Examining the Vibrational Couplings and Dynamics in Cyanide-Bridged Transition Metal Mixed Valence Complexes Using Ultrafast Nonlinear Infrared Spectroscopy

Karla M. Slenkamp

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

University of Washington

2015

Reading Committee:
Munira Khalil, Chair
Charles T. Campbell
Xiaosong Li

Program Authorized to Offer Degree:
Chemistry
University of Washington

Abstract

Examining the Vibrational Couplings and Dynamics in Cyanide-Bridged Transition Metal Mixed Valence Complexes Using Ultrafast Nonlinear Infrared Spectroscopy

Karla M. Slenkamp

Charge transfer processes are essential to any number of functions that are necessary for everyday life. Photosynthesis, respiration, and obtaining energy from solar cells are just a small sample of functions that rely on moving electrons around. Cyanide-bridged mixed valence complexes serve as useful models to study the coupling between electronic and nuclear motions during charge transfer reactions by using the cyanide stretching ($\nu_{\text{CN}}$) vibrations as probes. However, little is known about the $\nu_{\text{CN}}$ modes for these classes of molecules on the electronic ground state. Polarization-selective two-dimensional infrared (2D IR) spectroscopy was used to measure anharmonic couplings and angles between the transition dipole moments of the four $\nu_{\text{CN}}$ vibrations found in $[(\text{NH}_3)_5\text{Ru}^{\text{III}}\text{NCFe}^{\text{II}}(\text{CN})_5]^-$ (FeRu) and $[(\text{NC})_5\text{Fe}^{\text{II}}\text{CNPt}^{\text{IV}}(\text{NH}_3)_4\text{NCFe}^{\text{II}}(\text{CN})_5]^4^-$ (FePtFe) by fitting the experimental spectra. For this
study, FeRu was dissolved in both formamide and D$_2$O, and FePtFe was dissolved in D$_2$O. IR pump-probe and 2D IR spectroscopies were also used to study vibrational relaxation of the $v_{CN}$ modes in both complexes. The experimental 2D IR spectra of FeRu and FePtFe and their fits reveal a set of weakly coupled anharmonic $v_{CN}$ modes with the vibrational mode anharmonicities ranging from 14 to 28 cm$^{-1}$ and the mixed-mode anharmonicities ranging from 2 to 14 cm$^{-1}$. Measurement of the relative transition dipole moments of the four $v_{CN}$ modes reveal that FeRu appears almost linear in solution when dissolved in formamide, but bent when dissolved in D$_2$O. Fits of FePtFe dissolved in D$_2$O reveal that the molecule also appears bent. IR pump-probe experiments reveal that the vibrational lifetime of the radial, axial, and trans $v_{CN}$ modes are ~2 times faster when FeRu is dissolved in D$_2$O versus formamide, and the lifetimes of the $v_{CN}$ modes of FePtFe are ~4 times faster than FeRu when both are dissolved in D$_2$O. Finally, the synthesis and basic characterization of new cyanide containing Ru$^{II}$ polypyridine complexes are discussed. These complexes contain a metal-to-ligand charge transfer that relaxes into a longer lived triplet state. This longer lifetime will allow for a more in-depth analysis of vibrational modes on an electronic excited state as well as simplifying the analysis by reducing the number of $v_{CN}$ modes from the four seen in FeRu and FePtFe to one or two in the Ru polypyridine complexes.
# TABLE OF CONTENTS

List of Figures ........................................................................................................................................ vii

List of Tables .......................................................................................................................................... xi

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

1.1 Importance of Charge Transfer Reactions ...................................................................................... 1

1.2 Introduction to FeRu and FePtFe .................................................................................................... 3

1.3 Ultrafast Nonlinear Vibrational Spectroscopy .............................................................................. 8

1.4 Dissertation Outline ....................................................................................................................... 14

References ............................................................................................................................................. 15

Chapter 2 : Investigating Vibrational Anharmonic Couplings In Cyanide-Bridged Transition Metal
Mixed Valence Complexes Using Two-Dimensional Infrared Spectroscopy ........................................ 19

2.1 Introduction .................................................................................................................................... 20

2.2 Methods ....................................................................................................................................... 23

2.3 Results and Discussion .................................................................................................................. 29

   2.3.1 FTIR Spectra and Assignment of the $v_{cn}$ Modes ................................................................. 29

   2.3.2 2D IR Spectra of the $v_{CN}$ Modes in Mixed Valence Transition Metal
       Complexes ......................................................................................................................................... 32

   2.3.3 Extracting Anharmonic Couplings in the Site Basis Representation ...................................... 45

2.4 Summary ....................................................................................................................................... 47
Chapter 3: Investigating Vibrational Relaxation in Cyanide-Bridged Transition Metal Mixed Valence Complexes Using Two-Dimensional Infrared and Infrared Pump-Probe Spectroscopies

3.1 Introduction.................................................................................................................54
3.2 Methods.........................................................................................................................57
3.3 Results..........................................................................................................................59
  3.3.1 IR Pump-Probe .........................................................................................................60
  3.3.2 2D IR Relaxation Experiment ................................................................................65
3.4 Discussion......................................................................................................................72
  3.4.1 Vibrational Relaxation Comparison .........................................................................73
  3.4.2 Spectral Diffusion Comparison ................................................................................75
3.5 Conclusion.....................................................................................................................76
3.6 Additional 2D IR Spectra for FeRu ............................................................................77
References .........................................................................................................................81

Chapter 4: Synthesis and Characterization of Ruthenium (II) Polypyridine Cyanide Complexes

4.1 Introduction.....................................................................................................................84
4.2 Methods.........................................................................................................................85
  4.2.1 Spectra Collection ....................................................................................................85
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.2</td>
<td>Synthesis</td>
<td>86</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Sample Preparation</td>
<td>90</td>
</tr>
<tr>
<td>4.3</td>
<td>Compound Verification</td>
<td>91</td>
</tr>
<tr>
<td>4.3.1</td>
<td>$^1$H NMR Spectra</td>
<td>91</td>
</tr>
<tr>
<td>4.3.2</td>
<td>FTIR Spectra</td>
<td>93</td>
</tr>
<tr>
<td>4.3.3</td>
<td>UV/vis Spectra</td>
<td>95</td>
</tr>
<tr>
<td>4.4</td>
<td>Solvent Study</td>
<td>96</td>
</tr>
<tr>
<td>4.5</td>
<td>Conclusions and Future Work</td>
<td>98</td>
</tr>
<tr>
<td></td>
<td>References</td>
<td>99</td>
</tr>
<tr>
<td>Appendix A</td>
<td>Matlab Code for the Simulation of 2D IR Spectra</td>
<td>101</td>
</tr>
<tr>
<td>A.1</td>
<td>Six Level System with Homogeneous Lineshapes</td>
<td>101</td>
</tr>
<tr>
<td>A.1.1</td>
<td>sixLevelHomLimPolarComp.m</td>
<td>101</td>
</tr>
<tr>
<td>A.1.2</td>
<td>gaaHL.m</td>
<td>109</td>
</tr>
<tr>
<td>A.1.3</td>
<td>ga2aHL.m</td>
<td>109</td>
</tr>
<tr>
<td>A.1.4</td>
<td>g2a2aHL.m</td>
<td>109</td>
</tr>
<tr>
<td>A.1.5</td>
<td>gabHL.m</td>
<td>110</td>
</tr>
<tr>
<td>A.1.6</td>
<td>gaabHL.m</td>
<td>110</td>
</tr>
<tr>
<td>A.1.7</td>
<td>gababHL.m</td>
<td>111</td>
</tr>
<tr>
<td>A.2</td>
<td>Six Level System with Population Relaxation</td>
<td>111</td>
</tr>
<tr>
<td>A.2.1</td>
<td>sixLevel_poprlx.m</td>
<td>111</td>
</tr>
</tbody>
</table>
A.2.2  gaa.m .................................................................124
A.2.3  ga2a.m .................................................................125
A.2.4  g2a2a.m .................................................................125
A.2.5  gab.m .................................................................125
A.2.6  gaab.m .................................................................126
A.2.7  gabab.m .................................................................126
A.3  Fifteen Level System in the Homogeneous Limit .................................. 127
      A.3.1  fifteenLevelHomLim.m ........................................ 127
      A.3.2  Lineshape Functions ........................................... 136
A.4  Fitting 2D IR Simulations to Experimental Spectra .................................. 136
      A.4.1  fit_2D_FePtFe_angles_relative.m ................................ 137
      A.4.2  angle_const.m .................................................. 148
      A.4.3  fifteenLevelFit_angles_relative.m .............................. 149
      A.4.4  gttL.m ........................................................... 164
      A.4.5  gt2tL.m ........................................................... 164
      A.4.6  g2t2tL.m ........................................................... 165
      A.4.7  gtrL.m ........................................................... 166
      A.4.8  gttrL.m ........................................................... 167
      A.4.9  gttrtrL.m ........................................................... 167
A.5  specpack3 ............................................................. 168
Appendix B: Matlab Code for Determining Coupling Constants .................. 230

B.1 Main File – First Guesses for Local Site Basis Parameters ............... 230

B.2 Calculating Difference Between the Experimental and Fit Values .......... 235

B.3 Eigenstates of the Local Site Basis Hamiltonian .............................. 236

Appendix C: Matlab Code for Analyzing t-HDVE Raw Data .................... 240

C.1 Loading 3rd Order Data Files into Single Variable .......................... 240
C.2 Loading 5th Order Data Files into Single Variable .................................................. 245

C.3 Treating Raw t-HDVE Data....................................................................................... 249
LIST OF FIGURES

Figure 1.1: The molecular structures of (a) FeRu and (b) FePtFe. Colored arrows correspond to simplified depictions of four ν_{CN} modes. The transition dipole moments of the ν_{trans} (ν_{t}, blue), ν_{axial} (ν_{a}, green), and ν_{bridge} (ν_{b}, purple) modes lie along the MMCT axis and are perpendicular to the degenerate ν_{radial} (ν_{r}, red) mode. ................................................................. 3

Figure 1.2: Potential energy surface showing MMCT and BET in FeRu (a) and FePtFe (b) ...... 4

Figure 1.3: FTIR of FeRu in FA (a), FeRu in D_{2}O (b), and FePtFe in D_{2}O (c) ....................... 6

Figure 1.4: An example of an energy level diagram for a system of two coupled vibrations, modes p and q. Diagonal and mixed-mode anharmonicities are labeled................................. 9

Figure 1.5: Simulated FTIR and 2D IR spectra for a system of two coupled vibrations, ν_{p} and ν_{q}, with homogeneous lineshapes. Diagonal and mixed-mode anharmonicities are labeled ...... 10

Figure 1.6: A series of simulated 2D IR spectra for two coupled vibrations at increasing τ_{2} delays ........................................................................................................................................ 11

Figure 1.7: A series of IR pump-probe spectra for several τ_{2} points for a system with two modes .................................................................................................................................... 12

Figure 1.8: Simulated 2D IR spectra for coupled vibrations with two different relative angles between the transition dipole moments of ν_{p} and ν_{q}.................................................. 13

Figure 1.9: Simulated 2D IR spectrum for a system of four coupled vibrations ...................... 14

Figure 2.1: The molecular structures of (a) FeRu and (b) FePtFe. Colored arrows correspond to simplified depictions of four ν_{CN} modes .................................................................................................. 21

Figure 2.2: Cartoons representing the normal modes for the ν_{CN} vibrations ν_{trans}, ν_{radial}, ν_{axial}, and ν_{bridge} ......................................................................................................................... 22

Figure 2.3: FTIR spectra before being solvent subtracted.......................................................... 24
Figure 2.4: Experimental FTIR spectra (solid black) and spectra simulated (dashed green) using the lineshape parameters from our 2D fits presented in the captions for Figures 2.7, 2.8, and 2.10. The contribution to the simulated spectra from each mode is also shown: \( \nu_t \) (blue), \( \nu_r \) (red), \( \nu_a \) (green), and \( \nu_b \) (purple).

Figure 2.5: Solvent subtracted FTIR spectra (black line) of (a) FeRu in FA (b) FeRu in D\(_2\)O and (c) FePtFe in D\(_2\)O.

Figure 2.6: A generalized diagram of all the relevant vibrational states probed in the 2D IR experiment for four coupled vibrations.

Figure 2.7: 2D IR spectra of FeRu dissolved in FA at \( \tau_2 = 140 \) fs. (a) ZZZZ experiment, (b) YYZZ experiment, (c) ZZZZ simulated, and (d) YYZZ simulated.

Figure 2.8: 2D IR spectra of FeRu dissolved in D\(_2\)O at \( \tau_2 = 150 \) fs. (a) ZZZZ experiment, (b) YYZZ experiment, (c) ZZZZ simulated, and (d) YYZZ simulated.

Figure 2.9: Slices of the YYZZ experimental 2D spectra (normalized) at the \( \omega_1 \) frequency for each of the four modes: \( \nu_t \) (blue), \( \nu_r \) (red), \( \nu_a \) (green), and \( \nu_b \) (purple).

Figure 2.10: 2D IR spectra of FePtFe dissolved in D\(_2\)O at \( \tau_2 = 150 \) fs. (a) ZZZZ experiment, (b) YYZZ experiment, (c) ZZZZ simulated, and (d) YYZZ simulated.

Figure 3.1: (a) FeRu and (b) FePtFe with their \( \nu_{CN} \) modes illustrated by colored arrows \( \nu_{\text{trans}} \) (\( \nu_t \), blue), \( \nu_{\text{radial}} \) (\( \nu_r \), red), \( \nu_{\text{axial}} \) (\( \nu_a \), green), and \( \nu_{\text{bridge}} \) (\( \nu_b \), purple). Solvent subtracted FTIR spectra (black line) of (c) FeRu in FA (d) FeRu in D\(_2\)O and (e) FePtFe in D\(_2\)O.

Figure 3.2: The two panels of (a) represent spectra (left) of FeRu in FA at several \( \tau_2 \) delays and time traces (right) of the same signal for select frequencies shown as dashed lines in the left panel. (b) and (c) display the same information for FeRu in D\(_2\)O and FePtFe in D\(_2\)O, respectively, as described in (a).
Figure 3.3: Crossed polarization 2D IR spectra at selected $\tau_2$ delays for (a) FeRu in FA, (b) FeRu in D$_2$O, and (c) FePtFe .......................................................... 66

Figure 3.4: Plots of the center line slope (CLS) measuring spectral diffusion as a function of $\tau_2$ for selected modes from each sample.............................................................................. 70

Figure 3.5: The summed signal intensity over a small area of each 2D IR spectrum centered on the peaks indicated by the $\omega_1$ and $\omega_3$ frequencies is shown for each trace for the FeRu/FA samples (left), FeRu/D$_2$O sample (middle), and FePtFe sample (right) as a function of $\tau_2$ delay. ..................................................................................................................... 72

Figure 3.6: Parallel polarization 2D IR spectra for FeRu dissolved in FA for all time points collected ..................................................................................................................... 78

Figure 3.7: Crossed polarization 2D IR spectra for FeRu dissolved in FA for all time points collected ..................................................................................................................... 79

Figure 3.8: Parallel polarization 2D IR spectra for FeRu dissolved in D$_2$O for all time points collected ..................................................................................................................... 80

Figure 3.9: Crossed polarization 2D IR spectra for FeRu dissolved in D$_2$O for all time points collected ..................................................................................................................... 81

Figure 4.1: (a) $^1$H NMR spectrum of [Ru(tpy)(bpy)Cl]$^+$. (b) $^1$H NMR spectrum of [Ru(tpy)(bpy)CN]$^+$. (c) $^1$H NMR spectrum of [{Ru(tpy)(bpy)}$_2$CN]$^{3+}$. (d) $^1$H NMR spectrum for [Ru{CNRu(tpy)(bpy)}$_2$]$^{4+}$. (e) Cartoon figure of monomer. (f) Cartoon figure of dimer. (g) Cartoon of trimer.............................................................................................................................. 92

Figure 4.2: FTIR spectrum of the monomer (a) and dimer (b) in methanol................................. 94

Figure 4.3: Testing stability of monomer (a) and dimer (b) in methanol over a period of three days............................................................................................................................. 94
**Figure 4.4:** UV/vis spectra of the monomer (left panel) and dimer (right panel) in several solvents

.................................................................................................................................................. 95

**Figure 4.5:** FTIR spectrum of monomer (a) and dimer (b) in several solvents where the solvent backgrounds have been subtracted. .................................................................................................................................. 96
## LIST OF TABLES

**Table 2.1:** Best fit parameters of the FTIR spectra of FeRu and FePtFe ........................................... 32

**Table 2.2:** Best fit values for vibrational mode anharmonicities, mixed-mode anharmonicities, and angles between modes for all three samples ......................................................................................................................... 34

**Table 2.3:** Best fit values for the local mode site energies, vibrational mode anharmonicities, and bilinear coupling constants for all three systems .......................................................................................................................... 46

**Table 3.1:** Parameters from the exponential fits to experimental pump-probe data .................. 59

**Table 3.2:** Exponential fit values to CLS traces.......................................................................................... 68

**Table 4.1:** Best fit parameters for the frequency and width of FTIR peaks for the dimer in different solvents ........................................................................................................................................................................ 97

**Table 4.2:** Best fit parameters for the frequency and width of FTIR peaks for the monomer in different solvents ........................................................................................................................................................................ 98
ACKNOWLEDGEMENTS

So many people have supported me during my almost six years at the University of Washington. The work that made it into this dissertation, and all the work that didn’t, would not have been possible without them.

First I have to thank all of the members of the Khalil research group starting with Munira Khalil. I would like to thank her for allowing me to join her group and introducing me to the world of ultrafast spectroscopy. It was an intimidating world to walk into, but an exciting one as well. She always required excellence, and because of that, I can be proud of the work I completed in her group and the work I present in this dissertation.

I need to thank Jennifer Brookes and Anthony Reynolds. We joined the Khalil group together, and I’m not sure I would have made it to the end without them. They were always there when I had a ‘stupid’ question, and they never judged me…well not much anyway. Over the past six years, they have become members of my family, ones that I have grown with, ones that I have laughed, cried, and overcome difficulties with. They have made graduate school a more rewarding and enriching experience.

I also need to thank former group member Michael Lynch for sharing his experimental knowledge with me. We may not have gotten along in lab at all times, but Mike was still a great friend, and I learned so much from him. Ben Van Kuiken shared theoretical knowledge. Deciphering his answers could be a challenge at times, but he was always willing to hear my questions and answer more than the question I had asked. Two wonderful undergraduates, Laura Estergreen and Caitlin Bannan, synthesized most of the compounds I studied at UW. Thank you
to all other members of the Khalil group during my time at the University of Washington for being friends and valuable sources of information.

Outside of my group, there were numerous people within the chemistry department that helped me survive the trials of graduate school. Thank you to Glennis Rayermann and Joseph May for numerous Monday night nacho sessions. Thanks to Trevor James for being my bus buddy before we even officially met the very first day of orientation in September 2009. He made our longer than normal commutes more interesting. And thank you to other members of my incoming year for their support including Tom Porter, Michael Roberto, Sam Marionni, Sam Connelly, Megan Duda, Emilie Viglino, and Nathan Sylvain.

Any acknowledgements section would be incomplete without thanking the people in my life that had nothing to do with the University of Washington. I have to thank (and blame) my friend Sam Backus. He convinced me to apply to graduate school, and he has been listening to me complain ever since. He has heard every problem I’ve encountered the last six years, and I truly do owe my continuing sanity to him. I have to thank Kelsey Roe for also listening to me complain. And I have to thank Camille and Josh Lider, Bonnie McAllister, Nena Slocum, Bridgette Hannifin, Avery Stewart, and Rachel Unterreiner for reminding me the world at large doesn’t revolve around chemistry, even if mine has for almost 10 years now.

Last but not least, I have to thank my family. I have a large extended family, and thanking them individually here would take more pages than anyone would be willing to read. But they have all provided me with support and love and an escape from academic life. In particular, I need to thank my grandparents for emotional support. I need to thank my Grandma for sitting down and being actively engaged whenever I was discussing my research. I need to thank my sister, Shelby, for making me food and occasionally helping me clean my home when life got too hectic, even if she
grumbled the whole time. And finally, I need to thank my parents, Robert and Beverly Slenkamp. They have been amazing! I don’t have enough words to express my gratitude for all the ways they have supported me in reaching my goals, both emotionally and materially. I couldn’t have done it without you. I love you!
DEDICATION

To my family…

every last one of you!
Chapter 1:

Introduction

1.1 Importance of Charge Transfer Reactions

Charge transfer processes are essential to any number of functions that are necessary for everyday life. Photosynthesis, respiration, and obtaining energy from solar cells are just a small sample of functions that rely on moving electrons around. However, directly studying a system like photosynthesis is difficult because of its complexity; there is more to photosynthesis than moving one electron around. Instead, mixed valence transition metal complexes are much simpler systems, but they still exhibit these electron transfer processes. More famous examples of mixed valence compounds include Prussian Blue, used as a deep blue dye, and the Creutz-Taube ion, used primarily to study inner sphere electron transfer. Electron transfer can be initiated by photons with sufficient energy to reach the charge transfer bands. Cyanide-bridged mixed valence transition metal complexes in particular are a class of molecules that exhibit unique redox, spectroscopic, and charge transfer properties and are widely studied for applications in solar energy conversion, molecular electronics, or photocatalysis. Choosing cyanide as a bridging ligand between metal centers moves the metal-to-ligand charge transfer (MLCT) bands to higher energy keeping them from obscuring the metal-to-metal charge transfer (MMCT) bands in the near-infrared to visible region of the spectrum. Cyanide-bridged complexes also serve as useful models to study the coupling between electronic and nuclear motions during charge transfer reactions. Photochemical reactions in solution at room temperature can occur in less than 100
millionth-of-a-billionth-of-a-second, or in less than 100 femtoseconds ($10^{-15}$ s). This is on the order of molecular vibrations in solution which have frequencies in the range of $50 – 3500$ cm$^{-1}$ ($1.5 – 105$ THz) corresponding to periods of $10 – 680$ fs. Vibrational relaxation also occurs on a femtosecond to picosecond ($10^{-12}$ s) timescale. Many high-frequency vibrations ($> 1500$ cm$^{-1}$), cyanide included, can provide spatial information because the vibrations are relatively localized within one part of the molecule. This makes vibrations an ideal tool to study photoinduced charge transfer reactions.

The role high frequency vibrations play in electron transfer processes really started to interest groups in the 1980s.$^{32-36}$ Cyanide-bridged transition metal mixed valence dimer (two metal centers) systems of the form $[(\text{NH}_3)_5M^{\text{III}}\text{NCM}^{\text{II}}(\text{CN})_5]^-$, where $M = \text{Fe}, \text{Os}$ and $\text{Ru}$, have been widely studied since the late 1980s to follow the timescales of electron transfer kinetics, coherent wavepacket motion on the excited/ground electronic state, and vibrational excitation/relaxation of the cyanide stretching modes ($\nu_{\text{CN}}$) following MMCT excitation.$^{15,16,18,20-22,24,25,36-40}$ Transition metal mixed valence trimer systems like $[(\text{NC})_5M^{\text{II}}\text{CNPt}^{\text{IV}}(\text{NH}_3)_4\text{NCM}^{\text{II}}(\text{CN})_5]^+$, where $M = \text{Fe}$ or $\text{Ru}$, have been studied for similar purposes since the early 1990s.$^{11,26,41,42}$ The IR spectroscopy of the bridging cyanide ligand and its ability to probe the electron delocalization and coupling between the metal centers in mixed valence complexes is of great interest. Detailing how inter- and intramolecular vibrational energy flow in ultrafast photoinduced electron transfer processes relates to molecular conformational changes and the efficacy of the charge transfer process is critical for understanding basic and applied phenomena in chemistry, biology, and materials science. Using these relatively simple cyanide-bridged transition metal complexes to study the interplay between electronic transitions and vibrational motions allows for a better understanding
of the fundamental process of charge transfer without becoming overwhelmed by all that might be happening in a larger molecule like a protein.

1.2 Introduction to FeRu and FePtFe

Two commonly studied cyanide-bridged mixed valence dimer and trimer systems are \([\text{FeRu}] = [(\text{NH}_3)_5\text{Ru}^{III}\text{Fe}^{II}(\text{CN})_5]^-\), or FeRu, and \([\text{FePtFe}] = [(\text{NC})_5\text{Fe}^{II}\text{CNPt}^{IV}(\text{NH}_3)_4\text{NCFe}^{II}(\text{CN})_5]^{4-}\), or FePtFe. Both molecules are shown as cartoons in Figure 1.1. The FeRu molecule undergoes a photoinduced MMCT, transferring an electron from the iron atom to the ruthenium atom, when in solution and excited by ~1000 nm light.\(^{15,16}\)

\[
[(\text{NH}_3)_5\text{Ru}^{III}\text{NCFe}^{II}(\text{CN})_5]^- \xrightarrow{hv} [(\text{NH}_3)_5\text{Ru}^{II}\text{NCFe}^{III}(\text{CN})_5]^-
\]

This is followed almost immediately by back electron transfer (BET), which shifts the electron density back to the iron atom within ~100 fs. When photoexcited at 400 nm, FePtFe also undergoes MMCT moving an electron from one of the iron centers to the platinum atom, at which point the

---

**Figure 1.1:** The molecular structures of (a) FeRu and (b) FePtFe. Colored arrows correspond to simplified depictions of four \(v_{\text{CN}}\) modes. The transition dipole moments of the \(v_{\text{trans}}\) (\(v_t\), blue), \(v_{\text{axial}}\) (\(v_a\), green), and \(v_{\text{bridge}}\) (\(v_b\), purple) modes lie along the MMCT axis and are perpendicular to the degenerate \(v_{\text{radial}}\) (\(v_r\), red) mode.
electron is transferred back to the iron atom in \( \sim 100 \) fs with \( \sim 99\% \) efficiency to form a “hot” electronic ground state.

\[
[(\text{NC})_3\text{Fe}^{\text{II}}\text{CNPt}^{\text{IV}}(\text{NH}_3)_4\text{NCFe}^{\text{II}}(\text{CN})_5]^4- \xrightarrow{h\nu} [(\text{NC})_3\text{Fe}^{\text{III}}\text{CNPt}^{\text{III}}(\text{NH}_3)_4\text{NCFe}^{\text{II}}(\text{CN})_5]^4-
\]

The other \( \sim 1\% \) of the time, a second electron from the other iron atom is transferred to the platinum yielding \( \text{Fe}^{\text{III}}\text{Pt}^{\text{II}}\text{Fe}^{\text{III}} \). This product immediately decomposes into \( 2[\text{Fe}^{\text{III}}(\text{CN})_6]^3- \) and \( [\text{Pt}^{\text{II}}(\text{NH}_3)_4]^2+ \).\(^{11,41}\) Spectral electrochemistry has been used to study the nature of the MMCT bands in both the dimer and trimer molecules.\(^{11,21,26,40,43-49}\) This work depicted a simplified description of the energetics of the dimer and trimer molecules, where the potential energy is given by a two- or three-parabola potential energy surface in the Marcus-Hush picture, pictured in Figure 1.2 for both molecules. Resonance Raman data and time-dependent intensity analysis by several research

![Figure 1.2: Potential energy surface showing MMCT and BET in FeRu (a) and FePtFe (b). Timescales for FeRu dissolved in D\(_2\)O (FA).\(^{15,16}\) Figure adapted from Pfennig and Bocarsly.\(^{41}\)](image-url)
groups have shown that >40% of the inner sphere reorganization energy involves vibrational modes which lie along the intervalence charge transfer axis of the complex.\textsuperscript{26,41,42,49,50} Static and time-resolved experiments have been analyzed to show that the majority of the reorganization energy arises from solvent effects in transition metal mixed valence complexes.\textsuperscript{15,16,18} FeRu and FePtFe both have four $\nu_{\text{CN}}$ frequencies that are labeled $\nu_{\text{trans}}$ ($\nu_t$), $\nu_{\text{radial}}$ ($\nu_r$), $\nu_{\text{axial}}$ ($\nu_a$), and $\nu_{\text{bridge}}$ ($\nu_b$) (illustrated in Fig. 1.1a,b) of which three ($\nu_t$, $\nu_a$, and $\nu_b$) lie approximately along the MMCT axis.\textsuperscript{28,51}

Starting in the late 1980s, Barbara, Hupp, Walker, and Woodruff began studying mixed valence complexes using vibrational spectroscopy. Barbara and Hupp focused on Raman spectroscopy using it to back out parameters relevant to the electron transfer such as classical reorganization energy and electronic coupling.\textsuperscript{16,37} By studying FeRu with picosecond time-resolved infrared spectroscopy, Walker and coworkers found excitation in $\nu_{\text{CN}}$ modes after BET in when the molecule was dissolved in both formamide (FA) and deuterated water (D$_2$O).\textsuperscript{20} Three $\nu_{\text{CN}}$ modes were assigned which they called $\nu_{\text{bridge-CN}}$, $\nu_{\text{cis-CN}}$, and $\nu_{\text{trans-CN}}$. Walker and company’s resonance Raman work on FeRu showed that the solvent controls the degree of vibronic coupling for the trans $\nu_{\text{CN}}$ mode in the MMCT transition.\textsuperscript{21} They have also concluded that solvent induced changes in the structure of the molecule may result from the strong interaction between the solvent and $\nu_{\text{trans-CN}}$. In the similar $[(\text{NH}_3)_5\text{Ru}^{\text{III}}\text{NCOs}^{\text{II}}(\text{CN})_5]^-$ system, they found that vibrational levels $n = 0$, $n = 1$, and $n = 2$ of cis and trans CN were populated after BET.\textsuperscript{22} It has also been observed by Woodruff and coworkers that after BET in $[(\text{NH}_3)_3\text{Ru}^{\text{III}}\text{NCRu}^{\text{II}}(\text{CN})_5]^-$ there is population in $\nu_{\text{CN}}$ modes in vibrationally excited states $n = 17$.\textsuperscript{24} Around the same time, Endicott and coworkers studied a series of cyanide-bridged transition metal complexes. They found a simple vibronic coupling model could be used to understand the frequencies of the $\nu_{\text{CN}}$ within the complexes.\textsuperscript{29-31}
As mentioned above Bocarsly and co-workers first synthesized a transition metal mixed valence trimer system of the form, \([\text{NC}_5\text{M}^{\text{II}}\text{CNPt}^{\text{IV}}(\text{NH}_3)_4\text{NCM}^{\text{II}}(\text{CN})_5]^4\), where \(M = \text{Fe} \) or \(\text{Ru}\) in 1990, and showed that it undergoes an ultrafast multielectron transfer process following MMCT excitation.\(^{11,41}\) The multielectron transfer process discussed above made the trimer a potentially more interesting molecule than FeRu, but the small percentage that undergoes the second MMCT makes it very difficult to study that additional aspect.

Our group has obtained X-ray absorption spectroscopy results at the Ru L3-edge of FeRu.\(^ {52}\) These results highlighted the important role hydrogen-bonding plays in the solvent modulation of charge around transition metal ions in solution. We have also investigated the role of the \(\nu_{\text{CN}}\) modes in the MMCT excitation, and subsequent BET, reaction of FePtFe dissolved in D\(_2\)O.\(^ {27,28,53}\)

![Figure 1.3: FTIR of FeRu in FA (a), FeRu in D\(_2\)O (b), and FePtFe in D\(_2\)O (c). Colors correspond to the colors of the modes seen in Figure 1.1.](image-url)
The four \( \nu_{\text{CN}} \) modes (see Figure 1.1b) are a multidimensional probe that can be used to study vibrational energy relaxation following the MMCT transition. Using sub-100 fs transient infrared spectroscopy, we found that BET to the ground state occurs in \( \sim \)100 fs and it deposits greater than 6 quanta of vibrational energy into \( \nu_b \).\(^{28}\) Using transient heterodyne-detected dispersed vibrational echo (t-HDVE), a fifth order nonlinear visible-infrared spectroscopy, the dynamics of the \( \nu_{\text{CN}} \) modes can be monitored as a function of time after visible excitation (\( \tau_{\text{vis}} \)).\(^{53}\) In other words, the vibrational dynamics can be monitored as a photochemical reaction occurs.\(^{27}\) This technique provided experimental evidence that \( \nu_{\text{CN}} \) modes are both coherently and incoherently coupled to the charge transfer reaction. Vibrational relaxation timescales change as \( \tau_{\text{vis}} \) increases moving the system back towards equilibrium, and a change in the frequency of a \( \sim 57 \text{ cm}^{-1} \) oscillation indicates that the coupling between the probed \( \nu_{\text{CN}} \) modes changes \( \tau_{\text{vis}} < 500 \text{ fs} \). The Fourier transform infrared (FTIR) spectra of the \( \nu_{\text{CN}} \) modes (see Fig. 1.3a-c) of the bimetallic and trimetallic systems are strikingly different in terms of frequencies, amplitudes, and lineshapes. Why is this the case? What is the difference between the solvent or molecules that allows for this? Previous studies, both by our group and others, lack a detailed understanding of the electronic ground state in these systems.

These time-resolved and spectral studies have identified the importance of the solvent and vibrations in the MMCT process and measured the relevant rate constants. The goal of this dissertation is to understand how the solvent modulates the molecular geometry and the anharmonic couplings between the bridging and terminal cyanide stretching vibrations as well as the relaxation dynamics in cyanide-bridged bimetallic and trimetallic transition metal mixed valence complexes in the electronic ground state (without the MMCT). Molecular geometry and coupling affect how modes interact with each other as well as how modes interact with the rest of
their environment. Knowing the role these parameters play can have implications for the design of molecules or determining their spectroscopic properties. Vibrational relaxation dynamics reflect the differences in local environment and the interactions that occur between the solvent and the local vibrational probe. Changing dynamics between systems, or even different vibrational modes within a molecule, show differences in how energy is transferred from the vibrational mode to its environment. Ultrafast nonlinear infrared spectroscopy includes the most common techniques used to study vibrational coupling and relaxation in the condensed phase. We use infrared pump-probe and polarization-selective two-dimensional infrared spectroscopies to measure anharmonic couplings, angles between the transition dipole moments, and electronic ground state vibrational dynamics of the cyanide stretching vibrations of FeRu dissolved in D₂O and formamide (FA) and FePtFe dissolved in D₂O.

1.3 ULTRAFAST NONLINEAR VIBRATIONAL SPECTROSCOPY

One of the most comprehensive ways to gain information about a system’s vibrational potential energy surface is two-dimensional infrared spectroscopy (2D IR). Linear Fourier transform spectroscopy (FTIR) can give you some information like peak positions and relative amplitudes, but it is insensitive to coupling and distributions underneath the peak envelope. Two-dimensional infrared spectroscopy allows us to spread the spectra out in two dimensions (ω₁ and ω₃) and monitor how the spectra change as a function of time (τ₂). It allows for the measurement of vibrational anharmonicities, vibrational relaxation dynamics, vibrational frequency-frequency correlation functions (FFCF), and coupling between vibrational modes as well as determining how much of a peak’s width is due to homogeneous versus inhomogeneous broadening.

Figure 1.4 depicts an energy diagram for two coupled vibrational modes, νₚ and νₚ. Two coupled modes produce a six-level system consisting of a common ground state, two first-excited states,
two second-excited states (vibrational overtones), and a combination band that is a linear combination of both modes. There are three anharmonicities labeled in Figure 1.4 that relate the position of two-quantum states to the frequency of $v = 0 \rightarrow v = 1$ transition. The two that show how far from harmonic each mode is are called diagonal anharmonicities and are labeled $\Delta_{pp}$ and $\Delta_{qq}$. The last anharmonicity, $\Delta_{pq}$, is the mixed-mode anharmonicity, and it is a measure of how strongly modes $p$ and $q$ are coupled. The larger the value of $\Delta_{pq}$, the stronger the coupling.

Figure 1.5 shows a simulated 2D IR spectrum of a six-level system at a single $\tau_2$ point as well as the simulated FTIR spectrum. The positive (red) peaks represent transitions to one quantum states and the negative (blue) peaks represent transitions to two-quantum states. The separations between the positive and negative peaks in each pair reflect the anharmonicities of the system. In a perfectly harmonic system, the anharmonicities would be zero and the positive and negative peaks would cancel out. Peaks along the diagonal (where $\omega_1 = \omega_2$) represent the $v = 0 \rightarrow v = 1$ transition for each mode. The widths of the peaks along the diagonal tell us how inhomogeneous the environment for that mode is, while the anti-diagonal width tells us about homogeneous
broadening. In addition to the two sets of peaks along the diagonal, there are two additional sets of peaks seen in Figure 1.5 where $\omega_1 = \omega_p$ and $\omega_3 = \omega_q$, or vice versa. These are called crosspeaks, and their existence proves that the two modes are indeed coupled. While all of these parameters are useful to understand a system, if we want to learn about the system’s dynamics, we need to look at a series of spectra at multiple $\tau_2$ delays.

Figure 1.6 shows a series of 2D IR spectra as a function of $\tau_2$. Something in the system is changing as $\tau_2$ increases. The peaks “round-out”, the cross-peaks appear to grow and/or beat in relation to the diagonal peaks, and the overall amplitude of the spectra decrease. The peaks round out as a result of spectral diffusion, and the rate at which this happens is proportional to the FFCF for that mode. Frequency-frequency timescales measure how long it takes for the vibrational mode

![Diagram](image.png)

**Figure 1.5:** Simulated FTIR and 2D IR spectra for a system of two coupled vibrations, $v_p$ and $v_q$, with homogeneous lineshapes. Diagonal and mixed-mode anharmonicities are labeled. The 2D IR spectrum was generated with code found in Appendix A.
to sample all of its possible environments. Depending on the system being measured, FFCF timescales can be attributed to things like the breaking and forming of hydrogen bonds, or they can simply remark on the mode’s environment; maybe the environment for this mode is more polar than for another mode. Growth in the cross-peaks as $\tau_2$ increases measures population transfer between modes; energy starts out in one mode ($\omega_1 = \nu_p$) but ends up in another ($\omega_3 = \nu_q$). Following the rates of energy transfer gives important information about how energy is transferred within the molecule and the route it takes before escaping into the environment. The overall decrease in

**Figure 1.6:** A series of simulated 2D IR spectra (code in Appendix A) for two coupled vibrations at increasing $\tau_2$ delays. All spectra are plotted using the same contour levels and color scheme so amplitudes of peaks can be directly compared.
amplitude for all parts of the spectrum as $\tau_2$ increases is a result of population relaxation. As the modes relax back to their ground state, there is less and less signal to measure. Full 2D IR spectra take a significant amount of time to collect which limits the number of $\tau_2$ points obtained during each experiment. Because of this, population relaxation measurements are usually performed using another ultrafast technique, IR pump-probe. This technique collapses a 2D IR spectrum onto the $\omega_3$ axis, so all information can be collected in one dimension. An example of a series of IR pump-probe spectra for increasing $\tau_2$ delays are shown in Figure 1.7. It is easy to see the decay due to population relaxation. These are much faster to collect working in lab so more $\tau_2$ points can be measured giving a more detailed look at relaxation dynamics. All of the dynamics discussed in this paragraph are described in further detail in Chapter 3.

Finally, 2D IR can be used to calculate angles, $\theta_{pq}$, between the transition dipole moments of coupled modes. Figure 1.8 shows 2D IR spectra at two different polarizations, where the first and second pulse have polarizations either parallel (ZZZZ) or rotated 90° (YYZZ) to the third pulse.

Figure 1.7: A series of IR pump-probe spectra for several $\tau_2$ points for a system with two modes. Population relaxation dynamics can be measured by quantifying how fast each feature decays to zero. Negative going peaks represent $v = 0 \rightarrow v = 1$ transitions, while positive going peaks represent $v = 1 \rightarrow v = 2$ transitions.
and signal field, and for two different $\theta_{pq}$. In order to extract the angle, the ratio of the amplitudes between the same cross-peak in different polarizations must be compared. Because transition dipole moments at different orientations will react more strongly with light polarized in a similar orientation, the strength of the cross-peaks will change with different polarizations. This is seen in Figure 1.8, where the cross-peaks in the $ZZZZ$ geometry are much weaker when $\theta_{pq} = 90^\circ$ than when $\theta_{pq} = 0^\circ$.

All of the 2D IR examples presented above were theoretical spectra generated from a 2D IR simulation code written by our group. If the peaks are well-separated and highly anharmonic like...
the examples here, all of this information can be read directly from experimental spectra as well. But when modes become overlapped, or they are only slightly anharmonic, or there are a larger number of modes, it becomes difficult to extract useful information from the 2D IR spectra. When this is the case, it becomes necessary to fit the spectra, which is the reason we developed the simulation code in the first place. In the case of FeRu and FePtFe, all three of these issues become a problem. To illustrate, Figure 1.9 shows a simulated spectrum of four coupled modes that are well-separated and highly anharmonic. This alone leads to 16 sets of peaks and a more confusing spectrum. However, as the FTIR of both FeRu and FePtFe show, the four \( v_{\text{CN}} \) modes are not well separated, and as shown later, they are not highly anharmonic. This is the subject of Chapter 2.

1.4 **DISSERTATION OUTLINE**

Chapter 2 looks at gaining a better understanding of the vibrational coupling between four cyanide stretching modes in the ground electronic state of FeRu dissolved in D\(_2\)O or FA and FePtFe.
dissolved in D$_2$O using 2D IR spectroscopy. In order to obtain these couplings and other molecular properties, it was necessary to fit the 2D IR spectra. Chapter 3 also looks more closely at the ground electronic state of FeRu and FePtFe. Here, the focus is on using 2D IR and IR pump-probe spectroscopy to study vibrational relaxation dynamics within each molecule. Finally in Chapter 4, we move to new cyanide containing systems. This chapter focuses on developing syntheses and characterizing the vibrational spectra of three ruthenium (II) polypyridine cyanide systems for use in 2D IR experiments to characterize the cyanide stretching modes on the ground electronic state before using visible pump – IR probe experiments to study what role the high-frequency CN stretching modes play in a MLCT reaction.

REFERENCES


Chapter 2:
Investigating Vibrational Anharmonic Couplings In Cyanide-Bridged Transition Metal Mixed Valence Complexes Using Two-Dimensional Infrared Spectroscopy

The work in this chapter was published in the following article:

Using polarization-selective two-dimensional infrared (2D IR) spectroscopy, we measure anharmonic couplings and angles between the transition dipole moments of the four cyanide stretching ($\nu_{\text{CN}}$) vibrations found in $[(\text{NH}_3)_5\text{Ru}^\text{III}\text{NCFe}^\text{II}(\text{CN})_5]^+$ (FeRu) dissolved in D$_2$O and formamide and $[(\text{NC})_5\text{Fe}^\text{II}\text{CNPt}^\text{IV}(\text{NH}_3)_4\text{NCFe}^\text{II}(\text{CN})_5]^+$ (FePtFe) dissolved in D$_2$O. These cyanide-bridged transition metal complexes serve as model systems for studying the role of high frequency vibrational modes in ultrafast photoinduced charge transfer reactions. Here, we focus on the spectroscopy of the $\nu_{\text{CN}}$ modes in the electronic ground state. The FTIR spectra of the $\nu_{\text{CN}}$ modes of the bimetallic and trimetallic systems are strikingly different in terms of frequencies, amplitudes, and lineshapes. The experimental 2D IR spectra of FeRu and FePtFe and their fits reveal a set of weakly coupled anharmonic $\nu_{\text{CN}}$ modes. The vibrational mode anharmonicities of the individual $\nu_{\text{CN}}$ modes range from 14 to 28 cm$^{-1}$. The mixed-mode anharmonicities range from 2 to 14 cm$^{-1}$. In general, the bridging $\nu_{\text{CN}}$ mode is most weakly coupled to the radial $\nu_{\text{CN}}$ mode, which involves the terminal CN ligands. Measurement of the relative transition dipole moments
of the four $\nu_{CN}$ modes reveal that the FeRu molecule is almost linear in solution when dissolved in formamide, but it assumes a bent geometry when dissolved in D$_2$O. The $\nu_{CN}$ modes are modelled as bilinearly coupled anharmonic oscillators with an average coupling constant of 6 cm$^{-1}$. This study elucidates the role of the solvent in modulating the molecular geometry and the anharmonic vibrational couplings between the $\nu_{CN}$ modes in cyanide-bridged transition metal mixed valence complexes.

2.1 INTRODUCTION

Cyanide-bridged transition metal mixed valence complexes are a class of molecules which exhibit unique redox, spectroscopic, and charge transfer properties by virtue of bearing oxidizing and reducing moieties. They are widely studied for their applications in magnetism and photochemical energy conversion.$^{1-3}$ Additionally, these complexes have served as model systems to probe the coupling between electronic and vibrational motions during ultrafast photoinduced electron transfer reactions using equilibrium and transient spectroscopic techniques.$^{4-24}$ The goal of this work is to understand how the solvent modulates the molecular geometry and the anharmonic couplings between the bridging and terminal cyanide stretching vibrations in cyanide-bridged bimetallic and trimetallic transition metal mixed valence complexes. In this study, we use polarization-selective two-dimensional infrared (2D IR) spectroscopy to measure the anharmonic couplings and the angles between the transition dipole moments in the ground electronic state of the cyanide stretching ($\nu_{CN}$) vibrations of [(NH$_3$)$_5$Ru$^{III}$NCFe$^{II}$(CN)$_5$]$^-$ (FeRu) dissolved in D$_2$O and formamide (FA) and [(NC)$_5$Fe$^{II}$CNPt$^{IV}$(NH$_3$)$_4$NCFe$^{II}$(CN)$_5$]$^{4+}$ (FePtFe) dissolved in D$_2$O.
The molecular structures of the complexes and a simplified depiction of the $\nu_{\text{CN}}$ modes ($\nu_{\text{trans}}$, $\nu_{\text{radial}}$, $\nu_{\text{axial}}$, and $\nu_{\text{bridge}}$) are shown in Figure 2.1. A detailed description of the modes for FePtFe is provided in Figure 2.2 below. Our motivation to study the complexes shown in Figure 2.1 is based on previous IR and Raman spectroscopic studies of these molecules which have implicated some of the high frequency CN stretching vibrations in the ultrafast charge transfer processes. Transient IR studies of the FeRu molecule and its analogues find a high degree of vibrational excitation in the $\nu_{\text{CN}}$ modes upon ultrafast back electron transfer (BET). The role of the $\nu_{\text{CN}}$ modes in the metal-to-metal charge transfer (MMCT) excitation, and subsequent BET, of FePtFe dissolved in D$_2$O has also been recently studied using various nonlinear spectroscopies. These studies have found greater than 6 quanta of vibrational energy deposited into the $\nu_b$ mode following BET and provided experimental evidence that $\nu_{\text{CN}}$ modes are both coherently and incoherently coupled to the charge transfer reaction. Resonance Raman data on FePtFe and time-dependent intensity analysis by several research groups have shown that >40% of the inner sphere reorganization

**Figure 2.1:** The molecular structures of (a) FeRu and (b) FePtFe. Colored arrows correspond to simplified depictions of four $\nu_{\text{CN}}$ modes. The transition dipole moments of the $\nu_{\text{trans}}$ (blue), $\nu_{\text{axial}}$ (green), and $\nu_{\text{bridge}}$ (purple) modes lie along the MMCT axis and are perpendicular to the degenerate $\nu_{\text{radial}}$ (red) mode (see Figure 2.2). The metal-metal bond distance for FePtFe is 4.99 Å and ranges from 5.0 and 5.2 Å for FeRu.
energy involves vibrational modes which lie along the intervalence charge transfer axis of the complex.\textsuperscript{19,26-30} In this study, we aim to measure the anharmonic couplings and angles between the transition dipole moment of the $\nu_{CN}$ modes lying parallel and perpendicular to the charge transfer axis in the electronic ground state of the complexes shown in Figure 2.1. Our results will help illuminate how the molecular structure and geometry of cyanide-bridged transition metal mixed valence complexes affects ultrafast charge and energy transfer pathways following photoexcitation.

The role of the solvent in modulating the vibronic coupling in transition metal complexes has been studied for the trans $\nu_{CN}$ mode in FeRu (see Figure 2.1a).\textsuperscript{13} Recent X-ray absorption spectroscopy results at the Ru L$_3$-edge on FeRu have also shown the importance of hydrogen bonding effects of the solvent in modulating the charge on the transition metals.\textsuperscript{34} We will use 2D IR spectroscopy to measure how the solvent modifies the molecular geometry and the vibrational couplings of the $\nu_{CN}$ modes in FeRu and relate our results to previous studies. The experiment will measure the angles between the $\nu_{CN}$ modes and their vibrational mode and mixed-mode

Figure 2.2: Cartoons representing the normal modes for the $\nu_{CN}$ vibrations $\nu_{\text{trans}}$, $\nu_{\text{radial}}$, $\nu_{\text{axial}}$, and $\nu_{\text{bridge}}$. Normal mode symmetries are based on the $D_{4h}$ point group of the molecule. Red arrows represent symmetric displacements (not to scale). Similar modes are observed for the dimer RuFe.
anharmonicities by fitting simulated 2D IR spectra to the experimental spectra of FeRu and FePtFe. Using these normal mode anharmonicities, we also calculate bilinear coupling constants using the local mode basis for all three samples.

The remainder of the Chapter is organized as follows. Section 2.2 describes the methods used to synthesize the molecules, collect the 2D IR spectra, and simulate and fit the 2D IR spectra. Section 2.3 is split into three subsections. Section 2.3.1 describes the FTIR fitting and results. Section 2.3.2 contains the results and discussion of the experimental and simulated 2D IR spectra including the values for the angles and anharmonicities for FeRu in FA, FeRu in D$_2$O, and FePtFe in D$_2$O. Section 2.3.3 describes the method used to determine bilinear coupling constants and includes the discussion of these coupling constants. The final section, Section 2.4, contains the summary of the results and our conclusions.

2.2 Methods

Materials. Starting materials and solvents were purchased from Sigma Aldrich and used without further purification. Steady-state infrared spectra were collected with a JASCO FT/IR-4100 with a spectral resolution of 0.5 cm$^{-1}$. The solvents used in this study have absorption bands in the $\nu_{CN}$ region as shown in Figure 2.3. In order to remove the effect of the solvent before fitting the spectra shown in Figure 2.5, the solvent background was subtracted. For the case of FeRu dissolved in D$_2$O, an additional Voigt lineshape centered at 1979 cm$^{-1}$ with a width of 64 cm$^{-1}$ was subtracted from the FTIR spectrum to remove all contributions from the solvent. This subtraction has no effect on the fitting results tabulated in Table 2.1.
The mixed valence dimer, FeRu, was synthesized according to literature methods with a modified purification procedure. Briefly, an aqueous solution consisting of ~0.01 M $\text{K}_4[\text{Fe(CN)}_6]$ and ~0.01 M $[\text{RuCl(NH}_3)_5]\text{Cl}_2$ was heated at 60 °C for two hours. After removing solid impurities using vacuum filtration, methanol was added to the solution in a 2:1 volume ratio, and the solution was cooled overnight. The solid, green dimer was filtered out of solution, run through an ion-exchange column packed with Dowex, and then run through a size-exclusion column prepared with Bio-Gel P2 to remove starting materials. A saturated solution of FeRu was prepared in formamide (FA) with a maximum OD of 0.22 in the $\nu_{\text{CN}}$ region after solvent subtraction. A solution of FeRu was also prepared in deuterium oxide (D$_2$O) to a concentration of ~23 mM and maximum OD of 0.35 in the $\nu_{\text{CN}}$ region.

The mixed valence trimer, FePtFe, was synthesized following literature methods with additional purification steps. Briefly, aqueous $\text{K}_3[\text{Fe(CN)}_6]$ and $[\text{Pt(NH}_3)_4](\text{NO}_3)_2$ were mixed at room temperature, and the resulting solution was filtered using gravity filtration and purified by ion-exchange and size-exclusion chromatography. A solution of FePtFe dissolved in D$_2$O was prepared to a concentration of ~13 mM and a maximum OD of 0.41 for use in experiments.

![Figure 2.3: FTIR spectra before being solvent subtracted. The final experimental spectra presented in this paper in Figure 2.5 are the difference between the solution spectra (green) and the solvent spectra (blue).](image)
**2D IR Experiments.** The experimental set-up for performing 2D IR spectroscopy has been described recently.\textsuperscript{25,36} The polarization-selective experiments described here are performed with ~80 fs mid-IR pulses centered at a wavelength of 4.9 μm with a spectral bandwidth greater than 250 cm\textsuperscript{-1}. Each arm of the interferometer has a tunable half-wave plate (PO-TWP-L2-25-IR, Alphalas) and a wire-grid polarizer (WP25H-Z, Thorlabs) to adjust beam polarization and input power. The input beams, arranged in a “boxcar” geometry, are focused onto the sample with a spot size (1/e\textsuperscript{2}) of ~150 μm. The pulse energy in each arm of the three input beams is 0.4 μJ/pulse (0.5 μJ/pulse) for the FeRu (FePtFe) samples. The sample was contained in a home built sample cell equipped with two 1 mm thick CaF\textsubscript{2} windows and a 50 μm Teflon spacer for the D\textsubscript{2}O solutions or a 100 μm Teflon spacer for the FeRu in FA solution. The background free signal field is overlapped with the local oscillator beam on a 50:50 ZnSe beamsplitter for balanced heterodyne detection as described previously.\textsuperscript{25}

2D IR spectra were collected in two different polarization geometries: parallel (ZZZZ) where all input beams and the signal field have the same polarization, and crossed (YYZZ) where input beams 1 and 2 have a polarization that is rotated 90° to the polarization of beam 3 and the signal field. The ratio between peak amplitudes in the ZZZZ and YYZZ polarization geometry contains information about the angles between vibrational dipole moments.\textsuperscript{37,38} The IR coherence time delay (τ\textsubscript{1}) between pulses 1 and 2 and the waiting time delay (τ\textsubscript{2}) between pulses 2 and 3 are computer-controlled via Newport XMS50 linear stages. For FeRu dissolved in FA, the τ\textsubscript{1} delay was scanned from 0 to 3 ps in 3 fs steps for the rephasing (R) spectra and from 0 to 1.8 ps in 3 fs steps for the non-rephasing (NR) spectra at fixed τ\textsubscript{2} = 140 fs. For FeRu dissolved in D\textsubscript{2}O, the τ\textsubscript{1} delay was scanned from 0 to 2.5 ps in 3 fs steps for the R spectra and from 0 to 1.5 ps in 3 fs steps for the NR spectra at fixed τ\textsubscript{2} = 150 fs. For FePtFe, τ\textsubscript{1} was scanned from 0 to 2.2 ps in 4 fs steps
for the R spectra and from 0 to 1.5 ps in 4 fs steps for the NR spectra at fixed $\tau_2 = 150$ fs. The 2D IR data is collected at $\tau_2 \neq 0$ fs to reduce the contamination of the desired signal from additional signals present during the pulse-width of 80 fs and non-resonant signals from the sample cell windows. The data was zero padded to $2^{13}$ steps along $\tau_1$ and then Fourier transformed to obtain the $\omega_1$ axis. Rephasing and non-rephasing spectra were phased using a dispersed pump-probe trace at the same $\tau_2$ and added to construct absorptive 2D IR spectra for each sample as described previously.\textsuperscript{39} Each data point represents 1500 shots for the FeRu samples and 2000 shots for the FePtFe sample, and each spectrum was collected twice and averaged to obtain the experimental spectra presented here. Experimental spectral resolution along the $\omega_1$ and $\omega_3$ axes are 1.4 and $\sim 2.8$ cm$^{-1}$ respectively for the FeRu samples and 1.4 and $\sim 2.9$ cm$^{-1}$ for the FePtFe sample.

**2D IR Spectral Fitting.** One can simulate a 2D IR spectrum starting from the third order nonlinear signal that can be calculated using the nonlinear response formalism developed previously.\textsuperscript{39-41} The signal is calculated in the time domain as a function of $\tau_1$ and $\tau_3$ for a specific $\tau_2$ and Fourier transformed in both dimensions to obtain the $\omega_1$ and $\omega_3$ axes. Rephasing and non-rephasing spectra are calculated separately and then added in the frequency domain to obtain the absorptive spectrum. We assume a separation of time scales for vibrational and orientational dynamics when calculating the third order response functions. The rotational correlation function was written assuming a symmetric diffuser with an orientational diffusion constant of $D_{or}$.\textsuperscript{42}

The frequency-frequency correlation functions (FFCFs) used to simulate the 2D spectra are of the form:

\begin{align*}
C_{pp}(t) &= A_{pp}^2 \exp\left(-\frac{t}{\tau_{pp}}\right) + \frac{\delta(t)}{T'^2} \quad (2.1) \\
C_{pq}(t) &= A_{pp}A_{pq} \exp\left(-\frac{t}{\tau_{pq}}\right) + \frac{\delta(t)}{T'^*} \quad (2.2)
\end{align*}
Equation 2.1 is an autocorrelation function for mode \( p \) and Equation 2.2 is a cross correlation function between modes \( p \) and \( q \). In the above expressions, \( A_{pp} \) and \( \tau_{pp} \) are the amplitude and time scale of the Kubo model for mode \( p \), \( \tau_{pq} \) is the cross correlation time scale for modes \( p \) and \( q \), \( \delta(t) \) is the Dirac delta function, and \( T^*_2 \) is the time scale that represents the fast dynamics in a system leading to the homogeneous linewidth of peaks. All other FFCFs can be approximated as combinations of the two equations shown above.\(^{41}\) The following harmonic scaling relationships are used to limit the parameters necessary to simulate theoretical spectra:

\[
\begin{align*}
C_{2p,2p}(t) &= 2C_{2p,p}(t) = 2C_{p,2p}(t) = 4C_{pp}(t); \\
C_{p,pq}(t) &= C_{pp}(t) + C_{pq}(t); \\
C_{q,pq}(t) &= C_{qq}(t) + C_{pq}(t); \\
C_{pq,pq}(t) &= C_{pp}(t) + C_{qq}(t) + 2C_{pq}(t)
\end{align*}
\]

These relationships reduce the number of amplitude and timescale parameters needed for a system of four coupled vibrations from 60 to 14. We assume harmonic scaling for the magnitude of the vibrational transition dipole moment such that \( |\mu_{12}| = \sqrt{2} |\mu_{10}| \) and that the transition dipole moment for the \( n = 1 \rightarrow n = 2 \) transition is parallel to the transition dipole moment for the \( n = 0 \rightarrow n = 1 \) transition where \( n \) is the vibrational energy level of the system.

To determine orientational response functions each transition dipole moment is projected onto Cartesian coordinates with the transition dipole moment of the \( \nu_b \) mode acting as the arbitrary \( z \)-axis, the transition dipole moment of the \( \nu_r \) mode confined to the \( xz \)-plane at an angle of \( \theta_{\nu_b} \) from the \( \nu_b \) mode’s transition dipole moment, and the other two transition dipole moments determined in relation to the \( \nu_b \) and \( \nu_r \) modes by the angles between each transition dipole moment. We calculate the contribution of each affected transition dipole moment to the orientational response function for each possible set of light-matter interactions using the following equations from Reference 37:
Each $Y$ from Equation 2.4 represents the tensor element of the orientational response function for the case of four field interactions. Here $a$ represents transition dipole moments that are parallel to the first transition dipole excited, and $b$ represents transition dipole moments that are orthogonal to $a$.

The goal of simulating the 2D IR spectra is to extract vibrational mode and mixed-mode anharmonicities and the angles between the transition dipole moments of the $\nu_{CN}$ modes for each molecule. For all three complexes, a 15 energy-level system is used consisting of a common ground state, four first-excited one-quantum states, six combination two-quantum states, and four second-excited two-quantum states. A total of 120 third order response functions have to be calculated for each absorptive spectrum. To obtain a best fit spectrum, we iteratively change the lineshape parameters, the position of the two-quantum energy levels, and the transition dipole moment directions. We also fit an amplitude correction factor to each YYZZ spectrum to account for any experimental differences between the input power of beams 1 and 2 during the collection of the YYZZ and the ZZZZ spectra. The differences in input power arise from polarization-selective responses of some of the optics in the beam path. We stress that since we obtain the
angles by fitting both the ZZZZ and YYZZ data simultaneously for the four $\nu_{CN}$ modes, the correction factor does not affect our conclusions. The lineshape parameters are only fit to get the correct amplitude for each peak to enable an accurate determination of the transition dipole moment directions and the angles between them. These lineshape parameters were used to simulate the FTIR spectra displayed in Figure 2.4. The final lineshape parameters do not necessarily reflect the actual FFCF amplitudes and timescales for each sample and therefore will not be discussed in this chapter. The fits were completed in MATLAB using a nonlinear least-squares fitting routine.

![Figure 2.4: Experimental FTIR spectra (solid black) and spectra simulated (dashed green) using the lineshape parameters from our 2D fits presented in the captions for Figures 2.7, 2.8, and 2.10. The contribution to the simulated spectra from each mode is also shown: $\nu_t$ (blue), $\nu_r$ (red), $\nu_a$ (green), and $\nu_b$ (purple).](image)

2.3 RESULTS AND DISCUSSION

2.3.1 FTIR Spectra and Assignment of the $\nu_{CN}$ Modes

Figure 2.5 displays the solvent-subtracted FTIR spectra of the $\nu_{CN}$ region for FeRu dissolved in FA and D$_2$O and FePtFe dissolved in D$_2$O. Each FTIR spectrum was fit to four Voigt lineshapes for each $\nu_{CN}$ mode shown in Figure 2.1. Table 2.1 lists the center frequencies and spectral
linewidths of those fits for each sample. The fits to the spectra reveal significant changes in positions for the four ν\textsubscript{CN} modes between FeRu and FePtFe and as a function of solvent for FeRu. In addition to shifts in frequency, there is variation in the linewidths (Δν\textsubscript{FWHM}) and transition dipole strengths (|μp|²) in the FTIR spectra of the four ν\textsubscript{CN} modes shown in Figure 2.5. The linewidth of the νt mode is significantly larger in FeRu than in FePtFe. By contrast, the linewidths of the other three modes change by less than 3 cm\textsuperscript{-1} between all the samples. The relative transition dipole

**Figure 2.5:** Solvent subtracted FTIR spectra (black line) of (a) FeRu in FA (b) FeRu in D\textsubscript{2}O and (c) FePtFe in D\textsubscript{2}O. Each FTIR spectrum was fit to four Voigt lineshapes, and each peak is color coded to match the four modes shown in Figure 2.1 (ν\textsubscript{t} (blue), ν\textsubscript{a} (green), ν\textsubscript{b} (purple), and ν\textsubscript{r} (red)). Parameters of each fit are listed in Table 2.1. The final total fit is shown in dashed green.
strengths also exhibit a variation among the three samples. For FeRu in FA, the transition dipole moment strengths of all the $\nu_{CN}$ modes with respect to the $\nu_t$ mode are as follows: $|\mu_r| = 0.56|\mu_t|$, $|\mu_a| = 0.30|\mu_t|$, and $|\mu_b| = 0.22|\mu_t|$. When FeRu is dissolved in D$_2$O, the $|\mu_p|^2$ of all the $\nu_{CN}$ modes with respect to the $\nu_t$ mode are as follows: $|\mu_r| = 0.89|\mu_t|$, $|\mu_a| = 0.40|\mu_t|$, and $|\mu_b| = 0.12|\mu_t|$. For FePtFe in D$_2$O, the strongest peak is assigned to the radial mode and $|\mu_p|^2$ of all the $\nu_{CN}$ modes with respect to the $\nu_t$ mode are as follows: $|\mu_r| = 0.73|\mu_t|$, $|\mu_a| = 0.40|\mu_t|$, and $|\mu_b| = 0.42|\mu_t|$. It is instructive to compare the FTIR spectra shown in Figure 2.5 with previously reported FTIR and Raman spectra of FeRu and FePtFe. Previous work by Walker and co-workers had assigned three high-frequency IR active $\nu_{CN}$ modes for FeRu, two of which are also Raman active. The IR active modes were labelled as $\nu_{\text{bridge-CN}}$, $\nu_{\text{cis-CN}}$, and $\nu_{\text{trans-CN}}$ and they map to the modes labelled $\nu_b$, $\nu_r$, and $\nu_t$, respectively in Figure 2.1. The previous study did not assign an axial mode in FeRu, but a fourth $\nu_{CN}$ mode was acknowledged in the FTIR spectrum of [(NH$_3$)$_5$Ru$^{III}$NCOs$^{II}$(CN)$_5$]$^{-}$ at approximately 2070 cm$^{-1}$. We see clear evidence of the $\nu_a$ mode in the form of cross-peaks in 2D IR spectra (Figures 2.7-2.10) and in dispersed IR pump-probe data of the FeRu samples. We have measured the Raman spectrum of FeRu in FA showing that the $\nu_t$ mode is centered at 2000±2 cm$^{-1}$, and the $\nu_b$ mode is centered at 2088±2 cm$^{-1}$, similar to the IR spectrum. For FeRu in D$_2$O, the $\nu_t$ mode in the Raman spectrum is centered at 2022±4 cm$^{-1}$, and the $\nu_b$ mode is centered at 2099±2 cm$^{-1}$. Four high-frequency IR and Raman active modes are also observed for FePtFe. The previously reported Raman spectrum of FePtFe assigns the bridge mode at 2125 cm$^{-1}$, peaks at 2067 cm$^{-1}$ and 2081 cm$^{-1}$ are assigned to the $\nu_r$ mode, and the $\nu_a$ mode is observed at 2107 cm$^{-1}$. The discrepancy between the Raman and IR frequencies of the observed cyanide stretching modes in FePtFe is attributed to the structural complexity of the trimer molecule.
Table 2.1: Best fit parameters of the FTIR spectra of FeRu and FePtFe. The error bars are reported at 95% confidence interval.

<table>
<thead>
<tr>
<th></th>
<th>ν_{trans}</th>
<th>Δν_{FWHM}</th>
<th>ν_{max}</th>
<th>Δν_{FWHM}</th>
<th>ν_{max}</th>
<th>Δν_{FWHM}</th>
<th>ν_{max}</th>
<th>Δν_{FWHM}</th>
</tr>
</thead>
<tbody>
<tr>
<td>FeRu/FA</td>
<td>2002.0±0.1</td>
<td>20.5±0.1</td>
<td>2050.6±0.1</td>
<td>8.7±0.1</td>
<td>2065±1</td>
<td>11.2±0.9</td>
<td>2089.4±0.1</td>
<td>5.3±0.2</td>
</tr>
<tr>
<td>FeRu/D_{2}O</td>
<td>2026±1</td>
<td>26.8±0.8</td>
<td>2047.5±0.1</td>
<td>7.1±0.1</td>
<td>2060.9±0.6</td>
<td>9.5±0.6</td>
<td>2096±2</td>
<td>6±3</td>
</tr>
<tr>
<td>FePtFe/D_{2}O</td>
<td>2059.3±0.8</td>
<td>8±3</td>
<td>2049.3±0.5</td>
<td>9.4±0.5</td>
<td>2072±5</td>
<td>9±1</td>
<td>2116.3±0.2</td>
<td>8.2±0.3</td>
</tr>
</tbody>
</table>

The FTIR spectra in Figure 2.5 and the best fit results in Table 2.1 reveal that the largest variation in the spectral parameters (frequency, linewidth, and amplitude) occur for the ν_{CN} modes that lie along the charge transfer axis. These include the trans, axial, and bridge ν_{CN} modes. Our results are consistent with the Raman studies and transient IR studies which have found that the trans and the bridge ν_{CN} modes are coupled to the MMCT transition in these transition metal mixed valence complexes.\textsuperscript{13-15,18,20,21} The 2D IR spectra of the ν_{CN} modes described in the next section will measure the anharmonic coupling between these modes and the relative angles between them.

### 2.3.2 2D IR Spectra of the ν_{CN} Modes in Mixed Valence Transition Metal Complexes

Two-dimensional infrared spectroscopy (2D IR) is a third order nonlinear technique that is sensitive to the transitions involving the ground (n = 0), first excited (n = 1), and second excited (n = 2) vibrational states. An example of a generalized fifteen vibrational energy level diagram contributing to a 2D IR spectrum of four coupled ν_{CN} modes is shown in Figure 2.6. Two-dimensional infrared spectra of coupled vibrational modes consist of ‘diagonal’ and ‘cross’ peaks. Diagonal peaks, centered at ω_{1} = ω_{3}, occur when all three pulses interact with the same vibrational mode during τ_{1} and τ_{3}. Cross peaks located at ω_{1} = ν_{p} and ω_{3} = ν_{q}, only occur if modes p and q are anharmonically coupled and are found within the bandwidth of the IR pulses. There are also corresponding cross peaks between modes p and q across the diagonal at ω_{1} = ν_{q} and ω_{3} = ν_{p}. Each
diagonal and cross peak consists of positive and negative spectral features. The positive features involve transitions between \( n = 0 \) and \( n = 1 \) states, while the negative features involve a transition to a two-quantum state. The diagonal peaks are separated along \( \omega_3 \) by the vibrational mode anharmonicity \( (\Delta_{p(q)}) \), and the cross peaks are separated along \( \omega_3 \) by the mixed-mode anharmonicity \( (\Delta_{pq}) \). The mixed-mode anharmonicities measure the magnitude of the couplings between modes \( p \) and \( q \). When the vibrational anharmonicities are larger than the linewidths of the vibrational modes under study, the positive and negative features are well-separated in the \( \omega_3 \) dimension and the values of \( \Delta_{p(q)} \) and \( \Delta_{pq} \) can be read off the 2D IR spectrum. This is not the case for the \( \nu_{CN} \) modes in both FeRu and FePtFe, making it necessary to fit the 2D IR spectrum to obtain accurate values for the anharmonicities.

![Diagram of vibrational states](image)

**Figure 2.6**: A generalized diagram of all the relevant vibrational states probed in the 2D IR experiment for four coupled vibrations. The naming scheme is \( |n_t n_r n_a n_b| \) where \( n_i \) refers to the vibrational quantum number of the trans, radial, axial, and bridge \( \nu_{CN} \) modes.

The amplitudes of the diagonal peaks are proportional to \( |\mu_p|^4 \), and the cross peak amplitudes are proportional to \( |\mu_p|^2 |\mu_q|^2 \), where \( \mu_p \) is the magnitude of the transition dipole moment of mode \( p \). The amplitudes of the diagonal and cross peaks are also affected by the angle between the transition
dipole moments of the vibrations under study. The ratio of the amplitude of a cross peak in a 2D IR spectrum measured in the $ZZZZ$ (parallel) and the $YYZZ$ (cross) polarization geometries can be related to the relative angle between the transition dipole moments of the coupled vibrations. When the vibrations of interest are spectrally overlapped and the spectral linewidths are larger than the vibrational anharmonicities, it is not possible to obtain an accurate measurement of the angles between coupled vibrations by simply taking a ratio of their amplitudes under different polarization conditions. Attempting to do this for the 2D IR spectra shown in Figures 2.7, 2.8, and 2.10 resulted in imaginary values for several angles making it necessary to fit the spectra to obtain useful angles.

We have used the nonlinear response formalism described in the Section 2.2 to fit the polarization selective 2D IR spectra of FeRu in D$_2$O and FA and FePtFe in D$_2$O. Table 2.2 lists the best fit parameters for the vibrational anharmonicities and angles between the four coupled $\nu_{CN}$ modes. Below we discuss the experimental 2D IR spectra and the best fit results.

**Table 2.2**: Best fit values for vibrational mode anharmonicities, mixed-mode anharmonicities, and angles between modes for all three samples. The error bars are estimated to be $\pm 2$ cm$^{-1}$ and $\pm 5^\circ$.

<table>
<thead>
<tr>
<th></th>
<th>FeRu/FA</th>
<th>FeRu/D$_2$O</th>
<th>FePtFe/D$_2$O</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta_1$ (cm$^{-1}$)</td>
<td>14</td>
<td>15</td>
<td>24</td>
</tr>
<tr>
<td>$\Delta_2$ (cm$^{-1}$)</td>
<td>18</td>
<td>19</td>
<td>21</td>
</tr>
<tr>
<td>$\Delta_3$ (cm$^{-1}$)</td>
<td>27</td>
<td>28</td>
<td>19</td>
</tr>
<tr>
<td>$\Delta_4$ (cm$^{-1}$)</td>
<td>19</td>
<td>28</td>
<td>24</td>
</tr>
<tr>
<td>$\Delta_5$ (cm$^{-1}$)</td>
<td>2</td>
<td>14</td>
<td>2</td>
</tr>
<tr>
<td>$\Delta_6$ (cm$^{-1}$)</td>
<td>2</td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>$\Delta_{ab}$ (cm$^{-1}$)</td>
<td>4</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>$\Delta_{ba}$ (cm$^{-1}$)</td>
<td>6</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>$\Delta_{ab}$ (cm$^{-1}$)</td>
<td>5</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>$\Delta_{ab}$ (cm$^{-1}$)</td>
<td>13</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>$\theta_{12}$</td>
<td>85°</td>
<td>71°</td>
<td>73°</td>
</tr>
<tr>
<td>$\theta_{13}$</td>
<td>2°</td>
<td>5°</td>
<td>9°</td>
</tr>
<tr>
<td>$\theta_{14}$</td>
<td>6°</td>
<td>14°</td>
<td>25°</td>
</tr>
<tr>
<td>$\theta_{23}$</td>
<td>87°</td>
<td>74°</td>
<td>68°</td>
</tr>
<tr>
<td>$\theta_{24}$</td>
<td>88°</td>
<td>59°</td>
<td>50°</td>
</tr>
<tr>
<td>$\theta_{34}$</td>
<td>6°</td>
<td>19°</td>
<td>18°</td>
</tr>
</tbody>
</table>
2D IR spectroscopy of FeRu in Formamide. The ZZZZ and YYZZ experimental spectra of FeRu in FA at $\tau_2 = 140$ fs are shown in Figures 2.7a and 2.7b. The experimental spectra display three main peaks along the diagonal. The lowest energy diagonal peak involves the $\nu_t$ mode, the middle diagonal peak contains overlapping contributions from the $\nu_t$ and $\nu_a$ modes, and the highest energy peak results from exciting and probing the $\nu_b$ mode. The middle diagonal peak also has contributions from the cross peaks between the $\nu_t$ and the $\nu_a$ modes. There are three well resolved

![Figure 2.7: 2D IR spectra of FeRu dissolved in FA at $\tau_2 = 140$ fs. (a) ZZZZ experiment, (b) YYZZ experiment, (c) ZZZZ simulated, and (d) YYZZ simulated. Each spectrum is plotted using contour lines at the following positions: $\pm 0.018, \pm 0.055, \pm 0.1, \pm 0.2, \pm 0.35, \pm 0.5, \pm 0.65, \pm 0.8$, and $\pm 0.95$. The YYZZ spectra are multiplied by the numbers in the lower right corners. The lineshape parameters used to simulate the spectra in (c) and (d) are $\mu_t = 1.1, \mu_r = 0.60, \mu_a = 0.45, \mu_b = 0.30, A_{tt} = 2.0$ ps$^{-1}, A_{rr} = 1.6$ ps$^{-1}, A_{aa} = 2.9$ ps$^{-1}, A_{bb} = 1.0$ ps$^{-1}, \tau_{tt} = 9.5$ ps, $\tau_{rr} = 0.78$ ps, $\tau_{aa} = 2.8$ ps, $\tau_{bb} = 4.0$ ps, $\tau_{tr} = 0.14$ ps, $\tau_{ta} = 0.34$ ps, $\tau_{tb} = 0.84$ ps, $\tau_{ra} = 0.79$ ps, $\tau_{rb} = 0.85$ ps, $\tau_{ab} = 2.3$ ps, $\tau_{2t} = 0.21$ ps, and $T_{2t}^* = 0.21$ ns. For the $\nu_t$ mode, $|\mu_{12}| = 1.27|\mu_0|$.}
cross peaks in the $ZZZZ$ spectrum. Two cross peaks are located along $\omega_1 = \nu_b$, one between the $\nu_b$ and $\nu_t$ modes at $\omega_3 = \nu_t$ and one between the $\nu_b$ and $\nu_r$ modes at $\omega_3 = \nu_r$. Along $\omega_1 = \nu_t$, there is a cross peak between $\nu_t$ and $\nu_b$ at $\omega_3 = \nu_b$. Several more cross peaks are observed in the $YYZZ$ spectrum (Fig. 2.7b). Along $\omega_1 = \nu_r$, there is a cross peak between $\nu_r$ and $\nu_t$ at $\omega_3 = \nu_t$, and along $\omega_1 = \nu_t$, there is the corresponding cross peak between $\nu_r$ and $\nu_t$ at $\omega_3 = \nu_r$. The negative feature of the cross peak at $\omega_1 = \nu_t$, $\omega_3 = \nu_r$ is overlapped with the much stronger positive feature of the $\nu_t$ diagonal peak and is not visible on the spectrum. Along $\omega_1 = \nu_t$, the negative feature of a cross peak between $\nu_t$ and $\nu_b$ at $\omega_3 = \nu_b$ can be seen. The cross peaks that are visible in both polarization geometries are stronger in the $YYZZ$ spectrum relative to the diagonal peaks in comparison to the $ZZZZ$ spectrum indicating that the angles between those coupled modes are closer to $90^\circ$ than $0^\circ$.

We note that the cross peaks involving the bridge mode have a higher intensity than the diagonal peak for the bridge mode. This arises from the fact that the dipole moments for $\nu_t$, $\nu_r$, and $\nu_a$ are larger than the dipole for $\nu_b$ so $|\mu_p|^2 |\mu_b|^2$ for the cross peaks is greater than $|\mu_b|^4$ for the diagonal peak (where $p = t, r, a$).

As observed in the FTIR spectrum of FeRu in FA, the four cyanide stretches are overlapping with each other and this makes it challenging to assign all the peaks and read off the anharmonicities from the experimental 2D spectra. For example, the negative peak below the $\nu_b$ diagonal peak does not represent the vibrational mode anharmonicity of $\nu_b$. Instead it includes contributions from several combination bands. The indent in the contours of the cross peak at $\omega_1 = \nu_b$, $\omega_3 = \nu_a$ likely represents the position of the negative diagonal peak of $\nu_b$.

The $ZZZZ$ and $YYZZ$ 2D IR spectra simulated from the best fit parameters obtained using the methodology described in Section 2.2 are shown in Figure 2.7c and 2.7d. The fit spectra capture all the features of the experimental data in terms of spectral positions, amplitudes, and lineshapes.
Like in the experimental spectra, we see two main peaks along the diagonal with a weak diagonal bridge peak. Differences between the experimental and fit spectra are observed at $\omega_1 = \nu_t$ and $\omega_3 = \nu_b$, where the positive portion of the $\nu_t$ and $\nu_b$ cross peak is visible in the fit spectra and the negative portion of the cross peak has less amplitude in the fit. However, we point out that the differences among those contour lines is only $\sim 4\%$. The slight differences in amplitudes and lineshapes of the negative diagonal and cross-peak features between the experiment and simulation are attributed to the breakdown of the harmonic scaling approximation for the FFCFs, transition moment dipole strengths, and relative angles between the $n = 0 \rightarrow n = 1$ and $n = 1 \rightarrow n = 2$ transitions among all the $\nu_{CN}$ modes.

The vibrational anharmonicities and angles between the transition dipole moments determined from the fit are shown in column 1 of Table 2.2. The individual vibrational mode anharmonicities range from $14 \text{ cm}^{-1}$ for $\nu_t$ to $27 \text{ cm}^{-1}$ for $\nu_a$, and the mixed-mode anharmonicities range from $2 \text{ cm}^{-1}$ for $\Delta_{tr}$ and $\Delta_{ta}$ to $13 \text{ cm}^{-1}$ for $\Delta_{ab}$. The $13 \text{ cm}^{-1}$ value for $\Delta_{ab}$ is significantly larger than the other mixed-mode anharmonicities indicating that those two modes are strongly coupled. For a perfectly linear molecule (as FeRu is depicted in Figure 2.1a), we would expect all of the angles between the transition dipoles of the four $\nu_{CN}$ modes to be either $0^\circ$ or $90^\circ$. The results from the spectral fitting find that the angles between the transition dipole of the radial mode and the other three modes is $\sim 90^\circ$ and the transition dipoles of the modes lying along the charge transfer axis are approximately parallel to one another. The polarization-selective 2D IR spectra and their fits reveal that the FeRu molecule dissolved in FA exists in a linear geometry, and the four $\nu_{CN}$ modes are weakly coupled to each other.

2D IR spectroscopy of FeRu dissolved in $D_2O$. The ZZZZ and YYZZ 2D IR experimental spectra of FeRu in $D_2O$ at $\tau_2 = 150 \text{ fs}$ are shown in Figure 2.8a and 2.8b. In contrast to the 2D IR
spectra of FeRu in FA where there were three visible diagonal peaks, there is only one obvious diagonal peak here. This diagonal peak mainly represents the overlapped contributions from $\nu_r$ and $\nu_a$. In the $ZZZZ$ spectrum, there is also another positive feature at approximately $\omega_1 = \omega_3 = \nu_t$, that represents some of the contribution from the diagonal peak of $\nu_t$. There is no spectral amplitude in the region where $\nu_b$ should have a diagonal peak. Given the relative transition dipole moment strengths of the bridge and radial modes, we estimate that the magnitude of the $\nu_b$ diagonal peak would be $\sim 3000$ times smaller than that of the $\nu_t$ diagonal peak. The relatively weak transition

**Figure 2.8:** 2D IR spectra of FeRu dissolved in D$_2$O at $\tau_2 = 150$ fs. (a) $ZZZZ$ experiment, (b) $YYZZ$ experiment, (c) $ZZZZ$ simulated, and (d) $YYZZ$ simulated. Each spectrum is plotted using contour lines at the following positions: $\pm 0.01$, $\pm 0.04$, $\pm 0.1$, $\pm 0.2$, $\pm 0.35$, $\pm 0.5$, $\pm 0.65$, $\pm 0.8$, and $\pm 0.95$. The $YYZZ$ spectra are multiplied by the numbers in the lower right corners. The lineshape parameters used to simulate the spectra in (c) and (d) are $\mu_t = 1$, $\mu_r = 0.923$, $\mu_a = 0.458$, $\mu_b = 0.163$, $A_{tt} = 2.9$ ps$^{-1}$, $A_{rr} = 2.1$ ps$^{-1}$, $A_{aa} = 2.6$ ps$^{-1}$, $A_{bb} = 2.8$ ps$^{-1}$, $\tau_{tt} = 4.9$ ps, $\tau_{rr} = 0.61$ ps, $\tau_{aa} = 2.9$ ps, $\tau_{bb} = 1.3$ ps, $\tau_{tr} = 0.090$ ps, $\tau_{ta} = 0.79$ ps, $\tau_{tb} = 1.3$ ps, $\tau_{ra} = 0.62$ ps, $\tau_{rb} = 1.6$ ps, $\tau_{ab} = 1.4$ ps, $T_{2t}^* = 0.24$ ps, and $T_2^* = 3.0$ ns.
dipole moment strength of the bridge mode results in a lack of cross peaks in the 2D IR spectra.

Figure 2.9: Slices of the YYZZ experimental 2D spectra (normalized) at the $\omega_1$ frequency for each of the four modes: $v_t$ (blue), $v_r$ (red), $v_a$ (green), and $v_b$ (purple). Cross peaks are marked with the corresponding mode in $\omega_3$.

dipole moment strength of the bridge mode results in a lack of cross peaks in the 2D IR spectra.
We estimate that the cross peak between $\nu_r$ and $\nu_b$ would have an amplitude ~50 times smaller than the $\nu_r$ diagonal peak. Some of the cross peaks are more easily observed when taking slices of the 2D spectra. Slices at the $\omega_1$ frequency of each mode for each system are shown in Figure 2.9. The cross peaks observed when looking at the 2D spectra are spectrally overlapped with the diagonal peaks. The wing of the 2D IR lineshape at $\omega_3 = \nu_r$ seen in both polarizations is a cross peak between $\nu_r$ and $\nu_t$. It is much stronger in the YYZZ spectrum relative to the diagonal peak in comparison to the ZZZZ spectrum, and the negative feature is not visible in the ZZZZ spectrum. The shoulder at $\omega_1 = \nu_r$ off of the negative $\nu_r$ diagonal peak to lower energy seen in the YYZZ spectrum is the negative feature of the corresponding cross peak between $\nu_r$ and $\nu_t$.

The ZZZZ and YYZZ spectra of the fit are shown in Figure 2.8c and 2.8d. The fit has successfully captured all of the features of the experimental spectra. The most obvious differences between the experimental and fit spectra appear in the amplitude of the negative peaks due to our assumption of harmonic scaling for the magnitude of the transition dipole moments. The anharmonicities and angles obtained from this fit are in the second column of Table 2.2. The vibrational mode anharmonicities range from 15 cm$^{-1}$ for the $\nu_t$ mode to 28 cm$^{-1}$ for the $\nu_a$ and $\nu_b$ modes. The mixed-mode anharmonicity between the $\nu_t$ and $\nu_r$ modes is 14 cm$^{-1}$ which is the same as $\Delta_t$. The other mixed-mode anharmonicities range from 2 cm$^{-1}$ for $\Delta_{ta}$ to 11 cm$^{-1}$ for $\Delta_{ab}$. The results from the spectral fitting find that the angles between the transition dipoles of the radial mode and the other three modes are in the range of 60°-75°. The relative angles between the trans, axial, and bridge modes are in the range of 5°-20°. The polarization-selective 2D IR spectra and their fits reveal that the FeRu molecule dissolved in D$_2$O exists in a bent geometry, and the four $\nu_{CN}$ modes are weakly coupled to each other.
These results highlight the importance and strong effect of solute-solvent interactions in modulating the molecular geometry and in turn the couplings between the four coupled $\nu_{CN}$ modes. The large variation in the angles between the transition dipoles of the four $\nu_{CN}$ modes measured for FeRu dissolved in FA and D$_2$O demonstrates how the solute-solvent interactions affect the geometry of solute molecules. When FeRu is dissolved in FA, the measured angles between the four $\nu_{CN}$ modes reveal that the molecule is almost perfectly linear. However, when FeRu is dissolved in D$_2$O, a bent structure is preferred. While the diagonal anharmonicities for the $\nu_t$, $\nu_r$, and $\nu_a$ modes are within error of each other, the anharmonicity for the $\nu_b$ mode increases by 9 cm$^{-1}$ from FeRu in FA to FeRu in D$_2$O. Except for $\Delta_{tt}$, all of the mixed-mode anharmonicities are also within error of each other. The value for $\Delta_{tt}$ increases from 2 cm$^{-1}$ for FeRu in FA to 14 cm$^{-1}$ for FeRu in D$_2$O indicating the $\nu_t$ and $\nu_r$ modes are more strongly coupled in D$_2$O than in FA. The increased coupling between the trans and radial modes could result from the bent molecular geometry of FeRu in D$_2$O. As discussed earlier, previous Resonance Raman studies of FeRu in FA and D$_2$O have noted that the solvent determines which of the four $\nu_{CN}$ modes couple to the non-radiative back electron transfer process in this molecule.$^{13}$ The increase in the coupling between the trans and radial modes when FeRu is dissolved in FA and D$_2$O respectively provides evidence on how solvents can be used to manipulate vibrational energy relaxation and charge transfer pathways in complex molecular systems used for photochemical energy conversion.

2D IR spectroscopy of FePtFe dissolved in D$_2$O. The ZZZZ and YYZZ 2D IR experimental spectra of FePtFe in D$_2$O at $\tau_2 = 150$ fs are shown in Figure 2.10a and 2.10b. In contrast to FeRu in D$_2$O where there was only one visible diagonal peak, there are two diagonal peaks noticeable in both spectra for FePtFe. The more intense diagonal peak contains contributions from the $\nu_t$, $\nu_r$, and $\nu_a$ modes. The much smaller peak is the $\nu_b$ diagonal peak. Vibrational mode anharmonicities
for the $\nu_r$, $\nu_t$, and $\nu_a$ modes do not appear to be significantly different since the separation between the positive and negative features of the central diagonal peak does not seem to change much across its range of $\omega_1$ frequencies. The central diagonal peak also seems to be fairly elongated along the diagonal indicating a large inhomogeneous component to its lineshape, whereas the $\nu_b$ diagonal peak seems to be fairly well rounded indicating a more homogeneous lineshape. This is in contrast to the lineshape for $\nu_b$ in FeRu in FA, where $\nu_b$ does not have much anti-diagonal width.

Figure 2.10: 2D IR spectra of FePtFe dissolved in D$_2$O at $\tau_2 = 150$ fs. (a) $ZZZZ$ experiment, (b) $YYZZ$ experiment, (c) $ZZZZ$ simulated, and (d) $YYZZ$ simulated. Each spectrum is plotted using contour lines at the following positions: $\pm0.01, \pm0.045, \pm0.1, \pm0.2, \pm0.35, \pm0.5, \pm0.65, \pm0.8, \text{and } \pm0.95$. The YYZZ spectra are multiplied by the numbers in the lower right corners. The lineshape parameters used to simulate the spectra in (c) and (d) are $\mu_t = 0.69$, $\mu_r = 1$, $\mu_a = 0.34$, $\mu_b = 0.28$, $A_{tt} = 2.1$ ps$^{-1}$, $A_{rt} = 3.2$ ps$^{-1}$, $A_{aa} = 8.0$ ps$^{-1}$, $A_{bb} = 2.2$ ps$^{-1}$, $\tau_{tt} = 5.4$ ps, $\tau_{rt} = 1.0$ ps, $\tau_{aa} = 5.0$ ps, $\tau_{bb} = 0.27$ ps, $\tau_{ta} = 0.20$ ps, $\tau_{ta} = 1.3$ ps, $\tau_{tb} = 0.20$ ps, $\tau_{ra} = 1.1$ ps, $\tau_{rb} = 0.50$ ps, $\tau_{ab} = 1.0$ ps, and $T_z^* = 0.39$ ns.
There are only two observable cross peaks for FePtFe that can be seen in the spectra at both polarization geometries. One cross peak is along $\omega_1 = \nu_r$ at $\omega_3 = \nu_b$ and the other is $\omega_1 = \nu_b$ at $\omega_3 \approx \nu_r$. Both represent the cross peak between the $\nu_b$ mode and each of the other three modes $\nu_r$, $\nu_t$, and $\nu_a$. The cross peaks are once again much stronger in the YYZZ spectrum relative to the diagonal peaks than they are in the ZZZZ spectrum.

The ZZZZ and YYZZ spectra of the fit for FePtFe are shown in Figure 2.10c and 2.10d. We see that the fit reproduces all the spectral features seen in the experimental spectra. As with FeRu in D$_2$O, we see that most of the negative features have too much amplitude, proving that harmonic scaling is not an accurate assumption for these molecules dissolved in D$_2$O. The experimental spectra have what appears to be another small positive and negative set of peaks centered at $\omega_1 = \sim 1990$ cm$^{-1}$ and $\omega_3 = \nu_r$, and the fit does not reproduce this. The peaks could be from some slight contamination of starting material or from a product of sample degradation.

The vibrational anharmonicities and angles obtained from this fit are shown in the third column of Table 2.2. The individual vibrational mode anharmonicities are all around 20 cm$^{-1}$ with the smallest being 19 cm$^{-1}$ for the $\nu_a$ mode and the largest at 24 cm$^{-1}$ for both the $\nu_t$ and $\nu_b$ modes, which is a much smaller spread than the values obtained for FeRu in D$_2$O. The vibrational anharmonicities for the $\nu_r$, $\nu_t$, and $\nu_a$ modes listed in Table 2.2 are within experimental error to those previously reported.$^{21}$ The mixed-mode anharmonicities range from 2 to 15 cm$^{-1}$. The results from the spectral fitting find that the angles between the transition dipoles of the radial mode and the other three modes are in the range of 50°-73°. The relative angles between the trans, axial, and bridge modes are in the range of 9°-25°. The polarization-selective 2D IR spectra and their fits reveal that the FePtFe molecule dissolved in D$_2$O exists in a bent geometry, and the four $\nu_{CN}$ modes are weakly coupled to each other.
Comparing our fitting results for FeRu and FePtFe both dissolved in D$_2$O, we find that there are a lot of similarities. The vibrational mode anharmonicities for the $v_r$ and $v_b$ modes are the same within error, as are the mixed-mode anharmonicities for the $v_t$ and $v_b$, $v_r$ and $v_a$, and $v_a$ and $v_b$ modes. However, some of the diagonal vibrational mode anharmonicities are very different. For example, $\Delta_t$ shifts up 11 cm$^{-1}$ and $\Delta_a$ shifts down 9 cm$^{-1}$ from FeRu to FePtFe. The mixed-mode anharmonicities $\Delta_{tr}$, $\Delta_{ta}$, and $\Delta_{rb}$ also shift significantly between molecules with $\Delta_{tr}$ and $\Delta_{rb}$ larger in FeRu than in FePtFe.

Previous work on CN stretches find vibrational mode anharmonicities that range from about 15 to 36 cm$^{-1}$. Anharmonicities for Fe-CN and Os-CN span from $\sim$15 cm$^{-1}$ to $\sim$25 cm$^{-1}$. These values agree well with the vibrational mode anharmonicities presented here for Fe-CN systems which range from 14 to 28 cm$^{-1}$. Walker has reported anharmonicities for the $\nu_{\text{trans-CN}}$ and $\nu_{\text{cis-CN}}$ modes of the [(NH$_3$)$_5$Ru$^{III}$NCOs$^{II}$C(N)$_5$]$^-$ dimer in FA and D$_2$O. The $\nu_{\text{cis-CN}}$ mode corresponds to our $\nu_r$ mode, and their reported value of $\sim$20 cm$^{-1}$ matches well with the 18 – 21 cm$^{-1}$ we report for the $\nu_r$ in all three samples. However, the $\nu_{\text{trans-CN}}$ mode anharmonicity of $\sim$25 cm$^{-1}$, which corresponds to our $\nu_t$ mode, only matches the $\nu_t$ anharmonicity of 24 cm$^{-1}$ for the FePtFe system and is significantly larger than the $\nu_t$ anharmonicities for both FeRu systems.

Femtosecond transient IR spectroscopy on FePtFe dissolved in D$_2$O has measured rapid dissipation of vibrational energy along the axis of the Fe-Pt-Fe bond in the ground electronic state following MMCT excitation and subsequent ultrafast BET. The directionality of the rapid vibrational energy relaxation can be explained by the average anharmonic coupling of $\sim$11 cm$^{-1}$ between the trans, bridge, and axial modes lying along the charge transfer axis. In comparison, the average anharmonic coupling is $\sim$5 cm$^{-1}$ between the radial mode (whose transition dipole moment does not lie along the charge transfer axis) and the rest of the $\nu_{\text{CN}}$ modes. These results
suggest that appropriate placement of ligand architectures in charge transfer complexes can be used to direct charge and energy migration.

2.3.3 Extracting Anharmonic Couplings in the Site Basis Representation

The data in Table 2.2 suggests that we can effectively describe our system of four CN modes as weakly coupled anharmonic oscillators where the Hamiltonian is described in the site basis by Equation 2.5:48

\[
H = \sum_{i=a',a',a',b'} (\varepsilon_i m_i^\dagger m_i - \frac{\Delta_i}{2} m_i^\dagger m_i m_i) \\
+ \beta_{rr'} (m_r^\dagger m_r + m_{r^\prime}^\dagger m_{r^\prime}) + \beta_{ia'} (m_{a'}^\dagger m_{a'} + m_{a^\prime}^\dagger m_{a^\prime}) \\
+ \beta_{ib} (m_{b}^\dagger m_{b} + m_{b^\prime}^\dagger m_{b^\prime}) + \beta_{ia} (m_{a}^\dagger m_{a} + m_{a^\prime}^\dagger m_{a^\prime}) \\
+ \beta_{rb} (m_{b}^\dagger m_{b} + m_{b^\prime}^\dagger m_{b^\prime}) + \beta_{ib} (m_{a}^\dagger m_{a} + m_{a^\prime}^\dagger m_{a^\prime}) \\
\] (2.5)

Our choice of using four individual site energies to represent each of the four vibrational states observed experimentally is dictated by the molecular structure and the photochemistry of these molecules which strongly suggest that the five terminal CN ligands are non-equivalent. In the above expression, \( m_i^\dagger \) and \( m_i \) are the creation and annihilation operators of each vibrational mode respectively, \( \varepsilon_i \) is the site energy of each mode, \( \Delta_i \) is the vibrational mode anharmonicity, and \( \beta_{ij} \) is the bilinear coupling constant between modes \( i \) and \( j \). Our third order nonlinear experiment is capable of looking at interactions up to the two-exciton state. Here, the vCN modes are assumed to be weakly coupled, weakly anharmonic oscillators. This is a reasonable assumption as it was observed above that the modes are indeed coupled and that the magnitudes of the anharmonicities are on the order of only \( \sim 1\% \) of the fundamental frequencies.

In order to determine the coupling constants, the above Hamiltonian is diagonalized and the first fifteen values of the trace are taken. These correspond to the zero, one, and two-quantum states in
the normal mode basis. These values are compared to the energy levels we determined from the experimental spectra. This transformation process and comparison is repeated iteratively to minimize the difference between the experimentally determined normal mode anharmonicities and frequencies and the calculated anharmonicities and frequencies. The site energies and coupling constants from these best fits are reported in Table 2.3.

**Table 2.3:** Best fit values for the local mode site energies, vibrational mode anharmonicities, and bilinear coupling constants for all three systems. The error bars are estimated to be ±2 cm⁻¹ for the site energies and ±1 cm⁻¹ for the anharmonicities and coupling constants.

<table>
<thead>
<tr>
<th>(cm⁻¹)</th>
<th>FeRu/FA</th>
<th>FeRu/D₂O</th>
<th>FePtFe/D₂O</th>
</tr>
</thead>
<tbody>
<tr>
<td>εₜ′</td>
<td>2004</td>
<td>2032</td>
<td>2062</td>
</tr>
<tr>
<td>εₜ′</td>
<td>2052</td>
<td>2044</td>
<td>2051</td>
</tr>
<tr>
<td>εₜ′</td>
<td>2062</td>
<td>2057</td>
<td>2071</td>
</tr>
<tr>
<td>εₜ′</td>
<td>2087</td>
<td>2091</td>
<td>2113</td>
</tr>
<tr>
<td>Δₜ</td>
<td>18</td>
<td>21</td>
<td>22</td>
</tr>
<tr>
<td>Δₜ′</td>
<td>21</td>
<td>21</td>
<td>22</td>
</tr>
<tr>
<td>Δₜ′</td>
<td>20</td>
<td>21</td>
<td>26</td>
</tr>
<tr>
<td>Δₜ′</td>
<td>21</td>
<td>21</td>
<td>25</td>
</tr>
<tr>
<td>βₜₜ′</td>
<td>6</td>
<td>11</td>
<td>3</td>
</tr>
<tr>
<td>βₜₜ′</td>
<td>5</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>βₜₜ′</td>
<td>6</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>βₜₜ′</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>βₜₜ′</td>
<td>4</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>βₜₜ′</td>
<td>6</td>
<td>9</td>
<td>7</td>
</tr>
</tbody>
</table>

The site energies do not seem to shift far from their normal mode positions with the largest change being a 6 cm⁻¹ shift for the νₜ mode in FeRu in D₂O which had the largest changes for all modes overall. We also see that all the νCN modes have similar local mode vibrational anharmonicities. They range from 18 cm⁻¹ for the νₜ mode in FeRu in FA to 26 cm⁻¹ for the νₚ mode.
mode in FePtFe. The values of the coupling constants indicate that all of the $\nu_{\text{CN}}$ modes are weakly coupled to each other in the site basis showing that the vibrations are fairly localized on the CN ligands. The strongest coupling in the local mode basis set seems to occur between the $\nu_{t'}$ and $\nu_{b'}$ modes and the $\nu_{a'}$ and $\nu_{b'}$ modes. We also note that the $\nu_{r'}$ mode does not couple strongly with the $\nu_{a'}$ or $\nu_{b'}$ modes.

The site Hamiltonian calculated using Equation 2.5 agrees with previous transient IR work and resonant Raman work on FeRu and FePtFe which have revealed that the vibrational modes lying parallel to the charge transfer axis are crucial players in the photochemistry of these systems following MMCT excitation.$^{20,21}$ We have previously reported that the energy splitting between the trans and bridge modes in FePtFe decreases by about $\sim 7 \text{ cm}^{-1}$ in the ground electronic state following 400 nm MMCT excitation and back–electron transfer.$^{20}$ To see the effect that MMCT excitation has on the bilinear coupling of the trans and bridge modes in the local site basis, we move the frequency of the $\nu_{t}$ mode up by $7 \text{ cm}^{-1}$ and rerun the fitting routine described above making the assumption that the other modes and their anharmonicities have not changed. We find that in the electronic ground state the bilinear coupling constant between the bridge and trans mode, in the site basis is $8 \text{ cm}^{-1}$, but after photoexcitation, $\beta_{t'b'}$ becomes $7 \text{ cm}^{-1}$. This indicates that the two modes are less strongly coupled immediately following photoexcitation. The slight change in coupling could arise from differences in molecular structure and/or differences in anharmonic vibrational couplings to the solvent bath which could include other intramolecular and intermolecular vibrational modes.

2.4 SUMMARY

This study elucidates the role of the solvent in modulating the molecular structure and vibrational anharmonic couplings between the high frequency CN stretching modes in the ground
electronic state of bimetallic and trimetallic transition metal mixed valence complexes. We measure anharmonic couplings and angles between transition moment dipoles of the $\nu_{\text{CN}}$ modes in FeRu in FA and D$_2$O and FePtFe in D$_2$O by simulating and fitting experimental 2D IR spectra in two different polarization geometries. We stress that the accurate determination of these parameters would not have been possible without rigorously fitting the spectra due to overlapping vibrational modes and large linewidths relative to the vibrational anharmonicities. The values for anharmonic couplings reveal that all four modes are weakly coupled with each other in FeRu and FePtFe. The radial mode seems to be the least coupled to the other three vibrational modes which is attributed to the fact that it does not lie along the main MMCT axis. Previous results have shown that the modes lying along the charge transfer axis are involved in the ultrafast charge and energy transfer following MMCT excitation. The weak coupling of the radial mode to the trans, axial, and bridge modes explains why it is the least coupled to the charge transfer excitation. The measured angles reveal that the solvent plays a large role in determining the molecular geometry of the mixed valence complexes. FeRu in FA appears almost perfectly linear while both FeRu in D$_2$O and FePtFe in D$_2$O appear bent. The modulation of the molecular geometry and vibrational anharmonic couplings due to specific solute-solvent interactions in the ground electronic state help explain why the coupling of high frequency CN modes to the MMCT transition is a strong function of the solvent environment. These findings hold implications for designing new cyanide-bridged transition metal complexes and polymers and for effectively simulating the spectroscopic properties of these solvated molecules.
REFERENCES


Chapter 3:
Investigating Vibrational Relaxation In Cyanide-Bridged Transition Metal Mixed Valence Complexes Using Two-Dimensional Infrared And Infrared Pump-Probe Spectroscopies

The work presented in this chapter is in preparation for publication:

Using polarization-selective two-dimensional infrared (2D IR) and infrared pump-probe spectroscopies, we study vibrational relaxation of the four cyanide stretching ($\nu_{CN}$) vibrations found in $[(\text{NH}_3)_5\text{Ru}^{III}\text{Fe}^{II}(\text{CN})_5]^-$ (FeRu) dissolved in D$_2$O or formamide and $[(\text{NC})_5\text{Fe}^{II}\text{CNPt}^{IV}(\text{NH}_3)_4\text{NCFe}^{II}(\text{CN})_2]^4^-$ (FePtFe) dissolved in D$_2$O. These cyanide-bridged transition metal complexes serve as models for understanding the role high frequency vibrational modes play in metal-to-metal charge transfers over a bridging ligand, but there is currently little information about their electronic ground state. IR pump-probe experiments reveal that the vibrational lifetimes of the radial, axial, and trans $\nu_{CN}$ modes are ~2 times faster when FeRu is dissolved in D$_2$O versus formamide. They also reveal that the vibrational lifetimes of the $\nu_{CN}$ modes of FePtFe in D$_2$O are almost four times as long as for FeRu in D$_2$O indicating that the lifetimes are not strongly dependent on solvent, higher molecular complexity, or larger overall charge as these factors would all result in faster relaxation times for FePtFe. Combined with mode-specific relaxation dynamics measured from the 2D IR experiments, IR pump-probe experiments
also reveal that intramolecular vibrational relaxation is occurring in all three systems on \(~1\) ps timescale. Center line slope (CLS) dynamics, which have been shown to be a measure of the frequency-frequency correlation function, reveal that the radial, axial, and trans \(\nu_{CN}\) modes exhibit a \(~3\) ps timescale that the bridge \(\nu_{CN}\) mode does not display. This timescale is attributed to the forming and breaking of hydrogen bonds between each mode and the solvent. In contrast to previous work done by our group, there does not appear to be a large solvent dependence for vibrational relaxation dynamics in these systems.

3.1 INTRODUCTION

Vibrational relaxation dynamics reflect differences in the local environment and therefore depend on specific interactions between solvent molecules and the local vibrational probe. Changing dynamics between systems, or even different vibrational modes spanning different regions in a molecule, highlight how excess vibrational energy is transferred from a vibrational mode to its environment. Energy dissipation has implications for chemical reactivity and the mechanisms by which those reactions occur. In polyatomic molecules in solution, this energy dissipation can occur by intramolecular vibrational relaxation (IVR), where the energy is transferred to other modes within the same molecule, or by direct relaxation to solvent modes. Rates of relaxation can be affected by factors that include the frequency of the vibrational mode, the size, structure, and charge of the solute molecule, and similar features of the solvent. Ultrafast spectroscopy is the most common technique used to study vibrational relaxation in the condensed phase.

Cyanide-bridged transition metal mixed valence complexes serve as model systems to study charge transfer reactions. High-frequency vibrational modes in these systems function as a probe of the interaction between vibrational and electronic motions following photoinduced electron
transfer reactions. Previous studies of cyanide-bridged mixed valence complexes have shown high degrees of excitation, in some cases greater than 6 quanta, in high frequency CN stretching ($\nu_{CN}$) vibrations following back electron transfer subsequent to the metal-to-metal charge transfer (MMCT) reaction,\textsuperscript{1-6} but little is known of vibrational dynamics on the electronic ground state. The goal of this chapter is to determine what role solvent and other molecular properties play in the vibrational relaxation of cyanide-bridged bimetallic and trimetallic transition metal mixed valence complexes. Infrared pump-probe and two-dimensional infrared spectroscopies are used to measure ground state vibrational dynamics of the cyanide stretching vibrations of $[(\text{NH}_3)_5\text{Ru}^{\text{III}}\text{NCFe}^{\text{II}}(\text{CN})_5]^- \text{(FeRu)}$ dissolved in deuterium oxide (D$_2$O) or formamide (FA) and $[(\text{NC})_3\text{Fe}^{\text{II}}\text{CNPt}^{\text{IV}}(\text{NH}_3)_4\text{NCFe}^{\text{II}}(\text{CN})_5]^{4+} \text{(FePtFe)}$ dissolved in D$_2$O. FeRu and FePtFe both have four cyanide stretching frequencies that are labelled $\nu_{\text{trans}} (\nu_t)$, $\nu_{\text{radial}} (\nu_r)$, $\nu_{\text{axial}} (\nu_a)$, and $\nu_{\text{bridge}} (\nu_b)$ (illustrated in Fig. 3.1a,b) of which three ($\nu_t$, $\nu_a$, and $\nu_b$) lie approximately along the MMCT axis.\textsuperscript{7}

The FTIR spectra of the $\nu_{CN}$ modes (see Fig. 3.1c-e) of the bimetallic and trimetallic systems are strikingly different in terms of frequencies, amplitudes, and lineshapes. The four mode frequencies for the FeRu/FA system are centered at 2002, 2051, 2065, and 2089 cm$^{-1}$ for the $\nu_t$, $\nu_r$, $\nu_a$, and $\nu_b$ modes respectively. For the FeRu/D$_2$O system, they are centered at 2026, 2048, 2061, and 2096 cm$^{-1}$, and for FePtFe, they are centered at 2059, 2049, 2072, and 2116 cm$^{-1}$ for the $\nu_t$, $\nu_r$, $\nu_a$, and $\nu_b$ modes respectively. The higher oxidation state of Pt(IV) compared to Ru(III) explains why the frequency of the bridging mode is farther to the blue for FePtFe than for FeRu. The larger positive charge pulls more electron density out of the cyanide antibonding ($\pi^*$) orbital and increases the strength of the CN bond thus increasing its vibrational frequency.\textsuperscript{8-10} If the electron withdrawing effect carries along the MMCT axis, the strength of the bond could also increase in the trans and
axial cyanide ligands explaining why the \( \nu_t \) mode in FePtFe is at much higher frequencies than it is in FeRu and why the frequency ordering of the four \( \nu_{CN} \) modes change between systems.

Previous work by our group has shown the experimental 2D IR spectra of FeRu and FePtFe and their fits reveal a set of weakly coupled anharmonic \( \nu_{CN} \) modes. The vibrational mode anharmonicities of the individual \( \nu_{CN} \) modes range from 14 to 28 cm\(^{-1}\), and the mixed-mode anharmonicities range from 2 to 14 cm\(^{-1}\). In general, the bridging \( \nu_{CN} \) mode is most weakly coupled to the radial \( \nu_{CN} \) mode, which involves the terminal CN ligands approximately orthogonal to the main MMCT axis. Measurement of the relative transition dipole moments of the four \( \nu_{CN} \) modes reveal that the angles in FeRu in FA are significantly different than both FeRu in D\(_2\)O and FePtFe in D\(_2\)O suggesting structure might be a function of solvent environment.\(^7\) Here we will determine if the solvent plays a similar role in the vibrational dynamics in these systems.

The remainder of this chapter is organized as follows. Section 3.2 describes the methods used to prepare samples and collect both pump-probe and 2D IR spectra. Section 3.3 is split into two subsections. Section 3.3.1 describes the IR pump-probe fitting and results. Section 3.3.2 contains
the results of the 2D IR experiments including spectral diffusion and mode-specific dynamics. Section 3.4 contains the discussion of our results and is split into two subsections. Section 3.4.1 discusses vibrational relaxation and Section 3.4.2 discusses spectral diffusion dynamics. The final section, Section 3.5, contains the summary of the results and our conclusions.

3.2 METHODS

Materials. Starting materials and solvents were purchased from Sigma Aldrich and used without further purification. The mixed-valence complexes, FeRu and FePtFe, were synthesized as described previously.\textsuperscript{7,11-13} A saturated solution of FeRu was prepared in formamide (FA) with a maximum optical density (OD) of 0.22 in the $\nu_{CN}$ region after solvent subtraction. A solution of FeRu was also prepared in D\textsubscript{2}O to a concentration of ~25 mM and maximum OD of 0.36 in the $\nu_{CN}$ region. A solution of FePtFe dissolved in D\textsubscript{2}O was prepared to a concentration of ~15 mM and a maximum OD of 0.45 for use in experiments.

2D IR Experiments. The experimental parameters and layout of the 2D IR experiments have been described in detail in Ref. 7. 2D IR spectra of FeRu in FA were collected at the following $\tau_2$ delays (waiting times): 0.07, 0.14, 0.225, 0.3, 0.425, 0.475, 0.625, 1, 2.5, 5, 7.5, 10, 15, and 20 ps. 2D IR spectra of FeRu/D\textsubscript{2}O were collected at $\tau_2$ delays of 0.09, 0.15, 0.225, 0.3, 0.375, 0.65, 1, 3, 5, 7.5, 10, and 15 ps, and 2D IR spectra of FePtFe dissolved in D\textsubscript{2}O were taken at $\tau_2$ delays of 0.15, 0.34, 0.38, 1.36, 2.04, 5, 7.5, 10, and 15 ps. For each sample, 2D IR spectra were collected in both parallel ($S_||$) and crossed ($S_\perp$) polarization geometries. Magic angle spectra were constructed using the relationship: $S_\text{m} = \frac{1}{3}(S_|| + 2S_\perp)$.

IR Pump-Probe Experiments. The generation and manipulation of the mid-IR light pulses used in the IR pump-probe experiments was similar to the 2D IR experiments and has been
described earlier. The pump beam (chopped at 500 Hz) and the probe beam were spatially and temporally overlapped in the sample to generate the pump-probe signal. An additional mid-IR pulse was also sent through the sample to act as a reference beam. The input pump and probe beams had energies of ~0.5 and ~0.3 μJ/pulse for all three samples. All input beams were focused onto the sample with a spot size (1/e^2) of ~150 μm. The pump-probe signal and the reference beam were vertically offset before being dispersed by a spectrometer (Triax 190, Horiba Jobin Yvon) and detected on the upper and lower stripes of a 2×64 element mercury cadmium telluride array detector (IR0144, Infrared Systems Development).

Pump-probe spectra for the FeRu samples were collected at magic angle where the polarization of the probe beam was rotated 54.7° relative to the polarization of the pump beam to eliminate contributions from rotational dynamics. Collected data were generated by dividing both the pump on and pump off signals by their respective reference pulses and then subtracting the pump on signal from the pump off signal. Dividing each signal pulse by the reference helped mitigate the effects of shot-to-shot and long term noise fluctuations in the mid-IR field. The reference beam was set to come about 150 ps before the pump beam to ensure the sample had relaxed back to the ground state. For FeRu dissolved in FA and D₂O, the time delay between the pump and probe pulses was scanned from -150 to 150 fs in 10 fs steps, 150 to 2500 fs in 25 fs steps, 2.5 to 5 ps in 50 fs steps, 5 to 8 ps in 250 fs steps, 8 to 20 ps in 1 ps steps, 20 to 40 ps in 4 ps steps, and 40 to 100 ps in 10 ps steps. Each data point represents 1500 shots, and each spectrum was collected 100 times and averaged to obtain the experimental spectra presented here. The experimental spectral resolution is ~2.8 cm⁻¹ for the FeRu samples.

Spectra for FePtFe were collected in both parallel (\(S_\parallel\)) and crossed (\(S_\perp\)) polarization geometries and no reference beam was used for shot-to-shot normalization of the data. For FePtFe
dissolved in D₂O, the time delay between the pump and probe pulses was scanned from -1.5 to 3.5 ps in 25 fs steps, 3.5 to 6 ps in 250 fs steps, 6 to 20 ps in 1 ps steps, and 20 to 100 ps in 4 ps steps. Each data point represents 2000 shots, and each spectrum was collected 25 times and averaged to obtain the experimental spectra presented here. Magic angle spectra were constructed. Experimental spectral resolution is ~4.4 cm⁻¹ for the FePtFe sample.

3.3 RESULTS

**Table 3.1:** Parameters from the exponential fits to experimental pump-probe data. Error bars represent 95% confidence intervals.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Freq (cm⁻¹)</th>
<th>t₁ (ps) (A₁)</th>
<th>t₂ (ps) (A₂)</th>
<th>t₃ (ps) (A₃)</th>
<th>tᵣise (ps) (Aᵣise)</th>
<th>tosc (ps) (Aosci)</th>
<th>ωosc (cm⁻¹)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trans</td>
<td>2004</td>
<td>0.31±0.02</td>
<td>0.99±0.04</td>
<td>10.±1</td>
<td>0.2±0.1</td>
<td>50±10</td>
<td></td>
</tr>
<tr>
<td>FeRu/FA</td>
<td>Radial</td>
<td>2050</td>
<td>0.64±0.04</td>
<td>10.8±0.1</td>
<td>0.2±0.1</td>
<td>50±10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Axial</td>
<td>2064</td>
<td>9.32±0.09</td>
<td>0.7±0.3</td>
<td>90±4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bridge</td>
<td>2091</td>
<td>12.4±0.5</td>
<td>1.4±0.2</td>
<td>0.30±0.02</td>
<td>84±1</td>
<td></td>
</tr>
<tr>
<td>FeRu/D₂O</td>
<td>Trans</td>
<td>2028</td>
<td>0.42±0.03</td>
<td>5.8±0.05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Radial</td>
<td>2047</td>
<td>0.72±0.03</td>
<td>6.51±0.09</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Axial</td>
<td>2064</td>
<td>5.25±0.06</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bridge</td>
<td>2093</td>
<td>0.41±0.06</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FePtFe/D₂O</td>
<td>Trans</td>
<td>2060</td>
<td>0.85±0.07</td>
<td>17.2±0.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Radial</td>
<td>2050</td>
<td>0.5±0.1</td>
<td>19.4±0.2</td>
<td>0.34±0.06</td>
<td>33±4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Axial</td>
<td>2073</td>
<td>16.1±0.3</td>
<td>1.11±0.09</td>
<td>0.57±0.08</td>
<td>52±2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bridge</td>
<td>2115</td>
<td>18±1</td>
<td>0.82±0.08</td>
<td>0.46±0.02</td>
<td>53.6±0.6</td>
<td></td>
</tr>
</tbody>
</table>
3.3.1 IR Pump-Probe

The IR pump-probe spectra of FeRu/FA, FeRu/D₂O, and FePtFe/D₂O are shown in Figure 3.2. The time-dependence of the fundamental ν_{CN} frequencies of each sample are shown on semi-log plots. The data was fit to the following expression:

\[
\Delta A(\tau_2, \omega) = \left( \sum_i A_i \exp\left(-\frac{\tau_2}{t_i}\right) \right) + B \left(1 - \exp\left(-\frac{\tau_2}{t_{rise}}\right)\right) + C \exp\left(-\frac{\tau_2}{t_{osc}}\right) \cos(2\pi\omega_{osc}\tau_2 + \phi)
\]

(3.1)

where \( A_i(B) \) and \( t_i \) are the amplitude and timescale of each decay (rise) respectively, \( \omega_{osc} \) is the frequency of any oscillation that is present, and \( \phi \) is the phase of the oscillation. Because of large nonresonant signal, data begins at 200 fs and was only fit from this time forward. The best fit parameters are listed in Table 3.1.

**FeRu/FA.** Figure 3.2a displays \( \tau_2 \) dependent dispersed pump-probe traces of FeRu/FA in the right panel. The data reveal that the negative bleach signals (0→1 transitions) of the ν₁ (2051 cm⁻¹) and ν₉ (2065 cm⁻¹) modes are spectrally overlapped and the weaker axial mode appears as a shoulder to the blue side of the radial mode. It is also evident from Figure 3.2a that the ν₁ mode (2004 cm⁻¹) decays at a much faster rate than the other three modes. The positive absorption signals represent the 1→2 transitions to the various overtones and combination bands from the first excited vibrational states. We are not analyzing the 1→2 transitions due to the fact that they have highly overlapped contributions from both combination bands and overtones and therefore do not represent dynamics of any transition exclusively. The vibrational relaxation dynamics of the four fundamental transitions are plotted in the left panel of Figure 3.2a and the data is fit using Equation 3.1. The traces in Figure 3.2b and the best fit parameters listed in Table 3.1 reveal that the four fundamental ν_{CN} modes display different vibrational relaxation dynamics. The ν₁ mode.
decays on two main timescales: 0.31 and 0.99 ps. We attribute the faster timescale to intramolecular vibrational relaxation (IVR) processes where the vibrational energy leaks into other high-frequency $\nu_{CN}$ modes or low-frequency vibrational modes along the metal-ligand backbone. The 0.99 ps decay timescale is attributed to vibrational energy relaxation to the solvent from the

**Figure 3.2:** The two panels of (a) represent spectra (left) of FeRu in FA at several $\tau_2$ delays and time traces (right) of the same signal for select frequencies shown as dashed lines in the left panel. Note that the time axis follows a logarithmic scale. (b) and (c) display the same information for FeRu in D$_2$O and FePtFe in D$_2$O, respectively, as described in (a). Note that the bridge trace for each sample is multiplied by a static factor to make it easier to compare to the other modes.
excited νt mode. The fit of the νr mode reveals three timescales that consist of two exponential decays and a decaying cosine. We attribute the 10.8 ps decay to the vibrational population lifetime of the radial mode. The faster 0.64 ps decay is attributed to cross-population relaxation processes to other intramolecular vibrational modes. The oscillation at ~50 cm⁻¹ observed in the relaxation dynamics of the νr mode results from the coherent superposition of two coupled νCN modes. The amplitude of the oscillatory component is proportional to the anharmonic coupling and the transition dipole moment between coupled modes. The frequency difference between the radial and trans modes and between the radial and bridge modes is ~50 and ~40 cm⁻¹ respectively. We have previously measured similar anharmonic couplings and angles between the radial and trans modes and between radial and bridge modes suggesting that the oscillation arises from both of these coherent contributions. In addition, clear cross-peaks between the νt and νr and νt and νb modes can be seen in Figure 3.3a at ω3 = νr. The fit of the νa mode (2064 cm⁻¹) shows two timescales consisting of an exponential decay and a very weak oscillatory component. The 9.32 ps timescale is the vibrational lifetime of the axial mode while the faster component is attributed to IVR processes. The weak oscillatory component at ~90 cm⁻¹ likely results from spectral overlap with combination bands. The trace of the νb mode (2091 cm⁻¹) displays an exponential decay, a growth, and a decaying oscillation at ~84 cm⁻¹. The oscillation matches the difference in frequency between νt and νb indicating a superposition between the two modes which have parallel transition dipole moments. The decay of 12.4 ps is the vibrational lifetime of the bridge mode, and the growth at 1.4 ps is attributed to vibrational energy relaxation from the other three νCN modes. In summary, we note that the radial, axial, and bridge modes of FeRu/FA have vibrational lifetimes ranging from 9-12 ps. The trans mode is an outlier with a vibrational lifetime of ~1 ps. The lifetimes of the coherent oscillations found in the νr, νa, and νb are less than 1 ps.
**FeRu/D₂O.** Figure 3.2b shows the dispersed pump-probe spectra of the FeRu/D₂O sample as a function of ω₃ and τ₂, and the fit values for the fundamental traces can be seen in the FeRu/D₂O section of Table 3.1. We note that the νᵣ (2028 cm⁻¹), νᵣ (2047 cm⁻¹), and νₐ (2064 cm⁻¹) modes are spectrally overlapped and the νₐ mode (2093 cm⁻¹) is extremely weak. In contrast to the pump-probe data for FeRu in FA, the traces of FeRu in D₂O show no rises or oscillations. The fit of the νᵣ mode has two timescales, where the short timescale is attributed to IVR processes, and the 6.51 ps timescale is attributed to vibrational energy relaxation to the solvent. The νₐ mode only decays on a timescale of 5.25 ps, which represents its vibrational population lifetime. The fit of the νₐ mode consists of two exponential decays that have amplitude factors of opposite signs to the rest of the fundamental transitions. The fundamental transition for the νᵣ mode is overlapped with overtones of other modes as well as contributions from combination bands which results in the positive signal. This makes it difficult to attribute a vibrational population lifetime to the trans mode. Given that the vibrational lifetimes of the νᵣ and νₐ modes are approximately half of their measured values in the FeRu/FA sample, we tentatively assign the 0.42 ps decay to be the vibrational lifetime of the νᵣ mode. Since the νₐ mode region of the spectrum has such a low signal level, fit values for that trace likely have no physical meaning. In summary, we note that the radial and axial modes of FeRu/D₂O have lifetimes of ~6 ps, and the trans mode is again an outlier at ~400 fs. These timescales are twice as fast as those seen in FeRu/FA.

**FePtFe/D₂O.** Figure 3.2c shows τ₂ dependent dispersed pump-probe spectra of FePtFe and ω₃ dependent vibrational relaxation dynamics. The νᵣ (2050 cm⁻¹), νᵣ (2060 cm⁻¹), and νₐ (2073 cm⁻¹) modes are spectrally overlapped, but their individual contributions can be distinguished as shoulders in the strong bleach contribution in the dispersed pump-probe spectra. Similar to the FeRu sample dissolved in FA, we see that three of the modes in FePtFe have oscillations, but here
two modes have visible rises. From the fit values seen in Table 3.1, we can see that all four modes share a similar long time constant of ~18 ps representing vibrational population lifetimes for each mode. The $v_t$ mode does not display a coherent oscillation, but the fit shows a second decay in addition to its vibrational lifetime which is attributed to relaxation processes to other intramolecular vibrational modes. The fit for the $v_r$ mode also shows two exponential decays representing IVR relaxation and vibrational population relaxation to the solvent. Additionally, the $v_t$ mode displays an oscillation with a frequency of 33 cm$^{-1}$. Like seen in the FeRu/FA sample, this oscillation probably arises from contributions of both a $v_r/v_t$ superposition, where the difference in frequency is 10 cm$^{-1}$, and a $v_r/v_b$ superposition, where the difference in frequency is 67 cm$^{-1}$. A clear cross-peak can be seen between $v_t$ and $v_b$ in Figure 3.3c at $\omega_3 = v_t$ while the cross-peaks between $v_t$ and $v_t$ are not as easily distinguished due to overlap with the $v_t$ and $v_t$ diagonal peaks. The fit for the $v_a$ mode shows one exponential decay representing the vibrational lifetime, a rise, and a decaying oscillation. The rise of ~1.1 ps is attributed to energy relaxation from the other three $v_{CN}$ modes into the axial mode. A similar rise of ~800 fs is seen in the fit for the $v_b$ mode (2115 cm$^{-1}$) which also shows one exponential decay and a coherent oscillation. The oscillation frequency of ~54 cm$^{-1}$ is approximately the split in frequency between the $v_t$ and $v_b$ modes indicating a superposition between the two modes like that seen in the bridge mode of FeRu/FA. In summary, we note that all four $v_{CN}$ modes of FePtFe/D$_2$O have similar vibrational lifetimes ranging from 16-19 ps. This is in contrast to FeRu where both samples had significantly shorter lifetimes for the trans mode. The lifetimes of the coherent oscillations found in the $v_r$, $v_a$, and $v_b$ modes of FePtFe are ~500 fs.
3.3.2 2D IR Relaxation Experiment

*FeRu/FA.* Figure 3.3 displays representative crossed polarization 2D IR plots as a function of $\tau_2$ for all three samples. Note that contour levels do not follow a linear scale. The 2D IR data for FeRu in FA is shown in Figure 3.3a. There are three main features along the diagonal with positive (0→1 transitions) and negative (1→2 transitions) features. The spectral feature at lowest energy represents the $v_t$ mode, the feature in the middle of the spectrum represents overlapping contributions from the $v_r$ and the $v_a$ modes as well as their cross-peaks, and the one at highest energy represents the $v_b$ mode. It can be seen that the trans mode decays away relatively quickly compared to the other features. This is due to the faster vibrational relaxation time for the $v_t$ mode compared to the other $v_{CN}$ modes (see Table 3.1). The time-evolving 2D lineshapes and amplitudes measure the spectral diffusion and vibrational relaxation dynamics of the $v_{CN}$ modes in FeRu/FA. The 2D spectral feature of the $v_t$ mode has the largest anti-diagonal width and maintains its spectral elongation until after the time its amplitude has fully decayed ($\tau_2 \approx 5$ ps). The 2D peak representing both the radial and axial modes is elongated along the diagonal at $\tau_2 = 70$ fs and its spectral diffusion dynamics are complete by $\sim 5$ ps. In contrast to the $v_r$ and $v_a$ diagonal peaks, the 2D peak of the bridge mode does not appear to have any elongation along the diagonal axis at early waiting times. The diagonal and cross-peaks are amplitude modulated as a function of $\tau_2$ in a 2D IR spectrum of coupled vibrational modes. Oscillatory behavior is clearly seen for the $v_b$ mode in the 2D IR spectra where the diagonal peak has more amplitude at 300 fs than it does at 70 fs. The beats are also seen in the pump-probe data (Fig. 3.2a) which shows a clear oscillation at a frequency of 84 cm$^{-1}$. Recall that the dispersed pump-probe data at a particular $\tau_2$ delay is the projection of the 2D IR spectrum along $\omega_3$. Evidence of intramolecular vibrational energy redistribution among the $v_{CN}$ modes of FeRu/FA can be observed by the growth of three cross
peaks at \( (\omega_1 = \nu_r, \omega_3 = \nu_b) \), \( (\omega_1 = \nu_t, \omega_3 = \nu_b) \) and \( (\omega_1 = \nu_b, \omega_3 = \nu_r/\nu_a) \) relative to the radial diagonal peak. The other two obvious cross-peaks at \( (\omega_1 = \nu_r, \omega_3 = \nu_t) \) and \( (\omega_1 = \nu_t, \omega_3 = \nu_r) \) appear to decay at similar rates to the \( \nu_r \) and \( \nu_a \) modes.

**FeRu/D\(_2\)O.** The 2D IR spectra for FeRu in D\(_2\)O are shown in Figure 3.3b and look very different from the 2D IR spectra of FeRu in FA. The FTIR spectrum of FeRu/D\(_2\)O shows significant overlap...
of the \( \nu_t \), \( \nu_r \), and \( \nu_a \) modes and very weak amplitude in the \( \nu_b \) mode. The resultant 2D spectra of FeRu in \( \text{D}_2\text{O} \) show one main peak that represents overlapping contributions from the \( \nu_r \) and \( \nu_a \) modes as well as their cross peaks. At the lower left-hand corner of the central peak, there is a small feature that quickly disappears indicating the \( \nu_t \) mode whose vibrational lifetime is believed to be \( \sim \)400 fs (Table 3.1). A bridge diagonal peak is not distinguishable in any of the spectra due to the \( \nu_b \) mode’s very small transition dipole moment \( (|\mu_b| = 0.89|\mu_r|, |\mu_a| = 0.40|\mu_r|, \text{ and } |\mu_b| = 0.12|\mu_r|) \) as diagonal peak amplitudes scale with \( \mu_p^4 \). The central diagonal peak loses its spectral correlation and appears fully rounded by \( \sim \)5 ps. We note that it is difficult to disentangle the relative homogenous and inhomogeneous contributions from the radial and axial cross peak contributions. There are two cross peaks that show up as wings to the lower and higher energy sides of the central peak \( (\omega_1 = \nu_b/\nu_r, \omega_3 = \nu_r) \). The 2D spectra reveal a spectral feature growing in at higher \( \omega_3 \) frequencies for \( \tau \geq 3 \) ps. We attribute this to a solvent feature as there is a large \( \text{D}_2\text{O} \) band centered at \( \sim 2500 \text{ cm}^{-1} \). Temperature dependent FTIR spectra of \( \text{D}_2\text{O} \) show a growth in the transmission from \( \sim 2100 \) to \( 2150 \text{ cm}^{-1} \) as the temperature increases. In the 2D spectra, we see that the amplitude in the same spectral region increases.

\textbf{FePtFe/\text{D}_2\text{O}.} The 2D IR spectra for FePtFe are shown in Figure 3.3c. Here we see two main diagonal features, the central one representing overlapping contributions from the \( \nu_r \), \( \nu_t \), and \( \nu_a \) modes as well as their respective cross peaks, and the higher energy peak representing the \( \nu_b \) mode. The central feature starts off elongated along the diagonal with similar homogeneous width for all three modes (see 2D IR plot at \( \tau_2 = 0.15 \) ps) and the spectral diffusion dynamics are complete by 5 ps. Like seen in the 2D spectra for FeRu/FA, the 2D spectral feature of the bridge mode does not appear to have any elongation along the diagonal at early \( \tau_2 \) delays. The diagonal bridge peak exhibits amplitude beating with a time-period of 620 fs consistent with the beats seen in the
dispersed IR pump-probe data (Fig. 3.2c). We see that the obvious cross peaks, \((\omega_1 \approx \nu_r, \omega_3 = \nu_b)\) and \((\omega_1 = \nu_b, \omega_3 \approx \nu_t)\) grow in relative to the diagonal features and are most visible in the \(\tau_2 = 10\) ps 2D IR spectrum. Below we will discuss the spectral diffusion and vibrational relaxation dynamics measured in the 2D IR spectra.

3.3.2.1 Spectral Diffusion

Table 3.2: Exponential fit values to CLS traces. Error bars represent 90% confidence intervals.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Mode (t_1) (ps) (A_1)</th>
<th>Mode (t_2) (ps) (A_2)</th>
<th>(\omega_{osc}) (cm(^{-1}))(^a)</th>
<th>(A_\infty)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trans</td>
<td>4 ± 2 (0.31)</td>
<td>20 ± 10 (0.23)</td>
<td>0.07 ± 0.05</td>
<td></td>
</tr>
<tr>
<td>FeRu/FA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Radial</td>
<td>2 ± 1 (0.23)</td>
<td>0.30 (0.3)</td>
<td>84</td>
<td></td>
</tr>
<tr>
<td>Bridge</td>
<td>20 ± 40 (0.21)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FeRu/D(_2)O</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Radial</td>
<td>3 ± 1 (0.26)</td>
<td>0.6 ± 0.7 (0.12)</td>
<td>0.09 ± 0.03</td>
<td></td>
</tr>
<tr>
<td>Axial</td>
<td>4 ± 4 (0.09)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FePtFe</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Radial</td>
<td>3 ± 3 (0.83)</td>
<td>0.1 ± 0.3 (0.19)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Axial</td>
<td>2 ± 1 (0.19)</td>
<td>0.04 ± 0.02 (0.16)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bridge</td>
<td>11 ± 9 (0.16)</td>
<td>0.46 (0.7)</td>
<td>54</td>
<td></td>
</tr>
</tbody>
</table>

\(^a\)Oscillation frequencies are held constant at their respective pump-probe oscillation frequency.

Transition frequency fluctuations arise from the relative movements of molecules’ positions and orientations. Vibrational correlation functions can be used to characterize these fluctuations, and as such, these correlation functions are a measure of the system-bath interaction for each vibration. This gives us a tool to look at solvation dynamics. Timescales for spectral diffusion, a measure of the frequency-frequency correlation function (FFCF), can vary greatly between similar systems. There are various metrics available to extract FFCF timescales, but here we use the center line
slope (CLS) method to measure spectral diffusion of the CN stretches.\textsuperscript{14} CLS is one way to quantify how elongated the peak is along the diagonal at a particular $\tau_2$ delay. To accomplish this, several $\omega_1$ frequencies centered on the fundamental frequency of the vibrational mode are chosen, and the $\omega_3$ frequency at which maximum amplitude is reached is recorded for each $\omega_1$ frequency. These points determine a line on an $\omega_1$ versus $\omega_3$ plot, and the slope of that line is the CLS. This slope is calculated for each $\tau_2$, and the resulting decay is fit to the following expression (results in Table 3.2) to find timescales that are proportional to the FFCF:

$$\text{CLS}(\tau_2; \omega_1) = \sum_i A_i \exp\left(-\frac{\tau_2}{t_i}\right) + A_{\text{osc}} \exp\left(-\frac{\tau_2}{t_{\text{osc}}}\right) \cos(2\pi \omega_{\text{osc}} \tau_2 + \phi) + A_{\infty}$$

(3.2)

where $A$ and $t$ are the amplitude and timescale of each decay respectively, $\omega_{\text{osc}}$ is the frequency of the oscillation, and $\phi$ is the phase of the oscillation. The term $A_{\infty}$ represents a static component, dynamics that do not occur within the timescale of this experiment. The cosine is only included in the fit if there is an oscillatory component present, and the $t_{\text{osc}}$ and $\omega_{\text{osc}}$ variables are held constant at the values determined in the IR pump-probe fits at that $\omega_3$. Other oscillations that have appeared in peak shift measurements, another metric of the FFCF, have been attributed to underdamped hydrogen bond oscillation in HOD/D$_2$O,\textsuperscript{15} anharmonic coupling of a high frequency NH stretching vibration to low frequency interdimer modes that modulate the NH hydrogen-bonding strength,\textsuperscript{16} and coupling of the OH stretch to underdamped intermolecular coordinates HOD in D$_2$O.\textsuperscript{17}

Figure 3.4 shows the CLS decays determined from the parallel polarization 2D IR spectra for selected modes from each system. The $v_{\text{CN}}$ modes for which CLS analysis was performed were chosen carefully to minimize contributions from other close lying modes. Large error bars are due
to the limited number of $\tau_2$ points available for each sample. The CLS of the $v_r$ mode is the only one that can be compared between all three samples. The two CLS traces from the D$_2$O samples of both FeRu and FePtFe, have a short time constant around 3 ps and an offset that represents dynamics too long to be measured with the time resolution of this experiment. The FeRu/FA $v_r$ trace has two time constants and no offset. It still has a short time constant at 2 ps that is close to the D$_2$O samples, but its long time constant is ~20 ps. For the $v_a$ mode, only the two D$_2$O samples

**Figure 3.4:** Plots of the center line slope (CLS) measuring spectral diffusion as a function of $\tau_2$ for selected modes from each sample. Error bars represent 90% confidence intervals on the CLS value for each $\tau_2$ point. (a) and (b) are the $v_t$ and $v_r$ modes, respectively, for FeRu/FA. (c) and (d) are the $v_a$ and $v_r$ modes, respectively, for FeRu/D$_2$O. (e) and (f) are the $v_a$ and $v_b$ modes, respectively, for FePtFe.
can be compared. FeRu/D$_2$O has two timescales present in its fit: the shortest decay constant measured at $\sim$600 fs and a longer one at $\sim$4 ps that is close to the $\sim$3 ps decays seen in the $\nu_r$ mode. The FePtFe $\nu_a$ trace has one time constant at $\sim$2 ps and an offset. With the $\nu_b$ mode traces, we cannot compare across molecule or solvent due to low signal in the bridge region of the 2D spectra for FeRu/D$_2$O (Fig. 3.3b). The fits for the $\nu_b$ traces for the FeRu/FA and FePtFe samples reveal a matching decay time constant of $\sim$20 ps fit to them while keeping the decay for the beat constant. This long decay also matches the long time constant seen in the $\nu_r$ mode of FeRu/FA. Only the trans mode of the FeRu/FA sample was fully distinguishable from the other modes due to a lack of overlapping spectral features. This only allowed a determination of a CLS trace for one $\nu_t$ mode. Its one decay constant of $\sim$4 ps is close to the $\sim$3 ps time constant we see in all CLS fits that do not contain an oscillation.

3.3.2.2 Mode-Specific Vibrational Relaxation Dynamics

To look at the dynamics of specific modes, a small area of each 2D spectrum was chosen for each peak and all signal intensity within that area was summed. These volumes were then plotted as a function of $\tau_2$ to see peak specific relaxation. Each of these decays was then fit to a sum of exponentials like the formula used to fit the pump-probe traces. Because of the low number of $\tau_2$ points available, these fits have very large uncertainties and only a few fits are shown in Figure 3.5 for illustration. These fits are discussed in a qualitative manner below.

Several illuminating time-dependent cross-peak volume plots for FeRu in FA show growths in addition to their decays (Fig. 3.4, left panel). These rises indicate population transfer between modes on the order of one picosecond. Most of these cross-peaks also have multiple decay timescales with the longest timescales on the order of the vibrational lifetimes of the $\nu_{CN}$ modes. The mode-specific dynamics of FeRu in D$_2$O do not contain as many growths as the FA sample,
but the ones that are observed are also on ~1 ps timescales. Data for FePtFe is similar to that seen for FeRu in FA with multiple rises on the order of 1 ps, several decays, and long timescales on the order of the vibrational lifetimes for the $\nu_{CN}$ modes. All of the short decay timescales seen in the pump-probe match the ~1 ps rises seen in the 2D volume mode-specific relaxation dynamics giving more strength to the argument that IVR dynamics are happening in all three systems on this timescale.

3.4 DISCUSSION

Our results show the vibrational relaxation in the dephasing dynamics are mode dependent in the solvated molecules. Vibrational lifetimes change significantly between modes within the same system and between the same modes in different systems. Long spectral diffusion timescales also show large variations between the same mode in different systems and different modes within the same system. In this section, we will put this work in the context of other work done with CN stretches.
3.4.1 Vibrational Relaxation Comparison

The IR pump-probe experiments reveal that the vibrational lifetimes of the radial and axial modes are ~2 times faster when FeRu is dissolved in D$_2$O versus FA. This is likely due to the much larger overlap between the CN and D$_2$O IR bands than exists between the CN and FA bands. The D$_2$O band centered at ~2500 cm$^{-1}$ is much broader (~500 cm$^{-1}$) and more intense than the closest FA band (centered at ~2210 cm$^{-1}$ with a width of ~80 cm$^{-1}$) to the CN stretches. This is particularly obvious in the 2D spectra for the FeRu/D$_2$O sample where a large solvent response grows in at high energies along the $\omega_3$ axis. A greater overlap would lead to more efficient energy transfer between the solute and solvent resulting in faster relaxation times. The vibrational lifetime of the trans mode is also likely faster when FeRu is dissolved in D$_2$O versus FA, but it is hard to say by how much. A faster relaxation time for the $\nu_1$ mode in D$_2$O versus FA was also observed in a similar molecule, [(NH$_3$)$_5$Ru$^{III}$NCOs$^{II}$[(CN)$_5$]$^-$ after optical excitation and back electron transfer.$^4$ This was attributed to better overlap between the $\nu_1$ band and solvent bands in D$_2$O compared with FA. Shorter vibrational lifetimes for the $\nu_1$ mode in FeRu indicate the $\nu_1$ mode has stronger interactions with the solvent than the other three modes. Because the $\nu_1$ mode is on the end of the molecule, it is more exposed to the solvent than the other modes.

Due to the larger size and increased complexity of FePtFe, it is expected that the vibrational relaxation for the $\nu_{CN}$ modes would be faster than for FeRu. Greater complexity increases the number of IVR channels available for the $\nu_{CN}$ modes to dump their energy into leading to more efficient energy transport out of the $\nu_{CN}$ modes. FePtFe also has a larger charge than FeRu. Previous work with other ions in solution has shown that they relax more quickly with increasing charge, likely due to the increased strength of interaction with the solvent.$^{18}$ In addition to these molecular attributes, most of the FePtFe $\nu_{CN}$ modes have better overlap with D$_2$O IR bands than
the ν\textsubscript{CN} modes of FeRu. Again, this should increase the efficiency with which energy is funneled out of the ν\textsubscript{CN} modes and into the solvent. Given these factors, it is surprising to see that the vibrational lifetimes for most FePtFe ν\textsubscript{CN} modes are almost twice as long as the lifetimes for FeRu/FA and four times as long as the lifetimes for FeRu/D\textsubscript{2}O. The ν\textsubscript{t} mode certainly decays significantly slower in FePtFe than in either FeRu sample. The fact that the ν\textsubscript{t} mode in FePtFe does not have a shorter lifetime than the other modes would indicate it is not as strongly coupled to the solvent as it is in FeRu. Weaker coupling of the ν\textsubscript{CN} modes to the solvent could also explain the longer timescales seen in FePtFe in general. The larger charge on the Pt(IV) compared to the Ru(III) explains why the frequencies of the trans and bridge modes are farther to the blue for FePtFe than for FeRu. The larger positive charge pulls more electron density out of the cyanide antibonding orbital increasing the strength of that bond and increasing its vibrational frequency. This effect might also explain why the trans mode acts so differently in FePtFe than for the two FeRu systems. If the reduced metal-ligand interaction results in weaker coupling to other carbon-metal modes or other low frequency modes along the MMCT backbone, this would limit interactions with the solvent for FePtFe in general helping explain why the lifetimes observed are so much longer than those measured for the FeRu systems.

Vibrational lifetimes ranging from 2.3 to 38.2 ps were observed for OCN\textsuperscript{-}, SCN\textsuperscript{-}, and SeCN\textsuperscript{-} in D\textsubscript{2}O, a lifetime of 170 ps was measured for the ν\textsubscript{CN} mode in FeNO(CN)\textsubscript{5}\textsuperscript{2-}, and a lifetime of 4.2 ps was observed for a cyanide probe of the villin HP35 protein.\textsuperscript{18-20} Other simple metal-cyanide compounds have long vibrational lifetimes that range from 8.0 to 170 ps when dissolved in D\textsubscript{2}O.\textsuperscript{21} All of the lifetimes observed in the current study fall within this range with the possible exception of the ~400 fs lifetime measured for the trans mode in FeRu/D\textsubscript{2}O. Vibrational lifetimes for ν\textsubscript{CN} modes in FePtFe, all of which are ~18 ps, closely match the lifetime of 24 ps reported for
Fe$^{II}$(CN)$_6^{4-}$ in D$_2$O. However, the lifetimes for FeRu in D$_2$O and FA of ~6 ps and ~10 ps, respectively, much more closely match the lifetimes reported for Fe$^{III}$(CN)$_6^{3-}$ in D$_2$O and FA of 8 ps and 10 ps, respectively. This comparison would indicate that the oxidation state of the iron in FeRu might be +3 instead of the +2 reported or at least more delocalized than assumed. However, work by multiple groups has shown that the extra electron is fairly localized on the Ru and not on the Fe. In a study of single metal-cyanides of the form M(CN)$_x^{y-}$ dissolved in D$_2$O, it was found that the vibrational lifetimes for the $\nu_{CN}$ mode was split into two camps where the first group have long vibrational energy relaxation times of >100 ps, and the second group have times of <35 ps. The first group contains Pt(CN)$_4^{2-}$, a square planar molecule, while the second includes Ru(CN)$_6^{4-}$, Fe(CN)$_6^{2-}$, and Fe(CN)$_6^{3-}$, all octahedral molecules. While the oxidation state and molecular configuration of the platinum investigated in this study is different than the state of the platinum in FePtFe, the large discrepancy in relaxation rates between the Pt and the Fe containing species despite the species similar charges might provide a clue as to why we observe longer lifetimes for the $\nu_{CN}$ modes in FePtFe than in the $\nu_{CN}$ modes in FeRu.

3.4.2 Spectral Diffusion Comparison

Due to the limited number of points available to fit and the large error bars that result for the CLS timescales measured, the comparisons that can be made between the samples studied here and other systems are limited. The approximately 3 ps timescale seen repeatedly in the CLS measurements can be compared to similar timescales of 1-2 ps seen in the frequency-frequency correlation functions for several compounds: a 1.5 ps spectral diffusion constant for the $\nu_{CN}$ mode in Fe$^{II}$(CN)$_6^{4-}$ in D$_2$O, a ~2 ps timescale for $\nu_{NO}$ mode FeNO(CN)$_5^{2-}$ in several hydrogen bonding solvents, and a 1.6 ps timescale for the cyanide probe of a villin HP35 protein when water molecules were allowed into the pocket. In all of these cases, the timescale was compared to the
characteristic timescale for the forming and breaking of hydrogen bonds.\textsuperscript{17,26,27} In light of this, the lack of a \textasciitilde{}3 ps timescale for the bridge mode CLS is not surprising. The bridging cyanide is not capable of forming hydrogen bonds because the nitrogen is already bound to the ruthenium instead of being free to interact with the solvent like it is for the other modes. Comparing the $v_r$ traces between all three samples, we see there is a dependence on solvent in that both D$_2$O samples only have one decay and an offset whereas the FA sample has a similar short decay and a longer decay with no offset. We have previously shown that angles between transition dipole moments might imply that the D$_2$O samples are bent in solution, while the FA sample is linear.\textsuperscript{7} This would suggest that, at least for this mode, spectral diffusion rates could rely on the structure of the molecule. But we can see a difference between the molecules by comparing the $v_a$ traces between FeRu and FePtFe in D$_2$O. The FePtFe sample only shows the \textasciitilde{}1 ps decay and an offset. In contrast, the FeRu sample shows a shorter timescale, the \textasciitilde{}1 ps decay, and no offset.

3.5 CONCLUSION

Despite previous work showing correlations between solvent and molecular parameters completed by our group, this study fails to find a correlation between the solvent and electronic ground state vibrational relaxation dynamics of the cyanide stretching vibrations of bimetallic and trimetallic transition metal mixed-valence complexes. We measure vibrational lifetimes, IVR dynamics, and spectral diffusion dynamics using infrared pump-probe and two-dimensional infrared spectroscopies. The values for vibration lifetimes reveal the \textasciitilde{}18 ps lifetimes for the $v_{\text{CN}}$ modes of FePtFe dissolved in D$_2$O are approximately four times longer than the modes of FeRu dissolved in D$_2$O and two times longer than the modes in FeRu dissolved in FA. The difference in lifetimes between the two D$_2$O systems indicates that vibrational lifetimes are not strongly dependent on solvent, higher molecular complexity, or larger overall charge of the molecule all of
which would result in faster relaxation times for FePtFe. Vibrational lifetimes for the trans mode in both FeRu system are significantly faster than the other three modes, which is not the case for FePtFe. This is an indication that FePtFe has weaker solute-solvent interactions since the trans mode is the most accessible to the solvent. Mode specific dynamics from both the pump-probe and 2D IR experiments reveal that IVR is occurring in all three systems on the order of 1 ps. This confirms previous findings that the $\nu_{CN}$ modes are coupled, but again show no obvious solvent dependence. Spectral diffusion dynamics reveal that the radial, axial, and trans $\nu_{CN}$ modes show a $\sim$3 ps timescale that is indicative of the forming and breaking of hydrogen bonds between the solvent and each $\nu_{CN}$ mode that is not apparent in the bridge mode. The lack of a relationship between solvent and vibrational dynamics timescales in these systems points to a need to consider the specific metals used in the backbone of future cyanide-bridged transition metal systems, not just the ligands or the solvent used.

### 3.6 Additional 2D IR Spectra for FeRu

As mentioned in Section 3.2, 2D IR spectra were collected at more $\tau_2$ points than are shown in Figure 3.3. The additional spectra for FeRu in both FA and D2O are shown below in Figures 3.6 - 3.9. Each figure shows all time points collected for each sample in either the parallel (ZZZZ) or crossed (YYZZ) polarizations. They are plotted on the same respective contour levels as listed in the caption for Figure 3.3.
Figure 3.6: Parallel polarization 2D IR spectra for FeRu dissolved in FA for all time points collected. Each spectra has been normalized, and the relative amplitudes of each spectra are shown in the lower right corner. The frequencies of each $\nu_{CN}$ mode are shown by dashed lines and are color-coded to the modes shown in Figure 3.1.
Figure 3.7: Crossed polarization 2D IR spectra for FeRu dissolved in FA for all time points collected. Each spectra has been normalized, and the relative amplitudes of each spectra are shown in the lower right corner. The frequencies of each $\nu_{CN}$ mode are shown by dashed lines and are color-coded to the modes shown in Figure 3.1.
Figure 3.8: Parallel polarization 2D IR spectra for FeRu dissolved in D₂O for all time points collected. Each spectra has been normalized, and the relative amplitudes of each spectra are shown in the lower right corner. The frequencies of each νCN mode are shown by dashed lines and are color-coded to the modes shown in Figure 3.1.
Figure 3.9: Crossed polarization 2D IR spectra for FeRu dissolved in D₂O for all time points collected. Each spectra has been normalized, and the relative amplitudes of each spectra are shown in the lower right corner. The frequencies of each $\nu_{CN}$ mode are shown by dashed lines and are color-coded to the modes shown in Figure 3.1.

REFERENCES


81


Chapter 4:

Synthesis and Characterization of Ruthenium (II) Polypyridine Cyanide Complexes

4.1 INTRODUCTION

Ruthenium containing complexes have been widely studied for applications in catalysis and photovoltaics with regards to energy production.\textsuperscript{1-7} Usually these processes require some sort of charge transfer. While the ruthenium containing dimer (FeRu) discussed in Chapters 2 and 3 was used to study metal-to-metal charge transfer (MMCT) reactions, ruthenium polypyridyl derivatives have been investigated for their metal-to-ligand charge transfer (MLCT) reactions. Here, the electron is transferred from the ruthenium metal to the polypyridine ligand. One example shows a series of polypyridyl ruthenium (II) complexes containing 2,2’-bipyridine, 2,2’-bipyrazin or 2,2’-bipyrimidine as ligands where the electron transfer from the metal to the lowest lying π* orbital is found to be coupled to a proton transfer.\textsuperscript{1,4} Complexes like these are useful as models for several systems including the production of oxygen in photosystem II or for use in dye-sensitized solar cells to name a few.\textsuperscript{2,3}

Studying these MLCT reactions and how they are coupled to nuclear motions can be accomplished using nonlinear vibrational spectroscopy like was discussed in Chapter 1 of this dissertation. Previous studies have shown that high frequency vibrations are highly coupled to back electron transfer reactions in cyanide-bridged mixed valence systems.\textsuperscript{7-13} Just as the cyanide stretching vibrations were used as probes of the MMCT reactions in FeRu and FePtFe, high
frequency cyanide stretches could be used as probes of the MLCT reactions occurring in polypyridyl ruthenium complexes. Moving charges around the molecule can change its geometry as it comes to a new equilibrium, and the energy needed to move the electron has to leave the molecule somehow. Both of these processes will have an effect on molecular vibrations which can be monitored spectroscopically. Because each of these systems has fewer cyanide ligands than FeRu and FePtFe, it will be easier to interpret those vibrational spectra.

In this chapter, the syntheses of three polypyridyl ruthenium (II) complexes with cyanide ligands are described, and the complexes are characterized using $^1$H NMR, FTIR, and UV/vis spectroscopies. These complexes are $[\text{Ru}^{\text{II}}(\text{tpy})(\text{bpy})\text{CN}]^+$ (monomer), $[\{\text{Ru}^{\text{II}}(\text{tpy})(\text{bpy})\}_2\text{CN}]^{3+}$ (dimer), and $[\text{Ru}^{\text{II}}\{\text{CNRu}^{\text{II}}(\text{tpy})(\text{bpy})\}_2]^4+$ (trimer), where ‘tpy’ stands for 2,2':6',2''-terpyridine and ‘bpy’ is 2,2’-bipyridine. Section 4.2 includes procedures used to synthesize each complex as well as sample preparation used to obtain spectra. Section 4.3 contains spectroscopic evidence used for product verification, and Section 4.4 looks at FTIR and UV/vis spectra of those products in several solvents. The final section, Section 4.5, includes a summary of the work done to this point with these complexes and what work we wish to do in the future.

4.2 METHODS

4.2.1 Spectra Collection

Most $^1$H NMR spectra were collected in methanol-d$_4$ using an AV-300 300.13 MHz instrument. The trimer NMR spectra was collected in acetonitrile-d$_3$ using the same instrument. FTIR spectra were collected using a JASCO FT/IR-4100 with a spectral resolution of 0.5 cm$^{-1}$, and each final spectrum is an average of 64 spectra. Samples were placed in a cell between two BaF$_2$ windows separated by a 100 μm spacer. UV/vis spectra were collected at the same time as the FTIR spectra in the same sample cell using a JASCO V-630 with a spectral resolution of 0.2 nm.
4.2.2 *Synthesis*

Chemicals purchased were ruthenium(III) chloride hydrate (Sigma, Pressure Chemical), 2,2':6’2”-terpyridine (Sigma, Strem), 2,2’-bipyridyl (Sigma), potassium cyanide (Sigma), silver trifluoromethanesulfonate (AgOTf) 99% (Sigma), triethyl amine (Fisher), cis-dichlorobis(2,2’-bipyridine)ruthenium(II) hydrate (Strem), ethanol (Decon Lab), diethyl ether (Sigma), lithium chloride (Sigma), methanol (Sigma), LiOTf (Sigma).

The syntheses of the Ru(tpy)Cl$_3$, [Ru(tpy)Cl]$_2$Cl$_2$, [Ru(tpy)(bpy)Cl]Cl and [Ru(tpy)(bpy)OTf]OTf were based off previously documented procedures with slightly different ligands.$^{2,6}$ These reactions typically resulted in yields between 60-90%. The synthesis of [Ru(tpy)(bpy)CN]OTf and [{Ru(tpy)(bpy)}$_2$CN]OTf were procedures developed for the purposes of these experiments. The typical yields were in the range of 70-90%.

*Synthesis of Ru(tpy)Cl$_3$.*

\[
\text{RuCl}_3 \cdot x \text{H}_2\text{O} + \text{tpy} \rightarrow \text{Ru(tpy)}\text{Cl}_3
\]

RuCl$_3$·xH$_2$O (2.537 g, 12.2 mmol), 2,2’:6’,2”-terpyridine (tpy) (2.660 g, 11.4 mmol) and ethanol (150 mL) were charged to a 250 mL round bottom flask and brought to reflux under argon atmosphere. The reaction remained at reflux for 4 hours. The solution was allowed to cool to room temperature, and then the solution was filtered using a medium fritted funnel. The brown-orange solid was washed with ethanol until the wash appeared clear, and the solid was allowed to dry for several minutes. Then the solid was washed with ether for further drying, the solid Ru(tpy)Cl$_3$ product was collected (4.659 g, 10.6 mmol). The yield was ~92%.
Synthesis of \([\text{Ru(tpy)Cl}]_2\text{Cl}_2\).

\[
\text{Ru(tpy)Cl}_3 \xrightarrow{\text{Reflux,N(CH}_3\text{CH}_2)_3} [\text{Ru(tpy)Cl}]_2 \text{Cl}_2
\]

\(\text{Ru(tpy)Cl}_3\) (2.237 g, 5.07 mmol) was mixed with ethanol (120 mL) and triethyl amine (4 mL) in a 250 mL round bottom flask. The solution was flushed with argon and was left under argon for the remainder of the reaction. The solution was brought to reflux and left to reflux for 2 hours. The solution turned from a brown color to a dark brown-purple color. The product was cooled to room temperature and filtered. A dark purple solid was collected and washed with ethanol until the filtrate appeared clear. The product was then dried over vacuum and washed with ether to remove any remaining ethanol. The product was left to dry over vacuum for about 5 minutes. The purple solid was collected and stored (1.464 g, 1.89 mmol). The yield was \(\sim 75\%\).

Synthesis of \([\text{Ru(tpy})(bpy)\text{Cl}]\text{Cl}\).

\[
[\text{Ru(tpy)Cl}]_2 \text{Cl}_2 + 2(bpy) \xrightarrow{\text{Reflux}} [\text{Ru(tpy)(bpy)Cl}]\text{Cl}
\]

\([\text{Ru(tpy)Cl}]_2\text{Cl}_2\) (1.2341 g, 1.59 mmol) was mixed with two molar equivalents of 2,2’-bipyridine (bpy) (0.5098 g, 3.26 mmol) in a 2:1 ethanol/H\(_2\)O mixture (60 mL). The solution was brought to reflux under a nitrogen atmosphere and left at reflux for 4 hours. Once the reaction was complete the solution turned a dark red. A saturated solution of LiCl was injected into the solution through a septum with a syringe and the solution was left to reflux for an additional 10 minutes. The solution was then filtered hot using gravity filtration to catch any unreacted material. The filtrate was collected and allowed to cool overnight. The ethanol was removed by rotary evaporation. The remaining solid was submerged in ether and scraped out of the rotary flask then filtered over vacuum. The dark brown-red solid product was collected (1.0679 g, 1.90 mmol) and a \(^1\text{H NMR}\) was taken for verification of the product. The yield was \(\sim 60\%\).
**Synthesis of [Ru(tpy)(bpy)OTf]OTf.**

\[
[Ru(tpy)(bpy)Cl]Cl + 2\text{AgOTf} \xrightarrow{\text{methanol, 45 °C}} [Ru(tpy)(bpy)OTf]OTf
\]

[Ru(tpy)(bpy)Cl]Cl and 2.05 molar equivalents of AgOTf (the slight excess was used to help drive the reaction) were dissolved in methanol and allowed to stir with slight heating (oil bath temperature was set to 45 °C) overnight. The solution was removed from heat and allowed to cool back to room temperature. The solution was filtered over Celite and the AgCl precipitate was separated from the solution. The methanol was removed from the filtrate by rotary evaporation, and the precipitate was washed with ether. The red-brown solid was collected. The reaction resulted in a complete conversion.

**Synthesis of [Ru(tpy)(bpy)CN]OTf.**

\[
[Ru(tpy)(bpy)OTf]OTf + 3\text{KCN} \xrightarrow{\text{Reflux}} [Ru(tpy)(bpy)CN]OTf
\]

[Ru(tpy)(bpy)OTf]OTf (0.7337 g, 0.930 mmol) was mixed with 3 molar equivalents of KCN in a 2:1 ethanol/H$_2$O mixture (30 mL) in a 100 mL round bottom flask. The KCN excess was used to help drive the reaction and prevent any of the dimer from forming. The solution was flushed with nitrogen and remained under nitrogen for the entirety of the reflux reaction. The solution was heated to reflux and the reaction proceeded overnight. The now vibrant wine-red colored solution was injected with a concentrated solution of LiOTf and allowed to continue to reflux for another 10 minutes. The solution was cooled and the ethanol was removed by rotary evaporation. The remaining liquid solution was cooled in a refrigerator overnight and then filtered and washed with a small amount of water and ethanol. The bright red solid was dried over vacuum for several minutes before being washed with ether. The solid was collected (0.6038 g, 0.899 mmol) and verified as the desired product by $^1$H NMR and FTIR. The yield for this reaction was ~96%.
Synthesis of \([\{\text{Ru(tpy)(bpy)}\}_2\text{CN}\}\text{OTf}_3\).

\[
[\text{Ru(tpy)(bpy)OTf}]\text{OTf} + \frac{1}{2} \text{KCN} \xrightarrow{\text{Reflux}} [\{\text{Ru(tpy)(bpy)}\}_2\text{CN}\}\text{OTf}_3
\]

[\text{Ru(tpy)(bpy)OTf}]\text{OTf} (0.7664 g, 0.972 mmol) was mixed with 1/2 molar equivalent of KCN in a 2:1 ethanol/H\text{}_2\text{O} mixture (30 mL) in a 100 mL round bottom flask. The solution was flushed with nitrogen and remained under nitrogen for the entirety of the reflux reaction. The solution was heated to reflux and the reaction proceeded overnight. The now dark red colored solution was injected with a solution of LiOTf and allowed to continue to reflux for another 10 minutes. The solution was cooled and the ethanol was removed by rotary evaporation. The remaining liquid solution was cooled in a refrigerator overnight and filtered and washed with a small amount of water and ethanol. Then the dark brown-red solid was dried over vacuum for several minutes before being washed with ether. The solid was collected (0.6090 g, 0.419 mmol) and verified using \(^1\)H NMR and FTIR analysis. The reaction resulted in a yield of ~86%.

Synthesis of (bpy\(_2\)Ru(CN)\(_2\)).

\[
(\text{bpy}\)_2\text{RuCl}_2 + 2\text{AgOTf} \xrightarrow{50 \text{oC}} (\text{bpy}\)_2\text{RuOTf}_2 + \text{AgCl}
\]

\[
(\text{bpy}\)_2\text{RuOTf}_2 + 4\text{KCN} \xrightarrow{\text{Reflux}} (\text{bpy}\)_2\text{Ru(CN)}_2
\]

(\text{bpy}\)_2\text{RuCl}_2 (0.5839 g, 1.12 mmol) was mixed with a 2.05 molar equivalent of AgOTf in a 2:1 ethanol/H\text{}_2\text{O} mixture in a round bottom flask. The reaction mixture was heated slightly (~50 °C) and stirred under a nitrogen atmosphere in the dark for ~2 hours. The solution was then filtered over Celite and rinsed with methanol to remove the solid AgCl. The filtered solution was then poured into a clean round bottom flask and 5 molar equivalents of KCN (assuming complete conversion of (\text{bpy}\)_2\text{RuCl}_2 to (\text{bpy}\)_2\text{Ru(OTf)}_2) were added. Under nitrogen atmosphere, bring the solution to reflux (~75 °C) and let the reaction run for ~4 hours. The solution turned a bright red-orange in color. The ethanol/methanol solvent was removed by rotary evaporation, and the
remaining product/H\textsubscript{2}O mixture was refrigerated overnight to crystalize the product. Product was filtered using the same method as [Ru(tpy)(bpy)CN]OTf. The bright red-orange fine powder was collected (0.4015 g, 0.863 mmol). The yield for this reaction was ~75%.

**Synthesis of [Ru{CNRu(tpy)(bpy)}\textsubscript{2}]OTf\textsubscript{4}**

\[
(bpy)\textsubscript{2}Ru(CN)\textsubscript{2} + 2[Ru(tpy)(bpy)OTf]OTf \xrightarrow{\text{Reflux}} [Ru\{CNRu(tpy)(bpy)\}\textsubscript{2}]OTf\textsubscript{4}
\]

A solution of (bpy)\textsubscript{2}Ru(CN)\textsubscript{2} (0.304 g, 0.653 mmol) and 2.05 molar equivalents of [Ru(tpy)(bpy)OTf]OTf in a 2:1 ethanol/H\textsubscript{2}O mixture was placed in a round bottom flask. The round bottom flask was attached to a water-cooled condenser, and the solution was placed under nitrogen atmosphere. The solution was heated to ~75 °C and allowed to reflux overnight. The reaction mixture was poured into a rotary evaporator flask with a LiOTf/H\textsubscript{2}O solution. Ethanol was removed by rotary evaporation, and the mixture was placed in the refrigerator overnight. The product was scraped out of the flask (this may require some advanced-level scraping), and the flask was filtered with water rinses and dried over vacuum. The product was rinsed with ether before removing the vacuum. The product was collected (1.44 g, 0.609 mmol), and the reaction had a yield of ~90%.

**4.2.3 Sample Preparation**

Solutions of the monomer and dimer were made in the solvents listed below and studied using FTIR and UV/vis spectroscopies. The solvents used for the solvent study in Section 4.4 were methanol (Sigma), ethanol (Decon Lab), 1-butanol (Fischer), isopropanol (Mallinckrodt), acetone (Sigma), acetonitrile (EMD), benzonitrile (Alpha Aesar), and chloroform (Macron). All solutions were saturated with the exception of the monomer in methanol (6.25 mM) and the dimer in acetonitrile (3.44 mM).
4.3 **COMPOUND VERIFICATION**

To study MLCT reactions, and the role high frequency vibrations play in those reactions, we need to make sure we know what the complex is and that it does not have any impurities that might confuse the interpretation of any findings such as any unwanted RuCN combinations. Several spectroscopic techniques were used in conjunction with each other to verify the identity of each species and to try to gauge both the identity and amount of any impurities present. The primary method used to determine if the tpy and bpy ligands were attached to the ruthenium was $^1$H NMR.

4.3.1 $^1$H NMR Spectra

The $^1$H NMR spectra seen in Figure 4.1 were used as verification that the desired final products were made. The splitting patterns of the bipyridine and terpyridine seen in Figure 4.1 and differ from the literature for the isolated bipyridine and terpyridine complexes (12). In general, peaks are shifted further downfield, and there is more splitting. The perpendicular conformation of the bipyridine and terpyridine affects the environments of surrounding hydrogens on the ligands, and having the nitrogens attached to Ru(II) instead of completely unattached like in the isolated species also affects the electron density in the polypyridines. Both of these will disturb the chemical shift of the hydrogens. The $^1$H NMR spectra confirm that both the terpyridine and bipyridine were successfully coordinated to the ruthenium metal center.

In Figure 4.1a, the spectra of the chlorinated monomeric complex (pictured as a cartoon in Fig. 4.1e) displays a hydrogen that is shifted much further downfield than the other hydrogens. Due to the electronegativity of the Cl ligand and the fact that the chloride shares a plane with it, this hydrogen is less shielded than the others on the pyridyl ligands and will experience a shift farther down field ($\delta = 10.25$ ppm). The $^1$H NMR spectrum of the monomer is displayed in Figure 4.1b. Here the chloride has been replaced with the desired cyanide ligand. By comparing these two
The greatest indication of the ligand exchange is in the hydrogen that is furthest down field. The hydrogen moved from a chemical shift of $\delta = 10.23$ ppm to a shift of $\delta = 10.07$ ppm. Since this is the hydrogen that was in the plane with the chloride, a change in the shift

**Figure 4.1:** (a) $^1$H NMR spectrum of [Ru(tpy)(bpy)Cl]$^+$,  (b) $^1$H NMR spectrum of [Ru(tpy)(bpy)CN]$^+$.  (c) $^1$H NMR spectrum of [[Ru(tpy)(bpy)]$_2$CN]$^{3+}$.  (d) $^1$H NMR spectrum for [Ru{CNRu(tpy)(bpy)}$_2$]$. The small peak at $\delta = 10.23$ ppm is due to a monomer impurity in the sample.  (e) Cartoon figure of monomer. (f) Cartoon figure of dimer. (g) Cartoon of trimer.
indicates the coordinating ligand has also changed. Cyanide is less electronegative than the Cl ligand so the hydrogen should be shifted upfield, and this is what the spectra show.

The $^1$H NMR spectrum for the dimer (Fig. 4.1c) is similar to the monomeric complex (Fig. 4.1b) except that all of the peaks show further doublet splitting. Since the molecule is not perfectly symmetric, ‘matching’ hydrogens on the polypyridines have slightly dissimilar environments. There are now two hydrogens in the plane of the bridging CN and their shifts are even farther up field, $\delta = 9.4$ ppm since the electron withdrawing effects of the cyanide are now split between two sets of bpy and tpy ligands. Figure 4.1d shows the $^1$H NMR spectrum of the trimer. Here we see that the product is not entirely pure, because there is still a peak at a chemical shift of $\delta = 10.23$ ppm. This small amount of impurity can be assigned to the monomer, and further recrystallization should be able to remove it.

4.3.2 FTIR Spectra

While $^1$H NMR verified that bpy and tpy were successfully attached to the ruthenium, it only hinted at the fact that the other coordinating ligand is indeed cyanide. To more conclusively verify this, FTIR spectra were taken of the monomer and dimer. The CN ligands on the dimer and monomer should have stretching frequencies in the range of 2000 - 2150 cm$^{-1}$ based on previous infrared work on metal cyanide species in solution.$^7,^{10,11,13,15-18}$ Both FTIR spectra for the dimer and monomer in methanol (Figure 4.2) show a peak around 2100 cm$^{-1}$ confirming that the cyanide group was successfully coordinated to the ruthenium.

In addition to making sure that the cyanide ligand was attached to the ruthenium, we need to make sure that the compound will stay stable over a period of a couple of days, the length of time it takes to run a full set of nonlinear vibrational spectroscopy experiments. Solutions of monomer and dimer in methanol were left sitting in a vial for over two days to test this. Spectra taken at the
beginning and end of the period were compared to see if any significant changes had occurred.

Figure 4.3 shows the FTIR spectra taken on day one and day three. For the dimer in methanol (Fig. 4.3b), the absorbance increases, but the width and shape of the peak appear unchanged. Increased absorbance would indicate an increase of the concentration of the dimer solution. The
two spectra for the monomer (Fig. 4.3a) overlap completely. It appears the two complexes remain stable in solution over a period of several days.

4.3.3 UV/vis Spectra

Charge transfer transitions usually have spectroscopic signatures in the near-infrared to ultraviolet regions of the electromagnetic spectrum. The UV/vis spectra of the dimer and monomer were taken to visualize the MLCT from the ruthenium metal to the terpyridine ligand. This MLCT transition is of particular interest, because it relaxes directly into a long-lived triplet state\(^5\) where we can study vibrational dynamics before the system relaxes back to the ground electronic state. The charge transfer reaction for FePtFe previously studied by our group relaxes within ~200 fs which only allows for the study of ‘hot’ vibrational states on the electronic ground state. The UV/vis spectra show the MLCT band at ~610 nm for both the monomer (Fig. 4.4, left) and the dimer (Fig. 4.4, right). Previous work with these compounds has shown this transition at ~610 nm providing one more confirmation the desired product was made.\(^5\)

![UV/vis Spectra](image)

**Figure 4.4:** UV/vis spectra of the monomer (left panel) and dimer (right panel) in several solvents. Each inset shows a shoulder around 610 nm corresponding to the metal to ligand charge transfer of interest.
4.4 Solvent Study

There are no published FTIR for the monomer, dimer, or trimer as of this work. To find a solvent that would work for future experiments, both the dimer and monomer were placed in the following solvents to gauge their solubility: four alcohols (methanol, ethanol, isopropanol and 1-butanol), acetonitrile, benzonitrile, and chloroform, all of which are polar. These solvents were chosen for a combination of low absorbance in the IR region of interest, ease of access, and how easily they can be set up to flow through the sample cell for nonlinear experiments. With the exception of acetonitrile, the monomer was far more soluble in each solvent than the dimer. Ideally, an absorbance of 0.2 to 0.4 OD at the cyanide stretching frequency could be reached while minimizing solvent background. However, since the monomer and dimer are both charged and have large hydrophobic ligands, the compounds did not dissolve particularly well in highly polar solvents and hardly dissolve at all in the less polar solvents like the longer alcohol chains as can be seen in Figure 4.5. No solvent achieved greater than 0.2 OD for the dimer, and only methanol allowed

![Figure 4.5: FTIR spectrum of monomer (a) and dimer (b) in several solvents where the solvent backgrounds have been subtracted.](image)
the absorbance for the monomer to exceed this threshold. This greatly limits the choice of solvents available for experiments.

The cyanide stretching region of each FTIR spectrum shown in Figure 4.5 was fit using Voigt profiles. All dimer and the monomer samples not dissolved in alcohol were fit with a single peak. Monomer samples dissolved in alcohol were fit to two peaks. The resulting frequencies and widths along with their error bars are listed in Tables 4.1 and 4.2 for the dimer and monomer, respectively. Center frequencies for the dimer range from 2101 – 2107 cm\(^{-1}\), while the frequencies for the main monomer peak clump right around 2080 cm\(^{-1}\) with the notable exceptions of acetone and acetonitrile which have center frequencies around 2090 cm\(^{-1}\). It is interesting that the CN of the dimer, which should have less interaction with the solvent, has more variation in its frequencies than the CN of the monomer has for the same solvents. The second peak for the two monomers with enough amplitude to fit it, both have frequencies at 2100 cm\(^{-1}\). The widths for the dimer samples range from 9 – 11 cm\(^{-1}\) while the widths of the monomer peaks range from 4 – 9 cm\(^{-1}\). It is again interesting to note that the dimer has broader peaks than the monomer as this could indicate a more heterogeneous distribution of environments for the cyanide ligand when the cyanide ligand on the monomer should have more diverse interactions because it is readily available to the solvent. Of course, the broader peaks could be due to faster vibrational relaxation. Further experiments using 2D IR spectroscopy would help separate how much of the peak width comes from each component.

**Table 4.1:** Best fit parameters for the frequency and width of FTIR peaks for the dimer in different solvents. The error bars for these fits encompass a 95\% confidence interval.

<table>
<thead>
<tr>
<th></th>
<th>Frequency (cm(^{-1}))</th>
<th>Width (cm(^{-1}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methanol</td>
<td>2106.8 ± 0.1</td>
<td>9.3 ± 0.1</td>
</tr>
<tr>
<td>Acetone</td>
<td>2106.5 ± 0.1</td>
<td>10.2 ± 0.1</td>
</tr>
<tr>
<td>Acetonitrile</td>
<td>2106.9 ± 0.1</td>
<td>11.4 ± 0.2</td>
</tr>
<tr>
<td>Benzonitrile</td>
<td>2101.1 ± 0.1</td>
<td>10.9 ± 0.2</td>
</tr>
</tbody>
</table>
Table 4.2: Best fit parameters for the frequency and width of FTIR peaks for the monomer in different solvents. The error bars for these fits encompass a 95% confidence interval.

<table>
<thead>
<tr>
<th></th>
<th>Peak 1 Frequency (cm⁻¹)</th>
<th>Peak 1 Width (cm⁻¹)</th>
<th>Peak 2 Frequency (cm⁻¹)</th>
<th>Peak 2 Width (cm⁻¹)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methanol</td>
<td>2079.5 ± 0.1</td>
<td>9.21 ± 0.09</td>
<td>2099.2 ± 0.2</td>
<td>5.7 ± 0.2</td>
</tr>
<tr>
<td>Ethanol</td>
<td>2080.8 ± 0.1</td>
<td>7.8 ± 0.1</td>
<td>2100.3 ± 0.2</td>
<td>6.2 ± 0.2</td>
</tr>
<tr>
<td>Isopropanol</td>
<td>2080.3 ± 0.2</td>
<td>7.1 ± 0.3</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>1-butanol</td>
<td>2081.7 ± 0.1</td>
<td>7.0 ± 0.2</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>Acetone</td>
<td>2091.9 ± 0.1</td>
<td>4.4 ± 0.2</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>Acetonitrile</td>
<td>2088.1 ± 0.1</td>
<td>4.69 ± 0.08</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>Benzonitrile</td>
<td>2081.5 ± 0.2</td>
<td>9.1 ± 0.4</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

During the course of this solvent study, it was noticed that the monomer had two peaks in the alcohols. The second peak at ~2100 cm⁻¹ was initially thought to be a dimer impurity. However, the frequency for the peak is too low, as can be seen from comparing the frequencies in Tables 4.1 and 4.2. It is possible that this peak might represent CN hydrogen-bonded to the solvent through the nitrogen since it only occurs in the solvents that contain hydroxide groups. Further investigation of this feature using temperature dependent FTIR to see how the two peaks change relative to each other or 2D IR to look at spectral diffusion rates should provide more information about the difference in environment for each population of CN ligands.

4.5 CONCLUSIONS AND FUTURE WORK

Using ¹H NMR, FTIR, and UV/vis spectroscopies, we verified the synthesis of the monomer and dimer was successful. The trimer needs more purification, but its synthesis was also successful. The dimer and monomer were both found to be stable in solution over the period of a couple of days providing promise for future multi-day experiments. A MLCT band at ~610 nm was seen in the UV/vis spectra for both the monomer and the dimer in several solvents, and the FTIR spectra showed sufficient optical density in several solvents to see signal for even higher
nonlinear spectroscopy experiments. Further work will include synthesis of further purification of the trimer, as well as FTIR and UV/vis studies in multiple solvents. Temperature dependent FTIR studies of the monomer in methanol and ethanol would help elucidate the origin of the two cyanide stretching bands. Finally, nonlinear vibrational spectroscopy experiments could include a study of the ground electronic state using 2D IR and a study of the possible coupling of high-frequency cyanide stretches with the MLCT transition using vis-pump IR-probe experiments for the monomer, dimer, and trimer.

ACKNOWLEDGEMENTS

I would like to thank “my” undergraduate Laura Estergreen who actually did most of the synthesis, Dr. Michael Norris for his help in collecting NMR spectra and for helping us develop some of these syntheses, and Prof. Brandi Cossairt for kindly allowing us the use of her laboratory space for portions of these reactions.

REFERENCES


Appendix A:

Matlab Code for the Simulation of 2D IR Spectra

This Appendix contains all the code used to simulate the 2D IR spectra shown in Chapter 1 and Chapter 2. The first three sections contain basic examples of how to set-up code for two coupled modes and four coupled modes. Section A.4 contains an example of the code I used for fitting experimental spectra. Finally, Section A.5 contains the code for the background calculations that are the same for all systems. The basis of the code was written by Ben Van Kuiken, and I edited it for the examples shown here and for use with Matlab’s fitting routine as well as commenting it.

A.1 SIX LEVEL SYSTEM WITH HOMOGENEOUS LINESHAPES

This is the most basic code for two couple modes. It is written with homogeneous lineshape functions and parallel transition dipole moments. This code was used to produce Figures 1.5 and 1.8 both at $\tau_2 = 0$ fs.

A.1.1 sixLevelHomLimPolarComp.m

1  %% Six Level Test - Two Coupled Orthogonal Dipoles
2  % This is an example and test of specPack for a 6 level system using
3  % homogeneous limit lineshapes
4
5  %% Set Path
6  %This path allows several folders to access the same specPack file
7  addpath('C:\Users\Karla\Documents\GradSchool
8     Work\Research\Theses\Thesis\six-level examples for intro\specPack3')
%% Clear Memory

clear % clears memory (clears variables in workspace)
clc % clears command window

%% Define system parameters

% 'energies' contains all frequencies for each level in the system. The
% order does not matter as long as you follow through with the ordering
% in
% all other matrices and arrays. Frequencies have been shifted down by
% 1500
% in this example to speed calculation time (but anharmonicities are not
% scaled). This can be reversed in the final output by adding 1500 to
each
% frequency axis.
energies = [0.0 500 560 985 1045 1105];

% Transition dipole magnitudes (relative amplitudes - I've always defined
% one
% transition with an amplitude of 1, and all the other have amplitudes
% less
% than 1)
rt2 = sqrt(2.0);
mu = [ 0.0 1.0 1.0 0.0 0.0 0.0;...
  0.0 0.0 rt2 1.0 0.0;...
  0.0 0.0 0.0 1.0 rt2;
  rt2 0.0 0.0 0.0 0.0;...
  1.0 1.0 0.0 0.0 0.0;...]
%Define the number of states/levels your system contains
nstates = 6; % 6 for a system of two coupled modes
for i=1:nstates
    for j=1:nstates
        for k = 1:3
            muDir(i,j,k) = 0.0; %Defining a 3D matrix for transition dipole
directions
        end
    end
end

%Any nonzero elements of the mu direction matrix need to be defined here
%muDir(i,j,k) = m where i and j are relevant states, k is the approp
x,y, or z
%axis, and m is the magnitude along that axis
%The size of this list will change with the number of states in your system
muDir(1,2,1) = 1.0;
muDir(2,1,1) = 1.0;
muDir(1,3,1) = 1.0;
muDir(3,1,1) = 1.0;
muDir(2,4,1) = 1.0;
muDir(4,2,1) = 1.0; %These are currently written for parallel moments
muDir(2,5,1) = 1.0;
muDir(5,2,1) = 1.0;
muDir(3,5,1) = 1.0;
muDir(5,3,1) = 1.0;
muDir(3,6,1) = 1.0;
muDir(6,3,1) = 1.0;

% muDir(1,2,1) = 1.0;
% muDir(2,1,1) = 1.0;
% muDir(1,3,2) = 1.0;
% muDir(3,1,2) = 1.0;
% muDir(2,4,1) = 1.0;
% muDir(4,2,1) = 1.0; %These are currently written for perpendicular moments
% muDir(2,5,2) = 1.0;
% muDir(5,2,2) = 1.0;
% muDir(3,5,1) = 1.0;
% muDir(5,3,1) = 1.0;
% muDir(3,6,2) = 1.0;
% muDir(6,3,2) = 1.0;

% Define your initial populations. Here all population starts in the ground
% state. Size is 'nstates' dependent.
rho0 = [1.0 0.0 0.0 0.0 0.0 0.0 0.0];

% Defining a cell array for lineshape functions
hfun = cell(3);
for i = 1:nstates
    for j = 1:nstates
        hfun(i,j)=@g0;
    end
end
end

% All appropriate lineshapes functions for a 6-level system are called and
% assigned to their cell here. Lineshape functions need to be defined in
% their own function files. This list will grow as the number of states in
% the system grow.
hfun{2,2}=@gaaHL;
hfun{2,4}=@ga2aHL;
hfun{4,2}=@ga2aHL;
hfun{3,3}=@gbbHL;
hfun{3,6}=@gb2bHL;
hfun{6,3}=@gb2bHL;
hfun{4,4}=@g2a2aHL;
hfun{6,6}=@g2b2bHL;
hfun{2,3}=@gabHL;
hfun{3,2}=@gabHL;
hfun{2,5}=@gaabHL;
hfun{5,2}=@gaabHL;
hfun{3,5}=@gbabHL;
hfun{5,3}=@gbabHL;
hfun{5,5}=@gababHL;

% Orientational diffusion constant
dor = 2.0e-5;

% Initialize structs
% In this section all data about system, experiment and calculation options
% are packed in to c-style structures for convenient data handling

% Center frequencies for experiment's three "pulses" and local oscillator
field_freq = [530 530 530 530];

% Calls 'initSystem' function from specPack and passes along all the system
% information defined above
sys = initSystem( nstates, energies, mu, muDir, rho0, hfun, dor);

% Experiment specific information - Here it is initiating a 2DIR measurement
% in the ZZZZ polarization
expr = initExperiment(field_freq, '2DIR', 'ZZZZ');

%'sp_SetOptions' should always be opened and checked manually to make sure
% all parameters are appropriate for the system
opts = sp_SetOptions('tstep',8.0);

% Call 2D calculation function
% 'tau2' can be a single time point or a list (in femtoseconds)
tau2 = [0 50];
the first value passed to 'calc2D Impulse' is a user-defined named that all generated files will be saved under:

```
calc2DImpulse('sixLevel_ZZZZ', sys, expr, tau2, opts);
```

% Now calculate YYZZ Polarization

% Redefine the system parameters to calculate the YYZZ polarization and % recompute the spectrum

```
expr = initExperiment(field_freq, '2DIR', 'YYZZ');

calc2DImpulse('sixLevel_YYZZ', sys, expr, tau2, opts);
```

% Plot 2DIR Correlation Spectrum (ZZZZ Polarization)

```
cspecl = load('sixLevel_ZZZZ_t2-0_abs.mat');

figure(1)
contourf(cspecl.freq(1251:1312),cspecl.freq(1251:1312),... cspecl.abs1(1251:1312,1251:1312),20)
xlim([461 584])
ylim([461 584])
axis square
set(gca, 'Tickdir', 'out');
set(gca, 'FontWeight', 'bold')
set(gca,'FontSize',16)
set(gca, 'FontName','Arial')
set(gca,'TickLength', [0.03, 0.01])
set(gca,'LineWidth', 2)
xlabel('\omega_{1}/2\pi (cm^{-1})')
ylabel('\omega_{3}/2\pi (cm^{-1})')
title('ZZZZ, \tau_2 = 0')
clear cspec1

% Plot 2DIR Correlation Spectrum (YYZZ Polarization)

cspec1 = load('sixLevel_YYZZ_t2-0_abs.mat');

figure(2)
contourf(cspec1.freq(1251:1312),cspec1.freq(1251:1312),...
cspec1.abs1(1251:1312,1251:1312),20)
xlim([461 584])
ylim([461 584])
axis square
set(gca, 'Tickdir', 'out');
set(gca, 'FontWeight','bold')
set(gca,'FontSize',16)
set(gca,'FontName','Arial')
set(gca,'TickLength', [0.03, 0.01])
set(gca,'LineWidth', 2)
xlabel('\omega_{1}/2\pi (cm^{-1})')
ylabel('\omega_{3}/2\pi (cm^{-1})')
title('YYZZ, \tau_2 = 0')
clear cspec1
A.1.2  \textit{gaaHL.m}

This is a lineshape function, gbbHL.m follows the same format.

\begin{verbatim}
function y = gaaHL( t )
  %GAA Lineshape function \( g_{\text{aa}} \)
  % An example of a homogeneous lineshape function for mode a
  fac = 2000;
  y = abs(t)/fac;
end
\end{verbatim}

A.1.3  \textit{ga2aHL.m}

The lineshape function gb2bHL.m follows the same format.

\begin{verbatim}
function y = ga2aHL( t )
  %GAA Lineshape function \( g_{\text{a2a}} \)
  % An example of a homogeneous lineshape function for the transition from
  % a to 2a
  fac = 2000;
  y = 2.0*abs(t)/fac;
end
\end{verbatim}

A.1.4  \textit{g2a2aHL.m}

The lineshape function g2b2bHL.m follows the same format.

\begin{verbatim}
function y = g2a2aHL( t )
  %GAA Lineshape function \( g_{\text{aa}} \)
end
\end{verbatim}
An example of a homogeneous lineshape function for the 2a energy level

\[ y = \frac{4.0 \cdot \text{abs}(t)}{\text{fac}}; \]

A.1.5  \texttt{gabHL.m}

```
function y = gabHL( t )

% GAA Lineshape function \( g_{(ab)} \)

fac = 2000;

y = \text{abs}(t)/\text{fac};

end
```

A.1.6  \texttt{gaabHL.m}

The lineshape function \texttt{gbabHL.m} follows the same format.

```
function y = gaabHL( t )

%GAA Lineshape function \( g_{(aab)} \)

% An example of a homogeneous lineshape function for the transition from
% a to ab

fac1 = 2000;

fac2 = 2000;

y = \text{abs}(t)/\text{fac1}+\text{abs}(t)/\text{fac2};
```
A.1.7  *gababHL.m*

```matlab
function y = gababHL( t )
%GAA Lineshape function g_{abab}
% An example of a homogeneous lineshape function for the ab energy level

fac1 = 2000; %same as fac for gaa
fac2 = 2000; %same as fac for gab
fac3 = 2000; %same as fac for gbb

y = abs(t)/fac1+2.0*abs(t)/fac2+abs(t)/fac3;
end
```

A.2  **SIX LEVEL SYSTEM WITH POPULATION RELAXATION**

Make sure the `sp_SetOptions.m` file in SpecPack3 is updated to have a ‘true’ values for the population relaxation value. This code was used to make figure 1.6.

A.2.1  *sixLevel_poprlx.m*

```matlab
% Six Level Test - Two Coupled Orthogonal Dipoles
% This is an example and test of specPack for a 6 level system. Here
% population relaxation and 45 degree angle between dipole moments is
% shown. General comments on code can be found in the homogeneous limit
% example, and only comments on changes are noted.

%Warning!!! I believe there is an error in the implementation of the
%population lifetimes. Things don't seem to relax as quickly (or
%correctly) as they should.

%% Set Path
addpath('C:\Users\Karla\Documents\Grad School
Work\Research\Theses\Thesis\six-level examples for intro\specpack3')

%% Clear Memory
clear
clc

%% Define system parameters
energies = [0.0 500 560 985 1045 1105];
rt2 = sqrt(2.0);
mu = [ 0.0 1.0 1.0 0.0 0.0 0.0;...
0.0 0.0 rt2 1.0 0.0;...
0.0 0.0 0.0 1.0 rt2;
rt2 0.0 0.0 0.0 0.0;...
1.0 1.0 0.0 0.0 0.0;...
0.0 rt2 0.0 0.0 0.0];
nstates = 6;
for i=1:nstates
for j=1:nstates
for k = 1:3
muDir(i,j,k) = 0.0;
end
end
muDir(1,2,2) = 1.0/rt2;
muDir(2,1,2) = 1.0/rt2;
muDir(1,2,3) = 1.0/rt2;
muDir(2,1,3) = 1.0/rt2;
muDir(1,3,2) = 1.0;
muDir(3,1,2) = 1.0;
muDir(2,4,2) = 1.0/rt2;
muDir(4,2,2) = 1.0/rt2;
muDir(2,4,3) = 1.0/rt2;%This is an example of how to set up 45 degrees
btwn dipole moments
muDir(4,2,3) = 1.0/rt2;
muDir(2,5,2) = 1.0;
muDir(5,2,2) = 1.0;
muDir(3,5,2) = 1.0/rt2;
muDir(5,3,2) = 1.0/rt2;
muDir(3,5,3) = 1.0/rt2;
muDir(5,3,3) = 1.0/rt2;
muDir(3,6,2) = 1.0;
muDir(6,3,2) = 1.0;

rho0 = [1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0];

hfun = cell(3);
for i = 1:nstates
for j = 1:nstates
    hfun{i,j}=@g0;
end

hfun{2,2}=@gaa;
hfun{2,4}=@ga2a;
hfun{4,2}=@ga2a;
hfun{3,3}=@gbb;
hfun{3,6}=@gb2b;
hfun{6,3}=@gb2b;
hfun{4,4}=@g2a2a;
hfun{6,6}=@g2b2b;
hfun{2,3}=@gab;
hfun{3,2}=@gab;
hfun{2,5}=@gaab;
hfun{5,2}=@gaab;
hfun{3,5}=@gbab;
hfun{5,3}=@gbab;
hfun{5,5}=@gabab;

dor = 2.0e-5;

%This is where the timescales for population lifetimes are entered. There
%needs to be a value for each energy level, so here six timescales.
lifetimes = [1e10 10.0e3 5.0e3 10e3 7.5e3 5e3];
 %% Initialize structs

sys = initSystem( nstates, energies, mu, muDir, rho0, hfun, dor, lifetimes);

expr = initExperiment([500 500 500 500], '2DIR', 'ZZZZ');

opts = sp_SetOptions('prefacThresh',0.1,'tstep',10.0);

%% Call 2D calculation function

tau2 = [0,200,500,1000,2000,3500,5000,7000,10000];
calc2Dimpulse('sixLevel_test_45deg_poprlx_zzzz',sys,expr,tau2,opts);

%Below commented out are other ways to plot the data (for rephasing and non-rephasing, in time or freq domain)

% % % Create Rephasing Time Plot
%
% rph_time = load('sixLevel_test_t2-250_rt.mat');
% figure(1)
% contourf(rph_time.t1(1:400),rph_time.t3(1:400),...
% real(rph_time.response_time(1:400,1:400)),10)
% clear rph_time
%
% % % Create NonRephasing Time plot
%
% nr_time = load('sixLevel_test_t2-250_nrt.mat');
% figure(2)
% contourf(nr_time.t1(1:400),nr_time.t3(1:400),...
% nr_time.response_time(1:400,1:400),10)
% axis square
% clear nr_time

% % Plot rephasing frequency domain
% rf = load('sixLevel_test_t2-250_rf.mat');
% % Construct Frequency Axes
% f = 0.25/2*linspace(0,1,2048/2+1)/(2.998e-5);
% figure(3)
% contourf(rf.freq,rf.freq,rf.response_freq,10)
% clear rf

% % Plot Non-Rephasing Spectrum
% nrf = load('sixLevel_test_t2-250_nrf.mat');
% figure(4)
% contourf(nrf.freq,nrf.freq,nrf.response_freq,10)
% axis square
% clear nrf
% Plot 2DIR Correlation Spectrum

cspec1 = load('sixLevel_test_45deg_poprlx_zzzz_t2-0_abs.mat');
cspec2 = load('sixLevel_test_45deg_poprlx_zzzz_t2-200_abs.mat');
cspec3 = load('sixLevel_test_45deg_poprlx_zzzz_t2-500_abs.mat');
cspec4 = load('sixLevel_test_45deg_poprlx_zzzz_t2-1000_abs.mat');
cspec5 = load('sixLevel_test_45deg_poprlx_zzzz_t2-2000_abs.mat');
cspec6 = load('sixLevel_test_45deg_poprlx_zzzz_t2-3500_abs.mat');
cspec7 = load('sixLevel_test_45deg_poprlx_zzzz_t2-5000_abs.mat');
cspec8 = load('sixLevel_test_45deg_poprlx_zzzz_t2-7000_abs.mat');
cspec9 = load('sixLevel_test_45deg_poprlx_zzzz_t2-10000_abs.mat');

z = [-60000:8000:60000];

figure
contourf(cspec1.freq(1255:1310),cspec1.freq(1255:1310),... 
real(cspec1.abs1(1255:1310,1255:1310)),z)
xlim([470 580])
ylim([470 580])
caxis([-60000 60000])
axis square
set(gca, 'Tickdir', 'out');
set(gca, 'FontWeight','bold')
set(gca,'FontSize',16)
set(gca, 'FontName','Arial')
set(gca,'TickLength', [0.03, 0.01])
set(gca,'LineWidth', 2)
xlabel('\omega_{1}/2\pi (cm^{-1})')
ylabel('\omega_{3}/2\pi (cm^{-1})')
title('ZZZZ, \tau_2 = 0')

figure
contourf(cspec2.freq(1255:1310), cspec2.freq(1255:1310), ...
real(cspec2.abs1(1255:1310, 1255:1310)), z)
xlim([470 580])
ylim([470 580])
caxis([-60000 60000])
axis square
set(gca, 'Tickdir', 'out');
set(gca, 'FontWeight', 'bold')
set(gca, 'FontSize', 16)
set(gca, 'FontName', 'Arial')
set(gca, 'TickLength', [0.03, 0.01])
set(gca, 'LineWidth', 2)
xlabel('\omega_1/2 \pi (\text{cm}^{-1})')
ylabel('\omega_3/2 \pi (\text{cm}^{-1})')
title('ZZZZ, \tau_2 = 200')

figure
contourf(cspec3.freq(1255:1310), cspec3.freq(1255:1310), ...
real(cspec3.abs1(1255:1310, 1255:1310)), z)
xlim([470 580])
ylim([470 580])
caxis([-60000 60000])
axis square
set(gca, 'Tickdir', 'out');
set(gca, 'FontWeight', 'bold')
set(gca, 'FontSize', 16)
208 set(gca, 'FontName','Arial')
209 set(gca,'TickLength', [0.03, 0.01])
210 set(gca,'LineWidth', 2)
211 xlabel('\omega_{1}/2\pi (cm^{-1})')
212 ylabel('\omega_{3}/2\pi (cm^{-1})')
213 title('ZZZZ, \tau_2 = 500')

214

215 figure
216 contourf(cspec4.freq(1255:1310),cspec4.freq(1255:1310),... 
217 real(cspec4.abs1(1255:1310,1255:1310)),z)
218 xlim([470 580])
219 ylim([470 580])
220 caxis([-60000 60000])
221 axis square
222 set(gca, 'Tickdir', 'out');
223 set(gca, 'FontWeight','bold')
224 set(gca,'FontSize',16)
225 set(gca, 'FontName','Arial')
226 set(gca,'TickLength', [0.03, 0.01])
227 set(gca,'LineWidth', 2)
228 xlabel('\omega_{1}/2\pi (cm^{-1})')
229 ylabel('\omega_{3}/2\pi (cm^{-1})')
230 title('ZZZZ, \tau_2 = 1000')

231

232 figure
233 contourf(cspec5.freq(1255:1310),cspec5.freq(1255:1310),... 
234 real(cspec5.abs1(1255:1310,1255:1310)),z)
235 xlim([470 580])
236 ylim([470 580])
caxis([-60000 60000])
axis square
set(gca, 'Tickdir', 'out');
set(gca, 'FontWeight', 'bold')
set(gca, 'FontSize', 16)
set(gca, 'FontName', 'Arial')
set(gca, 'TickLength', [0.03, 0.01])
set(gca, 'LineWidth', 2)
xlabel('\omega_1/2 \pi (cm^{-1})')
ylabel('\omega_3/2 \pi (cm^{-1})')
title('ZZZZ, \tau_2 = 2000')

figure
contourf(cspec6.freq(1255:1310), cspec6.freq(1255:1310), ...)
real(cspec6.abs1(1255:1310, 1255:1310)), z)
xlim([470 580])
ylim([470 580])
caxis([-60000 60000])
axis square
set(gca, 'Tickdir', 'out');
set(gca, 'FontWeight', 'bold')
set(gca, 'FontSize', 16)
set(gca, 'FontName', 'Arial')
set(gca, 'TickLength', [0.03, 0.01])
set(gca, 'LineWidth', 2)
xlabel('\omega_1/2 \pi (cm^{-1})')
ylabel('\omega_3/2 \pi (cm^{-1})')
title('ZZZZ, \tau_2 = 3500')
figure
contourf(cspec7.freq(1255:1310), cspec7.freq(1255:1310), ...
real(cspec7.abs1(1255:1310, 1255:1310)), z)
xlim([470 580])
ylim([470 580])
caxis([-60000 60000])
axis square
set(gca, 'Tickdir', 'out');
set(gca, 'FontWeight', 'bold')
set(gca, 'FontSize', 16)
set(gca, 'FontName', 'Arial')
set(gca, 'TickLength', [0.03, 0.01])
set(gca, 'LineWidth', 2)
xlabel('\omega_1/2\pi (cm^{-1})')
ylabel('\omega_3/2\pi (cm^{-1})')
title('ZZZZ, \tau_2 = 5000')

figure
contourf(cspec8.freq(1255:1310), cspec8.freq(1255:1310), ...
real(cspec8.abs1(1255:1310, 1255:1310)), z)
xlim([470 580])
ylim([470 580])
caxis([-60000 60000])
axis square
set(gca, 'Tickdir', 'out');
set(gca, 'FontWeight', 'bold')
set(gca, 'FontSize', 16)
set(gca, 'FontName', 'Arial')
set(gca, 'TickLength', [0.03, 0.01])
set(gca,'LineWidth', 2)
xlabel('\omega_1/2\pi (cm^{-1})')
ylabel('\omega_3/2\pi (cm^{-1})')
title('ZZZZ, \tau_2 = 7000')

figure
contourf(cspec9.freq(1255:1310),cspec9.freq(1255:1310),...
real(cspec9.abs1(1255:1310,1255:1310)),z)
xlim([470 580])
ylim([470 580])
caxis([-60000 60000])
axis square
set(gca, 'Tickdir', 'out');
set(gca, 'FontWeight','bold')
set(gca,'FontSize',16)
set(gca,'FontName','Arial')
set(gca,'TickLength', [0.03, 0.01])
set(gca,'LineWidth', 2)
xlabel('\omega_1/2\pi (cm^{-1})')
ylabel('\omega_3/2\pi (cm^{-1})')
title('ZZZZ, \tau_2 = 10000')

figure(11)
subplot(2,3,1)
contourf(cspec1.freq(1220:1400),cspec1.freq(1220:1400),real(cspec1.abs1(1220:1400,1220:1400)),z)
axis square
xlim([470 580])
import matplotlib.pyplot as plt

plt.subplot(2, 3, 2)
plt.contourf(cspec2.freq[1220:1400], cspec2.freq[1220:1400], real(cspec2.abs1[1220:1400, 1220:1400]), z)
plt.axis('square')
plt.xlim([470, 580])
plt.ylim([470, 580])
plt.caxis([-40000, 40000])

plt.subplot(2, 3, 3)
plt.contourf(cspec3.freq[1220:1400], cspec3.freq[1220:1400], real(cspec3.abs1[1220:1400, 1220:1400]), z)
plt.axis('square')
plt.xlim([470, 580])
plt.ylim([470, 580])
plt.caxis([-40000, 40000])

plt.subplot(2, 3, 4)
plt.contourf(cspec4.freq[1220:1400], cspec4.freq[1220:1400], real(cspec4.abs1[1220:1400, 1220:1400]), z)
plt.axis('square')
plt.xlim([470, 580])
plt.ylim([470, 580])
plt.caxis([-40000, 40000])

plt.subplot(2, 3, 5)
contourf(cspec5.freq(1220:1400), cspec5.freq(1220:1400), real(cspec5.abs1(1220:1400,1220:1400)), z)

axis square
xlim([470 580])
ylim([470 580])
caxis([-40000 40000])
subplot(2,3,6)

contourf(cspec6.freq(1220:1400), cspec6.freq(1220:1400), real(cspec6.abs1(1220:1400,1220:1400)), z)

axis square
xlim([470 580])
ylim([470 580])
caxis([-40000 40000])

A.2.2 gaa.m

The lineshape function gbb.m follows a similar format with its own correlation amplitude and timescale (Figure 1.6 uses the same values for modes a and b).

function y = gaa( t )
% GAA Lineshape function g_{aa}
% See HL examples

t11 = 10.0e3; % Correlation time
s11 = 0.0005; % Amplitude from correlation function

y = s11^2*t11^2*(exp(-abs(t)/t11)-1.0+abs(t)/t11);
end
A.2.3  \texttt{ga2a.m}

The lineshape function \texttt{gb2b.m} follows a similar format.

\begin{verbatim}
1  function y = ga2a( t )
2  %GAA Lineshape function g\_{aa}
3
4  t11 = 10.0e3;
5  s11 = 0.0005;
6
7  y= 2.0*s11^2*t11^2*(exp(-abs(t)/t11)-1.0+abs(t)/t11);
8
9  end
\end{verbatim}

A.2.4  \texttt{g2a2a.m}

The lineshape function \texttt{g2b2b.m} follows a similar format.

\begin{verbatim}
1  function y = g2a2a( t )
2  %GAA Lineshape function g\_{aa}
3
4  t11 = 10.0e3;
5  s11 = 0.0005;
6
7  y= 4.0*s11^2*t11^2*(exp(-abs(t)/t11)-1.0+abs(t)/t11);
8
9  end
\end{verbatim}

A.2.5  \texttt{gab.m}

\begin{verbatim}
1  function y = gab( t )
2
3  %Gtt Lineshape function g\_{tt}
\end{verbatim}
A.2.6 gaab.m

The lineshape function gbab.m follows a similar format.

A.2.7 gabab.m
ta = 10.0e3;

sa = 0.0005;

tb = 10.0e3;

sb = 0.0005;

tab = 10.0e3;

sab = 0.0005;

rho = 1.0;

y = sa^2*ta^2*(exp(-abs(t)/ta)-1.0+abs(t)/ta)+...

sb^2*tb^2*(exp(-abs(t)/tb)-1.0+abs(t)/tb)+...

2.0*rho*sab^2*tab^2*(exp(-abs(t)/tab)-1.0+abs(t)/tab);

end

A.3 **Fifteen Level System in the Homogeneous Limit**

This is the most basic code for a system of four coupled vibrations. It is written with homogeneous lineshape functions for four parallel transition dipole moments. This code was used to produce Figure 1.9.

A.3.1 **fifteenLevelHomLim.m**

%% 15 Level Test - Four Coupled Dipoles

% This is an example and test of specPack for a 15 level system with
% homogeneous lineshapes. See the six level homogeneous example for
% more detailed comments.

%% Set Path

addpath('C:\Users\Karla\Documents\Grad School\Work\Research\Theses\Thesis\six-level examples for intro\specpack3')
%% Clear Memory

clear
clc

%% Define system parameters

% Energies (in cm^-1)
a=500; % 4 vibrations: a, b, c, d
b=540;
c=580;
d=620;
a2=970; % overtones
b2=1050;
c2=1130;
d2=1210;
ab=1020; % combination bands
ac=1060;
ad=1100;
bc=1100;
bd=1140;
cd=1180;

energies = [0 a b c d a2 b2 c2 d2 ab ac ad bc bd cd];

rt2 = sqrt(2.0);

mu = [ 0.0 1.0 1.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0;...
      %1
   0.0 0.0 0.0 0.0 rt2 0.0 0.0 0.0 1.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0;...
      %2
]
\begin{verbatim}
0.0 0.0 0.0 0.0 0.0 rt2 0.0 0.0 1.0 0.0 0.0 1.0 1.0 0.0; \ldots %3
0.0 0.0 0.0 0.0 0.0 rt2 0.0 0.0 1.0 0.0 0.0 1.0 0.0 1.0; \ldots %4
0.0 0.0 0.0 0.0 0.0 rt2 0.0 0.0 1.0 0.0 0.0 1.0 1.0 0.0; \ldots %5
rt2 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0; \ldots %6
0.0 rt2 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0; \ldots %7
0.0 0.0 rt2 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0; \ldots %8
0.0 0.0 0.0 rt2 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0; \ldots %9
1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0; \ldots %10
1.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0; \ldots %11
1.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0; \ldots %12
0.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0; \ldots %13
0.0 1.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0; \ldots %14
0.0 0.0 0.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0; \ldots %15

nstates = 15;
for i=1:nstates
    for j=1:nstates
        for k = 1:3
            muDir(i,j,k) = 0.0;
        end
    end
end

muDir(1,2,1) = 1.0;
muDir(2,1,1) = 1.0;
muDir(1,3,1) = 1.0;
muDir(3,1,1) = 1.0;
muDir(1,4,1) = 1.0;
muDir(4,1,1) = 1.0;
\end{verbatim}
muDir(1,5,1) = 1.0;
muDir(5,1,1) = 1.0;
muDir(2,6,1) = 1.0;
muDir(6,2,1) = 1.0;
muDir(3,7,1) = 1.0;
muDir(7,3,1) = 1.0;
muDir(4,8,1) = 1.0;
muDir(8,4,1) = 1.0;
muDir(5,9,1) = 1.0;
muDir(9,5,1) = 1.0;
muDir(2,10,1) = 1.0;
muDir(10,2,1) = 1.0;
muDir(2,11,1) = 1.0;
muDir(11,2,1) = 1.0;
muDir(2,12,1) = 1.0;
muDir(12,2,1) = 1.0;
muDir(3,10,1) = 1.0;
muDir(10,3,1) = 1.0;
muDir(3,13,1) = 1.0;
muDir(13,3,1) = 1.0;
muDir(3,14,1) = 1.0;
muDir(14,3,1) = 1.0;
muDir(4,11,1) = 1.0;
muDir(11,4,1) = 1.0;
muDir(4,13,1) = 1.0;
muDir(13,4,1) = 1.0;
muDir(4,15,1) = 1.0;
muDir(15,4,1) = 1.0;
muDir(5,12,1) = 1.0;
muDir(12,5,1) = 1.0;
muDir(5,14,1) = 1.0;
muDir(14,5,1) = 1.0;
muDir(5,15,1) = 1.0;
muDir(15,5,1) = 1.0;

rho0 = [1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0];

hfun = cell(3);
for i = 1:nstates
    for j = 1:nstates
        hfun{i,j}=@g0;
    end
end

hfun{2,2}=@gaaHL;
hfun{2,6}=@ga2aHL;
hfun{6,2}=@ga2aHL;
hfun{3,3}=@gbbHL;
hfun{3,7}=@gb2bHL;
hfun{7,3}=@gb2bHL;
hfun{4,4}=@gccHL;
hfun{4,8}=@gc2cHL;
hfun{8,4}=@gc2cHL;
hfun{5,5}=@gddHL;
hfun{5,9}=@gd2dHL;
123 \text{hfun}(9,5)=@gd2dHL;
124 \text{hfun}(6,6)=@g2a2aHL;
125 \text{hfun}(7,7)=@g2b2bHL;
126 \text{hfun}(8,8)=@g2c2cHL;
127 \text{hfun}(9,9)=@g2d2dHL;
128
129 \text{hfun}(2,3)=@gabHL;
130 \text{hfun}(3,2)=@gabHL;
131 \text{hfun}(2,10)=@gaabHL;
132 \text{hfun}(10,2)=@gaabHL;
133 \text{hfun}(3,10)=@gbabHL;
134 \text{hfun}(10,3)=@gbabHL;
135 \text{hfun}(10,10)=@gababHL;
136
137 \text{hfun}(2,4)=@gacHL;
138 \text{hfun}(4,2)=@gacHL;
139 \text{hfun}(2,11)=@gaacHL;
140 \text{hfun}(11,2)=@gaacHL;
141 \text{hfun}(4,11)=@gcacHL;
142 \text{hfun}(11,4)=@gcacHL;
143 \text{hfun}(11,11)=@gacacHL;
144
145 \text{hfun}(2,5)=@gadHL;
146 \text{hfun}(5,2)=@gadHL;
147 \text{hfun}(2,12)=@gaadHL;
148 \text{hfun}(12,2)=@gaadHL;
149 \text{hfun}(5,12)=@gdadHL;
150 \text{hfun}(12,5)=@gdadHL;
151 \text{hfun}(12,12)=@gadadHL;
152
153 \texttt{hfun(3,4)=@gbcHL;}
154 \texttt{hfun(4,3)=@gbcHL;}
155 \texttt{hfun(3,13)=@gbbcHL;}
156 \texttt{hfun(13,3)=@gbbcHL;}
157 \texttt{hfun(4,13)=@gcbcHL;}
158 \texttt{hfun(13,4)=@gcbcHL;}
159 \texttt{hfun(13,13)=@gbcbcHL;}
160
161 \texttt{hfun(3,5)=@gbdHL;}
162 \texttt{hfun(5,3)=@gbdHL;}
163 \texttt{hfun(3,14)=@gbbdHL;}
164 \texttt{hfun(14,3)=@gbbdHL;}
165 \texttt{hfun(5,14)=@gbdHL;}
166 \texttt{hfun(14,5)=@gbdHL;}
167 \texttt{hfun(14,14)=@gbdbhHL;}
168
169 \texttt{hfun(4,5)=@gcdHL;}
170 \texttt{hfun(5,4)=@gcdHL;}
171 \texttt{hfun(4,15)=@gccdHL;}
172 \texttt{hfun(15,4)=@gccdHL;}
173 \texttt{hfun(5,15)=@gdcHL;}
174 \texttt{hfun(15,5)=@gdcHL;}
175 \texttt{hfun(15,15)=@gcdcdHL;}
176
177 \texttt{dor = 2.0e-5;}
178
179 \texttt{\% Initialize structs}
180
sys = initSystem( nstates, energies, mu, muDir, rho0, hfun, dor);

expr = initExperiment([560 560 560 560], '2DIR', 'ZZZZ');

opts = sp_SetOptions('tstep',6.0);

%% Call 2D calculation function
tau2 = [0,200];
calc2DImpulse('fifteenLevel_test',sys,expr,tau2,opts);

%% Create Rephasing Time Plot
rph_time = load('fifteenLevel_test_t2-200_rt.mat');

figure(1)
contourf(rph_time.t1(1:400),rph_time.t3(1:400),...
real(rph_time.response_time(1:400,1:400)),10)
clear rph_time

%% Create NonRephasing Time plot
nr_time = load('fifteenLevel_test_t2-200_nrt.mat');

figure(2)
contourf(nr_time.t1(1:400),nr_time.t3(1:400),...
nr_time.response_time(1:400,1:400),10)
axis square
clear nr_time
rf = load('fifteenLevel_test_t2-200_rf.mat');

f = 0.25/2*linspace(0,1,2048/2+1)/(2.998e-5);

figure(3)
contourf(rf.freq,rf.freq,rf.response_freq,10)
clear rf

nrf = load('fifteenLevel_test_t2-200_nrf.mat');

figure(4)
contourf(nrf.freq,nrf.freq,nrf.response_freq,10)
axis square
clear nrf

cspec1 = load('fifteenLevel_test_t2-0_abs.mat');

z = [-65e3:4e3:65e3];

figure(5)
contourf(cspec1.freq(1194:1265),cspec1.freq(1194:1265),...
A.3.2 Lineshape Functions

The lineshape functions used to produce Figure 1.9 all follow that same format as the lineshape functions shown in Sections A.1.2 – A.1.7. Figure 1.9 used a ‘fac’ value of 2000 for all four modes for all possible lineshape functions.

A.4 Fitting 2D IR Simulations to Experimental Spectra

Following are the fitting files used to fit the FePtFe spectra in Chapter 2. They can be used as an example for setting up any fitting routine using this simulation code. There is an outside file where the parameters that are to be fit are defined and any constraints on them are placed. Then there is the function Matlab calls to which the experimental data is compared. Finally, this section has the file I used to constrain the angles between all four transition dipole moments.
A.4.1 fit_2D_FePtFe_angles_relative.m

This is the main front file where parameters to be fit are defined and the fitting function is called. This is also where all the plots to compare the simulated and experimental spectra are produced.

```matlab
1 clear all
2 clc %clear command window
3
4 global ym ys expPar energy
5
6 % Set Path - adds the path to the specPack3 function files
7 addpath('C:\Users\Karla\Documents\Documents\Grad School
            Work\Research\tests\Mikes 2D trimer data\Fitting\specPack3')
8
9 % timing starts
10 tic
11 starttime = fix(clock)
12
13 % This section loads experimental parameters and makes sure the
14 % variables match the spectra files. w1 and w3 are the freq axes, and
15 % c150_x variables are the spectra for the parallel and cross spectra
16 % for
17 % the 150 fs tau2 point.
18 load w3new
19 load ('w3_5220.txt')
20 w3_p=w3_5220(2:63);
21 w3_c=w3_5220(2:63);
22 load w1new
```
w1_p = w1new(46:178);
w1_c = w1new(46:178);
load c150new_p
load c150new_p2
load c150new_c
load c150new_c2

% Makes a 2D freq matrix that experimental spectra are interpolated to
[w1new1, w3new1] = meshgrid(w1new(46:178), w3new);

% Makes a 2D freq matrix from the original freq axes for the experimental
% spectra
[w1P, w3P] = meshgrid(w1_p, w3_p);
[w1C, w3C] = meshgrid(w1_c, w3_c);

% Averages the 2 spectra from each polarization, and then interpolates those
% new matrices to the new w1new1 and w3new1 freq axes
y1a = (c150new_p(:, 46:178) + c150new_p2(:, 46:178)) / 2; % this is the matrix of 2D spectra to fit
y2a = (c150new_c(:, 46:178) + c150new_c2(:, 46:178)) / 2;
y1 = interp2(w1new1, w3new1, y1a, w1P, w3P);
y2 = interp2(w1new1, w3new1, y2a, w1C, w3C);

ym(:, :, 1) = (y1 ./ max(max(y1))) + 0.01;
ym(:, :, 2) = y2 ./ max(max(y1));
ym(:, 106:end, :) = ym(:, 106:end, :) .* 4;
ym(50:end, :, :) = ym(50:end, :, :) .* 4;
%Takes slices from each averaged experimental spectra at the freq for each
%mode and puts them into a single matrix that the simulated spectra can be
%also be fit to if it gets difficult to fit the smaller dipole moments

ys(:,1)=y1(:,74)./max(y1(:,74));
ys(:,2)=y1(:,66)./max(y1(:,66));
ys(:,3)=y1(:,84)./max(y1(:,84));
ys(:,4)=y1(:,115)./max(y1(:,115));
ys(:,5)=y2(:,74)./max(y2(:,74));
ys(:,6)=y2(:,66)./max(y2(:,66));
ys(:,7)=y2(:,84)./max(y2(:,84));
ys(:,8)=y2(:,115)./max(y2(:,115));

name = 'FePtFe_fit2_angles'; %what you want your simulated files to be called
%this is user-defined

tau2 = 150; %the tau2 point of the spectra you're fitting the simulation too
dor=2.0e-5; %diffusion coefficient
%these are experimental parameters that are passed to the fitting function
expPar.w1P=w1P;
expPar.w3P=w3P;
expPar.w1C=w1C;
expPar.w3C=w3C;
expPar.name=name;
expPar.tau2=tau2;
expPar.dor=dor;

%This is the section where the simulation parameters are defined. Any or
%all of these might be fit. This examples is only fitting the dipole
%moments and the relative angles between the transition dipoles
A_t=0.0022;
A_r=0.0031;
A_a=0.007;  %the amplitudes for the correlation functions
A_b=0.0021;

tau_t=540;
tau_r=1035;
tau_a=4990;  %the timescales for the correlation functions
tau_b=251;

tau_tr=192;
tau_ta=1320;
tau_tb=177;
tau_ra=1100;
tau_rb=523;
tau_ab=1011;

T2=389200;  %the lorentzian component to each lineshape for each mode.
One of these can be defined for each mode.
% Energies (in cm^-1)
t=560; %4 vibrations: t,r,a,b
r=550;
a=574;
b=616;
an_t=25; %anharmonicities
an_r=21.5;
an_a=19;
an_b=24;
an_tr=2.2;
an_ta=13;
an_tb=9.6;
an_ra=9.3;
an_rb=2.01;
an_ba=10.6;
m_t=0.732; %dipole moment amplitudes
m_r=1;
m_a=0.313;
m_b=0.423;
%relative angles between most dipole transitions. For this example, the
%bridge was defined to be along the z axis, and the radial was constrained to the xz plane.
theta_rt = 89;
theta_bt = 3;
theta_ar = 82;
theta_br = 88;
theta_ba = 1;

%In case the amplitude between the parallel and crossed spectra isn't what you expected due to experimental issues. This can only be used when you have more than two coupled modes if you are trying to extract relative angles btwn transition dipoles
cross_corr = 0.882;

%the parameters that are currently being fit
f0 = [m_t m_a m_b ...
     theta_rt theta_bt theta_ar theta_br theta_ba];

%the upper and lower bounds placed on the parameters you're fitting
lb=[0.4, 0.2, 0.3,...
   50, 0, 50, 50, 0];
ub=[0.8, 0.4, 0.5...
   90, 40, 90, 90, 40];

%the options on for the fit
options=optimoptions(@fmincon,'Algorithm','interior-point','MaxFunEvals',1000,'MaxIter',2000,'Display','iter','tolfun',1E-100);

%Calling the fitting function with the parameters you're fitting, and their bounds, as well as any non-linear constraints placed on the parameters

[f,resnorm,residual,exitflag,output,lambda,jacobian]=fmincon('fifteenLevelFit_angles_relative',f0,[],[],[],[],lb,ub,@angle_const,options);

%timing ends
Toc
save f f
save output output

theta_rt = f(4);
theta_bt = f(5);
theta_ar = f(6);
theta_br = f(7);
theta_ba = f(8);

%calculates the final angle that wasn't necessary to fit
\[
\theta_{at} = \arccos\left(\cos(\theta_{ba})\cos(\theta_{bt}) + \left(\cos(\theta_{ar}) - \\
\cos(\theta_{ba})\cos(\theta_{br})\right)\left(\cos(\theta_{rt}) - \\
\cos(\theta_{br})\cos(\theta_{bt})\right)\right) / \left(\sin^2(\theta_{br})\right) \ldots
\]

\[
+ \sin(\theta_{ba})\sin(\theta_{bt})\sqrt{1 - \left(\cos(\theta_{ar}) - \\
\cos(\theta_{ba})\cos(\theta_{br})\right)^2 / \sin^2(\theta_{ba})\sin^2(\theta_{br})} \times \sqrt{1 - \left(\cos(\theta_{rt}) - \\
\cos(\theta_{br})\cos(\theta_{bt})\right)^2 / \sin^2(\theta_{br})\sin^2(\theta_{bt})}}
\]

175

% calculates confidence intervals on the fit parameters (kind of)
176
ci = nlarci(f, residual, 'jacobian', jacobian, 'alpha', 0.1);
177
cihalf = ((ci(:, 2) - ci(:, 1))./2)';
178
179
save ci ci
180
save cihalf cihalf
181
182
load yfit3 % matrix of simulated spectra along the same axes as the experimental spectra
183
184
185
% A whole bunch of figures to compare the fit to the experimental spectra
186
yfit4 = yfit3;
187
yfit4(:, 106:end, :) = yfit3(:, 106:end, :).*4;
188
yfit4(50:end, :, :) = yfit3(50:end, :, :).*4;
189
190
yfits(:, 1) = yfit4(:, 74, 1)./(max(yfit4(:, 74, 1)));
191
yfits(:, 2) = yfit4(:, 66, 1)./(max(yfit4(:, 66, 1)));
192
yfits(:, 3) = yfit4(:, 84, 1)./(max(yfit4(:, 84, 1)));
193
yfits(:, 4) = yfit4(:, 115, 1)./(max(yfit4(:, 115, 1)));
194
yfits(:,5)=yfit4(:,74,2)./max(yfit4(:,74,2));
yfits(:,6)=yfit4(:,66,2)./max(yfit4(:,66,2));
yfits(:,7)=yfit4(:,84,2)./max(yfit4(:,84,2));
yfits(:,8)=yfit4(:,115,2)./max(yfit4(:,115,2));

ym(:,:,2)=cross_corr*ym(:,:,2);

figure
plot(w3_p,ys(:,1),'o',w3_p,yfits(:,1))
title('trans, zzzz')

figure
plot(w3_p,ys(:,2),'o',w3_p,yfits(:,2))
title('radial, zzzz')

figure
plot(w3_p,ys(:,3),'o',w3_p,yfits(:,3))
title('axial, zzzz')

figure
plot(w3_p,ys(:,4),'o',w3_p,yfits(:,4))
title('bridge, zzzz')

figure
plot(w3_c,ys(:,5),'o',w3_c,yfits(:,5))
title('trans, yyzz')

figure
plot(w3_c,ys(:,6),'o',w3_c,yfits(:,6))
title('radial, yyzz')

figure
plot(w3_c,ys(:,7),'o',w3_c,yfits(:,7))
title('axial, yyzz')

figure
plot(w3_c,ys(:,8),'o',w3_c,yfits(:,8))
title('bridge, yyzz')

figure
plot(w3_p,yfit4(:,74,1),w3_p,ym(:,74,1),'o')
title('trans, zzzz')

figure
plot(w3_p,yfit4(:,66,1),w3_p,ym(:,66,1),'o')
title('radial, zzzz')

figure
plot(w3_p,yfit4(:,84,1),w3_p,ym(:,84,1),'o')
title('axial, zzzz')

figure
plot(w3_p,yfit4(:,115,1),w3_p,ym(:,115,1),'o')
title('bridge, zzzz')
figure
plot(w3_c,yfit4(:,74,2),w3_c,ym(:,74,2),'o')
title('trans, yyzz')

figure
plot(w3_c,yfit4(:,66,2),w3_c,ym(:,66,2),'o')
title('radial, yyzz')

figure
plot(w3_c,yfit4(:,84,2),w3_c,ym(:,84,2),'o')
title('axial, yyzz')

figure
plot(w3_c,yfit4(:,115,2),w3_c,ym(:,115,2),'o')
title('bridge, yyzz')

figure('Name','Fit')
subplot(1,2,1)
contourf(w1_p,w3_p,yfit4(:,:,1),15)
xlabel('\omega_1/2\pi (cm^{-1})')
ylabel('\omega_3/2\pi (cm^{-1})')
title('\tau_2 = 150, zzzz')
axis square
subplot(1,2,2)
contourf(w1_c,w3_c,yfit4(:,:,2),15)
xlabel('\omega_1/2\pi (cm^{-1})')
ylabel('\omega_3/2\pi (cm^{-1})')
title('\tau_2 = 150, yyzz')
A.4.2 angle_const.m

This is the file for some necessary nonlinear constraints on the values for the relative angles between transition dipole moments. If those angles are currently being fit, this function needs to be called when calling the fitting function as it is in the example from Section A.4.1.

```matlab
function [c, ceq] = angle_const(f)

% the angles that are being fit. Make sure the f(#) values are updated
if
 % the number of parameters you're fitting changes.
 theta_rt = f(4);
 theta_bt = f(5);
 theta_ar = f(6);
```
\begin{verbatim}
theta_br = f(7);
theta_ba = f(8);

%These inequalities have to be true from the geometry btwn 4 vectors
ineq1 = ((cosd(theta_ar) -
cosd(theta_ba)*cosd(theta_br))/(sind(theta_ba)*sind(theta_br)))^2;
ineq2 = ((cosd(theta_rt) -
cosd(theta_br)*cosd(theta_bt))/(sind(theta_br)*sind(theta_bt)))^2;

%Nonlinear inequality constraints (greater than zero)
c = [ineq1-1 ; ineq2-1];

%Nonlinear equality constraints
ceq = [];
\end{verbatim}

**A.4.3 fifteenLevelFit_angles_relative.m**

This is the fitting function that Matlab calls for the example given in Section A.4.1. It is currently only fitting the transition dipole moment amplitudes and the relative angles between them. Make sure all of the correct variables are made global, and that the correct values are updated when changing the parameters that are being fit.
global ym expPar energ A_t A_r A_a A_b tau_t tau_r tau_a tau_b tau_tr tau_ta tau_tb tau_ra tau_rb tau_ab T2

% Define Lineshape

Parameters

% Anything that isn't being fit should have an actual number. Anything that is being fit should have a value of fitvar(#) for this example. The # should correspond to the correct position that you defined in the main % front file where you called this fitting function.

A_t=0.0021;
A_r=0.0032;
A_a=0.0080;
A_b=0.0022;

tau_t=540;
tau_r=1030;
tau_a=4989;
tau_b=270;

tau_tr=200;
tau_ta=1313;
tau_tb=200;
tau_ra=1100;
tau_rb=500;
tau_ab=1000;
33 \ T2=389200;
34
35 \ cross\_corr = 0.87;
36 \ y=ym;
37 \ y(:, :, 2) = cross\_corr*y(:, :, 2); \% 'cross\_corr' can be a value that you fit
38
39 \ % Define system parameters
40 \ % Energies (in cm^-1)
41 \ % they are shifted down in energy by 1500 cm-1 to reduce calc time
42 \ t=energ.t; \% 4 vibrations: t, r, a, b
43 \ r=energ.r;
44 \ a=energ.a;
45 \ b=energ.b;
46 \ t2=2*t-24; \% overtones. All of the anharmonicities can also be fit and the values can be replaced with fitvar(\#)
47 \ r2=2*r-21;
48 \ a2=2*a-19;
49 \ b2=2*b-24;
50 \ tr=t+r-2.2; \% combination bands
51 \ ta=t+a-13.1;\%20;
52 \ tb=t+b-8.5;
53 \ ra=a+r-9;\%3;
54 \ rb=b+r-2;
55 \ ba=b+a-8.3;
56
57 \ % putting the above energies into a vector
58 \ energies = [0 t r a b t2 r2 a2 b2 tr ta tb ra rb ba];
% \mu values are being fit in this example

m_t = \text{fitvar}(1); % 0.732;
m_r = 1;
m_a = \text{fitvar}(2); % 0.313;
m_b = \text{fitvar}(3); % 0.423;

rt2 = \sqrt{2.0};
m2t = rt2 \times m_t;
m2r = rt2 \times m_r;
m2a = rt2 \times m_a;
m2b = rt2 \times m_b;

mu = [
  0.0   m_t   m_r   m_a   m_b   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0;
  m_t   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
]; %
nstates = 15;
for i=1:nstates
   for j=1:nstates
      for k = 1:3
         muDir(i,j,k) = 0.0;
      end
   end
end

%angles are also being fit in this example
theta_rt = fitvar(4);
theta_bt = fitvar(5);
theta_ar = fitvar(6);
theta_br = fitvar(7);
theta_ba = fitvar(8);

%Using geometry to assign the appropriate amplitude from each transition
dipole moment to the correct axes. The bridge mode is defined along
the
z-axis, and the radial mode is defined to the xz plane.
muDir(1,2,1) = (cosd(theta_rt) - cosd(theta_bt)*cosd(theta_br))./sind(theta_br);
muDir(2,1,1) = (cosd(theta_rt) - cosd(theta_bt)*cosd(theta_br))./sind(theta_br);
muDir(1,2,2) = sind(theta_bt)*sqrt(1-((cosd(theta_rt) - cosd(theta_br)*cosd(theta_bt))./(sind(theta_br)*sind(theta_bt)))^2);
\[\mu_{\text{Dir}}(2,1,2) = \sin(\theta_{bt}) \sqrt{1 - (\cos(\theta_{rt}) - \\
\cos(\theta_{br}) \cos(\theta_{bt})) / (\sin(\theta_{br}) \sin(\theta_{bt}))^2};\]

\[\mu_{\text{Dir}}(1,2,3) = \cos(\theta_{bt});\]

\[\mu_{\text{Dir}}(2,1,3) = \cos(\theta_{bt});\]

\[\mu_{\text{Dir}}(1,3,1) = \sin(\theta_{br});\]

\[\mu_{\text{Dir}}(3,1,1) = \sin(\theta_{br});\]

\[\mu_{\text{Dir}}(1,3,3) = \cos(\theta_{br});\]

\[\mu_{\text{Dir}}(3,1,3) = \cos(\theta_{br});\]

\[\mu_{\text{Dir}}(1,4,1) = (\cos(\theta_{ar}) - \\
\cos(\theta_{ba}) \cos(\theta_{br})) / \sin(\theta_{br});\]

\[\mu_{\text{Dir}}(4,1,1) = (\cos(\theta_{ar}) - \\
\cos(\theta_{ba}) \cos(\theta_{br})) / \sin(\theta_{br});\]

\[\mu_{\text{Dir}}(1,4,2) = \sin(\theta_{ba}) \sqrt{1 - (\cos(\theta_{ar}) - \\
\cos(\theta_{ba}) \cos(\theta_{br})) / (\sin(\theta_{ba}) \sin(\theta_{br}))^2};\]

\[\mu_{\text{Dir}}(4,1,2) = \sin(\theta_{ba}) \sqrt{1 - (\cos(\theta_{ar}) - \\
\cos(\theta_{ba}) \cos(\theta_{br})) / (\sin(\theta_{ba}) \sin(\theta_{br}))^2};\]

\[\mu_{\text{Dir}}(1,4,3) = \cos(\theta_{ba});\]

\[\mu_{\text{Dir}}(4,1,3) = \cos(\theta_{ba});\]

\[\mu_{\text{Dir}}(1,5,3) = 1.0;\]

\[\mu_{\text{Dir}}(5,1,3) = 1.0;\]

\[\mu_{\text{Dir}}(2,6,1) = (\cos(\theta_{rt}) - \\
\cos(\theta_{bt}) \cos(\theta_{br})) / \sin(\theta_{br});\]

\[\mu_{\text{Dir}}(6,2,1) = (\cos(\theta_{rt}) - \\
\cos(\theta_{bt}) \cos(\theta_{br})) / \sin(\theta_{br});\]
\begin{verbatim}
132  muDir(2,6,2) = sind(theta_bt)*sqrt(1-((cosd(theta_rt)-
cosd(theta_br)*cosd(theta_bt))./(sind(theta_br)*sind(theta_bt)))^2);
133  muDir(6,2,2) = sind(theta_bt)*sqrt(1-((cosd(theta_rt)-
cosd(theta_br)*cosd(theta_bt))./(sind(theta_br)*sind(theta_bt)))^2);
134  muDir(2,6,3) = cosd(theta_bt);
135  muDir(6,2,3) = cosd(theta_bt);
136
137  muDir(3,7,1) = sind(theta_br);
138  muDir(7,3,1) = sind(theta_br);
139  muDir(3,7,3) = cosd(theta_br);
140  muDir(7,3,3) = cosd(theta_br);
141
142  muDir(4,8,1) = (cosd(theta_ar)-
cosd(theta_ba)*cosd(theta_br))./sind(theta_br);
143  muDir(8,4,1) = (cosd(theta_ar)-
cosd(theta_ba)*cosd(theta_br))./sind(theta_br);
144  muDir(4,8,2) = sind(theta_ba)*sqrt(1-((cosd(theta_ar)-
cosd(theta_ba)*cosd(theta_br))./(sind(theta_ba)*sind(theta_br)))^2);
145  muDir(8,4,2) = sind(theta_ba)*sqrt(1-((cosd(theta_ar)-
cosd(theta_ba)*cosd(theta_br))./(sind(theta_ba)*sind(theta_br)))^2);
146  muDir(4,8,3) = cosd(theta_ba);
147  muDir(8,4,3) = cosd(theta_ba);
148
149  muDir(5,9,3) = 1.0;
150  muDir(9,5,3) = 1.0;
151
152  muDir(2,10,1) = sind(theta_br);
153  muDir(10,2,1) = sind(theta_br);
154  muDir(2,10,3) = cosd(theta_br);
\end{verbatim}
muDir(10,2,3) = cosd(theta_br);

muDir(2,11,1) = (cosd(theta_ar) -
    cosd(theta_pa)*cosd(theta_br))./sind(theta_br);
muDir(11,2,1) = (cosd(theta_ar) -
    cosd(theta_pa)*cosd(theta_br))./sind(theta_br);
muDir(2,11,2) = sind(theta_pa)*sqrt(1 - ((cosd(theta_ar) -
    cosd(theta_pa)*cosd(theta_br))./(sind(theta_pa)*sind(theta_br)))^2);
muDir(11,2,2) = sind(theta_pa)*sqrt(1 - ((cosd(theta_ar) -
    cosd(theta_pa)*cosd(theta_br))./(sind(theta_pa)*sind(theta_br)))^2);
muDir(2,11,3) = cosd(theta_pa);
muDir(11,2,3) = cosd(theta_pa);

muDir(2,12,3) = 1.0;
muDir(12,2,3) = 1.0;

muDir(3,10,1) = (cosd(theta_rt) -
    cosd(theta_bt)*cosd(theta_br))./sind(theta_br);
muDir(10,3,1) = (cosd(theta_rt) -
    cosd(theta_bt)*cosd(theta_br))./sind(theta_br);
muDir(3,10,2) = sind(theta_bt)*sqrt(1 - ((cosd(theta_rt) -
    cosd(theta_br)*cosd(theta_bt))./(sind(theta_br)*sind(theta_bt)))^2);
muDir(10,3,2) = sind(theta_bt)*sqrt(1 - ((cosd(theta_rt) -
    cosd(theta_br)*cosd(theta_bt))./(sind(theta_br)*sind(theta_bt)))^2);
muDir(3,10,3) = cosd(theta_bt);
muDir(10,3,3) = cosd(theta_bt);

muDir(3,13,1) = (cosd(theta_ar) -
    cosd(theta_pa)*cosd(theta_br))./sind(theta_br);
muDir(13,3,1) = (cosd(theta_ar) -
    cosd(theta钡)*cosd(theta钡) ./sind(theta钡);
muDir(3,13,2) = sind(theta钡)*sqrt(1 - ((cosd(theta_ar) -
    cosd(theta钡)*cosd(theta钡))./sind(theta钡)*sind(theta钡)))^2);
muDir(13,3,2) = sind(theta钡)*sqrt(1 - ((cosd(theta_ar) -
    cosd(theta钡)*cosd(theta钡))./sind(theta钡)*sind(theta钡)))^2);
muDir(3,13,3) = cosd(theta钡);
muDir(13,3,3) = cosd(theta钡);

muDir(3,14,3) = 1.0;
muDir(14,3,3) = 1.0;

muDir(4,11,1) = (cosd(theta_rt) -
    cosd(theta背)*cosd(theta背) ./sind(theta背);
muDir(11,4,1) = (cosd(theta_rt) -
    cosd(theta背)*cosd(theta背) ./sind(theta背);
muDir(4,11,2) = sind(theta背)*sqrt(1 - ((cosd(theta_rt) -
    cosd(theta背)*cosd(theta背))./sind(theta背)*sind(theta背)))^2);
muDir(11,4,2) = sind(theta背)*sqrt(1 - ((cosd(theta_rt) -
    cosd(theta背)*cosd(theta背))./sind(theta背)*sind(theta背)))^2);
muDir(4,11,3) = cosd(theta背);
muDir(11,4,3) = cosd(theta背);

muDir(4,13,1) = sind(theta背);
muDir(13,4,1) = sind(theta背);
muDir(4,13,3) = cosd(theta背);
muDir(13,4,3) = cosd(theta背);
muDir(4,15,3) = 1.0;
muDir(15,4,3) = 1.0;

muDir(5,12,1) = (cosd(theta_rt) -
cosd(theta_bt)*cosd(theta_br))./sind(theta_br);
muDir(12,5,1) = (cosd(theta_rt) -
cosd(theta_bt)*cosd(theta_br))./sind(theta_br);
muDir(5,12,2) = sind(theta_bt)*sqrt(1 - ((cosd(theta_rt) -
cosd(theta_br)*cosd(theta_bt))./(sind(theta_br)*sind(theta_br)))^2);
muDir(12,5,2) = sind(theta_bt)*sqrt(1 - ((cosd(theta_rt) -
cosd(theta_br)*cosd(theta_bt))./(sind(theta_br)*sind(theta_br)))^2);
muDir(5,12,3) = cosd(theta_bt);
muDir(12,5,3) = cosd(theta_bt);

muDir(5,14,1) = sind(theta_br);
muDir(14,5,1) = sind(theta_br);
muDir(5,14,3) = cosd(theta_br);
muDir(14,5,3) = cosd(theta_br);

muDir(5,15,1) = (cosd(theta_ar) -
cosd(theta_ba)*cosd(theta_br))./sind(theta_br);
muDir(15,5,1) = (cosd(theta_ar) -
cosd(theta_ba)*cosd(theta_br))./sind(theta_br);
muDir(5,15,2) = sind(theta_ba)*sqrt(1 - ((cosd(theta_ar) -
cosd(theta_ba)*cosd(theta_br))./(sind(theta_ba)*sind(theta_br)))^2);
muDir(15,5,2) = sind(theta_ba)*sqrt(1 - ((cosd(theta_ar) -
cosd(theta_ba)*cosd(theta_br))./(sind(theta_ba)*sind(theta_br)))^2);
muDir(5,15,3) = cosd(theta_ba);
muDir(15,5,3) = cosd(theta_ba);
all population starts out in the $v = 0$ state in this example

$$\rho_0 = [1.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0]$$;

$$hfun = \text{cell}(3);$$

for $i = 1:nstates$

for $j = 1:nstates$

$hfun(i,j) = @g0;$

end

end

All of the lineshape functions

$hfun(2,2) = @gttL;$$

$hfun(2,6) = @gt2tL;$$

$hfun(6,2) = @gt2tL;$$

$hfun(3,3) = @grrL;$$

$hfun(3,7) = @gr2rL;$$

$hfun(7,3) = @gr2rL;$$

$hfun(4,4) = @gaaL;$$

$hfun(4,8) = @ga2aL;$$

$hfun(8,4) = @ga2aL;$$

$hfun(5,5) = @gbbL;$$

$hfun(5,9) = @gb2bL;$$

$hfun(9,5) = @gb2bL;$$

$hfun(6,6) = @g2t2tL;$$

$hfun(7,7) = @g2r2rL;$$

$hfun(8,8) = @g2a2aL;$$

$hfun(9,9) = @g2b2bL;$$
hfun(2,3)=@gtrL;
hfun(3,2)=@gtrL;
hfun(2,10)=@gttrL;
hfun(10,2)=@gttrL;
hfun(3,10)=@grtrL;
hfun(10,3)=@grtrL;
hfun(10,10)=@gtrtrL;

hfun(2,4)=@gtbL;
hfun(4,2)=@gtbL;
hfun(2,11)=@gttbL;
hfun(11,2)=@gttbL;
hfun(5,12)=@gbtbL;
hfun(12,5)=@gbtbL;
hfun(12,12)=@gtbtbL;

hfun(3,4)=@graL;
hfun(4,3)=@graL;
hfun(3,13)=@grraL;
hfun(13,3)=@grraL;
hfun(4,13)=@garaL;
dor = expPar.dor;

% Initialize structs
sys = initSystem( nstates, energies, mu, muDir, rho0, hfun, dor);

expr = initExperiment([550 550 550 550], '2DIR', 'YYZZ');

opts = sp_SetOptions('tstep',8.0); %Open sp_SetOptions and manually adjust these values
% Call 2D calculation function

tau2 = expPar.tau2;
calc2DImpulse([expPar.name,'_YYZZ'],sys,expr,tau2,opts);

% Initialize structs

sys = initSystem( nstates, energies, mu, muDir, rho0, hfun, dor);

expr = initExperiment([550 550 550 550], '2DIR', 'ZZZZ');

opts = sp_SetOptions('tstep',8.0);

% Call 2D calculation function

tau2 = expPar.tau2;
calc2DImpulse([expPar.name,'_ZZZZ'],sys,expr,tau2,opts);

% Loads the simulated spectra just calculated

c = load([expPar.name,'_YYZZ','_t2-150_abs']);
p = load([expPar.name,'_ZZZZ','_t2-150_abs']);

% Makes a 2D matrix of the freq axes for the simulated spectra (the +1500 is
% to bring the freq axis back to the correct range - see line 42)

[fx,fy]=meshgrid(c.freq(1025:end)+1500);

% Puts simulated spectra onto the same freq axes as the experimental spectra


yfit1 = interp2(fx, fy, real(p.abs1(1025:end, 1025:end)), expPar.w1P, expPar.w3P);

yfit2 = interp2(fx, fy, real(c.abs1(1025:end, 1025:end)), expPar.w1C, expPar.w3C);

yfit3(:,:,1) = yfit1./max(max(yfit1));
yfit3(:,:,2) = yfit2./max(max(yfit1));

yfit4 = yfit3;

% trying to get a better fit on the bridge region by making its amplitude larger
yfit4(:,106:end,:) = yfit3(:,106:end,:).*4;
yfit4(50:end,:,:)=yfit3(50:end,:,:).*4;

save yfit3 yfit3
save fitvar fitvar

% Taking the difference between the simulated and experimental spectra
diff_f = (yfit4-y);

sq_diff = diff_f.^2;

% summing up the square of the difference along all three dimensions to get
% a single value to try to minimize
\begin{verbatim}
retf = sum(sum(sum(sq_diff))); 

A.4.4 gttL.m 
This is an example of the lineshape functions used FePtFe. The other three modes had lineshape functions with similar formats, but with their own A and tau values. These would include the grrL.m, gaaL.m, and gbbL.m functions.

function y = gttL(t)
    global A_t tau_t T2
    %If you aren't using the fitting files with defined global variables, you
    %need to explicitly define these values.
    % A_t=0.001;
    % tau_t=1000;
    % T2=135; %You can define different T2 values for each mode if they have
    % different Lorentzian percentages

    y = A_t^2*tau_t^2*(exp(-abs(t)/tau_t)+abs(t)/tau_t-1)... 
        +(abs(t)/T2)*(0.5);

end

A.4.5 gt2tL.m 

The other three modes had lineshape functions with similar formats, but again with their own A and tau values. These would include the gr2rL.m, ga2aL.m, and gb2bL.m functions.
\end{verbatim}
function y = gt2tL( t )

global A_t tau_t T2

%If you aren't using the fitting files with defined global variables, you
%need to explicitly define these values.
%A_t=0.001;
%tau_t=1000;
%T2t=135;

y = 2*A_t^2*tau_t^2*(exp(-abs(t)/tau_t)+abs(t)/tau_t-1)...
+2*(abs(t)./T2)*(0.5);
end

A.4.6 g2t2tL.m

The other three modes had lineshape functions with similar formats, but again with their own A and tau values. These would include the g2r2rL.m, g2a2aL.m, and g2b2bL.m functions.

function y = g2t2tL( t )

global A_t tau_t T2

%If you aren't using the fitting files with defined global variables, you
%need to explicitly define these values.
%A_t=0.001;
%tau_t=1000;
% T2t=135;

d y = 4*A_t^2*tau_t^2*(exp(-abs(t)/tau_t)+abs(t)/tau_t-1)...
  +4*(abs(t)./T2)*(0.5);
end

A.4.7 \textit{gtrL.m}

There will be five other lineshape functions with similar formats: \textit{gtaL.m}, \textit{gtbL.m}, \textit{graL.m}, \textit{grbL.m}, and \textit{gabL.m}. Each of these will have two A values, the same as used in the lineshape functions in Sections A.4.5 – A.4.7, and a new tau value.

function y = gtrL( t )

global A_t A_r tau_tr T2

%If you aren't using the fitting files with defined global variables, you
%need to explicitly define these values.
%A_t=0.001;
%A_r=0.0018;
%tau_tr=1000;
%T2=135;

y = A_t*A_r*tau_tr^2*(exp(-abs(t)/tau_tr)+abs(t)/tau_tr-1)...
 + (abs(t)./T2)*(0.5);
end
A.4.8  \textit{gttrL.m}

There will be 11 other lineshape functions with similar formats. All values for this will have been used in other types of lineshape functions so make sure the correct variables are called.

\begin{verbatim}
function y = gttrL( t )
global A_t A_r tau_t tau_tr T2

%If you aren't using the fitting files with defined global variables, you
%need to explicitly define these values.
%A_t=0.001;
%A_r=0.0018;%0.0018
%tau_t=1000;
%tau_tr=1000;
%T2=6000;

y = A_t^2*tau_t^2.*(exp(-abs(t)./tau_t)+abs(t)./tau_t-1)...
+A_t*A_r*tau_tr^2.*(exp(-abs(t)./tau_tr)+abs(t)./tau_tr-1)...
+(abs(t)./T2)*(1);
end
\end{verbatim}

A.4.9  \textit{gtrtrL.m}

Five other lineshape functions follow the same format: gtataL.m, gtbtbL.m, graraL.m, grbrbL.m, and gababL.m. Again, all variables should match the values used in other lineshape functions.

\begin{verbatim}
function y = gtrtrL( t )
global A_t A_r tau_t tau_r tau_tr T2

y = A_t^2*tau_t^2.*(exp(-abs(t)./tau_t)+abs(t)./tau_t-1)...
+A_t*A_r*tau_tr^2.*(exp(-abs(t)./tau_tr)+abs(t)./tau_tr-1)...
+(abs(t)./T2)*(1);
end
\end{verbatim}
%If you aren't using the fitting files with defined global variables, you
%need to explicitly define these values.

% A_t=0.001;
% A_r=0.0018;%0.0018
% tau_t=1000;
% tau_r=1800;
% tau_tr=1000;
% T2=6000;

y = A_t^2*tau_t^2.*(exp(-abs(t)./tau_t)+abs(t)./tau_t-1)...
+A_r^2*tau_r^2.*(exp(-abs(t)./tau_r)+abs(t)./tau_r-1)...
+2*A_t*A_r*tau_tr^2.*(exp(-abs(t)./tau_tr)+abs(t)./tau_tr-1)...
+(abs(t)./T2)*(2);
end

A.5 SPECPACK3

These code files are used for all types of systems with minimal changes needed between types. Updates need to be made to sp_SetOptions.m. Updates may need to be made to the threshold in buildRot.m. The function files are presented in alphabetical order.

A.5.1 buildEcho.m

This function file builds the echo (rephasing) response functions.

function rfn = buildEcho(sys, expr, rwaThresh, prefacThresh, l_poprlx)
%BUILDDECHO construct photon echo response function
% Description: This function takes system and experimental parameters, and it constructs a response function for the ks = -k1+k2+k3 phase matching

% Parameters: sys - Structure containing system parameters
exp - Structure containing experimental parameters
rwaThresh - Threshold for resonance for RWA (units = cm^-1)
prefacThresh - product of rho0 and dipoles is assumed to be 0 if smaller than prefacThresh
l_poprlx - logical controlling population relaxation

% Returns: rfn - structure with response function parameters
rfn.nterm - number of non-zero terms in response function
rfn.rtype - array with rfn types
rfn.aa - array with 1st eigenstate for each function
rfn.bb - array with 2nd eigenstate for each function
rfn.cc - array with 3rd eigenstate for each function
rfn.dd - array with 4th eigenstate for each function
rfn.w1 - array with t1 frequency for each function
rfn.w2 - array with t2 frequency for each function
rfn.w3 - array with t3 frequency for each function
rfn.rho0 - population in state a at t = 0
rfn.T1_1 - array with population relaxation contrib to t1
rfn.T1_2 - array with population relaxation contrib to t2
rfn.T1_3 - array with population relaxation contrib to t3
rfn.nterm = 0;  % Number of terms in Response function
sizeGuess = sys.nStates*6;  % Guess of number of rfn terms

% Begin Timing
ticID = tic;

% Preallocated rfs parameter arrays
rfn.rtype = zeros(1,sizeGuess);
rfn.aa = zeros(1,sizeGuess);
rfn.bb = zeros(1,sizeGuess);
rfn.cc = zeros(1,sizeGuess);
rfn.dd = zeros(1,sizeGuess);
rfn.w1 = zeros(1,sizeGuess);
rfn.w2 = zeros(1,sizeGuess);
rfn.w3 = zeros(1,sizeGuess);
rfn.rho0 = zeros(1,sizeGuess);
rfn.nYTerms = zeros(1,sizeGuess);

% Checks for each type of response function, R1 through R8 (JPCA, 2003, 107,
% 5258-5279). Compares prefactors and differences btwn frequencies against
% prefacThresh and RWA threshold.
for a=1:sys.nStates
for b=1:sys.nStates
for c=1:sys.nStates
for d=1:sys.nStates
% Check for R1 Members
prefac = sys.mu(c,d)*sys.mu(b,c)*sys.mu(a,b)*sys.mu(a,d)*...
sys.rho0(a);
if (prefac > prefacThresh)
  if (abs(sys.w(d,c) - expr.w(4)) <= rwaThresh)
    if (abs(sys.w(d,b) - (expr.w(2) - expr.w(1))) ... <= rwaThresh)
      if (abs(sys.w(d,a) + expr.w(1)) <= rwaThresh)
        rfn.nterm = rfn.nterm + 1;
        rfn.rtype(rfn.nterm) = 1;
        rfn.aa(rfn.nterm) = a;
        rfn.bb(rfn.nterm) = b;
        rfn.cc(rfn.nterm) = c;
        rfn.dd(rfn.nterm) = d;
        rfn.w1(rfn.nterm) = sys.w(d,a);
        rfn.w2(rfn.nterm) = sys.w(d,b);
        rfn.w3(rfn.nterm) = sys.w(d,c);
        rfn.rho0(rfn.nterm) = sys.rho0(a);
        [nYTmp, yFacTmp, yTypeTmp] = ...
        buildRot(a,b,c,d,sys.muDir,...
        rfn.rtype(rfn.nterm), expr.polar);
        rfn.nYTerms(rfn.nterm) = nYTmp;
        rfn.yFac(rfn.nterm,1:nYTmp) = yFacTmp;
        rfn.yType(rfn.nterm,1:nYTmp) = yTypeTmp;
        rfn.prefac(rfn.nterm) = prefac;
        if l_poprlx %if set to true in sp_SetOptions.m, then adds in pop rlx component
          rfn.T1_1(rfn.nterm) = 0.5/sys.popRlx(d) + 0.5/sys.popRlx(a);
          rfn.T1_2(rfn.nterm) = 0.5/sys.popRlx(d) + 0.5/sys.popRlx(b);
          rfn.T1_3(rfn.nterm) = 0.5/sys.popRlx(d) + 0.5/sys.popRlx(c);
          end
84     end
85     end
86     end
87     end
88     end
89     end
90     end
91     end
92     end
93     end
94     end
95     end
96     end
97     end
98     end
99     end
100    end
101    end
102    end
103    end
104    end
105    end
106    end
107    end
108    end
109    end
110    end
111    end
112    end

% Check for R2 Members
prefac = sys.mu(c,b)*sys.mu(d,c)*sys.mu(b,a)*sys.mu(a,d)*...
sys.rho0(a);
if (prefac > prefacThresh)
    if (abs(sys.w(b,c) - expr.w(4)) <= rwaThresh)
        if (abs(sys.w(d,b) + (expr.w(2) - expr.w(1))) ...) <= rwaThresh)
            if (abs(sys.w(d,a) - expr.w(1)) <= rwaThresh)
                rfn.nterm = rfn.nterm + 1;
                rfn.rtype(rfn.nterm) = 2;
                rfn.aa(rfn.nterm) = a;
                rfn.bb(rfn.nterm) = b;
                rfn.cc(rfn.nterm) = c;
                rfn.dd(rfn.nterm) = d;
                rfn.w1(rfn.nterm) = sys.w(d,a);
                rfn.w2(rfn.nterm) = sys.w(d,b);
                rfn.w3(rfn.nterm) = sys.w(b,c);
                rfn.rho0(rfn.nterm) = sys.rho0(a);
                [nYTmp, yFacTmp, yTypeTmp] = ...
                buildRot(a,b,c,d,sys.muDir,...
                rfn.rtype(rfn.nterm), expr.polar);
                rfn.nYTerms(rfn.nterm) = nYTmp;
                rfn.yFac(rfn.nterm,1:nYTmp) = yFacTmp;
                rfn.yType(rfn.nterm,1:nYTmp) = yTypeTmp;
                rfn.prefac(rfn.nterm) = prefac;
if l_popRlx
  rfn.T1_1(rfn.nterm) = 0.5/sys.popRlx(d) + 0.5/sys.popRlx(a);
  rfn.T1_2(rfn.nterm) = 0.5/sys.popRlx(d) + 0.5/sys.popRlx(b);
  rfn.T1_3(rfn.nterm) = 0.5/sys.popRlx(b) + 0.5/sys.popRlx(c);
end
end
end
end

% Check for R3 Members
prefac = sys.mu(c,b)*sys.mu(b,a)*sys.mu(d,c)*sys.mu(d,a)*...
    sys.rho0(a);
if (prefac > prefacThresh)
  if (abs(sys.w(b,c) - expr.w(4)) <= rwaThresh)
    if (abs(sys.w(c,a) + (expr.w(2) - expr.w(1))) ...
        <= rwaThresh)
      if (abs(sys.w(d,a) - expr.w(1)) <= rwaThresh)
        rfn.nterm = rfn.nterm + 1;
        rfn.rtype(rfn.nterm)= 3;
        rfn.aa(rfn.nterm)= a;
        rfn.bb(rfn.nterm)= b;
        rfn.cc(rfn.nterm)= c;
        rfn.dd(rfn.nterm)= d;
        rfn.w1(rfn.nterm)= sys.w(d,a);
        rfn.w2(rfn.nterm)= sys.w(c,a);
        rfn.w3(rfn.nterm)= sys.w(b,c);
        rfn.rho0(rfn.nterm)= sys.rho0(a);
        [nYTmp, yFacTmp, yTypeTmp] = ... 
        buildRot(a,b,c,d,sys.muDir,...

173
rfn.rtype(rfn.nterm), expr.polar);
rfn.nYTerms(rfn.nterm) = nYTmp;
rfn.yFac(rfn.nterm,1:nYTmp) = yFacTmp;
rfn.yType(rfn.nterm,1:nYTmp) = yTypeTmp;
rfn.prefac(rfn.nterm) = prefac;
if l_poprlx
  rfn.T1_1(rfn.nterm) = 0.5/sys.popRlx(d) + 0.5/sys.popRlx(a);
  rfn.T1_2(rfn.nterm) = 0.5/sys.popRlx(a) + 0.5/sys.popRlx(c);
  rfn.T1_3(rfn.nterm) = 0.5/sys.popRlx(b) + 0.5/sys.popRlx(c);
end
end
end
end
%
Check for R4 Members
prefac = sys.mu(a,b)*sys.mu(b,c)*sys.mu(c,d)*sys.mu(d,a)*...
sys.rho0(a);
if (prefac > prefacThresh)
  if (abs(sys.w(b,a) - expr.w(4)) <= rwaThresh)
    if (abs(sys.w(c,a) - (expr.w(2) - expr.w(1))) ...
      <= rwaThresh)
      if (abs(sys.w(d,a) + expr.w(1)) <= rwaThresh)
        rfn.nterm = rfn.nterm + 1;
        rfn.rtype(rfn.nterm)= 4;
        rfn.aa(rfn.nterm)= a;
        rfn.bb(rfn.nterm)= b;
        rfn.cc(rfn.nterm)= c;
        rfn.dd(rfn.nterm)= d;
        rfn.w1(rfn.nterm)= sys.w(d,a);
rfn.w2(rfn.nterm) = sys.w(c,a);
rfn.w3(rfn.nterm) = sys.w(b,a);
rfn.rho0(rfn.nterm) = sys.rho0(a);

[nYTmp, yFacTmp, yTypeTmp] = ...
buildRot(a,b,c,d,sys.muDir,...
rfn.rtype(rfn.nterm), expr.polar);
rfn.nYTerms(rfn.nterm) = nYTmp;
rfn.yFac(rfn.nterm,1:nYTmp) = yFacTmp;
rfn.yType(rfn.nterm,1:nYTmp) = yTypeTmp;
rfn.prefac(rfn.nterm) = prefac;
if l_poprlx
rfn.T1_1(rfn.nterm) = 0.5/sys.popRlx(d) + 0.5/sys.popRlx(a);
rfn.T1_2(rfn.nterm) = 0.5/sys.popRlx(c) + 0.5/sys.popRlx(a);
rfn.T1_3(rfn.nterm) = 0.5/sys.popRlx(b) + 0.5/sys.popRlx(a);
end
end
end
end
% Check for R5 (R1*) Members
prefac = sys.mu(c,d)*sys.mu(b,c)*sys.mu(a,b)*sys.mu(a,d)*...
sys.rho0(a);
if (prefac > prefacThresh)
if (abs(sys.w(d,c) + expr.w(4)) <= rwaThresh)
if (abs(sys.w(d,b) + (expr.w(2) - expr.w(1))) ...
<= rwaThresh)
if (abs(sys.w(d,a) - expr.w(l)) <= rwaThresh)
rfn.nterm = rfn.nterm + 1;
rfn.rtype(rfn.nterm) = 5;
rfn.aa(rfn.nterm) = a;
rfn.bb(rfn.nterm) = b;
rfn.cc(rfn.nterm) = c;
rfn.dd(rfn.nterm) = d;
rfn.w1(rfn.nterm) = sys.w(d,a);
rfn.w2(rfn.nterm) = sys.w(d,b);
rfn.w3(rfn.nterm) = sys.w(d,c);
rfn.rho0(rfn.nterm) = sys.rho0(a);
[nYTmp, yFacTmp, yTypeTmp] = ...
buildRot(a,b,c,d,sys.muDir,...
rfn.rtype(rfn.nterm), expr.polar);
rfn.nYTerms(rfn.nterm) = nYTmp;
rfn.yFac(rfn.nterm,1:nYTmp) = yFacTmp;
rfn.yType(rfn.nterm,1:nYTmp) = yTypeTmp;
rfn.prefac(rfn.nterm) = prefac;
if l_poprlx
rfn.T1_1(rfn.nterm) = 0.5/sys.popRlx(d) + 0.5/sys.popRlx(a);
rfn.T1_2(rfn.nterm) = 0.5/sys.popRlx(d) + 0.5/sys.popRlx(b);
rfn.T1_3(rfn.nterm) = 0.5/sys.popRlx(d) + 0.5/sys.popRlx(c);
end
end
end
end
% Check for R6 (R2*) Members
prefac = sys.mu(c,b)*sys.mu(d,c)*sys.mu(b,a)*sys.mu(a,d)*...
sys.rho0(a);
if (prefac > prefacThresh)
if (abs(sys.w(b,c) + expr.w(4)) <= rwaThresh)
if (abs(sys.w(d,b) - (expr.w(2) - expr.w(1))) ... 
<= rwaThresh)
if (abs(sys.w(d,a) + expr.w(1)) <= rwaThresh)
rfn.nterm = rfn.nterm + 1;
rfn.rtype(rfn.nterm)= 6;
rfn.aa(rfn.nterm)= a;
rfn.bb(rfn.nterm)= b;
rfn.cc(rfn.nterm)= c;
rfn.dd(rfn.nterm)= d;
rfn.w1(rfn.nterm)= sys.w(d,a);
rfn.w2(rfn.nterm)= sys.w(d,b);
rfn.w3(rfn.nterm)= sys.w(b,c);
rfn.rho0(rfn.nterm)= sys.rho0(a);
[nYTmp, yFacTmp, yTypeTmp] = ... 
buildRot(a,b,c,d,sys.muDir,...
rfn.rtype(rfn.nterm), expr.polar);
rfn.nYTerms(rfn.nterm) = nYTmp;
rfn.yFac(rfn.nterm,1:nYTmp) = yFacTmp;
rfn.yType(rfn.nterm,1:nYTmp) = yTypeTmp;
rfn.prefac(rfn.nterm) = prefac;
if l_poprlx
rfn.T1_1(rfn.nterm) = 0.5/sys.popRlx(d) + 0.5/sys.popRlx(a);
rfn.T1_2(rfn.nterm) = 0.5/sys.popRlx(d) + 0.5/sys.popRlx(b);
rfn.T1_3(rfn.nterm) = 0.5/sys.popRlx(b) + 0.5/sys.popRlx(c);
end
end
end
end
end
% Check for R7 (R3*) Members
prefac = sys.mu(c,b)*sys.mu(b,a)*sys.mu(d,c)*sys.mu(d,a)*...
sys.rho0(a);
if (prefac > prefacThresh)
    if (abs(sys.w(b,c) + expr.w(4)) <= rwaThres)
        if (abs(sys.w(c,a) - (expr.w(2) - expr.w(1))) ...
            <= rwaThresh)
            if (abs(sys.w(d,a) + expr.w(1)) <= rwaThresh)
                rfn.nterm = rfn.nterm + 1;
                rfn.rtype(rfn.nterm)= 7;
                rfn.aa(rfn.nterm)= a;
                rfn.bb(rfn.nterm)= b;
                rfn.cc(rfn.nterm)= c;
                rfn.dd(rfn.nterm)= d;
                rfn.w1(rfn.nterm)= sys.w(d,a);
                rfn.w2(rfn.nterm)= sys.w(c,a);
                rfn.w3(rfn.nterm)= sys.w(b,c);
                rfn.rho0(rfn.nterm)= sys.rho0(a);
                [nYTmp, yFacTmp, yTypeTmp] = ...
                buildRot(a,b,c,d,sys.muDir,...
                rfn.rtype(rfn.nterm), expr.polar);
                rfn.nYTerms(rfn.nterm) = nYTmp;
                rfn.yFac(rfn.nterm,1:nYtmp) = yFacTmp;
                rfn.yType(rfn.nterm,1:nYtmp) = yTypeTmp;
                rfn.prefac(rfn.nterm) = prefac;
    if l_poprlx
        rfn.T1_1(rfn.nterm) = 0.5/sys.popRlx(d) + 0.5/sys.popRlx(a);
        rfn.T1_2(rfn.nterm) = 0.5/sys.popRlx(c) + 0.5/sys.popRlx(a);
        rfn.T1_3(rfn.nterm) = 0.5/sys.popRlx(b) + 0.5/sys.popRlx(c);
end

end

end

% Check for R8 (R4*) Members

defac = sys.mu(a,b)*sys.mu(b,c)*sys.mu(c,d)*sys.mu(d,a)*...
sys.rho0(a);

if (prefac > prefacThresh)
  if (abs(sys.w(b,a) + expr.w(4)) <= rwaThresh)
    if (abs(sys.w(c,a) + (expr.w(2) - expr.w(1))) ...
      <= rwaThresh)
      if (abs(sys.w(d,a) - expr.w(1)) <= rwaThresh)
        rfn.nterm = rfn.nterm + 1;
        rfn.rtype(rfn.nterm) = 8;
        rfn.aa(rfn.nterm) = a;
        rfn.bb(rfn.nterm) = b;
        rfn.cc(rfn.nterm) = c;
        rfn.dd(rfn.nterm) = d;
        rfn.w1(rfn.nterm) = sys.w(d,a);
        rfn.w2(rfn.nterm) = sys.w(c,a);
        rfn.w3(rfn.nterm) = sys.w(b,a);
        rfn.rho0(rfn.nterm) = sys.rho0(a);
        [nYTmp, yFacTmp, yTypeTmp] = ... buildRot(a,b,c,d,sys.muDir, ...
        rfn.rtype(rfn.nterm), expr.polar);
        rfn.nYTerms(rfn.nterm) = nYTmp;
        rfn.yFac(rfn.nterm,1:nYTmp) = yFacTmp;
        rfn.yType(rfn.nterm,1:nYTmp) = yTypeTmp;
rfn.prfac(rfn.nterm) = prefac;
if l_poprlx
  rfn.T1_1(rfn.nterm) = 0.5/sys.popRlx(d) + 0.5/sys.popRlx(a);
  rfn.T1_2(rfn.nterm) = 0.5/sys.popRlx(c) + 0.5/sys.popRlx(a);
  rfn.T1_3(rfn.nterm) = 0.5/sys.popRlx(b) + 0.5/sys.popRlx(a);
end
end
end
end
end
end
end
end
end
end
%
if (rfn.nterm < sizeGuess)
  rfn.rtype(rfn.nterm+1:sizeGuess) = [];
  rfn.aa(rfn.nterm+1:sizeGuess) = [];
  rfn.bb(rfn.nterm+1:sizeGuess) = [];
  rfn.cc(rfn.nterm+1:sizeGuess) = [];
  rfn.dd(rfn.nterm+1:sizeGuess) = [];
  rfn.w1(rfn.nterm+1:sizeGuess) = [];
  rfn.w2(rfn.nterm+1:sizeGuess) = [];
  rfn.w3(rfn.nterm+1:sizeGuess) = [];
  rfn.rho0(rfn.nterm+1:sizeGuess) = [];
  rfn.nYTerms(rfn.nterm+1:sizeGuess) = [];
end
% Convert wave numbers to inverse femtoseconds (for each type of tau
delay)

\[ \text{cm2fs} = 2\pi \times 2.998 \times 10^{-5}; \]

\[ \text{rfn.w1} = \text{rfn.w1} \times \text{cm2fs}; \]

\[ \text{rfn.w2} = \text{rfn.w2} \times \text{cm2fs}; \]

\[ \text{rfn.w3} = \text{rfn.w3} \times \text{cm2fs}; \]

% Add Lineshape functions to response function

\[ \text{rfn.hfun} = \text{sys.hfun}; \]

% Add Rotational Diffusion coefficient to Response function

\[ \text{rfn.dor} = \text{sys.dor}; \]

% End timing

\[ \text{time_elapsed} = \text{toc(ticID)}; \]

\[ \text{outStr} = \text{strcat('Time elapsed for buildEcho: '},\text{num2str(time_elapsed)}); \]

\[ \text{disp(outStr)}; \]

end % buildEcho

A.5.2 \textit{buildRot.m}

This function file builds the orientational response portion of the response function.

\begin{verbatim}
function [nYTerms, yFac, yType] = buildRot(a, b, c, d,...
dipoleDir, rType, polarization)
% BUILDROT - build oritentational contribution to the response funciton
% Each of the pieces in this code can be viewed in a more straight-forward
% manner in Table II of JCP, 2001, 115, 297. The code view each possible
\end{verbatim}
% contribution as orthogonal transition dipole moments, which is why the
% contribution each dipole made to the 3 cardinal directions had to be
% calculated in the matrix muDir when inputing the system parameters.

% termThresh is currently hard coded to be 1.0%, but this can be
% changed if
% needed.
termThresh = 0.01;

%For each case (which are determined in buildEcho.m, it calculates the
%product of the directions for each transition in all 3 dimensions
%(molecules can be oriented in any direction). This is 'yCosTerm'.

%Determines the type of orientation of orientational contribution based
%on
%allowed directions/dimensions for polarizations of light
interactions).
%This is in 'selectYtype' towards bottom of file.

%Terms are added for each case that meets all the requirements.
nYTerms = 0;
switch rType
  case 1
    for ii = 1:3
      for jj = 1:3
        for kk = 1:3
          for ll = 1:3
            yCosTerm = dipoleDir(c,d,ll)*dipoleDir(b,c,kk)*...
183

dipoleDir(a, b, jj) * dipoleDir(d, a, ii);

type = selectYType(ii, jj, kk, ll, polarization);

if ((yCosTerm > termThresh) && (type ~= 0))

nYTerms = nYTerms + 1;

yFac(nYTerms) = calcYFac(yCosTerm, type);

yType(nYTerms) = type;

end

end

end

end

case 2

for ii = 1:3

for jj = 1:3

for kk = 1:3

for ll = 1:3

yCosTerm = dipoleDir(c, b, ii) * dipoleDir(d, c, jj) * ...

dipoleDir(b, a, kk) * dipoleDir(a, d, ll);


type = selectYType(ii, jj, kk, ll, polarization);

if ((yCosTerm > termThresh) && (type ~= 0))

nYTerms = nYTerms + 1;

yFac(nYTerms) = calcYFac(yCosTerm, type);

yType(nYTerms) = type;

end

end

end

end

case 3
for ii = 1:3
for jj = 1:3
for kk = 1:3
for ll = 1:3
    yCosTerm = dipoleDir(c,b,ii)*dipoleDir(b,a,jj)*...dipoleDir(d,c,kk)*dipoleDir(a,d,ll);
type = selectYType(ii,jj,kk,ll,polarization);
    if ((yCosTerm > termThresh) && (type ~= 0))
        nYTerms = nYTerms + 1;
        yFac(nYTerms) = calcYFac(yCosTerm, type);
        yType(nYTerms) = type;
    end
end
end
end
end

ca se 4
for ii = 1:3
for jj = 1:3
for kk = 1:3
for ll = 1:3
    yCosTerm = dipoleDir(a,b,ii)*dipoleDir(b,c,jj)*...dipoleDir(c,d,kk)*dipoleDir(a,d,ll);
type = selectYType(ii,jj,kk,ll,polarization);
    if ((yCosTerm > termThresh) && (type ~= 0))
        nYTerms = nYTerms + 1;
        yFac(nYTerms) = calcYFac(yCosTerm, type);
        yType(nYTerms) = type;
    end
end
end
end
end

184
end
end
end
end
case 5
for ii = 1:3
for jj = 1:3
for kk = 1:3
for ll = 1:3
yCosTerm = dipoleDir(c,d,ll)*dipoleDir(b,c,kk)*...
dipoleDir(a,b,jj)*dipoleDir(d,a,ii);
type = selectYType(ii,jj,kk,ll,polarization);
if ((yCosTerm > termThresh) && (type ~= 0))
nYTerms = nYTerms + 1;
yFac(nYTerms) = calcYFac(yCosTerm, type);
yType(nYTerms) = type;
end
end
end
end
case 6
for ii = 1:3
for jj = 1:3
for kk = 1:3
for ll = 1:3
yCosTerm = dipoleDir(c,b,ii)*dipoleDir(d,c,jj)*...
dipoleDir(b,a,kk)*dipoleDir(a,d,ll);
type = selectYType(ii,jj,kk,ll,polarization);
if ((yCosTerm > termThresh) && (type ~= 0))
    nYTerms = nYTerms + 1;
    yFac(nYTerms) = calcYFac(yCosTerm, type);
    yType(nYTerms) = type;
end

case 7
    for ii = 1:3
        for jj = 1:3
            for kk = 1:3
                for ll = 1:3
                    yCosTerm = dipoleDir(c,b,ii)*dipoleDir(b,a,jj)*...
                        dipoleDir(d,c,kk)*dipoleDir(a,d,ll);
                    type = selectYType(ii,jj,kk,ll,polarization);
                    if ((yCosTerm > termThresh) && (type ~= 0))
                        nYTerms = nYTerms + 1;
                        yFac(nYTerms) = calcYFac(yCosTerm, type);
                        yType(nYTerms) = type;
                    end
                end
            end
        end
    end
case 8
    for ii = 1:3
        for jj = 1:3
for kk = 1:3
  for ll = 1:3
    yCosTerm = dipoleDir(a,b,ii)*dipoleDir(b,c,jj)*dipoleDir(c,d,kk)*dipoleDir(a,d,ll);
    type = selectYType(ii,jj,kk,ll,polarization);
    if ((yCosTerm > termThresh) && (type ~= 0))
      nYTerms = nYTerms + 1;
      yFac(nYTerms) = calcYFac(yCosTerm, type);
      yType(nYTerms) = type;
    end
  end
end

otherwise
  error('specPack:buildRot','No Response function matches this type');
end

function type = selectYType(ii,jj,kk,ll,polarization)
  if strcmpi(polarization,'ZZZZ')
    if (ii == jj) && (ii == kk) && (ii == ll)
      type = 1;
    elseif (ii == jj) && (ii ~= kk) && (kk == ll)
      type = 2;
    elseif (ii == jj) && (ii ~= kk) && (kk ~= ll)
      type = 3;
    elseif (ii == jj) && (jj == kk) && (kk ~= ll)
      type = 4;
    elseif (ii == jj) && (jj == kk) && (kk == ll)
      type = 5;
    elseif (ii == jj) && (ii == kk) && (jj ~= ll)
      type = 6;
    elseif (ii == jj) && (ii == kk) && (jj == ll)
      type = 7;
    elseif (ii == jj) && (ii ~= kk) && (kk == ll)
      type = 8;
    elseif (ii == jj) && (ii ~= kk) && (kk ~= ll)
      type = 9;
    elseif (ii == jj) && (ii == kk) && (kk == ll)
      type = 10;
    end
  end
end
elseif (ii ~= jj) && (jj == kk) && (ii == ll)
    type = 3;
elseif (ii ~= jj) && (jj == ll) && (ii == kk)
    type = 3;
else
    type = 0;
end
elseif strcmpi(polarization,'YYZZ')
    if (ii == jj) && (ii == kk) && (ii == ll)
        type = 2;
    elseif (ii == jj) && (ii ~= kk) && (kk == ll)
        type = 4;
    elseif (ii ~= jj) && (jj == kk) && (ii == ll)
        type = 5;
    elseif (ii ~= jj) && (jj == ll) && (ii == kk)
        type = 5;
    else
        type = 0;
    end
else
    error('specPack:selectYType','Polarization must be ZZZZ or YYZZ')
end

function y = calcYFac(cosTerm, yType)
%CALCYFAC Calculate prefactor for rotational contribution to spectrum
%If 'yCosTerm' is larger than threshold and 'yType' is not zero, this
% determines the orientational factor, which is just 'yCosTerm' divided by
% the factor determined by the 'yType'.

switch yType
    case 1
        y = cosTerm/9.0;
    case 2
        y = cosTerm/9.0;
    case 3
        y = cosTerm/15.0;
    case 4
        y = cosTerm/9.0;
    case 5
        y = -cosTerm/30.0;
    otherwise
        error('specPack:calcYFac','Unknown yType');
end

end

A.5.3 buildVirtEcho.m

This function file builds the virtual echo (non-rephasing) response functions.

function rfn = buildVirtEcho(sys, expr, rwaThresh, prefacThresh,
    l_poprlx)
    %BUILDVIRTECHO construct virtual photon echo response function
    % Description: This function takes system and experimental parameters,
% it constructs a response function for the ks = k1-k2+k3 phase
    matching

% condition. Additional comments can be found in buildEcho.m

% Parameters: sys - Structure containing system parameters
%             exp - Structure containing experimental parameters
%             rwaThresh - Threshold for resonance for RWA ( units =
% cm^-1)
%            dipoleThresh - product of rho0 and dipoles is assumed
% to be 0 if smaller than prefacThresh
%            l_poprlx - logical controlling population relaxation

% Returns: rfn - structure with response function parameters
%          rfn.nterms - number of non-zero terms in response function
%          rfn.rtype - array with rfn types
%          rfn.aa - array with 1st eigenstate for each function
%          rfn.bb - array with 2nd eigenstate for each function
%          rfn.cc - array with 3rd eigenstate for each function
%          rfn.dd - array with 4th eigenstate for each function
%          rfn.w1 - array with t1 frequency for each function
%          rfn.w2 - array with t2 frequency for each function
%          rfn.w3 - array with t3 frequency for each function
%          rfn.rho0 - population in state a at t = 0
%          rfn.T1_1 - array with population relaxation contrib to t1
%          rfn.T1_2 - array with population relaxation contrib to t2
%          rfn.T1_3 - array with population relaxation contrib to t3

% Start Timing
timeID = tic;
rfn.nterm = 0;
sizeGuess = sys.nStates*6;

% Preallocated rfs parameter arrays
rfn.rtype = zeros(1, sizeGuess);
rfn.aa = zeros(1, sizeGuess);
rfn.bb = zeros(1, sizeGuess);
rfn.cc = zeros(1, sizeGuess);
rfn.dd = zeros(1, sizeGuess);
rfn.w1 = zeros(1, sizeGuess);
rfn.w2 = zeros(1, sizeGuess);
rfn.w3 = zeros(1, sizeGuess);
rfn.rho0 = zeros(1, sizeGuess);
rfn.nYTerms = zeros(1, sizeGuess);

for a=1:sys.nStates
    for b=1:sys.nStates
        for c=1:sys.nStates
            for d=1:sys.nStates

            % Check for R1 Members
            prefac = sys.mu(c,d)*sys.mu(b,c)*sys.mu(a,b)*sys.mu(a,d)*...
sys.rho0(a);
            if (prefac > prefacThresh)
                if (abs(sys.w(d,c) - expr.w(4)) <= rwaThresh)
                    if (abs(sys.w(d,b) - (expr.w(2) - expr.w(1))) ...
                        <= rwaThresh)
                        if (abs(sys.w(d,a) - expr.w(2)) <= rwaThresh)
                            rfn.nterm = rfn.nterm + 1;
                            rfn.rtype(rfn.nterm) = 1;
            end
        end
    end
end
rfn.aa(rfn.nTerm) = a;
rfn.bb(rfn.nTerm) = b;
rfn.cc(rfn.nTerm) = c;
rfn.dd(rfn.nTerm) = d;
rfn.w1(rfn.nTerm) = sys.w(d, a);
rfn.w2(rfn.nTerm) = sys.w(d, b);
rfn.w3(rfn.nTerm) = sys.w(d, c);
rfn.rho0(rfn.nTerm) = sys.rho0(a);

[nYTmp, yFacTmp, yTypeTmp] = ...
bbuildRot(a, b, c, d, sys.muDir, ...
rfn.rtype(rfn.nTerm), expr.polar);
rfn.nYTerms(rfn.nTerm) = nYTmp;
rfn.yFac(rfn.nTerm, 1:nYTmp) = yFacTmp;
rfn.yType(rfn.nTerm, 1:nYTmp) = yTypeTmp;
rfn.prefac(rfn.nTerm) = prefac;
if l_poprlx
  rfn.T1_1(rfn.nTerm) = 0.5/sys.popRlx(d) + 0.5/sys.popRlx(a);
  rfn.T1_2(rfn.nTerm) = 0.5/sys.popRlx(d) + 0.5/sys.popRlx(b);
  rfn.T1_3(rfn.nTerm) = 0.5/sys.popRlx(d) + 0.5/sys.popRlx(c);
end
end
end
end
% Check for R2 Members
prefac = sys.mu(c, b)*sys.mu(d, c)*sys.mu(b, a)*sys.mu(a, d)* ...
sys.rho0(a);
if (prefac > prefacThresh)
if (abs(sys.w(b, c) - expr.w(4)) <= rwaThresh)
89  if (abs(sys.w(d,b) + (expr.w(2) - expr.w(1))) ... 
90    <= rwaThresh)
91  if (abs(sys.w(d,a) + expr.w(2)) <= rwaThresh)
92    rfn.nterm = rfn.nterm + 1;
93    rfn.rtype(rfn.nterm) = 2;
94    rfn.aa(rfn.nterm) = a;
95    rfn.bb(rfn.nterm) = b;
96    rfn.cc(rfn.nterm) = c;
97    rfn.dd(rfn.nterm) = d;
98    rfn.w1(rfn.nterm) = sys.w(d,a);
99    rfn.w2(rfn.nterm) = sys.w(d,b);
100   rfn.w3(rfn.nterm) = sys.w(b,c);
101   rfn.rho0(rfn.nterm) = sys.rho0(a);
102   [nYTmp, yFacTmp, yTypeTmp] = ...
103   buildRot(a,b,c,d,sys.muDir,...
104   rfn.rtype(rfn.nterm), expr.polar);
105   rfn.nYTerms(rfn.nterm) = nYTmp;
106   rfn.yFac(rfn.nterm,1:nYTmp) = yFacTmp;
107   rfn.yType(rfn.nterm,1:nYTmp) = yTypeTmp;
108   rfn.prefac(rfn.nterm) = prefac;
109  if l_poprlx
110    rfn.T1_1(rfn.nterm) = 0.5/sys.popRlx(d) + 0.5/sys.popRlx(a);
111    rfn.T1_2(rfn.nterm) = 0.5/sys.popRlx(d) + 0.5/sys.popRlx(b);
112    rfn.T1_3(rfn.nterm) = 0.5/sys.popRlx(b) + 0.5/sys.popRlx(c);
113 end
114 end
115 end
116 end
117 end

% Check for R3 Members
prefac = sys.mu(c,b)*sys.mu(b,a)*sys.mu(d,c)*sys.mu(d,a)*...
sys.rho0(a);
if (prefac > prefacThresh)
  if (abs(sys.w(b,c) - expr.w(4)) <= rwaThresh)
    if (abs(sys.w(c,a) + (expr.w(2) - expr.w(1))) <= rwaThresh)
      if (abs(sys.w(d,a) + expr.w(2)) <= rwaThresh)
        rfn.nterm = rfn.nterm + 1;
        rfn.rtype(rfn.nterm)= 3;
        rfn.aa(rfn.nterm)= a;
        rfn.bb(rfn.nterm)= b;
        rfn.cc(rfn.nterm)= c;
        rfn.dd(rfn.nterm)= d;
        rfn.w1(rfn.nterm)= sys.w(d,a);
        rfn.w2(rfn.nterm)= sys.w(c,a);
        rfn.w3(rfn.nterm)= sys.w(b,c);
        rfn.rho0(rfn.nterm)= sys.rho0(a);
        [nYTmp, yFacTmp, yTypeTmp] = ... buildRot(a,b,c,d,sys.muDir,...
        rfn.rtype(rfn.nterm), expr.polar);
        rfn.nYTerms(rfn.nterm) = nYTmp;
        rfn.yFac(rfn.nterm,1:nYTmp) = yFacTmp;
        rfn.yType(rfn.nterm,1:nYTmp) = yTypeTmp;
        rfn.prefac(rfn.nterm) = prefac;
if l_poprlx
  rfn.T1_1(rfn.nterm) = 0.5/sys.popRlx(d) + 0.5/sys.popRlx(a);
  rfn.T1_2(rfn.nterm) = 0.5/sys.popRlx(a) + 0.5/sys.popRlx(c);
  rfn.T1_3(rfn.nterm) = 0.5/sys.popRlx(b) + 0.5/sys.popRlx(c);
% Check for R4 Members
prefac = sys.mu(a,b)*sys.mu(b,c)*sys.mu(c,d)*sys.mu(d,a)*...
sys.rho0(a);
if (prefac > prefacThres)
  if (abs(sys.w(b,a) - expr.w(4)) <= rwaThresh)
    if (abs(sys.w(c,a) - (expr.w(2) - expr.w(1))) ...
      <= rwaThresh)
        if (abs(sys.w(d,a) - expr.w(2)) <= rwaThresh)
          rfn.nterm = rfn.nterm + 1;
          rfn.rtype(rfn.nterm)= 4;
          rfn.aa(rfn.nterm)= a;
          rfn.bb(rfn.nterm)= b;
          rfn.cc(rfn.nterm)= c;
          rfn.dd(rfn.nterm)= d;
          rfn.w1(rfn.nterm)= sys.w(d,a);
          rfn.w2(rfn.nterm)= sys.w(c,a);
          rfn.w3(rfn.nterm)= sys.w(b,a);
          rfn.rho0(rfn.nterm)= sys.rho0(a);
          [nYTmp, yFacTmp, yTypeTmp] = ...
          buildRot(a,b,c,d,sys.muDir,...
          rfn.rtype(rfn.nterm), expr.polar);
          rfn.nYTerms(rfn.nterm) = nYTmp;
          rfn.yFac(rfn.nterm,1:nYTmp) = yFacTmp;
          rfn.yType(rfn.nterm,1:nYTmp) = yTypeTmp;
rfn.prefac(rfn.nterm) = prefac;
if l_poprlx
rfn.T1_1(rfn.nterm) = 0.5/sys.popRlx(d) + 0.5/sys.popRlx(a);
rfn.T1_2(rfn.nterm) = 0.5/sys.popRlx(c) + 0.5/sys.popRlx(a);
rfn.T1_3(rfn.nterm) = 0.5/sys.popRlx(b) + 0.5/sys.popRlx(a);
end
end
dend

dend
% Check for R5 (R1*) Members
prefac = sys.mu(c,d)*sys.mu(b,c)*sys.mu(a,b)*sys.mu(a,d)*...
sys.rho0(a);
if (prefac > prefacThresh)
if (abs(sys.w(d,c) + expr.w(4)) <= rwaThresh)
if (abs(sys.w(d,b) + (expr.w(2) - expr.w(1))) ...
<= rwaThresh)
if (abs(sys.w(d,a) + expr.w(2)) <= rwaThresh)
rfn.nterm = rfn.nterm + 1;
rfn.rtype(rfn.nterm)= 5;
rfn.aa(rfn.nterm)= a;
rfn.bb(rfn.nterm)= b;
rfn.cc(rfn.nterm)= c;
rfn.dd(rfn.nterm)= d;
rfn.w1(rfn.nterm)= sys.w(d,a);
rfn.w2(rfn.nterm)= sys.w(d,b);
rfn.w3(rfn.nterm)= sys.w(d,c);
rfn.rho0(rfn.nterm)= sys.rho0(a);
[nYTmp, yFacTmp, yTypeTmp] = ...
buildRot(a,b,c,d,sys.muDir,...)
rfn.rtype(rfn.nterm), expr.polar);
rfn.nYTerms(rfn.nterm) = nYTmp;
rfn.yFac(rfn.nterm,1:nYTmp) = yFacTmp;
rfn.yType(rfn.nterm,1:nYTmp) = yTypeTmp;
rfn.prefac(rfn.nterm) = prefac;
if l_poprlx
rfn.T1_1(rfn.nterm) = 0.5/sys.popRlx(d) + 0.5/sys.popRlx(a);
rfn.T1_2(rfn.nterm) = 0.5/sys.popRlx(d) + 0.5/sys.popRlx(b);
rfn.T1_3(rfn.nterm) = 0.5/sys.popRlx(d) + 0.5/sys.popRlx(c);
end
end
end
end
end
end
% Check for R6 (R2*) Members
prefac = sys.mu(c,b)*sys.mu(d,c)*sys.mu(b,a)*sys.mu(a,d)*...
sys.rho0(a);
if (prefac > prefacThresh)
if (abs(sys.w(b,c) + expr.w(4)) <= rwaThresh)
if (abs(sys.w(d,b) - (expr.w(2) - expr.w(1))) <= rwaThresh)
if (abs(sys.w(d,a) - expr.w(2)) <= rwaThresh)
rfn.nterm = rfn.nterm + 1;
rfn.rtype(rfn.nterm)= 6;
rfn.aa(rfn.nterm)= a;
rfn.bb(rfn.nterm)= b;
rfn.cc(rfn.nterm)= c;
rfn.dd(rfn.nterm)= d;
rfn.w1(rfn.nterm) = sys.w(d,a);
rfn.w2(rfn.nterm) = sys.w(d,b);
rfn.w3(rfn.nterm) = sys.w(b,c);
rfn.rho0(rfn.nterm) = sys.rho0(a);
[nYTmp, yFacTmp, yTypeTmp] = ... buildRot(a,b,c,d,sys.muDir,...
rfn.rtype(rfn.nterm), expr.polar);
rfn.nTerms(rfn.nterm) = nYTmp;
rfn.yFac(rfn.nterm,1:nYTmp) = yFacTmp;
rfn.yType(rfn.nterm,1:nYTmp) = yTypeTmp;
rfn.prefac(rfn.nterm) = prefac;
if l_poprlx
  rfn.T1_1(rfn.nterm) = 0.5/sys.popRlx(d) + 0.5/sys.popRlx(a);
  rfn.T1_2(rfn.nterm) = 0.5/sys.popRlx(d) + 0.5/sys.popRlx(b);
  rfn.T1_3(rfn.nterm) = 0.5/sys.popRlx(b) + 0.5/sys.popRlx(c);
end
end
end
end
% Check for R7 (R3*) Members
prefac = sys.mu(c,b)*sys.mu(b,a)*sys.mu(d,c)*sys.mu(d,a)*...
sys.rho0(a);
if (prefac > prefacThresh)
  if (abs(sys.w(b,c) + expr.w(4)) <= rwaThresh)
    if (abs(sys.w(c,a) - (expr.w(2) - expr.w(1))) ... <= rwaThresh)
      if (abs(sys.w(d,a) - expr.w(2)) <= rwaThresh)
        rfn.nterm = rfn.nterm + 1;
      end
    end
  end
end
263  rfn.rtype(rfn.nterm) = 7;
264  rfn.aa(rfn.nterm) = a;
265  rfn.bb(rfn.nterm) = b;
266  rfn.cc(rfn.nterm) = c;
267  rfn.dd(rfn.nterm) = d;
268  rfn.w1(rfn.nterm) = sys.w(d,a);
269  rfn.w2(rfn.nterm) = sys.w(c,a);
270  rfn.w3(rfn.nterm) = sys.w(b,c);
271  rfn.rho0(rfn.nterm) = sys.rho0(a);
272  [nYTmp, yFacTmp, yTypeTmp] = ...
273  buildRot(a,b,c,d,sys.muDir,...
274  rfn.rtype(rfn.nterm), expr.polar);
275  rfn.nYTerms(rfn.nterm) = nYTmp;
276  rfn.yFac(rfn.nterm,1:nYTmp) = yFacTmp;
277  rfn.yType(rfn.nterm,1:nYTmp) = yTypeTmp;
278  rfn.prefac(rfn.nterm) = prefac;
279  if l_poprlx
280   rfn.T1_1(rfn.nterm) = 0.5/sys.popRlx(d) + 0.5/sys.popRlx(a);
281   rfn.T1_2(rfn.nterm) = 0.5/sys.popRlx(a) + 0.5/sys.popRlx(c);
282   rfn.T1_3(rfn.nterm) = 0.5/sys.popRlx(b) + 0.5/sys.popRlx(c);
283  end
284  end
285  end
286  end
287  end
288  % Check for R8 (R4*) Members
289  prefac = sys.mu(a,b)*sys.mu(b,c)*sys.mu(c,d)*sys.mu(d,a)*...
290  sys.rho0(a);
291  if (prefac > prefacThresh)
if (abs(sys.w(b,a) + expr.w(4)) <= rwaThresh)
if (abs(sys.w(c,a) + (expr.w(2) - expr.w(1))) ...
if (abs(sys.w(d,a) + expr.w(2)) <= rwaThresh)
rfn.nterm = rfn.nterm + 1;
rfn.rtype(rfn.nterm) = 8;
rfn.aa(rfn.nterm) = a;
rfn.bb(rfn.nterm) = b;
rfn.cc(rfn.nterm) = c;
rfn.dd(rfn.nterm) = d;
rfn.w1(rfn.nterm) = sys.w(d,a);
rfn.w2(rfn.nterm) = sys.w(c,a);
rfn.w3(rfn.nterm) = sys.w(b,a);
rfn.rho0(rfn.nterm) = sys.rho0(a);
[nYTmp, yFacTmp, yTypeTmp] = ...
builtRot(a,b,c,d,sys.muDir,...
rfn.rtype(rfn.nterm), expr.polar);
rfn.nYTerms(rfn.nterm) = nYTmp;
rfn.yFac(rfn.nterm,1:nYTmp) = yFacTmp;
rfn.yType(rfn.nterm,1:nYTmp) = yTypeTmp;
rfn.prefac(rfn.nterm) = prefac;
if 1_poprlx
rfn.T1_1(rfn.nterm) = 0.5/sys.popRlx(d) + 0.5/sys.popRlx(a);
rfn.T1_2(rfn.nterm) = 0.5/sys.popRlx(c) + 0.5/sys.popRlx(a);
rfn.T1_3(rfn.nterm) = 0.5/sys.popRlx(b) + 0.5/sys.popRlx(a);
end
end
end
% Deallocate Unused Memory
if (rfn.nterm < sizeGuess)
  rfn.rtype(rfn.nterm+1:sizeGuess) = [];
  rfn.aa(rfn.nterm+1:sizeGuess) = [];
  rfn.bb(rfn.nterm+1:sizeGuess) = [];
  rfn.cc(rfn.nterm+1:sizeGuess) = [];
  rfn.dd(rfn.nterm+1:sizeGuess) = [];
  rfn.w1(rfn.nterm+1:sizeGuess) = [];
  rfn.w2(rfn.nterm+1:sizeGuess) = [];
  rfn.w3(rfn.nterm+1:sizeGuess) = [];
  rfn.rho0(rfn.nterm+1:sizeGuess) = [];
  rfn.nYTerms(rfn.nterm+1:sizeGuess) = [];
end

% Convert wave numbers to inverse femtoseconds
cm2w = 2*pi*2.998e-5;
rfn.w1 = rfn.w1*cm2w;
rfn.w2 = rfn.w2*cm2w;
rfn.w3 = rfn.w3*cm2w;

% Set lineshape functions
rfn.hfun = sys.hfun;

% Set Orientational Diffusion Coefficient
rfn.dor = sys.dor;

% End Timing
time_elapsed = toc(timeID);

% Print Timing
outstr = strcat('Time elapsed for buildVirtEcho: 
',num2str(time_elapsed));
disp(outstr)

end

A.5.4 calc2DImpulse.m

function calc2DImpulse(basename, sys, expr, tau2, opts)

%CALC2DIMPULSE -
% Description: This function provides a wrapper for calculating 2DIR spectra by automating the calculation of rephasing and nonrephasing contributions. All laser pulses are assumed to be delta functions for this calculation.

% Parameters: basename - This is the string that defines the basename for all output files
% sys - struct containing molecular system parameters
% expr - struct containing exprimental parameters
% tau2 - vector of tau2 values for the spectra
% opts - struct containing calculation options and thresholds
% Outputs: '[BASENAME]_t2-[VAL]_rt.mat' - rephasing response in time domain
%          '[BASENAME]_t2-[VAL]_rf.mat' - rephasing spectrum time domain
%          '[BASENAME]_t2-[VAL]_nrt.mat' - nonrephasing response in time domain
%          '[BASENAME]_t2-[VAL]_nrf.mat' - nonrephasing spectrum in frequency domain
%          '[BASENAME]_t2-[VAL]_abs.mat' - 2DIR correlation spectrum

% evaluate rephasing signals
rfn = buildEcho(sys, expr, opts.rwaThresh, opts.prefacThresh,...
opt.s.l_poprlx);
printRandY(rfn,strcat(basename,'_Echo_fn'));
for ii=1:length(tau2)
tau2str = num2str(tau2(ii));
%adds tau2 time and rt (rephasing signal in the time domain) to file name
tname = strcat(basename,'_t2-',tau2str,'_rt');
fname = strcat(basename,'_t2-',tau2str,'_rf'); %rephasing freq
eval2DImpulse(rfn, tname, fname, tau2(ii), opts);
if opts.l_cspecOnly %this is an option in sp_SetOptions to save file space
delete(tname)
end
end
clear rfn

% evaluate nonrephasing signals
rfn = buildVirtEcho(sys, expr, opts.rwaThresh, opts.prefacThresh,...
opts.l_poprlx);
printRandY(rfn,strcat(basename,'_VirtEcho_fn'));
for ii=1:length(tau2)
    tau2str = num2str(tau2(ii));
    tname = strcat(basename,'_t2-',tau2str,'_nrt'); %non-rephasing time
    fname = strcat(basename,'_t2-',tau2str,'_nrf'); %non-rephasing freq
    eval2DImpulse(rfn, tname, fname, tau2(ii), opts);
    if opts.l_cspecOnly
        delete(tname)
    end
end
clear rfn

% Make correlation spectra
timeID = tic;
for ii = 1:length(tau2)
    tau2str = num2str(tau2(ii));
    fname1 = strcat(basename,'_t2-',tau2str,'_rf');
    fname2 = strcat(basename,'_t2-',tau2str,'_nrf');
    absname = strcat(basename,'_t2-',tau2str,'_abs');
    fr = load(fname1);
    a = size(fr.response_freq,1);
    fr2 = zeros(a+1);
    fr2(1:a,1:a) = fr.response_freq;
    clear fr
    fnr = load(fname2);
    fnr2 = zeros(a+1);
    fnr2(1:a,1:a) = fnr.response_freq;
abs1 = fliplr(fr2) + fnr2;
freq = fnr.freq;
freq2 = zeros(a+1,1);
freq2(1:a) = freq;
freq2(a+1) = -freq2(1);
freq = freq2;
save(absname, 'abs1','freq');
if opts.l_cspecOnly
    delete(fname1)
delete(fname2)
end
end

time_elapsed = toc(timeID);
outstr = strcat('Time elapsed for correlation spectrum construction: ',...
num2str(time_elapsed));
disp(outstr)
end

A.5.5 createHomLimlineShape.m

This function can be used to automate the creation of homogeneous lineshape functions (I think).

It is not called by any of the other simulation functions as far as I can tell.

function hfun = createHomLimLineShape(dephasingTimes)

%CREATEHOMLIMLINESHAPE - Generate Lineshape function for Hom. Limit
% Description: This function creates a cell containing the function handles for linshape functions in the homogeneous limit given a vector
% of dephasing times. It also create each individual function
% Parameters: dephasingTimes - matrix of dephasing times for each coherence the ij element of the matrix should be the dephasing time for the coherence between states i and j. Function assumes ij and ji dephasing times are the same

% Output: hfun - cell containing lineshape function handles

% gij.m - lineshape function files

% Create g0

fid = fopen('g0.m','w');
fprintf(fid,'function y = g0(t)\n');
fprintf(fid,'%%G0 - Lineshape function to return 0');
fprintf(fid,'\ny = 0;');
close(fid);

shape = size(dephasingTimes);

if (shape(1) ~= shape(2))
    error('specPack: createHomLimLineShape','Argument must be square matrix');

end

0 thresh = 1.0

for i=1:shape(1)
    for j=1:shape(1)
        if (dephasingTimes(i,j) > 0)
            fname = strcat('g',num2str(j),num2str(j),num2str(j))
A.5.6 eval2DImpulse.m

This evaluates the response function for the 2D IR spectrum, and saves files for the time and frequency domain responses.

```matlab
function eval2DImpulse( rfn, tname, fname, tau2, opts )

%EVAL2DIMPULSE Evaluates response function for 2DIR with Delta Fn pulses
% Description: This function evaluates a response function on a 2D time
% matrix and performs the fourier transform. time and frequency domain
% results are saved in .mat files
%
% Parameters: rfn - response function
%             tname - file name for time domain data
%             fname - file name for frequency domain data
%             tau2 - tau2 value
%             opts - struct of calculation options
%
% Output:     [tname].mat - time domain data
%             [fname].mat - frequency domain data
%
time_width = opts.nsteps+1;
time_end = opts.nsteps*opts.tstep;
t1 = 0:opts.tstep:time_end;
t1mat = repmat(t1, [time_width 1]);
```
t3 = (0:opts.tstep:time_end)';
t3mat = repmat(t3, [1 time_width]);

%If poprlx is 'true' in sp_SetOptions.m, call evalrfn_rlpx, otherwise calls evalrfn.
if opts.l_poprlx
    response_time = evalrfn_rlpx(rfn, t1mat, tau2, t3mat);
else
    response_time = evalrfn(rfn, t1mat, tau2, t3mat);
end

save(tname, 'response_time', 't1', 't3');

fsize = opts.fftsize;
fsizep1 = opts.fftsize+1;

response_freq = fft2(real(response_time), fsize, fsize);
response_freq = fftshift(response_freq);
clear response_time

% Make Frequency axes
N = opts.fftsize;
freq = (−N/2):(N/2−1);
freq = freq/(N*opts.tstep*2.998e-5);

save(fname, 'response_freq','freq');
function response = evalrfn( rfn, t1, t2, t3 )

%EVALRFN Evaluates a response function returning a matrix of values
% Description: This function evaluates the response function for
% that has been previously constructed
%
% Parameters: rfn - response function struct
% t1 - matrix of t1 times
% t2 - t2 delay (usually scalar)
% t3 - matrix of t3 times
%
% Output: response - response in time domain
%
% Begin timing
ticID = tic;

%The type was decided in buildEcho.m, and each term is subsequently
% added
%to the same y value
response = zeros(size(t1));
for ii = 1:rfn.nterm
switch rfn.rtype(ii);
    case 1
        response = response + r1val(rfn, ii, t1, t2, t3);
    case 2
        response = response + r2val(rfn, ii, t1, t2, t3);
end
case 3
response = response + r3val(rfn, ii, t1, t2, t3);
case 4
response = response + r4val(rfn, ii, t1, t2, t3);
case 5
response = response - conj(r1val(rfn, ii, t1, t2, t3));
case 6
response = response - conj(r2val(rfn, ii, t1, t2, t3));
case 7
response = response - conj(r3val(rfn, ii, t1, t2, t3));
case 8
response = response - conj(r4val(rfn, ii, t1, t2, t3));
end
end

% End Timing
time_elapsed = toc(ticID);
outStr = strcat('Time elapsed during evalrfn: ',num2str(time_elapsed));
disp(outStr)
End

function y = r1val(rfn, ii, t1, t2, t3)
%R1VAL - returns value of r1
%Each part of the response function is added to each other here,
%orientational, dephasing, the prefactor, and the time component. This
%the same for each r value.
y = zeros(size(t1));

% Comput rotational contribution

%Most of these values were determined in buildRot.m
for j=1:rfn.nYTerms(ii)
y = y+yVal(t1,t2,t3, rfn.yType(ii,j),rfn.dor)*rfn.yFac(ii,j);
end

% First Calculate dephasing functions (JCP, 2001, 115, 9266)

y = y.*exp(-(conj(rfn.hfun{rfn.cc(ii),rfn.cc(ii)}(t3))...+
+conj(rfn.hfun{rfn.bb(ii),rfn.bb(ii)}(t2))...+
+rfn.hfun{rfn.dd(ii),rfn.dd(ii)}(t1+t2+t3)...+
+conj(rfn.hfun{rfn.bb(ii),rfn.cc(ii)}(t3+t2)...-rfn.hfun{rfn.bb(ii),rfn.cc(ii)}(-t3)...%conj?
-rfn.hfun{rfn.bb(ii),rfn.cc(ii)}(-t2)...%conj
+rfn.hfun{rfn.cc(ii),rfn.dd(ii)}(t1+t2)...-
-rfn.hfun{rfn.cc(ii),rfn.dd(ii)}(t1+t2+t3)...-
-rfn.hfun{rfn.cc(ii),rfn.dd(ii)}(-t3)...+
+conj(rfn.hfun{rfn.bb(ii),rfn.dd(ii)}(-t1)...-
-rfn.hfun{rfn.bb(ii),rfn.dd(ii)}(-t3-t2)...%conj, time ordering-
-rfn.hfun{rfn.bb(ii),rfn.dd(ii)}(+t1+t2)...%conj+
+rfn.hfun{rfn.bb(ii),rfn.dd(ii)}(-t3)))); %conj

% compute final values
y = rfn.prefac(ii)*exp(1i*(-rfn.w3(ii)*t3-rfn.w2(ii)*t2-
rfn.w1(ii)*t1)).*y;
end
function y = r2val(rfn, ii, t1, t2, t3)
%R2VAL - returns value of r2

y = zeros(size(t1));

% Comput rotational contribution
for j=1:rfn.nYTerms(ii)
    y = y+yVal(t1,t2,t3, rfn.yType(ii,j),rfn.dor)*rfn.yFac(ii,j);
end

% First calculate dephasing
y = y.*exp(conj(-rfn.hfun{rfn.cc(ii),rfn.cc(ii)}(t3)...
+rfn.hfun{rfn.dd(ii),rfn.dd(ii)}(t1+t2)...
+conj(rfn.hfun{rfn.bb(ii),rfn.bb(ii)}(t2+t3))...
+rfn.hfun{rfn.bb(ii),rfn.cc(ii)}(-t2)...
-rfn.hfun{rfn.bb(ii),rfn.cc(ii)}(t2)...
-rfn.hfun{rfn.bb(ii),rfn.cc(ii)}(-t2-t3)...
+rfn.hfun{rfn.cc(ii),rfn.dd(ii)}(t1+t2+t3)...
-rfn.hfun{rfn.cc(ii),rfn.dd(ii)}(t1+t2)...
-rfn.hfun{rfn.cc(ii),rfn.dd(ii)}(t3)...
+rfn.hfun{rfn.bb(ii),rfn.dd(ii)}(t1)...
-rfn.hfun{rfn.bb(ii),rfn.dd(ii)}(t3+t1+t2)...
+rfn.hfun{rfn.bb(ii),rfn.dd(ii)}(t3))));

y = rfn.prefac(ii)*exp(1i*(-
    rfn.w3(ii)*t3+rfn.w2(ii)*t2+rfn.w1(ii)*t1)).*y;
function y = r3val(rfn, ii, t1, t2, t3)
%R3VAL - returns value of r3

y = zeros(size(t1));

% Compute rotational contribution
for j=1:rfn.nYTerms(ii)
    y = y+yVal(t1,t2,t3, rfn.yType(ii,j),rfn.dor)*rfn.yFac(ii,j);
end

% Calculate Dephasing
y = y.*exp(conj(-
    rfn.hfun{rfn.bb(ii),rfn.bb(ii)}(t3)...
    +rfn.hfun{rfn.cc(ii),rfn.cc(ii)}(t2+t3)...
    +rfn.hfun{rfn.dd(ii),rfn.dd(ii)}(t1)...
    +rfn.hfun{rfn.cc(ii),rfn.dd(ii)}(t2+t3+t1)...
    -rfn.hfun{rfn.cc(ii),rfn.dd(ii)}(t2)...)
    rfn.w3(ii)*t3+rfn.w2(ii)*t2+rfn.w1(ii)*t1).*y;
function y = r4val(rfn, ii, t1, t2, t3)

% R4VAL - returns value of r1
y = zeros(size(t1));

% Compute rotational contribution
for j=1:rfn.nYTerms(ii)
    y = y+yVal(t1,t2,t3, rfn.yType(ii,j),rfn.dor)*rfn.yFac(ii,j);
end

% Calculate Dephasing
y = y.*exp(-rfn.hfun{rfn.bb(ii),rfn.bb(ii)}(t3)...%time ordering
+rfn.hfun{rfn.cc(ii),rfn.cc(ii)}(t2)...%time ordering
+rfn.hfun{rfn.dd(ii),rfn.dd(ii)}(t1)...%time ordering
+rfn.hfun{rfn.bb(ii),rfn.cc(ii)}(t3+t2)...%time ordering
-rfn.hfun{rfn.bb(ii),rfn.cc(ii)}(t3)...%time ordering
-rfn.hfun{rfn.bb(ii),rfn.cc(ii)}(t2)...%time ordering
+rfn.hfun{rfn.cc(ii),rfn.dd(ii)}(t1+t2)...%time ordering
-rfn.hfun{rfn.cc(ii),rfn.dd(ii)}(t2)...%time ordering
+rfn.hfun{rfn.bb(ii),rfn.dd(ii)}(t3+t2+t1)...%time ordering
-rfn.hfun{rfn.bb(ii),rfn.dd(ii)}(t3+t2)...%time ordering
-rfn.hfun{rfn.bb(ii),rfn.dd(ii)}(t2+t1)...%time ordering
+rfn.hfun{rfn.bb(ii),rfn.dd(ii)}(t2));

y = rfn.prefac(ii)*exp(1i*(-rfn.w3(ii)*t3-rfn.w2(ii)*t2-rfn.w1(ii)*t1)).*y;
end
A.5.8  evalrfn_rlx.m

This is the same as evalrfn.m except for a couple of lines of code. Refer to evalrfn.m for more
detailed comments. The population relaxation portion of this simulation code needs more work.

function response = evalrfn( rfn, t1, t2, t3 )
%EVALRFN Evaluates a response function returning a matrix of values
% Description: This function evaluates the response function for
% that has been previously constructed. Only comments on portions
% different from evalrfn.m are included.

%!!Warning!! - I think the population relaxation portion of this code
%needs further work. Modes do not seem to decay properly when plotting
%spectral feature over multiple tau2 points.

% Parameters: rfn - response function struct
% t1 - matrix of t1 times
% t2 - t2 delay (usually scalar)
% t3 - matrix of t3 times

% Output: response - response in time domain

% Begin timing
ticID = tic;

response = zeros(size(t1));

for ii = 1:rfn.nterm
  switch rfn.rtype(ii);

end
case 1
response = response + r1val(rfn, ii, t1, t2, t3);

case 2
response = response + r2val(rfn, ii, t1, t2, t3);

case 3
response = response + r3val(rfn, ii, t1, t2, t3);

case 4
response = response + r4val(rfn, ii, t1, t2, t3);

case 5
response = response - conj(r1val(rfn, ii, t1, t2, t3));

case 6
response = response - conj(r2val(rfn, ii, t1, t2, t3));

case 7
response = response - conj(r3val(rfn, ii, t1, t2, t3));

case 8
response = response - conj(r4val(rfn, ii, t1, t2, t3));
end
end

% End Timing
time_elapsed = toc(ticID);
outStr = strcat('Time elapsed during evalrfn: ',num2str(time_elapsed));
disp(outStr)

End

function y = r1val(rfn, ii, t1, t2, t3)
%R1VAL - returns value of r1
y = zeros(size(t1));

% Compute rotational contribution
for j=1:rfn.nYTerms(ii)
    y = y+yVal(t1,t2,t3, rfn.yType(ii,j),rfn.dor)*rfn.yFac(ii,j);
end

% First Calculate dephasing functions
y = y.*exp(-conj(rfn.hfun{rfn.cc(ii),rfn.cc(ii)}(t3)) + conj(rfn.hfun{rfn.bb(ii),rfn.bb(ii)}(t2)) + rfn.hfun{rfn.dd(ii),rfn.dd(ii)}(t1+t2+t3) + conj(rfn.hfun{rfn.bb(ii),rfn.cc(ii)}(t3+t2) - rfn.hfun{rfn.bb(ii),rfn.cc(ii)}(t3) - rfn.hfun{rfn.bb(ii),rfn.cc(ii)}(t2)) + rfn.hfun{rfn.cc(ii),rfn.dd(ii)}(t1+t2) - rfn.hfun{rfn.cc(ii),rfn.dd(ii)}(t1+t2+t3) - rfn.hfun{rfn.cc(ii),rfn.dd(ii)}(-t3) + conj(rfn.hfun{rfn.bb(ii),rfn.dd(ii)}(-t1) - rfn.hfun{rfn.bb(ii),rfn.dd(ii)}(t2+t3) - rfn.hfun{rfn.bb(ii),rfn.dd(ii)}(-t1-t2) + rfn.hfun{rfn.bb(ii),rfn.dd(ii)}(t3)));

% Compute population relaxation component – this is the only difference % from evalrfn.m
y = y.*exp(-rfn.T1_1(ii)*t1-rfn.T1_2(ii)*t2-rfn.T1_3(ii)*t3);

% compute final values
y = rfn.prefac(ii)*exp(1i*(-rfn.w3(ii)*t3-rfn.w2(ii)*t2-rfn.w1(ii)*t1)).*y;
function y = r2val(rfn, ii, t1, t2, t3)

% R2VAL - returns value of r2

y = zeros(size(t1));

% Comput rotational contribution
for j=1:rfn.nYTerms(ii)
    y = y+yVal(t1,t2,t3, rfn.yType(ii,j),rfn.dor)*rfn.yFac(ii,j);
end

% First calculate dephasing
y = y.*exp(conj(-rfn.hfun{rfn.cc(ii),rfn.cc(ii)}(t3)...
    +rfn.hfun{rfn.dd(ii),rfn.dd(ii)}(t1+t2)...
    +conj(rfn.hfun{rfn.bb(ii),rfn.bb(ii)}(t2+t3))...
    +rfn.hfun{rfn.bb(ii),rfn.cc(ii)}(-t2)...
    -rfn.hfun{rfn.bb(ii),rfn.cc(ii)}(t3)...
    -rfn.hfun{rfn.bb(ii),rfn.cc(ii)}(-t2-t3)...
    +rfn.hfun{rfn.cc(ii),rfn.dd(ii)}(t1+t2+t3)...
    -rfn.hfun{rfn.cc(ii),rfn.dd(ii)}(t1+t2)...
    -rfn.hfun{rfn.cc(ii),rfn.dd(ii)}(t3)...
    +rfn.hfun{rfn.bb(ii),rfn.dd(ii)}(t1)...
    -rfn.hfun{rfn.bb(ii),rfn.dd(ii)}(t3+t1+t2)...
    -rfn.hfun{rfn.bb(ii),rfn.dd(ii)}(-t2)...
    +rfn.hfun{rfn.bb(ii),rfn.dd(ii)}(t3))));

% Compute population relaxation component
y = y.*exp(-rfn.T1_1(ii)*t1-rfn.T1_2(ii)*t2-rfn.T1_3(ii)*t3);
219

y = rfn.prefac(ii)*exp(1i*(-
    rfn.w3(ii)*t3+rfn.w2(ii)*t2+rfn.w1(ii)*t1)).*y;

end

function y = r3val(rfn, ii, t1, t2, t3)
%R3VAL - returns value of r3

y = zeros(size(t1));

% Comput rotational contribution
for j=1:rfn.nYTerms(ii)
    y = y+yVal(t1,t2,t3, rfn.yType(ii,j),rfn.dor)*rfn.yFac(ii,j);
end

y = y.*exp(conj((-conj(rfn.hfun{rfn.bb(ii),rfn.bb(ii)}(t3))...
    +rfn.hfun{rfn.cc(ii),rfn.cc(ii)}(t2+t3)...
    +rfn.hfun{rfn.dd(ii),rfn.dd(ii)}(t1)...
    +rfn.hfun{rfn.cc(ii),rfn.dd(ii)}(t2+t3+t1)...
    -rfn.hfun{rfn.cc(ii),rfn.cc(ii)}(t2+t3)...
    -rfn.hfun{rfn.cc(ii),rfn.cc(ii)}(t1)...
    -rfn.hfun{rfn.bb(ii),rfn.cc(ii)}(t3+t2)...
    -rfn.hfun{rfn.bb(ii),rfn.cc(ii)}(-t3)...
    +rfn.hfun{rfn.bb(ii),rfn.cc(ii)}(t2)...
    -rfn.hfun{rfn.bb(ii),rfn.dd(ii)}(t3+t2+t1)...
    +rfn.hfun{rfn.bb(ii),rfn.dd(ii)}(t3+t2)...
    +rfn.hfun{rfn.bb(ii),rfn.dd(ii)}(t2+t1)...
    -rfn.hfun{rfn.bb(ii),rfn.dd(ii)}(t2))));

% Compute population relaxation component
y = y.*exp(-rfn.T1_1(ii)*t1-rfn.T1_2(ii)*t2-rfn.T1_3(ii)*t3);

% Compute final value
y = rfn.prefac(ii)*exp(1i*(-rfn.w3(ii)*t3+rfn.w2(ii)*t2+rfn.w1(ii)*t1)).*y;
end

function y = r4val(rfn, ii, t1, t2, t3)
%R4VAL - returns value of r1
y = zeros(size(t1));

% Compute rotational contribution
for j=1:rfn.nYTerms(ii)
y = y+yVal(t1,t2,t3, rfn.yType(ii,j),rfn.dor)*rfn.yFac(ii,j);
end

y = y.*exp(-(rfn.hfun{rfn.bb(ii),rfn.bb(ii)}(t3) +rfn.hfun{rfn.cc(ii),rfn.cc(ii)}(t2) +rfn.hfun{rfn.dd(ii),rfn.dd(ii)}(t1) +rfn.hfun{rfn.bb(ii),rfn.cc(ii)}(t3+t2) -rfn.hfun{rfn.bb(ii),rfn.cc(ii)}(t3) -rfn.hfun{rfn.bb(ii),rfn.cc(ii)}(t2) +rfn.hfun{rfn.cc(ii),rfn.dd(ii)}(t1+t2) -rfn.hfun{rfn.cc(ii),rfn.dd(ii)}(t2) +rfn.hfun{rfn.bb(ii),rfn.dd(ii)}(t3+t2+t1) -rfn.hfun{rfn.bb(ii),rfn.dd(ii)}(t3+t2) -rfn.hfun{rfn.bb(ii),rfn.dd(ii)}(t2+t1) +rfn.hfun{rfn.bb(ii),rfn.dd(ii)}(t2)))

$y = y \cdot \exp(-\text{rfn.T1}_1(ii) \cdot t1 - \text{rfn.T1}_2(ii) \cdot t2 - \text{rfn.T1}_3(ii) \cdot t3);$ 

$y = \text{rfn.prfac}(ii) \cdot \exp(\imath (\text{rfn.w3}(ii) \cdot t3 - \text{rfn.w2}(ii) \cdot t2 - \text{rfn.w1}(ii) \cdot t1)) \cdot y;$ 

end

A.5.9 initExperiment.m

This function puts all of the experimental parameters together in one place so they are easily transferred to other functions.

function expr = initExperiment(fieldFreq, measurement, polarization)
    % INITEXPERIMENT Creates object holding all experimental parameters
    % Description: Creates a structure that holds all of the parameters relevant to experimental setup
    % Parameters: fieldFreq - 4 element vector containing center frequency of input and local oscillator fields
    % measurement - String denoting desired experiment, right now that is only a 2D IR experiment
    % polarization - field polarizations 'ZZZZ' or 'YYZZ'
    expr.w = fieldFreq;
    expr.measurement = measurement;
    expr.polar = polarization;
    end
A.5.10 initSystem.m

This function puts all of the system parameters together in one place so they can easily be transferred to other functions.

```matlab
function sys = initSystem(nStates, energies, dipoles, dipoleDir, rho0,...
    hfun, dor, popRlx)
% initSystem Create structure that hold parameters of molecular system
% Description - Function creates a object that stores all parameter
% of the material system required to calculate nonlinear signal
% Parameters: nStates - Number of system eigenstates
% energies - vector containing eigenstate energies (cm^-1)
% dipoles - matrix of transition dipole moments
% dipoleDir - 3d Matrix defining projection of dipoles onto each of the cartesian axes in the molecular frame
% hfun - matrix of lineshape functions
% dor - orientational diffusion coefficient
% popRlx - optional vector of population relaxation times
% Returns: sys - structure containing system information
% sys.nStates
% sys.w - transition frequency matrix, the difference between frequencies of each energy level
% sys.mu - transition dipole matrix
% sys.muDir - directional cosine information
```
% sys.hfun - lineshape functions
% sys.dor - orientational diffusion coefficient

sys.nStates = nStates;
sys.mu = dipoles;
sys.muDir = dipoleDir;
sys.rho0 = rho0;
sys.hfun = hfun;
sys.dor = dor;

% Optional poprlx
if nargin == 8
    sys.popRlx = popRlx;
end

% Construct transition frequency matrix
sys.w = zeros(nStates,nStates);
for i = 1:nStates
    for j = i:nStates
        sys.w(i,j) = energies(i) - energies(j);
        sys.w(j,i) = -sys.w(i,j);
    end
end

A.5.11 printRandY.m

This function is most useful as a check of all parts of the orientational response function.

function printRandY( rfn, outFile )
%This function files prints out of the orientational components that are calculated. This can be used to check and make sure that both your relative angles are correctly distributed to the x, y, z axes and that all terms for the orientational response have been calculated.

fid = fopen(strcat(outFile, '.txt'),'w');
for ii = 1:rfn.n_term
fprintf(fid,'R%d(%d,%d,%d,%d)\n',rfn.r_type(ii),rfn.aa(ii),...
rfn.bb(ii),rfn.cc(ii),rfn.dd(ii));
fprintf(fid,'Num Y Terms: %d\n',rfn.nY_Terms(ii));
for jj = 1:rfn.nY_Terms(ii)
fprintf(fid,'Type: %d       Prefactor: %d\n',rfn.y_type(ii,jj),...
rfn.y_fac(ii,jj));
end
end
fclose(fid);

A.5.12 printTerms.m

This function is also most useful for checking that all response functions are accounted for.

function printTerms( rfn, outFile )
%PRINTTERMS Prints the terms in the response function
% Description: Prints the terms of a response function i.e.
% 'R1(0,1,0,1)'
%
% Parameters: rfn - struct containing response function
% outFile - baseName for output. response functions will be
% printed in the file '[outFile].txt'

fid = fopen(strcat(outFile, '.txt'),'w');
for ii = 1:rfn.nterm
fprintf(fid,'R%d(%d,%d,%d,%d)\n',rfn.rtype(ii),rfn.aa(ii),...
rfn.bb(ii),rfn.cc(ii),rfn.dd(ii));
end
fclose(fid);

A.5.13 sp_SetOptions.m

This file has to be manually updated for each system that 2D IR spectra are simulated for.

function opts = sp_SetOptions( varargin )
%SP_SETOPTIONS Returns a structure containing all options for simulation
% Description: For newer versions of MATLAB, updates the specified
% paramenters automatically (here that is the commented out code), and then
% returns the structure containing all parameters. For older versions of
% MATLAB, this file has to be opened and modified by hand (as it is written now).
%
% Parameters: prefacThresh - Threshold for cut-off of terms with a
% prefactor less than this value. (Should be set at least below the quadruple of lowest dipole moment, mu^4.)

rwaThresh - Rotating wave approximation cut-off.

Frequency differences larger than this value will show no coupling in simulation.

tstep - Size of step used in the t1 and t3 time axes

nsteps - Number of time steps taken in the t1 and t3 time axes

fftsize - Size that each time axis will be zero-padded to before Fourier transform. This value should be a power of 2.

l_poprlx - Option to turn on population relaxation. Either 'true' or 'false'.

l_cspecOnly - Option to only save the correlation spectrum, and not the rephasing or non-rephasing spectra, to save disk space. Either 'true' or 'false'.

if (mod(nargin,2) ~= 0)
error('sp_SetOptions:ArgChk',...'
'Odd number of parameters. Arguments must be in key-value pairs')
End
% Set defaults
keySet = {'prefacThresh','rwaThresh','tstep', 'nsteps','fftsize',...
   'l_poprlx','l_cspecOnly'};
defaults = [0.2, 80, 10.0, 2000, 2048, false, false];
% optMap = containers.Map(keySet, defaults);
%
% for ii = 1:(nargin/2)
%   if optMap.isKey(varargin{2*ii-1})
%     optMap('isKey') = varargin{2*ii};
%   else
%     error('sp_SetOptions:KeyVal','...
%        'Option not found')
%   end
% end
%
% opts.prefacThresh = optMap('prefacThresh');
% opts.rwaThresh = optMap('rwaThresh');
% opts.tstep = optMap('tstep');
% opts.nsteps = optMap('nsteps');
% opts.fftsize = optMap('fftsize');
% opts.l_poprlx = optMap('l_poprlx');
% opts.l_cspecOnly = optMap('l_cspecOnly');

opts.prefacThresh = 0.08;
opts.rwaThresh = 120;
opts.tstep = 8;
opts.nsteps = 2000;
opts.fftsize = 2048;
opts.l_poprlx = false;
opts.l_cspecOnly = false;

A.5.14 yVal.m

This function is called by buildRot.m.

function y = yVal(t1, t2, t3, type, dor)

%YVAL returns the value of the dynamic orientational contribution term
% Description: This function calculates the time-dependent
% contribution to the orientational response. Again these terms come from Table
% II in JCP, 2001, 115, 297.
%
% Parameters: t1 - t1 time
% t2 - t2 time
% t3 - t3 time
% type - type of Y response
% dor - Orientational diffusion coefficient
%
% Case is determined in buildRot.m
switch type
    case 1
        y = c1(t1,dor).*c1(t3,dor).*(1.0+0.8*c2(t2,dor));
    case 2
        y = c1(t1,dor).*c1(t3,dor).*(1.0-0.4*c2(t2,dor));
end
case 3
y = c1(t1,dor).*c1(t3,dor).*c2(t2,dor);
case 4
y = c1(t1,dor).*c1(t3,dor).*(1.0+0.2*c2(t2,dor));
case 5
y = c1(t1,dor).*c1(t3,dor).*c2(t2,dor);
otherwise
error('specPack:yVal', 'No response function of this type')
end
end

function y = c1(tt,dor)
%C1 Function for orientational response
y = exp(-2.0*dor*tt);
end

function y = c2(tt,dor)
%C2
y = exp(-6.0*dor*tt);
end
Appendix B:

Matlab Code for Determining Coupling Constants

This was the code used to determine the coupling constants reported in Chapter 2. There is the main file that defines the initial guesses for the energies and coupling constants in the local site basis.

B.1 MAIN FILE – FIRST GUESSES FOR LOCAL SITE BASIS PARAMETERS

The function file, coupling_fit_main.m, is where the first guesses for local site basis parameters are entered and where the experimental energies are compared to the diagonalized values from the local mode Hamiltonian.

```matlab
1   clear all
2   global tren1
3
4   %These sections have the experimental values for the energies of each mode
5   %and their anharmonicities (in the normal mode basis)
6
7   % % % %trimer
8   % % % r = 2050;
9   % % % t = 2067;
10  % % % a = 2074;
11  % % % b = 2116;
12  % % % an_r = 21;
13  % % % an_t = 24;
```
an_a = 19;
an_b = 24;
an_rt = 2;
an_ra = 9;
an_rb = 3;
an_ta = 15;
an_tb = 9;
an_ab = 10;

dimer in D2O
r = 2048;
t = 2026;
a = 2061;
b = 2096;
an_r = 19;
an_t = 15;
an_a = 28;
an_b = 28;
an_rt = 14;
an_ra = 10;
an_rb = 9;
an_ta = 2;
an_tb = 6;
an_ab = 11;

dimer in FA
r = 2051;
t = 2002;
a = 2065;
% b = 2090;
% an_r = 18;
% an_t = 14;
% an_a = 27;
% an_b = 19;
% an_rt = 2;
% an_ra = 6;
% an_rb = 5;
% an_ta = 2;
% an_tb = 4;
% an_ab = 13;

% computes the energies of the two quantum states
r2 = 2*r-an_r;
t2 = 2*t-an_t;
a2 = 2*a-an_a;
b2 = 2*b-an_b;
rt = r+t-an_rt;
ra = r+a-an_ra;
rb = r+b-an_rb;
ta = a+t-an_ta;
tb = b+t-an_tb;
ab = a+b-an_ab;

tren = [0 r t a b r2 t2 a2 b2 rt ra rb ta tb ab]';
tren1 = sort(tren); % puts them in terms of increasing energy (no longer following the format in line 67)
Here is where the first guesses for energies in the local mode basis are made:

Er1 = 2032;
Et1 = 2004;
Ea1 = 2053;
Eb1 = 2089;
Vrt1 = 25;
Vra1 = 18;
Vrb1 = 19;
Vta1 = 17;
Vtb1 = 22;
Vab1 = 35;
Dr1 = 21;
Dt1 = 18;
Da1 = 20;
Db1 = 21;

dimer in D2O:
Er1 = 2037;
Et1 = 2003;
Ea1 = 2064;
Eb1 = 2090;
Vrt1 = 6;
Vra1 = 6;
Vrb1 = 6;
Vta1 = 6;
Vtb1 = 6;
Vab1=6;
Dr1=21;
Dt1=21;
Da1=21;
Db1=22;

% Constraints on the values

% trimer
lb = [2000,2000,2000, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2];
ub = [2080,2104,2146,30,30,30,30,30,40,40,40,40,40];

% dimer/D2O
% lb = [2000,2000,2000,2000, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2];
% ub = [2080,2090,2104,2146,30,30,30,30,30,40,40,40,40,40];

f0=[Er1 Ea1 Eb1 Vrt1 Vra1 Vrb1 Vta1 Vtb1 Vab1  Dr1  Dt1  Da1  Db1];
options=optimset('MaxFunEvals', 500000, 'MaxIter', 100000, 'TolFun',
1e-200, 'TolX', 1e-350);
[f,resnorm,residual] = lsqnonlin('fit_coupling',f0,[],[],options);

load y

compmat=zeros(15,4);
% commat(2:15,1)=f;
com endif;
compmat(3:15,1)=f;
compmat(2,1)=f(1);
compmat(2,2)=tren1;
B.2 Calculating Difference Between the Experimental and Fit Values

This function file, fit_coupling.m, calls a function that diagonalizes the Hamiltonian, and then it calculates the difference between the experimental and fit values. The sum of these differences are minimized.

```matlab
function s = fit_coupling(f)
    global tren1

    % Calls another function here that diagonalizes the local site basis matrix
    y = fifteenLevHam(f);

    % Difference between experimental values and the values from the diagonalized matrix
    s1 = abs(y(1) - tren1(1));
    s2 = abs(y(2) - tren1(2));
    s3 = abs(y(3) - tren1(3));
    s4 = abs(y(4) - tren1(4));
    s5 = abs(y(5) - tren1(5));
    s6 = abs(y(6) - tren1(6));
    s7 = abs(y(7) - tren1(7));
    s8 = abs(y(8) - tren1(8));
```
s9 = abs(y(9) - tren1(9));
s10 = abs(y(10) - tren1(10));
s11 = abs(y(11) - tren1(11));
s12 = abs(y(12) - tren1(12));
s13 = abs(y(13) - tren1(13));
s14 = abs(y(14) - tren1(14));
s15 = abs(y(15) - tren1(15));

% Trying to minimize the sum of the absolute value of the differences, with
% more emphasis on some of the differences
s = s1 + 4*s2 + 4*s3 + 4*s4 + 4*s5 + s6 + s7 + s8 + s9 + s10 + s11 + s12 + s13 + s14 + s15;

### B.3 Eigenstates of the Local Site Basis Hamiltonian

This function file, fifteenLevHam.m, diagonalizes the Hamiltonian after the local mode energies are entered.

```matlab
function y = fifteenLevHam(f)

% Er = f(1);
% Et = f(2);
% Ea = f(3);
% Eb = f(4);
% Vrt = f(5);
% Vra = f(6);
% Vrb = f(7);
% Vta = f(8);
% Vtb = f(9);
% Vab = f(10);
```
% Placing guesses in the Hamiltonian matrix

Htr = zeros(15, 15);
Htr(2, 2) = Er;
Htr(3, 3) = Et;
Htr(4, 4) = Ea;
Htr(5, 5) = Eb;
Htr(6, 6) = 2*Er - Dr;
Htr(7, 7) = 2*Et - Dt;
Htr(8, 8) = 2*Ea - Da;
Htr(9, 9) = 2*Eb - Db;
Htr(10,10)=Er+Et;
Htr(11,11)=Er+Ea;
Htr(12,12)=Er+Eb;
Htr(13,13)=Et+Ea;
Htr(14,14)=Et+Eb;
Htr(15,15)=Eb+Ea;
Htr1=zeros(15,15);
Htr1(2,3)=Vrt;
Htr1(2,4)=Vra;
Htr1(2,5)=Vrb;
Htr1(3,4)=Vta;
Htr1(3,5)=Vtb;
Htr1(4,5)=Vab;
Htr1(6,10)=sqrt(2)*Vrt;
Htr1(6,11)=sqrt(2)*Vra;
Htr1(6,12)=sqrt(2)*Vrb;
Htr1(7,10)=sqrt(2)*Vrt;
Htr1(7,13)=sqrt(2)*Vta;
Htr1(7,14)=sqrt(2)*Vtb;
Htr1(8,11)=sqrt(2)*Vra;
Htr1(8,13)=sqrt(2)*Vta;
Htr1(8,15)=sqrt(2)*Vab;
Htr1(9,12)=sqrt(2)*Vrb;
Htr1(9,14)=sqrt(2)*Vtb;
Htr1(9,15)=sqrt(2)*Vab;
Htr2=Htr1';
Htr3=Htr1+Htr+Htr2;

%Diagonalizing the Hamiltonian
[~,D] = eig(Htr3);
y=diag(D);
save y y
Appendix C:

Matlab Code for Analyzing t-HDVE Raw Data

The raw data from the transient heterodyne detected vibrational echo (t-HDVE) experiments has to be treated before system information can be extracted. For the data presented in Lynch et al., JPCA, 2012, 116, 7023 and Lynch et al., JCP, 2012, 136, 241101, this is how the data was treated:

- Interpolate the $\omega_3$ axis to equal frequency steps
- Fourier transform into the time domain
- Remove the negative time points
- Apply a window that multiplies the data points before the signal by zero, and multiplies all of the other time points by one
- Fourier transform back into the time domain, zero-padding to 64 points
- To obtain the correct frequency axis, compare it to an analysis where both Fourier transforms have been heavily zero-padded (8192 points), and add the needed offset to the frequency axis

The function files following include two that show how all the data files that LabView saves are loaded into the same matrix for easy use in the function file in Section C.3 where the raw data is actually treated as above.

C.1 LOADING 3RD ORDER DATA FILES INTO SINGLE VARIABLE

The function file, pepsmat_3rd.m, contains the method used to load the data files for the 3rd order signal into a single variable.

```matlab
1 clear all;
```
%This file loads all points saved in different files by LabView and puts %them together in the same variable. This version is for the 3rd order %signal.

file='CN_mat_avg_3rd151001.dat';
file1='CN_mat_avg_3rd15101.dat';
filet='CN_time151001.dat';
filet1='CN_time15101.dat';

firstfile=1;
i=firstfile;
k=i;

lastfile=5;
m=lastfile;
l=m;

for t=k:1:
    if t<10
        sth=num2str(t);
        file1(18)=sth;
        filet1(11)=sth;
    end
    data=load(file1);
    time=load(filet1);
    matrixd11(:,:,t)=data;
    matrixt11(:,:,t)=time;
else
    tstr=num2str(t);
    file(18)=tstr(1);
    file(19)=tstr(2);
    filet(11)=tstr(1);
    filet(12)=tstr(2);
    data=load(file);
    time=load(filet);
    matrixd11(:,:,t)=data;
    matrixt11(:,:,t)=time;
end
end

file='CN_mat_avg_3rd152001.dat';
file1='CN_mat_avg_3rd15201.dat';
filet='CN_time152001.dat';
filet1='CN_time15201.dat';

firstfile=1;
i=firstfile;
k=i;

lastfile=28;
m=lastfile;
l=m;

for t=k:l;
if t<10
    sth=num2str(t);
    file1(18)=sth;
    filet1(11)=sth;
    data=load(file1);
    time=load(filet1);
    matrixd21(:,:,t)=data;
    matrixt21(:,:,t)=time;
else
    tstr=num2str(t);
    file(18)=tstr(1);
    file(19)=tstr(2);
    filet(11)=tstr(1);
    filet(12)=tstr(2);
    data=load(file);
    time=load(filet);
    matrixd21(:,:,t)=data;
    matrixt21(:,:,t)=time;
end

file='CN_mat_avg_3rd153001.dat';
file1='CN_mat_avg_3rd15301.dat';
filet='CN_time153001.dat';
filet1='CN_time15301.dat';
firstfile=1;
i=firstfile;
k=i;

lastfile=5;
m=lastfile;
l=m;

for t=k:l;
if t<10
    sth=num2str(t);
    file1(18)=sth;
    file1(11)=sth;
    data=load(file1);
    time=load(filet1);
    matrixd31(:,:,t)=data;
    matrixt31(:,:,t)=time;
else
    tstr=num2str(t);
    file(18)=tstr(1);
    file(19)=tstr(2);
    filet(11)=tstr(1);
    filet(12)=tstr(2);
    data=load(file);
    time=load(filet);
    matrixd31(:,:,t)=data;
    matrixt31(:,:,t)=time;
end
end
\[ [a \ b \ c] = \text{size(matrixd11)}; \]
\[ [a1 \ b1 \ c1] = \text{size(matrixd21)}; \]
\[ [a2 \ b2 \ c2] = \text{size(matrixd31)}; \]
\[ [at \ bt \ c] = \text{size(matrixt11)}; \]
\[ [at1 \ bt1 \ c1] = \text{size(matrixt21)}; \]
\[ [at2 \ bt2 \ c2] = \text{size(matrixt31)}; \]

pepsmatd15 = zeros(a, b, c + c1 + c2);
pepsmatt15 = zeros(at, bt, c + c1 + c2);

pepsmatd15(:,:,1:c) = matrixd11;
pepsmatd15(:,:,c+1:c+c1) = matrixd21;
pepsmatd15(:,:,c+c1+1:end) = matrixd31;

pepsmatt15(:,:,1:c) = matrixt11;
pepsmatt15(:,:,c+1:c+c1) = matrixt21;
pepsmatt15(:,:,c+c1+1:end) = matrixt31;

save pepsmatt15 pepsmatt15
save pepsmatd15 pepsmatd15

C.2 Loading 5th Order Data Files into Single Variable

The function file, pepsmat_5th.m, contains the method used to load the data files for the 3rd order signal into a single variable.

clear all;
This file loads all points saved in different files by LabView and puts them together in the same variable. This version is for the 5th order signal.

```
file='CN_mat_avg_5th151001.dat';
file1='CN_mat_avg_5th15101.dat';

firstfile=1;
i=firstfile;

k=i;

lastfile=5;
m=lastfile;
l=m;

for t=k:l;
  if t<10
    sth=num2str(t);
    file1(18)=sth;
  end
  data=load(file1);
  matrixd11(:,:,t)=data;
  else
    tstr=num2str(t);
    file(18)=tstr(1);
    file(19)=tstr(2);
```
data=load(file);
matrixd11(:,:,t)=data;
end
end

file='CN_mat_avg_5th152001.dat';
file1='CN_mat_avg_5th15201.dat';

firstfile=1;
i=firstfile;
k=i;

lastfile=28;
m=lastfile;
l=m;

for t=k:l;
if t<10
sth=num2str(t);
file1(18)=sth;
data=load(file1);
matrixd21(:,:,t)=data;
else
tstr=num2str(t);
file(18)=tstr(1);
file(19)=tstr(2);
data=load(file);
matrixd21(:,:,t)=data;
file='CN_mat_avg_5th153001.dat';
file1='CN_mat_avg_5th15301.dat';

firstfile=1;
i=firstfile;

for t=k:l;
    if t<10
        sth=num2str(t);
        file1(18)=sth;
        data=load(file1);
        matrixd31(:,:,t)=data;
    else
        tstr=num2str(t);
        file(18)=tstr(1);
        file(19)=tstr(2);
        data=load(file);
        matrixd31(:,:,t)=data;
    end
end
end
end
This function file, pepsmat_analysis_0pad_time.m, contains the method used to treat the raw data from a t-HDVE experiment.

```matlab
1 clear all
2 tic
3
4 %from pepsmat_3rd.m or pepsmat_5th.m
5 load pepsmat2
6 dat=pepsmat2;
7 load pepsmatt2
8 load w3
9 t1=-pepsmatt2(3,:,:);
```
[a1,b1,c1]=size(dat);
c=2.99792458E10; %speed of light in cm/s
f3=w3.*c; %converting from cm^-1 to s^-1
data2=sortrows(f3);
datan=f3;
xstep=f3(2)-f3(1);
x=round(datan(1)):xstep:round(datan(end));
px=length(x);

for m=1:c1; %m => tau2
    for n=1:b1; % n => tau1, 11 is tau1=0 for t-HDVE_02
        %interpolates so there is equal spaces in the freq domain
        y(:,n,m)=spline(f3,dat(:,n,m),x);
    end
end
newx=x;
newy=zeros(length(newx),n,m);
newy(:,n,m)=y(:,n,m);
%first FT from freq to time domain (no zero padding)

ffm(:,n,m)=ifft(newy(:,n,m));

ffm1(:,n,m)=fftshift(ffm(:,n,m));

a=length(ffm(:,1,1));

%Defining the time axis after the first FT

time1=zeros(1,a);

for k=1:a
    time1(k)=k/(a*xstep);
end

time2=time1(1:a/2);
time3=flipdim(time2,2).*(-1);
time=zeros(1,a+1);
time(1:a/2)=time3;
time(a/2+2:a+1)=time2;

%Putting first FT into the same points as new time axis
ffm3=zeros(a+1,n,m);
ffm3(1:a/2,n,m)=ffm1(1:a/2,n,m);
ffm3(a/2+2:end,n,m)=ffm1(a/2+1:end,n,m);
ffm2(:,:,n,m)=ffm3(a/2+1:end,n,m);
time(1:a/2+1)=[];

%This is the filter - a shifted heaviside function
fil=ones(length(ffm2(:,1)),n,m);
fil(1:6,n,m)=0; %This is where the filter is shifted, change '1:6'
gtime=time;

%The filter is applied to the data in the time domain here
filt(:,:,n,m)=ffm2(:,:,n,m).*fil(:,:,n,m);

%Second FT - zero padded to 64 to give the same number of points in
%the freq domain that we started with
stuff(:,:,n,m)=fft(filt(:,:,n,m),64);
fl=length(stuff(:,:,1,1));

%Defining the new freq axis - this needs to be compared to a
%version where the second FT was heavily zero-padded and the
%frequency is more accurately determined
newfreq=zeros(size(stuff(:,:,1,1)));
time_step=gtime(3)-gtime(2);
for p = 1:fl;
    newfreq(p)=p/(fl*time_step);
end
w3new=newfreq./c;
%calculation of the pump-probe

dvepp(:,n,m)=stuff(:,n,m);

%calculation of the dispered vibrational echo

dve(:,n,m)=(abs(dvepp(:,n,m))).^2;

end
end

figure
plot(f3,dat(:,21,1),'linewidth',2)
axis square
set(gca, 'Tickdir', 'out');
set(gca, 'FontWeight','bold')
set(gca,'FontSize',14)
set(gca, 'FontName','Arial')
set(gca,'TickLength', [0.03, 0.01])
set(gca,'LineWidth', 2)
xlabel('freq (Hz)')
title('Raw Data')

figure
plot(time,abs(ffm2(:,21,1)),time,ffm2(:,11,1),gtime,filt(:,11,1),'linewidth',2)
xlim([0 4e-12])
axis square
set(gca, 'Tickdir', 'out');
set(gca, 'FontWeight','bold')
set(gca,'FontSize',14)
set(gca, 'FontName','Arial')
set(gca,'TickLength', [0.03, 0.01])
set(gca,'LineWidth', 2)
xlabel('time (s)')
title('After First FT from freq to time')
legend('absolute value','before filter','after filter')
figure
plot(w3new,dve(:,21,1),'linewidth',2)
axis square
set(gca, 'Tickdir', 'out');
set(gca, 'FontWeight','bold')
set(gca,'FontSize',12)
set(gca, 'FontName','Arial')
set(gca,'TickLength', 
[0.03, 0.01])
set(gca,'LineWidth', 2)
xlabel('\omega_{3}')
title('Calculated DVE')

figure
plot(t1(1,:,1),dve(44,:,1),'linewidth',2)
axis square
set(gca, 'Tickdir', 'out');
set(gca, 'FontWeight','bold')
set(gca,'FontSize',12)
set(gca, 'FontName','Arial')
set(gca,'TickLength', [0.03, 0.01])
set(gca,'LineWidth', 2)
xlabel('\tau_1 (fs)')
title('Calculated DVE')

dvepp2_0pad_time_sfthv=dvepp;
dvecalc2_0pad_time_sfthv=dve;
w3new_0pad_time=w3new;
%These are the files that are saved in the 'raw' data file folder.

This is

%the data Mike started with for the analysis in

%The Journal of Physical Chemistry A 116 (2012): 7023-7032. and

save dvepp2_0pad_time_sfthv dvepp2_0pad_time_sfthv

save dvecalc2_0pad_time_sfthv dvecalc2_0pad_time_sfthv

save w3new_0pad_time w3new_0pad_time

toc % 1 min
VITA

Karla M. Slenkamp was born in Seattle, WA in 1987. After completing her work at Henry M. Jackson High School in 2005, she entered Western Washington University where she studied chemistry. There she received her Bachelor of Science, *Magna Cum Laude*, in June 2009, and entered the Graduate School at the University of Washington in September 2009. She completed her Doctor of Philosophy in Chemistry at UW in June 2015.