydberg atom interacting with a target gas of density $\rho$, the total probability of electron attachment is given by

$$P(t) = 1 - e^{-\rho kt}$$ (4.4)

where $k$ is the rate constant and $t$ is the interaction time. For $\rho kt << 1$,

$$P(t) = \rho kt$$ (4.5)

So, for a Rydberg atom produced at time $t_R$ after the beginning of the laser pulse, the probability that it will undergo electron attachment is

$$P(t_R) = \rho k(t_c + t_c - t_R)$$ (4.6)

where $t_L$ is the length of the laser pulse and $t_C$ is the time between the end of the laser pulse and the extraction pulse. So, from these considerations, we see that the distribution of formation times for Rydberg atoms that lead to electron attachment is
\[ f(t) = \begin{cases} C(t_L + t_C - t) & t \leq t_L \\ 0 & t \geq t_L \end{cases} \quad (4.7) \]

where

\[ C = \frac{2}{(t_L^2 + 2t_L t_C)} \quad (4.8) \]

From Eq. 4.5, the probability per unit time of electron capture is constant:

\[ \frac{dP}{dt} = \rho k \quad (4.9) \]

So, the distribution of electron capture times is a constant, and is given by

\[ f(t) = \begin{cases} \frac{1}{(t_L + t_C - t_R)} & t_R \leq t \leq t_L + t_C \\ 0 & t < t_R, t > t_L + t_C \end{cases} \quad (4.10) \]

where \( t_R \) is obtained from Eq. 4.7. So, to determine the ion formation time, the time of Rydberg atom creation is chosen from Eq. 4.7 and is used in Eq. 4.10 to obtain the time of electron capture.

The location of the electron capture event relative to the Rydberg atom core is the exception to the Monte Carlo procedure described above. The probability of electron capture is assumed to be proportional to the electron probability density. Because the radial probability distribution is not smooth, containing \( n-\ell-1 \) nodes, the procedure outlined above would not produce a set of representative values. Instead, the electron capture event is chosen to occur at a random location within a box centered on the Rydberg atom. The dimensions of the box are chosen to contain some fraction, typically 99.99%, of the total Rydberg electron probability density. The electron probability distribution at this location is determined from the hydrogenic wave function and is then used as a weighting factor for each collision event.
4.3 Dissociation

Following selection of the initial conditions, the motion of the ion pair is followed using classical orbital mechanics until the intermediate ion dissociates at time \( t \). The dissociation time is chosen from the distribution

\[
f(t) = \frac{1}{\tau} e^{-\frac{t}{\tau}} \tag{4.11}
\]

where \( \tau \) is the mean lifetime.

The energy of the dissociating molecule is chosen from a distribution based on the decay mechanism. For a very short intermediate lifetime (\( \tau \ll \text{typical vibrational period} \)), the dissociation will occur before significant distribution of the excess energy of reaction among internal vibrational modes can take place. In this event, all of the excess energy should appear in translation, leading to a narrow translational energy release distribution centered on the available excess energy. For this case, the energy of the dissociating molecule is chosen from a Gaussian

\[
f(\varepsilon) = \frac{1}{\sqrt{2\pi\sigma_{\varepsilon}}} e^{-\frac{(\varepsilon-\bar{\varepsilon})^2}{2\sigma_{\varepsilon}^2}} \tag{4.12}
\]

where \( \bar{\varepsilon} \) is the mean energy, typically the excess energy of reaction, and \( \sigma_{\varepsilon} \) is the width. The dissociation energy may also be chosen to have zero width:

\[
f(\varepsilon) = \delta(\varepsilon - \bar{\varepsilon}) \tag{4.13}
\]

If efficient distribution of energy occurs prior to dissociation, unimolecular decay theory predicts an exponential translational energy release distribution.\(^{37,38}\) For this case, the energy of the dissociating molecule is chosen from the distribution

\[
f(\varepsilon) = \frac{1}{\bar{\varepsilon}} e^{-\frac{\varepsilon}{\bar{\varepsilon}}} \tag{4.14}
\]

where \( \bar{\varepsilon} \) is the mean translational energy release.
4.4 Post-Dissociation

Following dissociation, the total relative energy of the product ion pair is calculated. If the energy is negative, the pair is bound and is unable to separate. If the energy is positive, the positions and velocities of the final products at the end of the remaining interaction time are determined from classical orbital theory. If only those product ion pairs traveling in the $xy$ plane are of interest, ions with $z$-components of velocity outside of a narrow range are discarded. The further transverse displacement that occurs during ion passage through the TOFMS to the PSD is calculated and is used to determine the ion arrival position at the PSD. The arrival position of the ion is weighted by the electron probability density at the capture location. Ion arrival position distributions are built up by accumulating the weighted ion arrival positions that result from a large number of collision events.

4.5 Classical Orbital Mechanics

Prior to and following dissociation, the product ions interact electrostatically. Finding the final position and velocity of an ion from the initial position and velocity, the prediction problem, has no analytical solution. This simulation uses two approaches to solving the prediction problem. For the majority of trajectories, the elliptic and hyperbolic Kepler equations are used to obtain the positions and velocities of the product ions.\textsuperscript{39-41} The Kepler equations are readily solved using standard numerical techniques, such as the Newton-Raphson method. However, the Kepler equations converge poorly and suffer a loss of accuracy for near parabolic trajectories.\textsuperscript{42} For this case, the universal variable formulation may be applied.\textsuperscript{43,44} The universal variable formulation is general for all trajectories; however, it is used only for near parabolic cases in this application because it involves solving for a variable that increases without bound for open trajectories. In the present simulation, the variable tends to become too large to be handled computationally. However, it has been observed that the universal formulation remains manageable for
near parabolic trajectories, and thus may used for the trajectories for which the Kepler equations converge poorly.
Chapter 5
Data Analysis

The radial and angular position distributions of the products of electron transfer reactions of the type
\[ K(np) + AB \rightarrow K^+ + AB^* \rightarrow K^+ + A^- + B \]  
(5.1)
are compared with Monte Carlo calculations to obtain information about the decay energetics and an estimate of the \( AB^{*-} \) intermediate lifetime.

5.1 Radial Distribution of Position Data

At high-\( n \), electrostatic interactions between the product \( K^+ \)-negative ion pairs are negligible. Because the ions are formed moving isotropically in a small volume, the negative ion arrival position distribution on the PSD is radially symmetric about the interaction volume. For ions that do not dissociate, the observed position distribution is governed by the thermal velocity distribution of the parent molecules. For product ions that result from dissociation, the observed distribution results from a convolution of the dissociation energetics of the intermediate with the thermal velocity distribution of the parent molecules.

To analyze the radial distribution of ion arrival positions on the PSD, the data are binned into rings. The sum in each ring is then divided by the area of the ring. The position distributions are recorded in a rectangular array, typically 128 x 128 elements. The problem is to distribute data in square elements into rings centered on an arbitrary point. Fig. 5.1 illustrates this problem. The most straightforward method is to assign the signal in a particular PSD element to a ring if the center of the element is contained within the ring. However, a significant number of PSD elements are contained in multiple rings. To correct for this problem, the signal in a particular PSD element is assigned to multiple rings in proportion to the area of the element within each ring. These two methods are illustrated in Figs. 5.2 and 5.3. Figure 5.2 shows a circular
FIG. 5.1. Superposition of rings on the PSD. The signal contained in each PSD element must be assigned to one or more rings. One method is to assign the signal in a PSD element to a ring based on the location of the midpoint of the element, for which the signal in the shaded element is assigned to ring 2. A second method is to assign the signal to multiple rings based on the fractional area of the PSD element occupying each ring, for which the signal in the shaded element is assigned proportionally to rings 1 and 2.
FIG. 5.2. For (a) a circular distribution on the PSD, (b) (-----), (▲), and (■) show radial distributions that are exact, calculated by assigning the signal in each PSD element to a radial bin based on the midpoint of the element, and calculated by assigning the signal in each PSD element to a radial bin based on the fraction of the element contained in the bin, respectively.
FIG. 5.3. For (a) a gaussian distribution on the PSD, (b) (---), (▼), and (■) show radial distributions that are exact, calculated by assigning the signal in each PSD element to a radial bin based on the midpoint of the element, and calculated by assigning the signal in each PSD element to a radial bin based on the fraction of the element contained in the bin, respectively.
FIG. 5.4. Flowchart of method to calculate the radial distribution of arrival position data on the PSD.
distribution on the PSD and the corresponding radial distribution computed with each method, along with the exact distribution. Figure 5.3 shows the same quantities for a gaussian distribution on the PSD. As is evident, the method assigning the signal in each PSD element to multiple rings yields better results and is the one used in the analysis of the present data. A flow chart of this method is shown in Fig. 5.4, and an example algorithm in C is given in appendix B.

5.2 Angular Distribution of Position Data

At low-n, electrostatic interactions between the product K*-negative ion pairs may be significant. As a result, the trajectories of the product ions may be significantly perturbed and the arrival position distributions of negative ions will not be radially symmetric. It is therefore more useful to determine the distribution as a function of the angle between the initial Rydberg atom velocity vector (the atomic beam direction) and the final negative ion trajectory in the xy plane.

To analyze the angular distribution of ion arrival positions on the PSD, the data are binned into wedges. The problem is to distribute data in square elements into wedges centered on an arbitrary point. Fig. 5.5 illustrates this problem. The most straightforward method is to assign the signal in a particular PSD element to a wedge if the center of the element is contained within the wedge. However, a significant number of elements lie on a boundary. To correct for this problem, the signal in a particular PSD element is assigned to multiple wedges, in proportion to the area of the element within each wedge. These two methods are illustrated in Figs. 5.6 and 5.7. Figure 5.6 shows the angular distribution computed with each method along with the exact distribution for the circular and gaussian position distributions shown in Figs. 5.2 and 5.3. Figure 5.7 shows a circular distribution modulated by a cosine function on the PSD and the angular distribution computed with each method along with the exact distribution. As is evident, the method assigning the signal in each PSD element to multiple wedges yields better
FIG. 5.5. Superposition of angular bins on the PSD. The signal contained in each PSD element must be assigned to one or more wedges. One method is to assign the signal in a PSD element to a wedge based on the location of the midpoint of the element, for which the signal in the shaded element is assigned to wedge 2. A second method is to assign the signal to multiple wedges based on the fractional area of the PSD element occupying each wedge, for which the signal in the shaded element is assigned proportionally to wedges 1, 2, and 3.
FIG. 5.6. ( ), (v), and (■) show angular distributions that are exact, calculated by assigning the signal in each PSD element to an angular bin based on the midpoint of the element, and calculated by assigning the signal in each PSD element to an angular bin based on the fraction of the element contained in the bin, respectively, for (a) a circular distribution and (b) a gaussian distribution on the PSD.
FIG. 5.7. For (a) a circular distribution modulated by $\sin^2 \theta$ on the PSD, (b) (---), (▼), and (●) show angular distributions that are exact, calculated by assigning the signal in each PSD element to an angular bin based on the midpoint of the element, and calculated by assigning the signal in each PSD element to an angular bin based on the fraction of the element contained in the bin, respectively.
FIG. 5.8. Flowchart of method to calculate the angular distribution of arrival position data on the PSD.
results and is the one used in the analysis of the present data. A flow chart of this method is shown in Fig. 5.8, and an example algorithm in C is given in appendix C.

5.3 Instrumental Width

The transverse position distribution for negative ions resulting from the electron transfer reaction

\[ K(55p) + SF_6 \rightarrow K^+ + SF_6^- \]  \hspace{1cm} (5.2)

is shown in Fig. 5.9, along with the expected distribution based on the thermal one-dimensional Boltzmann distribution

\[ f(\nu) = \frac{2}{\nu \sqrt{\pi \nu_p}} e^{-\frac{\nu^2}{\nu_p^2}} \]  \hspace{1cm} (5.3)

The observed distribution is wider than predicted due to instrumental broadening. This broadening is most likely due to spurious deflections of the ions produced during passage through the fine-mesh grids separating regions in the mass-spectrometer. Slight inhomogeneities in the accelerating fields in the TOFMS may also produce deflections. To account for this width, the measured width of ions due to reaction 5.2 is assumed to be the convolution of the thermal width and an instrumental width characteristic of the experimental system. It is important to note that the instrumental width is dependent on the TOFMS voltage conditions. The instrumental width is included when comparing the results of Monte Carlo simulation calculations with experimental measurements.

5.4 Decay Energetics

Information about the decay energetics of the $AB^{-*}$ intermediate may be obtained from measurements at high-$n$ where electrostatic interactions between the product ions are negligible.

For a very short intermediate lifetime (lifetime $\ll$ typical vibrational period), the dissociation will occur before significant distribution of the excess energy of reaction among internal vibrational modes can take place. In this event, all of the energy should
FIG. 5.9. Transverse arrival position distribution of SF$_6^-$ ions formed in K(55p)/SF$_6$ collisions. ■, experimental data; the solid line shows the expected thermal distribution.
appear in translation, leading to a narrow translational energy release distribution centered on the available excess energy. The excess energy of reaction is given by

$$E = EA(A^-) - D_0(A-B) + E_{\text{int}} + E_k(e^-)$$  \hspace{1cm} (5.4)

where $EA(A^-)$ is the electron affinity of $A$, $D_0(A-B)$ is the $A-B$ bond dissociation energy, $E_k$ is the median kinetic energy of the attached electron, and $E_{\text{int}}$ is the usable internal energy in the parent molecule. If the dissociation is rapid, $E_{\text{int}}$ should simply comprise the thermal energy in the $A-B$ stretch vibration plus the thermal energy of rotation about an axis perpendicular to the dissociation axis. Comparison of model calculations assuming a narrow translational energy release distribution centered on the excess energy with experiment will demonstrate if all excess energy appears in translation.

If efficient redistribution of energy occurs prior to dissociation, unimolecular decay theory predicts a translational energy release distribution of the form$^{38}$

$$f(e) = \frac{1}{\bar{e}} e^{-\frac{e}{\bar{e}}}$$  \hspace{1cm} (5.5)

where $\bar{e}$ is the mean translational energy release. A simple estimate of $\bar{e}$ may be obtained from

$$\bar{e} = \frac{E^*}{n}$$  \hspace{1cm} (5.6)

where $E^*$ is the energy in excess of the minimum required for dissociation and $n$ is the number of vibrational modes of the intermediate.$^{45}$ $E^*$ is given by Eq. 5.4 where $E_{\text{int}}$ comprises the total thermal vibrational and rotational energy in $AB$. Comparison of model calculations assuming an exponential translational energy release distribution with experiment will demonstrate if the release distribution is exponential.

5.5 Lifetime

The lifetime of the $AB^{-*}$ intermediate is investigated at low-$n$ where electrostatic interactions between the $K^+AB^{-*}$ product ions are important. If the $AB^{-*}$ intermediate decays rapidly (on a time period very short compared to the typical $K^+AB^{-*}$ orbital
period), the initial K$^+$ trajectory is not significantly perturbed. Following dissociation, A$^-$ ions formed traveling parallel to the atomic beam direction have smaller relative velocities than A$^-$ ions formed traveling antiparallel to the beam direction. As a consequence, ions formed traveling parallel to the beam are likely to be electrostatically bound and not be observed. Conversely, ions formed traveling antiparallel to the beam are less likely to be bound, resulting in an asymmetric arrival position distribution. If the AB$^{-*}$ intermediate is long lived, the electrostatic interaction between the K$^+$–AB$^{-*}$ ion pair may result in significant deflection of the K$^+$ trajectory prior to dissociation, thus reducing the asymmetry. To obtain an estimate of the intermediate lifetime, experimental measurements of the negative ion arrival position distributions are compared with model calculations assuming different intermediate lifetimes.
Chapter 6
Results and Discussion

Experimental measurements of negative ion distributions resulting from the
dissociative electron attachment processes

\[ K(np) + CF_3I \rightarrow K^+ + CF_3I^- \rightarrow K^+ + I^- + CF_3 \]  \hspace{1cm} (6.1)

\[ K(np) + CCl_4 \rightarrow K^+ + CCl_4^- \rightarrow K^+ + Cl^- + CCl_3 \]  \hspace{1cm} (6.2)

are compared with model calculations. Information about the decay energetics is
obtained and the lifetime of each intermediate is estimated.

6.1 CF₃I

Arrival position distributions for ions formed through the dissociative electron
transfer reaction

\[ K(55p) + CF_3I \rightarrow K^+ + CF_3I^- \rightarrow K^+ + I^- + CF_3 \]  \hspace{1cm} (6.3)

are shown in Fig.6.1 for a Rydberg atom velocity of 600 m/s. The data in Fig. 6.1 (and in
later figures) are for K⁺–I⁻ ion pairs traveling in the xy plane. The shading indicates the
relative I⁻ ion signal at different elements of the PSD. If the lifetime of the intermediate
is very short, dissociation will occur before significant distribution of the excess energy
of reaction among internal vibrational modes of the intermediate can occur. All of the
excess energy should then appear in translation, leading to a narrow translational energy
release distribution centered on the available excess energy. Figure 6.1 includes the
results of model calculations undertaken assuming that the lifetime of the CF₃-I⁻
intermediate is zero and gaussian translational energy release distributions of 50 and 250
meV FWHM centered on a mean value \( \bar{\epsilon} = 800 \) meV. This value is the estimated excess
energy of reaction which is given by

\[ E = EA(I^-) - D_0(CF_3 - I) + E_{int} + E_k(e^-) \]  \hspace{1cm} (6.4)

where \( EA(I^-) \) is the electron affinity of I (3.063 eV), \( D_0(CF_3-I) \) is the CF₃-I bond
dissociation energy (2.30 ± 0.02 eV)⁴⁶, \( E_k \) is the median kinetic energy of the attached
FIG. 6.1. Arrival position distributions of $I^-$ ions formed in CF$_3$I/K(55p) collisions for a K(55p) velocity of 600 m/s. The model predictions assume a translational energy release of the form (a) a gaussian of 50 meV FWHM centered on $\epsilon = 0.8$ eV, (b) a gaussian of 250 meV FWHM centered on $\epsilon = 0.8$ eV, (c) $\exp(-\epsilon/\epsilon)$ with $\epsilon = 0.105$ eV. The shading indicates the relative $I^-$ signal at different elements of the PSD; the light gray, mid-gray, and black areas correspond to 10-40%, 40-70%, and 70-100% of the peak signal, respectively.
electron (1 meV at n = 55), and $E_{\text{int}}$ is the usable internal energy in the parent molecule. For a rapid dissociation, $E_{\text{int}}$ should comprise simply the thermal energy in the C-I stretch vibration (0.012 eV) plus the thermal energy of rotation about an axis perpendicular to the molecular symmetry axis (0.013 eV). As is evident, the data point to a narrow translational energy release distribution centered on the excess energy. This conclusion is consistent with a previous measurement made in this laboratory which determined a translational energy release $\epsilon = 820 \pm 100$ meV.\textsuperscript{15,45} This result suggests that the dissociation of the CF$_3$I$^-$ intermediate is rapid and that no significant distribution of the excess energy of reaction among internal vibrational modes occurs prior to dissociation. If, however, efficient distribution of excess energy occurs prior to dissociation, unimolecular decay theory predicts an exponential translational energy release distribution given by Eq. 5.5. An estimate of the mean translational energy from Eq. 5.6 is $\bar{\epsilon} = 105$ meV. Note that, for this case, $E_{\text{int}}$ comprises the total thermal rotational and vibrational energy of the parent CF$_3$I molecule. The $\Gamma^-$ arrival position distribution calculated using such a form is also shown in Fig. 6.1, and is clearly different from the experimental data.

The sensitivity of the computed arrival position distributions to the assumed mean translational energy release $\bar{\epsilon}$ is illustrated in Fig. 6.2. This figure shows the radial arrival position distribution for $\Gamma^-$ ions formed in K(55p)/CF$_3$I collisions at a K(55p) velocity of 600 m/s. The horizontal error bars indicate the uncertainty in arrival position associated with the spatial calibration of the PSD. Figure 6.2 also includes model predictions derived assuming values of $\bar{\epsilon}$ of 0.80, 0.75 and 0.70 eV. Inspection of the data indicates that, to within experimental error, the mean translational energy release is equal to the excess energy of reaction. Further, because the energies associated with the symmetric C-F stretch in the CF$_3$ radical (0.135 eV) and the CF$_3$ umbrella motion (0.087 eV) are quite large, this suggests that energy is not coupled into these motions.
FIG. 6.2. Radial arrival position distributions of $\Gamma^-$ ions formed in K(5p)/CF$_3$I collisions with a K(5p) velocity of 600 m/s. ■, experimental data; the curves show model predictions assuming a gaussian translational energy release distribution of 50 meV FWHM centered on mean translational energy releases $\epsilon$ of (——) 0.81 eV, (-------) 0.75 eV, and (-----) 0.70 eV.
during dissociation of the CF$_3$I$^{-*}$ intermediate.

The lifetime of the CF$_3$I$^{-*}$ intermediate is investigated at low-$n$ where electrostatic interactions between the K$^+$/CF$_3$I$^{-*}$ product ions are important. Arrival position distributions for I$^-$ ions formed through the dissociative electron transfer reaction

$$K(14p) + CF_3I \rightarrow K^+ + CF_3I^{-*} \rightarrow K^+ + I^- + CF_3$$  \hspace{1cm} (6.5)

are shown in Fig. 6.3 for Rydberg atom velocities of 350, 650 and 950 m/s. The I$^-$ distribution is markedly asymmetric; the majority of the detected ions have significant velocity components antiparallel to the initial velocity of the Rydberg atoms, the effect being most pronounced at low Rydberg atom velocities. This is a direct consequence of post-attachment electrostatic interactions$^{18}$ and results because the mean velocity of I$^-$ ions formed by dissociative capture is similar to that of the K$^+$ core ions. Thus if the K$^+$ and I$^-$ ions are initially traveling in the same general direction their kinetic energy of relative motion will be small and a sizable fraction of the ion pairs will remain electrostatically bound and not be detected. If, however, the ions are initially traveling in opposite directions their relative kinetic energy will be much greater permitting a larger fraction of the ion pairs to separate.

The sensitivity of calculated arrival position distributions to the assumed lifetime of the CF$_3$I$^{-*}$ intermediate is illustrated in Fig. 6.4 which compares the experimental data recorded for a K(14p) velocity of 350 m/s with model predictions derived assuming CF$_3$I$^{-*}$ lifetimes of 0.2, 5 and 10 ps. As $\tau$ increases the asymmetry in the arrival position distribution is reduced. This results because electrostatic interactions between the K$^+$ core ion and CF$_3$I$^{-*}$ intermediate cause significant K$^+$ deflection prior to dissociation. Following dissociation, interactions between the product K$^+$ and I$^-$ ions still occur but the initial velocities of the K$^+$ ions at the time of dissociation are more random resulting in a
FIG. 6.3. Arrival position distributions of $I^{-}$ ions formed in CF$_3$I/K(14p) collisions for K(14p) velocities of 350, 650, and 950 m/s. The model predictions assume a gaussian translational energy release of 50 meV FWHM centered on $\varepsilon = 0.8$ eV and a CF$_3$I$^{-*}$ lifetime of 0 ps. The shading indicates the relative $I^{-}$ signal at different elements of the PSD; the light gray, mid-gray, and black areas correspond to 10-40%, 40-70%, and 70-100% of the peak signal, respectively.
FIG. 6.4. Arrival position distributions of $\Gamma$ ions formed in $\text{CF}_3I/K(14p)$ collisions with a $K(14p)$ velocity of 350 m/s. The model calculations assume a gaussian translational energy release of 50 meV FWHM centered on $\varepsilon = 0.8$ eV and a $\text{CF}_3I^{-}$ lifetime of (a) 0, (b) 2, (c) 5, and (d) 10 ps. The shading indicates the relative $\Gamma$ signal at different elements of the PSD; the light gray, mid-gray, and black areas correspond to 10-40%, 40-70%, and 70-100% of the peak signal, respectively.
FIG. 6.5. Angular distributions of $\Gamma^-$ ions formed in K(14p)/CF$_3$I collisions for K(14p) velocity of 350 m/s. ■, experimental data; the solid lines show model predictions for assumed CF$_3$I$^{\Gamma^-}$ mean lifetimes of (a) 0, (b) 2, (c) 5, and (d) 10 ps.
less asymmetric $\Gamma^-$ distribution. The sensitivity of the $\Gamma^-$ distribution to $\tau$ is also
demonstrated in Fig. 6.5 which shows the measured and calculated $\Gamma^-$ angular
distributions expressed as a function of the angle between the initial Rydberg atom and
the final $\Gamma^-$ trajectories. The data are normalized to equal scattered $\Gamma^-$ signals. Inspection
of the data in Figs. 6.4 and 6.5 indicates that the angular distribution is quite sensitive to
the assumed lifetime and suggests that the lifetime of the $\text{CF}_3\Gamma^{-*}$ intermediates is very
short, $< 2 \text{ ps}$. 

6.2 CCl$_4$

Arrival position distributions for ions formed through the dissociative electron
transfer reaction

$$K(55p) + \text{CCl}_4 \rightarrow K^+ + \text{CCl}_4^{-*} \rightarrow K^+ + \text{Cl}^- + \text{CCl}_3$$

(6.6)

are shown in Fig. 6.6 for a Rydberg atom velocity of 375 m/s. The data in Fig. 6.6 (and in
later figures) are for $K^+-\text{Cl}^-$ ion pairs traveling in the $xy$ plane. The shading indicates the
relative $\text{Cl}^-$ ion signal at different elements of the PSD. The corresponding radial
distributions are shown in Fig. 6.7. As is the case for $\text{CF}_3\Gamma^{-*}$, if the lifetime of the
intermediate is very short, dissociation will occur before significant distribution of the
excess energy of reaction among internal vibrational modes of the intermediate can occur.
All of the excess energy should then appear in translation, leading to a narrow
translational energy release distribution centered on the available excess energy. Figures
6.6 and 6.7 include the results of a model calculation undertaken assuming that the
lifetime of the $\text{CCl}_4^{-*}$ intermediate is zero and that the translational energy release
distribution is a gaussian of 50 meV FWHM centered on a mean value $\bar{\epsilon} = 610 \text{ meV}$.
This value is the estimated excess energy of reaction which is given by

$$E = EA(\text{Cl}^-) - D_0(\text{CCl}_3 - \text{Cl}) + E_{\text{int}} + E_k(\epsilon^-)$$

(6.7)
FIG. 6.6. Arrival position distributions of Cl\textsuperscript{−} ions formed in K(55p)/CCl\textsubscript{4} collisions a K(55p) velocity of 375 m/s. The model predictions assume translational energy release distributions of the form (a) a gaussian of 50 meV FWHM centered on \( \varepsilon = 0.61 \) eV, (b) \( \exp(-\varepsilon/\bar{\varepsilon}) \) with \( \bar{\varepsilon} = 0.1 \) eV, (c) a constant extending from 0 to 0.3 eV. The shading indicates the relative Cl\textsuperscript{−} signal at different elements of the PSD; the light gray, mid-gray, and black areas correspond to 10-40%, 40-70%, and 70-100% of the peak signal, respectively.
FIG. 6.7. Radial arrival position distributions of Cl ions formed in K(55p)/CCl₄ collisions with a K(55p) velocity of 375 m/s. ■, experimental data; the curves show model predictions assuming translational energy release distributions of the form (· · · · · ·) a gaussian of 50 meV FWHM centered on ε = 0.61 eV, (- - - - - ) a constant extending from 0 to 0.3 eV, and (-----) exp(-ε/ ε) with ε = 0.1 eV.
where $EA(\text{Cl}^-)$ is the electron affinity of Cl (3.613 eV), $D_0(\text{CICl}_3$-Cl) is the CCl$_3$-Cl bond dissociation energy (3.02 eV)$^{45}$, $E_k$ is the median kinetic energy of the attached electron (1 meV at $n = 55$), and $E_{int}$ is the usable internal energy in the parent molecule. For a rapid dissociation, $E_{int}$ should comprise simply the thermal energy of rotation about an axis perpendicular to the molecular symmetry axis (0.013 eV) plus the thermal energy in the C-Cl stretch vibration. CCl$_4$ has four C-Cl stretching modes. An estimate of the vibrational energy in the dissociating bond coordinate is the mean energy in the four stretching modes, 0.004 eV. Comparison of model calculations undertaken assuming that the lifetime of the CCl$_4$–* intermediate is zero and that the translational energy release distribution is a narrow Gaussian of 50 meV FWHM centered on $\bar{\epsilon} = 0.61$ eV are included in Figs. 6.6 and 6.7. The calculations are in poor agreement with experiment. In particular, the measured radial distribution of ion arrival positions is much tighter indicating that the Cl$^-$ ions must be formed with much smaller kinetic energies. Thus the lifetime of the CCl$_4$–* intermediate must be sufficient to allow distribution of the excess energy of reaction among internal vibrational modes of the intermediate prior to dissociation.

If efficient redistribution of excess energy occurs prior to dissociation, unimolecular decay theory predicts an exponential translational energy release distribution given by Eq. 5.5 of the form $\exp(-\bar{\epsilon}/\bar{\epsilon})$ where $\bar{\epsilon}$ is the mean translational energy release. Cl$^-$ arrival position distributions and radial distributions calculated using such a distribution are included in Figs. 6.6 and 6.7, respectively. Based on earlier measurements and theoretical predictions,$^{14,47}$ a mean translational energy release $\bar{\epsilon}$ of 0.1 eV is assumed. Agreement between experiment and the model calculations is excellent, confirming that efficient distribution of the excess energy of reaction occurs prior to dissociation. Although the present results are consistent with the earlier value of $\bar{\epsilon}$, the computed distributions were found to be relatively insensitive to the exact value of
selected. However, the calculated radial distributions are quite sensitive to the form of the translational energy release distribution. For example, as illustrated in Fig. 6.7, the data are not well fit using a translational energy release distribution that is rectangular in shape and extends from 0 to 0.3 eV. This distribution was chosen to represent an intermediate case where the excess energy is not completely distributed prior to dissociation.

The lifetime of the CCl$_4^{-*}$ intermediate is investigated at low-$n$ where electrostatic interactions between the K$^+\cdot$CCl$_4^{-*}$ product ions are important. Arrival position distributions for Cl$^-$ ions formed through the dissociative electron transfer reaction

$$K(14p) + CCl_4 \rightarrow K^+ + CCl_4^{-*} \rightarrow K^+ + Cl^- + CCl_3$$

for a Rydberg atom velocity of 375 m/s is shown in Fig. 6.8, expressed as a function of the angle $\theta$ between the initial Rydberg atom velocity vector and the final Cl$^-$ trajectory in the laboratory. To improve the statistics, measurements at scattering angles of $\pm \theta$ are averaged. The results reveal a minimum in the angular distribution in the forward direction. This is a direct consequence of post attachment electrostatic interactions between the product ions.

Calculated angular distributions for the Cl$^-$ fragment are quite sensitive to the assumed lifetime of the CCl$_4^{-*}$ intermediate. This is illustrated in Fig. 6.8 which includes angular distributions derived assuming CCl$_4^{-*}$ lifetimes $\tau$ of 0, 5, 7.5, 10, 12.5 and 15 ps. The asymmetry in the calculated angular distributions becomes less pronounced as the assumed mean lifetime increases. This results because greater deflection of the K$^+$ core ion trajectory takes place prior to dissociation of the CCl$_4^{-*}$ intermediate. Post-attachment interactions between the product K$^+\cdot$Cl$^-$ ion pairs still occur but the K$^+$ ion velocity at the time of dissociation is more random, thus reducing the angular asymmetry. The quality of the fits between theory and experiment is indicated in
FIG. 6.8. Angular distributions of Cl\textsuperscript{−} ions formed in K(14p)/CCl\textsubscript{4} collisions for K(14p) velocity of 375 m/s. ■, experimental data; the solid lines show model predictions for assumed CCl\textsubscript{4}− mean lifetimes of (a) 0, (b) 5, (c) 7.5, (d) 10, (e) 12.5, and (f) 15 ps.
FIG. 6.9. Dependence of $\chi^2$ on the assumed lifetime for angular distributions of Cl$^-$ ions formed in K(14p)/CCl$_4$ collisions and a Rydberg atom velocity of 375 m/s.
Fig. 6.9 which shows the dependence of $\chi^2$ on the assumed lifetime. Inspection of the data in Figs. 6.8 and 6.9 points to a CCl$_4^-$ lifetime of $7.5 \pm 2.5$ ps. The uncertainty is primarily due to statistical uncertainty and error in determination of the K$^+$ formation position, which is required to calculate the angular distribution.

Angular distributions were also measured for higher Rydberg atom velocities and larger values of $n$. In each case the experimental data were optimally fit by model calculations that assume a CCl$_4^-$ lifetime of 7.5 ps. The quality of the fits is illustrated in Fig. 6.10 which includes data for K(14p) collisions at higher Rydberg atom velocities and in Fig. 6.11 which shows data for K(15p) and K(16p) collisions at Rydberg atom velocity 375 m/s. Both figures include results of model calculations that assume a CCl$_4^-$ lifetime of 7.5 ps. Interestingly, both theory and experiment show that the form of the Cl$^-$ angular distribution is quite sensitive to the initial Rydberg atom velocity, and for velocities $\geq 800$ m/s becomes forward peaked. In this regime, the Rydberg atom velocity exceeds that typical of Cl$^-$ ions formed by CCl$_4^-$ dissociation. Thus a substantial fraction of even those Cl$^-$ ions initially formed traveling antiparallel to the K$^+$ core ion will, because of the center-of-mass motion of the K$^+$-Cl$^-$ system, acquire a forward component of velocity. It should be noted, however, that calculated angular distributions become less sensitive to the assumed lifetime as the Rydberg atom velocity is increased.

It is instructive to compare the present lifetime with that expected on the basis of Rice-Ramsberger-Kassel-Marcus (RRKM) theory.$^{48,49}$ Such theory suggests CCl$_4^-$ lifetimes that decrease from $\sim 40$ to $\sim 15$ ps as the energy of the attached Rydberg electron increases from 0 to 200 meV.$^4$ (For a 14p state the time-averaged kinetic energy of the Rydberg electron is $\sim 94$ meV, but the median energy of those Rydberg electrons that actually attach is $\sim 19$ meV.$^{13}$) The lifetimes calculated using RRKM theory, however, are sensitive to the assumed parameters including the C-Cl stretching frequencies for the
CCl$_4^-$ intermediate (which can only be estimated) and the number of modes active in the transition complex. Although uncertainties in these quantities preclude accurate calculation of the CCl$_4^-$ lifetime, it is, nonetheless, apparent that the lifetime deduced by analysis of the experimental data is in reasonable accord with that expected theoretically.
FIG. 6.10. Velocity dependence of the angular distributions of Cl\textsuperscript{−} ions formed in K(55p)/CCl\textsubscript{4} collisions. ■, ○, ●, experimental results for K(14p) velocities of 375, 600, and 900 m/s, respectively. The solid lines show model predictions for CCl\textsubscript{4}\textsuperscript{−} mean lifetime of 7.5 ps and K(14p) velocities of (a) 200, (b) 375, (c) 600, (d) 800, (e) 900, and (f) 1000 m/s.
FIG. 6.11. Angular distributions of Cl$^-$ ions formed in K(np)/CCl$_4$ collisions. The K(np) velocity is 375 m/s. ■, experimental data for (a) K(15p) and (b) K(16p); the solid lines show model predictions for assumed CCl$_4$$^{18}$ mean lifetime of 7.5 ps.
Chapter 7
Future Work

The techniques developed in this work can be used in additional experiments, the most straightforward of which is the determination of the lifetimes of other transient negative ions. More speculative experiments include investigation of the production and properties of exotic negative ions.

7.1 Transient Negative Ions

The present work has not exhausted the set of transient negative ions that can be studied by this technique. CH₃I and CF₃Br are possibilities; however, they are uninteresting. These molecules are similar to CF₃I in that all of the excess energy of reaction appears in translation, which implies that they dissociate immediately. A more promising candidate for study is CFCl₃, which is similar to CCl₄. Previous measurements have demonstrated that the intermediate CFCl₃⁻ ion decays with an exponential energy release distribution, which implies that it has a non-zero lifetime.

One promising target that is different from the two studied in the present work is CF₂Br₂, which has two dissociation channels:

\[ \text{K(np)} + \text{CF}_2\text{Br}_2 \rightarrow \text{K}^+ + \text{CF}_2\text{Br} + \text{Br}^- \]  \hspace{1cm} (7.1)

\[ \text{K(np)} + \text{CF}_2\text{Br}_2 \rightarrow \text{K}^+ + \text{CF}_2 + \text{Br}_2^- \]

Previous studies indicate that the CF₂Br₂⁻ intermediate has a non-zero lifetime. Measurements of the lifetime of the intermediate ion can probe the details of the attachment process. One possibility is that electron attachment proceeds into two separate precursor states that dissociate into the products:

\[ \text{K(np)} + \text{CF}_2\text{Br}_2 \rightarrow \text{K}^+ + \text{CF}_2\text{Br}_2^- \rightarrow \text{K}^+ + \text{CF}_2\text{Br} + \text{Br}^- \]  \hspace{1cm} (7.2)

\[ \text{K(np)} + \text{CF}_2\text{Br}_2 \rightarrow \text{K}^+ + \text{CF}_2\text{Br}_2^- \rightarrow \text{K}^+ + \text{CF}_2 + \text{Br}_2^- \]  \hspace{1cm} (7.3)
For this case, each dissociation channel has a different lifetime. Another possibility is that, following the initial attachment,

\[ K(np) + CF_2Br_2 \rightarrow K^+ + CF_2Br_2^- \]  

(7.4)

the Br⁻ production results from a two step process:

\[ CF_2 + Br_2^- \rightarrow CF_2 + Br + Br^- \]

(7.5)

\[ CF_2Br_2^- \rightarrow CF_2 + Br^- \]

To investigate this possibility, the Monte Carlo simulation must be modified to model a two step dissociation. In all cases, because the product of the second channel is polyatomic, there may be internal-to-translational energy transfer that would render determination of the lifetime by this method difficult. This possibility must be considered when undertaking measurements with CF₂Br₂.

Other possibilities for study that have multiple dissociation channels include SF₅Cl and SF₅NCO. SF₅Cl has two channels following low energy electron attachment:\(^{51}\)

\[ e^- + SF_5Cl \rightarrow SF_5Cl^- \]  

(7.6)

\[ e^- + SF_5Cl \rightarrow SF_5^+ + Cl^- \]  

(7.7)

The translational energy release upon dissociation is small, suggesting a non-zero intermediate lifetime. SF₅NCO has several dissociation channels with small translational energy release following low energy electron attachment:\(^{10}\):

\[ e^- + SF_5NCO \rightarrow SF_4NCO^- + F \]  

(7.8)

\[ e^- + SF_5NCO \rightarrow SF_5^- + NCO \]  

(7.9)

\[ e^- + SF_5NCO \rightarrow SF_3N^- + F_2CO \]  

(7.10)

\[ e^- + SF_5NCO \rightarrow FNCO^- + SF_4 \]  

(7.11)

As with CF₂Br₂, any measurements undertaken with these targets must include the possibility of internal-to-translational energy transfer.
This list of potential molecules for study is not complete. A comprehensive survey of molecules that dissociate following low-energy electron attachment will certainly uncover many interesting possibilities for study by the techniques developed in the present work.

7.2 Exotic Negative Ions

Recent work has shown that exotic negative ions are formed in collisions of neutral molecules with Rydberg atoms that are not observed in collisions with free electrons. Of recent interest are dipole-bound negative ions and electron capture by $\text{C}_6\text{O}$.

If the dipole moment of a polar molecule exceeds a critical value (~2-2.5 D), it will support one or more bound states. Such ions are not formed in collisions with free electrons because there is no mechanism to remove the excess energy. However, dipole-bound negative ions are formed in collisions of polar molecules with Rydberg atoms because the core ion is available to carry away excess energy. Because the core ion is necessary to stabilize the negative ion, the free electron model does not apply to these processes. A curve crossing model has been developed that provides good agreement with experimental measurements of the rate constants. The curve crossing treatment can be readily incorporated into the Monte Carlo simulation described here. Comparisons of the velocity and angular distributions with calculations based on the curve crossing model may yield insights into the nature of the production of dipole-bound negative ions.

Low energy free electron capture by $\text{C}_6\text{O}$ is not observed; however Rydberg electron capture by $\text{C}_6\text{O}$ is observed. It has been proposed that the resulting $\text{C}_6\text{O}^-$ ion is an image-charge-bound state. $\text{C}_6\text{O}$ has a large polarizability and calculations suggest that the electron-$\text{C}_6\text{O}$ polarization potential is sufficient to support a bound state. However, unlike dipole-bound negative ions, the electron attachment process occurs at high-$\pi$ indicating that the Rydberg atom core ion is not required to carry away the excess
energy. Study of the velocity and angular distributions of the product ions may illuminate the nature of the attachment process.
Chapter 8
Conclusion

In the present work, techniques were developed to determine the lifetime of transient negative ions formed through Rydberg electron transfer. Velocity and angular data derived from arrival position distributions were compared with a Monte Carlo simulation. For CF$_3$I, the lifetime of the intermediate CF$_3$I$^{*}$ is determined to be $< 2$ ps, which is consistent with prior measurements of its decay energetics. For CCl$_4$, the intermediate CCl$_4$-$^{*}$ lifetime has been determined to be $7.5 \pm 2.5$ ps. The techniques developed may be used to measure the lifetimes of other transient species formed through electron capture and can be extended to further work, including the study of exotic negative ions.
Appendix A
Validity of the Free Electron Model for Low-Energy Electron Attachment

In order to describe the Rydberg electron transfer reaction

\[ K(n\ell) + ABC \rightarrow K^+ + ABC^- \] \[ \rightarrow K^+ + ABC + e^- \] (A.1)

in terms of a binary electron-molecule interaction, two criteria must be satisfied. First, the range of the electron-molecule interaction must be much less than the typical Rydberg electron-core separation. Second, the electron-molecule interaction time must be much shorter than the Rydberg electron orbital period. To determine the validity of the free-electron model, estimates of the range of the electron-molecule interaction and interaction time must be compared with the Rydberg electron core separation and orbital period.

The mean electron-core separation is

\[ r_e = \frac{3}{2} n^2 a_0 \] (A.2)

where \( a_0 \) is the Bohr radius, \( 5.29 \times 10^{-11} \) cm. An estimate of the typical electron-molecule interaction distance is obtained from the electron attachment cross section, \( \sigma \):

\[ r_{int} = \sqrt{\frac{\sigma}{\pi}} \] (A.3)

At low electron energies, the electron attachment cross section is inversely proportional to the electron velocity, \( v_e \):

\[ \sigma = \frac{k}{v_e} \] (A.4)

Substituting the rms velocity of the Rydberg electron for \( v_e \) yields an estimate of the electron-molecule interaction distance:

\[ r_{int} = \sqrt{\frac{kn}{\pi v_0}} \] (A.5)
where $v_0$ is the atomic unit of velocity, 2.19×10^8 cm/s. Thus, the ratio of interaction distance to electron-core separation is given by

$$\frac{r_{int}}{r_e} = \frac{2}{3n^2a_0} \sqrt{\frac{k}{\pi v_0}}$$  \hspace{1cm} (A.6)

The Rydberg electron orbital period is

$$\tau_e = 2\pi n^3 \tau_0$$  \hspace{1cm} (A.7)

where $\tau_0$ is the atomic unit of time, 2.42×10^{-17} s. An estimate of the electron-molecule interaction time is obtained from the electron-molecule interaction distance and the velocity of the electron, $v_e$:

$$\tau_{int} = \frac{r_{int}}{v_e} = \frac{1}{v_e} \sqrt{\frac{k}{\pi v_0}}$$  \hspace{1cm} (A.8)

Substituting the rms velocity of the Rydberg electron for $v_e$ yields an estimate of the electron-molecule interaction time:

$$\tau_{int} = \frac{n^4}{v_0^2} \sqrt{\frac{k}{\pi}}$$  \hspace{1cm} (A.9)

where $v_0$ is the atomic unit of velocity, 2.19×10^8 cm/s. Thus, the ratio of interaction time to the Rydberg electron orbital period is given by

$$\frac{\tau_{int}}{\tau_e} = \frac{1}{2\pi n^3 \tau_0 v_0^2} \sqrt{\frac{k}{\pi}}$$  \hspace{1cm} (A.10)

For the free-electron model to be valid, the following two criteria must be satisfied:

$$\frac{r_{int}}{r_e} \ll 1$$  \hspace{1cm} (A.11)

$$\frac{\tau_{int}}{\tau_e} \ll 1$$  \hspace{1cm} (A.12)

Table A.1 summarizes the results and presents numerical estimates based on the SF$_6$ cross section. SF$_6$ has a large cross section for low-energy electron attachment,
k = 5 \times 10^{-7} \text{ cm}^3/\text{s}^{64}, and thus is a good test case. It is evident that the two criteria are satisfied and that the free-electron model is valid for low-energy electron attachment for $n \geq 10$.

Table A.1 Free-electron model parameters for SF$_6$

<table>
<thead>
<tr>
<th>Property</th>
<th>$n$ dependence (atomic units)</th>
<th>$n = 1$</th>
<th>$n = 10$</th>
<th>$n = 50$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rydberg atom mean radius</td>
<td>$\frac{3}{2}n^2$</td>
<td>8 $\times 10^{-9}$ cm</td>
<td>8 $\times 10^{-7}$ cm</td>
<td>2 $\times 10^{-5}$ cm</td>
</tr>
<tr>
<td>Rydberg electron orbital period</td>
<td>$2\pi n^3$</td>
<td>1.5 $\times 10^{-16}$ s</td>
<td>1.5 $\times 10^{-13}$ s</td>
<td>2 $\times 10^{-11}$ s</td>
</tr>
<tr>
<td>Typical electron-molecule interaction distance</td>
<td>$\sqrt{\frac{kn}{\pi}}$</td>
<td>2.7 $\times 10^{-8}$ cm</td>
<td>8.5 $\times 10^{-8}$ cm</td>
<td>1.9 $\times 10^{-7}$ cm</td>
</tr>
<tr>
<td>Typical electron-molecule interaction time</td>
<td>$n^{\frac{3}{2}}\sqrt{\frac{k}{\pi}}$</td>
<td>1.2 $\times 10^{-16}$ s</td>
<td>3.8 $\times 10^{-15}$ s</td>
<td>4.2 $\times 10^{-14}$ s</td>
</tr>
<tr>
<td>Interaction distance Mean radius</td>
<td>$\frac{2}{3n^{\frac{3}{2}}}\sqrt{\frac{k}{\pi}}$</td>
<td>3.4</td>
<td>0.1</td>
<td>0.01</td>
</tr>
<tr>
<td>Interaction time Orbital Period</td>
<td>$\frac{1}{2\pi n^3}\sqrt{\frac{k}{\pi}}$</td>
<td>0.81</td>
<td>0.03</td>
<td>0.002</td>
</tr>
</tbody>
</table>
Appendix B
Radial Distribution Calculation Algorithm in C

/**/
******************************************************************************
/*
/* Radial distribution code
/*
/* This set of functions bins data contained in a rectangular
/* array of dimensions k_PSDsize x k_PSDsize into a radial
/* distribution of rings (# of rings in the distribution
/* = k_rings). Note that the width (delta r) of the rings is
/* k_PSDsize / k_rings. The prototype of the function is
/*
/* double *RadialDist( double [], double, double ) *
/*
/* The arguments are the array and the (x,y) center of the rings. *
/* It returns a pointer to an array of size k_rings containing *
/* the results. Note that the memory for the array containing *
/* the results is allocated by this function and that it is the *
/* responsibility of the user to free this memory if necessary. *
/*
******************************************************************************
/**/

#include <stdlib.h>
#include <math.h>

#define k_PSDsize 128
#define k_rings k_PSDsize

#define MAX( a, b ) ( ( ( a ) > ( b ) ) ? ( a ) : ( b ) )
#define MIN( a, b ) ( ( ( a ) < ( b ) ) ? ( a ) : ( b ) )

struct box
{
    double left;
    double right;
    double top;
    double bottom;
};

struct point
{
    double x;
    double y;
};

static double pi( void )
{
    return( 3.14159265358979323846 );
}

/**/
static double area( struct box therect )
{
    double result;
    result = ( therect.top - therect.bottom ) * (therect.right - therect.left);
    return( result );
}

static double segment( struct point one, struct point two, double r )
{
    double c;
    double theta;
    double result;

    if ( r == 0.0 )
    {
        return( 0.0 );
    };

    c = ( one.x - two.x ) * ( one.x - two.x ) + ( one.y - two.y ) * ( one.y - two.y );
    c = sqrt( c );
    theta = 2.0 * asin( c / ( 2.0 * r ));
    result = r * r * ( theta - sin( theta ) ) / 2.0;
    return( result );
}
static double fractional_area( struct box rect, double radius )
{
    struct point  il;   /* il-2 are points @ which the circle */
    struct point  i2;   /* intersects the box. */
    struct point  cross1;  /* cross1-4 are the points @ which the */
    struct point  cross2;  /* circle crosses each of the */
    struct point  cross3;  /* boundary lines of the box. */
    struct point  cross4;
    struct point  corner[4];  /* array containing the box corners */
    double    radius1;  /* radius 1-4 are radii to corners of */
    double    radius2;  /* the PSD bin */
    double    radius3;
    double    radius4;
    double    fraction;  /* fraction of area @ less than radius */
    double    maxx1;     /* maxx1-2 & maxy1-2 are used in */
    double    maxy1;     /* calculating an area bounded by 2 */
    double    maxx2;     /* corners & 2 intersection points. */
    double    maxy2;
    int    numcorners;  /* Number of box corners inside circle */
    int    num = 0;      /* Number of intersection points. */
    int    Should be 0 or 2;

    /* Calculate radii to each PSD bin corner */
    radius1 = sqrt( rect.bottom * rect.bottom + rect.left * rect.left );
    radius2 = sqrt( rect.top * rect.top + rect.left * rect.left );
    radius3 = sqrt( rect.bottom * rect.bottom + rect.right * rect.right);
    radius4 = sqrt( rect.top * rect.top + rect.right * rect.right );

    /* Check if the circle passes through the bin. If not, */
    /* determine if the bin is inside or outside the box. */
    /* If it does pass through the bin, find the */
    /* intersection points.

    if ( ( radius >= radius1 ) && ( radius >= radius2 ) &&
         ( radius >= radius3 ) && ( radius >= radius4 ) )
    {
        /* Bin is inside the circle */

        numcorners = 4;
    }
    else
    {
        /* Circle passes through the bin */

        /* First find the points where the circle intersects */
        /* the box. */
cross1.x = rect.left;
cross1.y = radius * radius - cross1.x * cross1.x;
cross2.x = rect.right;
cross2.y = radius * radius - cross2.x * cross2.x;
cross3.y = rect.bottom;
cross3.x = radius * radius - cross3.y * cross3.y;
cross4.y = rect.top;
cross4.x = radius * radius - cross4.y * cross4.y;

if ( cross1.y >= 0 )
{
    cross1.y = sqrt( cross1.y );
    if ( ( cross1.y >= rect.bottom ) && ( cross1.y <= rect.top ) )
    {
        i1 = cross1;
        num = 1;
    }
    else if ( ( - cross1.y >= rect.bottom ) &&
              ( - cross1.y <= rect.top ) )
    {
        cross1.y = - cross1.y;
        i1 = cross1;
        num = 1;
    }
};

if ( cross2.y >= 0 )
{
    cross2.y = sqrt( cross2.y );
    if ( ( cross2.y >= rect.bottom ) && ( cross2.y <= rect.top ) )
    {
        if ( num == 0 )
        {
            i1 = cross2;
            num = 1;
        }
        else if ( ( i1.x != cross2.x ) && ( i1.y != cross2.y ) )
        {
            i2 = cross2;
            num = 2;
        }
    }
    else if ( ( - cross2.y >= rect.bottom ) &&
              ( - cross2.y <= rect.top ) )
    {
        cross2.y = - cross2.y;
        if ( num == 0 )
        {
            i1 = cross2;
            num = 1;
        }
        else if ( ( i1.x != cross2.x ) && ( i1.y != cross2.y ) )
        {
            i2 = cross2;
num = 2;
);
);
);

if ( cross3.x >= 0 )
{
    cross3.x = sqrt( cross3.x );
    if ( ( cross3.x <= rect.right ) && ( cross3.x >= rect.left ) )
    {
        if ( num == 0 )
        {
            i1 = cross3;
            num = 1;
        }
        else if ( ( i1.x != cross3.x ) && ( i1.y != cross3.y ) )
        {
            i2 = cross3;
            num = 2;
        }
    }
    else if ( ( - cross3.x <= rect.right ) &&
              ( - cross3.x >= rect.left ) )
    {
        cross3.x = - cross3.x;
        if ( num == 0 )
        {
            i1 = cross3;
            num = 1;
        }
        else if ( ( i1.x != cross3.x ) &&
                  ( i1.y != cross3.y ) )
        {
            i2 = cross3;
            num = 2;
        }
    }
);

if ( cross4.x >= 0 )
{
    cross4.x = sqrt( cross4.x );
    if ( ( cross4.x <= rect.right ) && ( cross4.x >= rect.left ) )
    {
        if ( num == 0 )
        {
            i1 = cross4;
            num = 1;
        }
        else if ( ( i1.x != cross4.x ) && ( i1.y != cross4.y ) )
        {
            i2 = cross4;
            num = 2;
        }
    }
    else if ( ( - cross4.x <= rect.right ) &&
              ( - cross4.x >= rect.left ) )
{  
    cross4.x = -cross4.x;
    if (num == 0)
    {
        i1 = cross4;
        num = 1;
    }
    else if ((i1.x != cross4.x) && (i1.y != cross4.y))
    {
        i2 = cross4;
        num = 2;
    }
}

/* Find the corners of the box that, along with the */
/* intersection points, bound the area of interest. */

numcorners = 0;

if (radius1 <= radius)
{
    corner [numcorners].x = rect.left;
    corner [numcorners].y = rect.bottom;
    numcorners++;
}

if (radius2 <= radius)
{
    corner [numcorners].x = rect.left;
    corner [numcorners].y = rect.top;
    numcorners++;
}

if (radius3 <= radius)
{
    corner [numcorners].x = rect.right;
    corner [numcorners].y = rect.bottom;
    numcorners++;
}

if (radius4 <= radius)
{
    corner [numcorners].x = rect.right;
    corner [numcorners].y = rect.top;
    numcorners++;
}

/* if num = 1, the circle intersects a corner */
/* and no area is enclosed, so set numcorners */
/* to zero. */

if (num == 1)
{
    numcorners = 0;
}
};

/* Calculate the area. */

switch(numcorners)
{
    case 0:
        fraction = 0.0;
        break;
    case 1:
        fraction = fabs(i2.x - i1.x) * fabs(i2.y - i1.y) / 2.0;
        fraction += segment(i1, i2, radius);
        break;
    case 2:
        fraction = fabs(i2.x - i1.x) * fabs(i2.y - i1.y) / 2.0;
        maxx1 = MAX(fabs(i1.x - corner[0].x),
                    fabs(i1.x - corner[1].x));
        maxy1 = MAX(fabs(i1.y - corner[0].y),
                    fabs(i1.y - corner[1].y));
        maxx2 = MAX(fabs(i2.x - corner[0].x),
                    fabs(i2.x - corner[1].x));
        maxy2 = MAX(fabs(i2.y - corner[0].y),
                    fabs(i2.y - corner[1].y));
        fraction += MIN(maxx1, maxx2) * MIN(maxy1, maxy2);
        fraction += segment(i1, i2, radius);
        break;
    case 3:
        fraction = area(rect);
        fraction -= fabs(i2.x - i1.x) * fabs(i2.y - i1.y) / 2.0;
        fraction += segment(i1, i2, radius);
        break;
    case 4:
        fraction = area(rect);
        break;
}

fraction /= area(rect);

return(fraction);

/**
 ********************************************
 */

/**
  annulus
 */

/**
  This function calculates the
  fraction of area of a box that
  is contained between 2 circles
  at angles rlow and rhigh.
 */

/**
 ********************************************
 */

static double annulus(struct box bin, double rlow, double rhigh)
int i;
double result;
struct box smallbin[4];

/* make sure that rlow is smaller than rhigh. */
/* if not, swap them. */

if ( rlow > rhigh )
{
    result = rhigh;
    rhigh = rlow;
    rlow = result;
};

/* There are 3 tricky cases - if the origin is in the bin */
/* under consideration, or if the bin straddles the x or y */
/* axes. So, first we check to see if this is the case and */
/* take care of it by breaking the bin into smaller ones. */

if ( ( bin.left < 0.0 ) && ( bin.right > 0.0 ) &&
     ( bin.bottom < 0.0 ) && ( bin.top > 0.0 ) )
{
    smallbin[0] = bin;
    smallbin[0].right = 0.0;
    smallbin[0].bottom = 0.0;
    smallbin[1] = bin;
    smallbin[1].right = 0.0;
    smallbin[1].top = 0.0;
    smallbin[2] = bin;
    smallbin[2].left = 0.0;
    smallbin[2].bottom = 0.0;
    smallbin[3] = bin;
    smallbin[3].left = 0.0;
    smallbin[3].top = 0.0;

    /* Now calculate the fractional area subtended. */

    for ( i = 0, result = 0.0; i < 4; i++ )
    {
        result += fractional_area( smallbin[i], rhigh ) *
                   area( smallbin[i] );
        result -= fractional_area( smallbin[i], rlow ) *
                   area( smallbin[i] );
    }

    result /= area( bin );
}

else if ( bin.left < 0.0 && bin.right > 0.0 )
{
    smallbin[0] = bin;
    smallbin[0].right = 0.0;
    smallbin[1] = bin;
    smallbin[1].left = 0.0;

    /* Now calculate the fractional area subtended. */
for ( i = 0, result = 0.0; i < 2; i++ )
{
    result += fractional_area( smallbin[i], rhigh ) * 
               area( smallbin[i] );
    result -= fractional_area( smallbin[i], rlow ) * 
               area( smallbin[i] );
}
result /= area( bin );
}
else if ( bin.bottom < 0.0 && bin.top > 0.0 )
{
    smallbin[0] = bin;
    smallbin[0].bottom = 0.0;
    smallbin[1] = bin;
    smallbin[1].top = 0.0;
    /* Now calculate the fractional area subtended. */
    for ( i = 0, result = 0.0; i < 2; i++ )
    {
        result += fractional_area( smallbin[i], rhigh ) * 
                  area( smallbin[i] );
        result -= fractional_area( smallbin[i], rlow ) * 
                  area( smallbin[i] );
    }
    result /= area( bin );
}
/* If the origin is not contained in the bin, the result is */
/* straightforward to calculate. */
else
{
    result = fractional_area( bin, rhigh );
    result -= fractional_area( bin, rlow );
}
return( result );
}

/**/
/*****************************/
/*
/* RadialDist */
/*
/* Determines the radial */
/* distribution of the PSD */
/* array. */
/*
/*****************************/
/**/

double *RadialDist( double PSD[][k_PSDsize], double xcenter,
    double ycenter )
{  
  int  xindex;  /* PSD array index */  
  int  yindex;  /* PSD array index */  
  int  index;  /* Array index */  
  int  indexlow;  /* Low radius index for PSD element */  
  int  indexhigh;  /* High radius index for PSD element */  
  double x;  /* X-displacement */  
  double y;  /* Y-displacement */  
  double sum = 0.0;  /* Normalization factor */  
  double *dist;  /* Array containing radial */  
                          /* distribution */  
  double radiuslow;  /* Low radius of radial bin */  
  double radiushigh;  /* High radius of radial bin */  
  double radius1;  /* radius 1-4 are radii to corners */  
  double radius2;  /* of PSD element */  
  double radius3;  
  double radius4;  
  double dradius;  /* Size of radial bin */  
  double f;  /* Fraction of area within radial bin */  
  struct box psdbin;  /* PSD element boundaries */  
  
  /* Allocate the memory for the distribution. */  
  /* If memory allocation fails, return a */  
  /* null pointer. */  
  
  dist = malloc( (size_t) k_rings * (size_t) sizeof( double ) );  
  if ( dist == (double *) NULL )  
    return( dist );  
  
  for( index = 0 ; index < k_rings ; index++ )  
  {  
    dist[index] = 0.0;  
  }  
  
  /* Sort through PSD array and calculate radial distribution. */  
  dradius = (double) k_PSDsize / (double) k_rings;  
  
  for ( xindex = 0; xindex < k_PSDsize; xindex++ )  
  {  
    x = (double) xindex - xcenter;  
    for ( yindex = 0; yindex < k_PSDsize; yindex++ )  
    {  
      y = (double) yindex - ycenter;  
      /* Setup PSD bin */  
      psdbin.left = x;  
      psdbin.bottom = y;  
      psdbin.right = x + 1.0;  
      psdbin.top = y + 1.0;  
      
      /* Calculate radii to each PSD bin corner */  
      
      radius1 = sqrt( psdbin.bottom * psdbin.bottom +  
                      psdbin.left * psdbin.left );  
  
  
  /* Calculate radii to each PSD bin corner */  
  
  radius1 = sqrt( psdbin.bottom * psdbin.bottom +  
                  psdbin.left * psdbin.left );  
}
radius2 = sqrt( psdbin.top * psdbin.top +
              psdbin.left * psdbin.left );
radius3 = sqrt( psdbin.bottom * psdbin.bottom +
              psdbin.right * psdbin.right );
radius4 = sqrt( psdbin.top * psdbin.top +
              psdbin.right * psdbin.right );

/* Find minimum and maximum radii and  */
/* calculate index bounds */
radiuslow = MIN( MIN( radius1, radius2 ),
                 MIN( radius3, radius4 ) );
radiushigh = MAX( MAX( radius1, radius2 ),
               MAX( radius3, radius4 ) );

indexlow = (int)( radiuslow / dradius );
indexhigh = (int)( radiushigh / dradius ) + 1;

if ( indexhigh >= k_rings )
  { indexhigh = k_rings - 1; }

/* If the bin contains the origin the above procedure */
/* does not correctly calculate the low index, so set it */
/* to 0 */
if ( ( ( psdbin.top > 0.0 ) && ( psdbin.bottom < 0.0 ) ) ||
     ( ( psdbin.right > 0.0 ) && ( psdbin.left < 0.0 ) ) )
  { indexlow = 0; }

/* Go through all radial bins and increment contents */
/* accordingly. */
for( index = indexlow ; index <= indexhigh ; index++ )
  {
    /* Calculate low and high radii of the radial bin */
    radiuslow = (double) index * dradius;
radiushigh = radiuslow + dradius;

    /* Increment radial bin and normalization sum. */
    f = annulus( psdbin, radiuslow, radiushigh );
dist[index] += f * PSD[index][yindex] / ( pi() *
         ( radiushigh * radiushigh - radiuslow *
           radiuslow ) );
    sum += f * PSD[index][yindex] / ( pi() *
         ( radiushigh * radiushigh - radiuslow *
           radiuslow ) );
  }

/* Normalize distribution. */
for ( index = 0 ; index < k_rings ; index++ )
    dist[index] /= sum;

return( dist );
}
/**
 ******************************
 /*
 /* Angular distribution code */
 /*
 /* This set of functions bins data contained in a rectangular */
 /* array of dimensions k_PSDsize x k_PSDsize into an angular */
 /* distribution of wedges (# of wedges in the distribution */
 /* = k_pieslices). The prototype of the function is */
 /*
 /* double *AngularDist( double [[k_PSDsize], double, double ) */
 /*
 /* The arguments are the array and the (x,y) center of the */
 /* wedges. It returns a pointer to an array of size k_pieslices */
 /* containing the results. Note that the memory for the array */
 /* containing the results is allocated by this function and that */
 /* it is the responsibility of the user to free this memory if */
 /* necessary. */
 /*
 ******************************
 /**

#include <stdlib.h>
#include <math.h>

#define k_PSDsize 128
#define k_pieslices 36

#define MAX( a, b ) ( ( ( a ) > ( b ) ) ? ( a ) : ( b ) )
#define MIN( a, b ) ( ( ( a ) < ( b ) ) ? ( a ) : ( b ) )

struct box
{
    double left;
    double right;
    double top;
    double bottom;
};

struct point
{
    double x;
    double y;
};

static double pi( void )
{
    return( 3.14159265358979323846 );
}

static double myatan2( double x, double y)
{  
double result;
    
    result = atan2(x,y);
    if( result < 0.0 )
    {
        result += 2.0 * pi();
    }
    
    return( result );
}

/**/
/**-----------------------------------------------*/
/*
/*    *
/*    *
/*    *
/*    *
/*    *
/*    *
/*    *
/*-----------------------------------------------*/
/**

static double area( struct box therect )
{
    double result;
        
    result = ( therect.top - therect.bottom ) *
            (therect.right - therect.left );

    return( result );
}

/**/
/**-----------------------------------------------*/
/*
/*    *
/*    *
/*    *
/*    *
/*    *
/*    *
/*    *
/*-----------------------------------------------*/
/**

static double fractional_area( struct box rect, double phi )
{
    struct point  i1;  /* i1-2 are points @ which the line */
    struct point  i2;  /* intersects the box. */
    struct point  cross1; /* cross1-4 are the points @ which the */
    struct point  cross2; /* line crosses each of the */
    struct point  cross3; /* boundary lines of the box. */
    struct point  cross4;

struct point  corner[4]; /* array containing the box corners */
/* below the line */
double  theta1;  /* Theta 1-4 are angles to corners */
double  theta2; /* of PSD element */
double  theta3;
double  theta4;
double  fraction; /* fraction of area @ less than phi */
double  maxx1; /* maxx1-2 & maxy1-2 are used in */
double  maxy1; /* calculating an area bounded by 2 */
double  maxx2; /* corners & 2 intersection points. */
double  maxy2;
int  numcorners; /* Number of box corners below phi */
int  num = 0; /* Number of intersection points.. */
/* Should be 0 or 2 */

/* Calculate angles to each PSD bin corner */
theta1 = myatan2( rect.bottom, rect.left );
theta2 = myatan2( rect.top, rect.left );
theta3 = myatan2( rect.bottom, rect.right );
theta4 = myatan2( rect.top, rect.right );

/* If the origin is one corner, adjust the angles */
/* so that the correct area is determined. */
if ( ( rect.left == 0.0 ) && ( rect.bottom == 0.0 ) )
  theta1 = 2.0 * pi();
if ( ( rect.right == 0.0 ) && ( rect.bottom == 0.0 ) )
  theta3 = 2.0 * pi();
if ( ( rect.right == 0.0 ) && ( rect.top == 0.0 ) )
  theta4 = 2.0 * pi();
if ( ( rect.left == 0.0 ) && ( rect.top == 0.0 ) )
  theta4 = 2.0 * pi();

/* If the box is on the x-axis, adjust the angles */
/* so that the correct area is determined. */
if ( rect.top == 0.0 )
{
  if ( rect.left > 0.0 )
  {
    theta2 = theta4 = 2.0 * pi();
  }
  else if ( rect.right > 0.0 )
  {
    theta4 = 2.0 * pi();
  }
};

/* Check if the angle passes through the bin. If not, */
/* determine if the angle is above or below the box. */
/* if it does pass through the bin, find the */
/* intersection points.
if ( (phi <= theta1) && (phi <= theta2) &&
    (phi <= theta3) && (phi <= theta4))
{
    /* Angle is below the bin */
    numcorners = 0;
}
else if ( (phi >= theta1) && (phi >= theta2) &&
    (phi >= theta3) && (phi >= theta4))
{
    /* Angle is above the bin */
    numcorners = 4;
}
else if (phi == (pi() / 2.0))
{
    /* handle phi = pi/2 explicitly because tan(pi/2) is infinite */
    numcorners = 2;
    i1.x = i2.x = 0.0;
    i1.y = rect.bottom;
    i2.y = rect.top;
    corner[0].x = corner[1].x = rect.right;
    corner[0].y = rect.bottom;
    corner[1].y = rect.top;
}
else if (phi == (3.0 * pi() / 2.0))
{
    /* handle phi = 3pi/2 explicitly because tan(3pi/2) is infinite */
    numcorners = 2;
    i1.x = i2.x = 0.0;
    i1.y = rect.bottom;
    i2.y = rect.top;
    corner[0].x = corner[1].x = rect.left;
    corner[0].y = rect.bottom;
    corner[1].y = rect.top;
}
else if (phi == pi())
{
    /* handle phi = pi explicitly because tan(pi) is zero */
    numcorners = 2;
    i1.y = i2.y = 0.0;
    i1.x = rect.right;
    i2.x = rect.left;
    corner[0].y = corner[1].y = rect.top;
    corner[0].x = rect.left;
    corner[1].x = rect.right;
}
else
{
    /* First find the two points where the line intersects */
    /* the box. */
    cross1.x = rect.left;
cross1.y = cross1.x * tan( phi );
cross2.x = rect.right;
cross2.y = cross2.x * tan( phi );
cross3.y = rect.bottom;
cross3.x = cross3.y / tan( phi );
cross4.y = rect.top;
cross4.x = cross4.y / tan( phi );

if ( ( cross1.y > rect.bottom ) && ( cross1.y < rect.top ) )
{
    i1 = cross1;
    num++;
};

if ( ( cross2.y > rect.bottom ) && ( cross2.y < rect.top ) )
{
    if ( num == 1 )
    {
        i2 = cross2;
        num++;
    }
    else
    {
        i1 = cross2;
        num++;
    }
};

if ( ( cross3.x <= rect.right ) && ( cross3.x >= rect.left ) )
{
    if ( num == 1 )
    {
        i2 = cross3;
        num++;
    }
    else if ( num == 0 )
    {
        i1 = cross3;
        num++;
    }
};

if ( ( cross4.x <= rect.right ) && ( cross4.x >= rect.left ) )
{
    if ( num == 1 )
    {
        i2 = cross4;
        num++;
    }
};

/* If the angle cuts the box at a corner, num = 1. */
/* To properly handle this special case, the second crossing point must equal the first. */
if ( num == 1 )
{
    i2 = i1;
};

/* Find the corners of the box that, along with the */
/* intersection points, bound the area of interest. */

numcorners = 0;

if ( theta1 <= phi )
{
    corner [ numcorners ].x = rect.left;
    corner [ numcorners ].y = rect.bottom;
    numcorners++;
}

if ( theta2 <= phi )
{
    corner [ numcorners ].x = rect.left;
    corner [ numcorners ].y = rect.top;
    numcorners++;
}

if ( theta4 <= phi )
{
    corner [ numcorners ].x = rect.right;
    corner [ numcorners ].y = rect.top;
    numcorners++;
}

if ( theta3 <= phi )
{
    corner [ numcorners ].x = rect.right;
    corner [ numcorners ].y = rect.bottom;
    numcorners++;
}

};

/* Calculate the area. */

switch( numcorners )
{
    case 0 :
        fraction = 0.0;
        break;
    case 1 :
        fraction = fabs( i2.x - i1.x ) * fabs( i2.y - i1.y ) / 2.0;
        break;
    case 2 :
        fraction = fabs( i2.x - i1.x ) * fabs( i2.y - i1.y ) / 2.0;
        maxxl = MAX( fabs( i1.x - corner[0].x ),
                     fabs( i1.x - corner[1].x ) );
        maxyl = MAX( fabs( i1.y - corner[0].y ),
                     fabs( i1.y - corner[1].y ) );

        break;
}

else
{
    // Handle the case where numcorners is greater than 2
    // Calculate the area accordingly
}

/* Other code here */

*/
maxx2 = MAX( fabs( i2.x - corner[0].x ),
        fabs( i2.x - corner[1].x ) );
maxy2 = MAX( fabs( i2.y - corner[0].y ),
        fabs( i2.y - corner[1].y ) );
fraction += MIN( maxx1, maxx2 ) * MIN( maxy1, maxy2 );
break;
case 3 :
    fraction = area( rect );
    fraction -= fabs( i2.x - i1.x ) * fabs( i2.y - i1.y ) / 2.0;
    break;
case 4 :
    fraction = area( rect );
    break;
};

fraction /= area( rect );

return( fraction );

/**/
/*****************************************************************************/
/*
*/
/* subtend
*/
/*
*/
/* This function calculates the
*/
/* fraction of area of a box that
*/
/* is contained between 2 lines at
*/
/* angles philow and phihigh.
*/
/* This function does not handle
*/
/* philow in quadrant IV and
*/
/* phihigh in quadrant I.
*/
/*
*/
/*****************************************************************************/
/**/

static double subtend( struct box bin, double philow, double phihigh )
{
    int i;
    double result;
    double phi[2];
    double thearea[2];
    struct box smallbin[2];

    phi[0] = philow;
    phi[1] = phihigh;

    /* There are 2 tricky cases. The first is if the origin is in the bin under consideration. So, first we check to see if this is the case and take care of it by breaking the bin into smaller ones. */
    if ( ( bin.left < 0.0 ) && ( bin.right > 0.0 ) &&
        ( bin.bottom < 0.0 ) && ( bin.top > 0.0 ) )
    {
        for( i = 0 ; i < 2 ; i++ )
        {


```c
{
    smallbin[i] = bin;
    if ( phi[i] < pi() / 2.0 )
    {
        thearea[i] = 0.0;
        smallbin[i].bottom = smallbin[i].left = 0.0;
    }
    else if ( phi[i] < pi() )
    {
        smallbin[i].bottom = smallbin[i].left = 0.0;
        thearea[i] = area( smallbin[i] );
        smallbin[i] = bin;
        smallbin[i].bottom = smallbin[i].right = 0.0;
    }
    else if ( phi[i] < 3.0 * pi() / 2.0 )
    {
        smallbin[i].bottom = 0.0;
        thearea[i] = area( smallbin[i] );
        smallbin[i] = bin;
        smallbin[i].top = smallbin[i].right = 0.0;
    }
    else
    {
        thearea[i] = area( smallbin[i] );
        smallbin[i].top = smallbin[i].left = 0.0;
        thearea[i] -= area( smallbin[i] );
    }
};

/* Now calculate the fractional area subtended. */
thearea[0] += fractional_area( smallbin[0], phi[0] ) * 
    area( smallbin[0] );
thearea[1] += fractional_area( smallbin[1], phi[1] ) * 
    area( smallbin[1] );
result = thearea[1] - thearea[0];
result /= area( bin );
}

/* The other tricky case is if the bin straddles quadrants */
/* I & IV. If this is the case, take care of it by breaking */
/* the bin into 2 smaller ones. */
else if ( bin.left >= 0.0 && bin.bottom < 0.0 && bin.top > 0.0 )
{
    smallbin[0] = smallbin[1] = bin;
    smallbin[0].top = smallbin[1].bottom = 0.0;

    result = fractional_area( smallbin[0], phi[1] ) * 
        area( smallbin[0] );
    result -= fractional_area( smallbin[0], phi[0] ) * 
        area( smallbin[0] );
    result += fractional_area( smallbin[1], phi[1] ) * 
        area( smallbin[1] );
    result -= fractional_area( smallbin[1], phi[0] ) * 
        area( smallbin[1] );
```
result /= area( bin );
}

/* If the origin is not contained in the bin, the result is */
/* straightforward to calculate. */
else
{
    result = fractional_area( bin, phi[1] );
    result -= fractional_area( bin, phi[0] );
};

return( result );

/*****************************************************************************/
/* AngularDist */
/* Determines the angular */
/* distribution on the PSD */
/* array. */
/*****************************************************************************/

double *AngularDist( double PSD[][k_PSDsize], double xcenter,
                     double ycenter )
{
    int xindex; /* PSD array index */
    int yindex; /* PSD array index */
    int index; /* Array index */
    int indexlow; /* Low angle index for PSD element */
    int indexhigh; /* High angle index for PSD element */
    double x; /* X-displacement */
    double y; /* Y-displacement */
    double sum = 0.0; /* Normalization factor */
    double *dist; /* Array containing angular */
                 /* distribution */
    double thetalow; /* Low angle of angular bin */
    double thetahigh; /* High angle of angular bin */
    double theta1; /* theta 1-4 are angles to corners */
    double theta2; /* of PSD element */
    double theta3;
    double theta4;
    double dtheta; /* Size of angular bin */
    double f; /* Fraction of area within angular bin */
    struct box psdbin; /* PSD element boundaries */

    /* Allocate the memory for the distribution. */
    /* If memory allocation fails, return a */
    /* null pointer. */
    dist = malloc( (size_t) k_pieslices * (size_t) sizeof( double ) );
    if ( dist == (double *) NULL )
return( dist );

for( index = 0 ; index < k_pieslices ; index++ )
{
    dist[index] = 0.0;
}

/* Sort through PSD array and calculate angular distribution. */
dtheta = 2.0 * pi() / (double) k_pieslices;

for ( xindex = 0; xindex < k_PSDsize; xindex++ )
{
    x = (double) xindex - xcenter;
    for ( yindex = 0; yindex < k_PSDsize; yindex++ )
    {
        y = (double) yindex - ycenter;

        /* Setup PSD bin */
        psdbin.left = x;
        psdbin.bottom = y;
        psdbin.right = x + 1.0;
        psdbin.top = y + 1.0;

        /* Calculate angles to each PSD bin corner */
        theta1 = myatan2( psdbin.bottom, psdbin.left );
        theta2 = myatan2( psdbin.top, psdbin.left );
        theta3 = myatan2( psdbin.bottom, psdbin.right );
        theta4 = myatan2( psdbin.top, psdbin.right );

        /* If the PSD bin is on the x-axis, correct angles */
        if ( psdbin.top == 0.0 )
        {
            if ( psdbin.left >= 0.0 )
            {
                theta2 = theta4 = 2.0 * pi();
            }
            else if ( psdbin.right == 0.0 )
            {
                theta2 = theta4 = pi();
            }
            else if ( psdbin.right > 0.0 )
            {
                theta4 = 2.0 * pi();
            }
        }

        if ( ( psdbin.bottom == 0.0 ) && ( psdbin.right == 0.0 ) )
        {
            theta1 = theta3 = pi();
        }

        /* Find minimum and maximum angles and calculate */
        /* index bounds */
thetalow = MIN( MIN( theta1, theta2 ),
             MIN( theta3, theta4 ) );

thetahigh = MAX( MAX( theta1, theta2 ),
               MAX( theta3, theta4 ) );

indexlow = (int)( thetalow / dtheta );
indexhigh = (int)( thetahigh / dtheta ) + 1;

if ( indexhigh >= k_pieslices )
{
    indexlow = 0;
    indexhigh = k_pieslices - 1;
};

/* If the bin contains the origin or straddles quadrants I & IV the above procedure does not work, so for these cases use the brute force approach and calculate all indices. */

if ( ( psdbin.top > 0.0 ) && ( psdbin.bottom < 0.0 ) &&
    ( psdbin.right > 0.0 ) )
{
    indexlow = 0;
    indexhigh = k_pieslices - 1;
};

/* Go through all angular bins and increment contents accordingly. */

for( index = indexlow ; index <= indexhigh ; index++ )
{
    /* Calculate low and high angles of the angular bin */
    /* The first bin straddles QI & QIV, which needs to be handled by splitting it into 2 bins. */
    /*
    if ( index == 0 )
    {
        thetalow += 2.0 * pi();
        f = subtend( psdbin, 0.0, thetahigh );
        f += 1.0 - subtend( psdbin, 0.0, thetalow );
    }
    else
    {
        f = subtend( psdbin, thetalow, thetahigh );
    }
    */
    f = subtend( psdbin, thetalow, thetahigh );
    /* Increment angular bin and normalization sum. */
    dist[index] += f * PSD[xindex][yindex];
    sum += f * PSD[xindex][yindex];
};
});
});

/* Normalize distribution. */
for (index = 0; index < k_pieslices; index++)
    dist[index] /= sum;

return (dist);
References

28. R. Adler, IEEE spectrum 4, 42 (May 1967)
35. For a brief discussion of Monte Carlo simulation, see P. R. Bevington and D. K. Robinson, Data Reduction and Error Analysis for the Physical Sciences 2d ed. (McGraw-Hill, New York, 1992) Chap. 5
36. See, for example, F. Reif, Fundamentals of statistical and thermal physics (McGraw-Hill, New York, 1965) Chap. 7, pp. 265-269


45. C. W. Walter, Ph.D. Thesis, Rice University, 1989


