

**Development of Frequency Domain Multidimensional Spectroscopy
with Applications in Semiconductor Photophysics**

By
Blaise Jonathan Thompson

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

Doctor of Philosophy
(Chemistry)

at the
UNIVERSITY OF WISCONSIN - MADISON
2018

Date of final oral examination: April 23, 2018

This dissertation is approved by the following members of the Final Oral Committee:

John C. Wright, Professor, Analytical Chemistry
Randall Goldsmith, Professor, Physical Chemistry
Tim Bertram, Professor, Analytical Chemistry
Kyoung-Shin Choi, Professor, Materials Chemistry

Contents

List of Figures	ix
List of Tables	xiii
Acknowledgments	xv
Abstract	xvii
1 Introduction	1
I Background	9
2 Spectroscopy	11
2.1 Light-matter interaction	12
2.2 Types of spectroscopy	15
2.2.1 Linear vs multidimensional	15
2.2.2 Frequency vs time domain	16
2.2.3 Homodyne vs heterodyne	17
3 Software	19
3.1 Science needs software	20
3.2 Challenges in scientific software development	22
3.3 Good-enough practices	24
3.4 Object oriented programming	27
3.5 Hierarchical data format	30
3.6 Scientific Python	32

II	Development	33
4	Processing	35
4.1	Introduction to WrightTools	37
4.2	Creating a data object	47
4.3	Collections	50
4.3.1	From directory	51
4.4	Visualizing a data object	52
4.4.1	Strategies for 2D visualization	52
4.4.2	Quick	57
4.5	Variables and channels	60
4.6	Axes	61
4.7	Math	63
4.7.1	In-place operators	63
4.7.2	Clip	64
4.7.3	Symmetric root	64
4.7.4	Log	65
4.7.5	Trim	65
4.7.6	Smooth	66
4.8	Dimensionality manipulation	68
4.8.1	Chop	68
4.8.2	Collapse	69
4.8.3	Split	69
4.8.4	Join	70
4.9	Fitting	72
4.9.1	Function objects	72
4.9.2	Fitter	74
4.10	Construction, maintenance, and distribution	77
4.10.1	Licensing	77
4.10.2	Version control	78
4.10.3	Unit tests	80
4.10.4	Distribution	80

4.10.5	Documentation	81
5	Acquisition	83
5.1	Introduction	84
5.2	Graphical user interface	87
5.3	Internal structure	92
5.3.1	Multithreading	92
5.3.2	Abstraction and inheritance	93
5.3.3	Core classes of PyCMDS	95
5.4	Hardware	101
5.4.1	Hardware inheritance	101
5.4.2	Delays	104
5.4.3	Spectrometers	106
5.4.4	OPAs	108
5.5	Sensors (devices)	110
5.5.1	The DAQ card	110
5.5.2	Multidimensional sensors	114
5.6	Autonomic	116
5.7	Somatic	118
5.7.1	Queue manager	118
5.7.2	Scans	119
5.7.3	Acquisition modules	120
5.8	Conditional validity	124
5.9	Integrations	125
5.10	Future directions	127
5.10.1	Spectral delay correction module	127
5.10.2	“Headless” hardware, sensors	127
5.10.3	Ideal Axis Positions	127
5.10.4	Simultaneous acquisitions	132
5.10.5	Hotswappable hardware	132
5.10.6	wt5 savefile	132

6	Active correction	133
6.1	Introduction	134
6.2	Spectral delay correction	135
6.3	Poynting correction	138
6.4	Chopping	140
6.4.1	Scatter	140
6.4.2	Fibrillation	148
7	A robust, automated strategy to collect high quality OPA tuning curves	149
7.1	Introduction	150
7.2	Curves	151
7.3	TOPAS-C	152
7.4	Preamp	156
7.5	Poweramp	160
7.6	Mixers	165
7.7	Generalizability	166
8	Disentangling material and instrument response	169
8.1	Introduction	170
8.2	Theory	173
8.2.1	Characteristics of driven and impulsive response	179
8.2.2	Inhomogeneity	181
8.3	Methods	185
8.3.1	Liouville pathway parameters	185
8.3.2	Matrix formulation	187
8.3.3	Heun method	192
8.3.4	Scripts & raw output	196
8.3.5	Simulation package	196
8.4	Results	204
8.4.1	Evolution of single coherence	204
8.4.2	Evolution of single Liouville pathway	212
8.4.3	Temporal pathway discrimination	218

8.4.4	Multidimensional line shape dependence on pulse delay time	221
8.4.5	Inhomogeneous broadening	227
8.5	Discussion	236
8.5.1	An intuitive picture of pulse effects	236
8.5.2	Conditional validity of the driven limit	237
8.5.3	Extracting true material correlation	239
8.6	Conclusion	242
III	Applications	243
9	Resonant third-order susceptibility of PbSe quantum dots determined by standard dilution and transient grating spectroscopy	245
9.1	Introduction	246
9.2	Theory	248
9.2.1	Extraction of susceptibility	248
9.2.2	Optical bleaching and dependencies on experimental conditions	250
9.3	Experimental	252
9.4	Results	255
9.4.1	Solvent and window response	255
9.4.2	Concentration-dependent corrections	258
9.4.3	Quantum dot response	260
9.5	Discussion	266
9.6	Conclusion	271
10	Global Analysis of Transient Grating and Transient Absorption of PbSe Quantum Dots	273
10.1	Introduction	274
10.2	Theory	277
10.2.1	The bleach nonlinearity	280
10.2.2	TG/TA scaling	280
10.2.3	The absorptive third-order susceptibility	283
10.3	Methods	285
10.4	Results	287

10.4.1	Pump-probe 3D acquisitions for TA and TG	287
10.4.2	The skewed TG probe spectrum	288
10.5	Discussion	290
10.5.1	Comparison of TA and TG line shapes	290
10.5.2	Determination of state filling factor	297
10.5.3	Inhomogeneity and the pulse overlap response	298
10.6	Conclusion	301
11	Measurement of Ultrafast Excitonic Dynamics of Few-Layer MoS₂ using State-Selective Coherent Multidimensional Spectroscopy	303
11.1	Introduction	305
11.2	Methods	309
11.3	Results and discussion	318
11.4	Conclusions	332
11.5	Errata	333
12	Measurement of ultrafast dynamics within PEDOT:PSS using three-pulse photon echo spectroscopy	335
12.1	Introduction	336
12.2	Methods	337
12.3	Results	339
12.4	Discussion	342
12.4.1	Assignment of zero delay	342
12.4.2	Numerical model	347
12.5	Conclusions	352
IV	Appendix	353
A	Procedures	355
A.1	“Six-month” maintenance	356
A.2	Lytron Kodiak RC006	358
A.3	PolyScience 6000 Series	359
A.4	NesLab Merlin M33	360
A.5	Calibrating the 407A	361

A.6	Millenia	362
A.6.1	Startup	362
A.6.2	Toggling service mode	362
A.7	Spitfire Pro	363
A.7.1	Startup	363
A.7.2	Common alignment	363
A.7.3	Stretcher alignment	366
A.7.4	Compressor alignment	366
A.8	TOPAS-C	367
A.8.1	Common alignment	367
A.8.2	Full alignment	368
A.9	MicroHR Monochromator	372
B	Instrumental response function	373
C	Quantitative differential absorbance	375
D	Abandon the random phase approximation	379
E	Colophon	383

List of Figures

4.1	Colormap lightness.	54
4.2	Colormap comparison.	55
4.3	Interpolation types in 2D plotting.	56
4.4	quick1D example.	58
4.5	quick2D example.	59
4.6	Example using transform.	62
4.7	Outlier detection using a z-test.	67
4.8	Join example.	71
4.9	Fitting a Gaussian.	73
4.10	Fitting as dimensionality reduction.	76
5.1	PyCMDS at startup.	89
5.2	PyCMDS queue.	90
5.3	PyCMDS while scanning.	91
5.4	Parent to hardware and sensors.	97
5.5	Parent class of all drivers.	98
5.6	Parent hardware class.	102
5.7	Hardware inheritance.	103
5.8	Representative delay stage advanced menu.	105
5.9	Representative spectrometer advanced menu.	107
5.10	Representative OPA advanced menu.	109
5.11	PCI-6251 samples tab.	111
5.12	PCI-6251 shots tab.	112
5.13	Simple shots processing script for dual chopping.	113

5.14	Array detector serving as an axis.	115
5.15	Autonomic tab.	117
5.16	Representative aqn file.	121
5.17	Slack screenshot.	126
5.18	Ideal axis positions, exponential decay.	131
6.1	Spectral delay correction.	137
6.2	Poynting correction.	139
6.3	Simulated interference paterns in old delay parameterization.	141
6.4	Simulated interference paterns in current delay parameterization.	143
6.5	Comparison of single, dual chopping.	147
7.1	TOPAS-C interaction ranges.	153
7.2	TOPAS-C internal optics and beam path.	154
7.3	TOPAS-C interaction range output powers.	155
7.4	TOPAS-C preamp motortune.	157
7.5	Preamp tune procedure flowchart.	158
7.6	Preamp tuning output.	159
7.7	TOPAS-C poweramp motortune.	161
7.8	Poweramp tune procedure flowchart.	162
7.9	Poweramp D2 tuning output.	163
7.10	Poweramp C2 tuning output.	164
7.11	Picosecond OPA motortune.	167
8.1	Sixteen triply-resonant Liouville pathways.	177
8.2	Overview of the MR-CMDS simulation.	178
8.3	Convolution overview.	184
8.4	Liouville pathway network.	189
8.5	Integration technique comparison.	195
8.6	Relative importance of FID and driven response for a single quantum coherence.	208
8.7	Pulsed excitation of a single quantum coherence and its dependance on pulse detuning.	209
8.8	Frequency domain representation of a single quantum coherence vs pulse detuning.	210

8.9	Time-gated amplitude of a single quantum coherence vs pulse detuning.	211
8.10	2D frequency response of a single Liouville pathway at different delay values.	215
8.11	2D frequency response of a single Liouville pathway without a tracking monochromator.	216
8.12	2D delay response for different relative dephasing rates.	220
8.13	Evolution of the 2D frequency response.	224
8.14	Evolution of the 2D frequency response, with all contours shown.	225
8.15	Simulated Wigner spectra.	226
8.16	3PEPS tutorial.	229
8.17	2D delay response with inhomogeneity.	230
8.18	2D delay response for all combinations of inhomogeneity, dephasing rate.	231
8.19	Spectral evolution of an inhomogeneous system.	232
8.20	Spectral evolution of an inhomogeneous system, with all contours shown.	233
8.21	Eccentricity at zero delay.	234
8.22	Eccentricity at large population time.	235
8.23	Conditional validity of the driven limit.	238
8.24	Metrics of correlation.	241
9.1	Absorption spectra of QD aliquots used in this study.	253
9.2	CMDS amplitude of neat carbon tetrachloride.	257
9.3	Absorption effects in QD dilution study.	259
9.4	FWM at pulse overlap, and integral thereof.	262
9.5	CMDS signal with different concentrations of PbSe.	264
10.1	Similarities between transient grating and transient absorption measurements.	276
10.2	Model system for the 1S band of PbSe quantum dots.	279
10.3	TA and TG 2D spectra as function of delay.	289
10.4	Kramers-Kronig analysis of TA, TG spectra.	291
10.5	Fits of probe traces.	295
10.6	Global simulation.	300
11.1	Introduction to CMDS.	308
11.2	Schematic of sulfidation reaction chamber.	310

11.3 OPA outputs at each color explored.	312
11.4 Spectral delay correction.	313
11.5 Mask and epi vs transmissive.	315
11.6 MoS ₂ post processing.	317
11.7 Few-layer MoS ₂ thin film characterization.	319
11.8 MoS ₂ absorbance.	320
11.9 MoS ₂ frequency-frequency slices.	322
11.10 MoS ₂ ω_1 Wigner progression.	324
11.11 MoS ₂ ω_2 Wigner progression.	325
11.12 MoS ₂ transients.	329
11.13 MoS ₂ frequency-frequency slices near pulse overlap.	331
12.1 Phase matching mask for 3PE, 3PE*.	338
12.2 PEDOT:PSS transmission and reflectance spectra.	340
12.3 Raw 3PE data.	341
12.4 3PE, 3PE* delay space.	344
12.5 Delay offsets.	345
12.6 Peak shift traces drawn in delay space.	346
12.7 3PEPS parameter space.	349
12.8 Agreement between simulation and experiment.	351
D.1 Auto-interference vs cross-interference.	381
D.2 Cross interference over 100 seconds.	382

List of Tables

4.1	Components of WrightTools.	40
4.2	Attributes and methods of Data.	42
4.3	Attributes and methods of Channel and Variable.	44
4.4	Attributes and methods of Axis.	46
4.5	WrightTools data import functions.	49
4.6	Commits to WrightTools.	79
6.1	Shot-types in phase shifted parallel modulation.	145
8.1	Parameters of each Liouville Pathway.	186
8.2	Parameters used in large NISE simulation.	198
8.3	Description of four-dimensional simulation coordinates.	200
8.4	Simulation output format (polarization).	201
8.5	Simulation output format (measured).	203
8.6	Conditions for peak intensity at different pulse delays for pathway I γ	217
9.1	Parameters and extracted values from the global line shape fit.	265
9.2	Non-linear parameters relative to carbon tetrachloride hyperpolarizability.	269
9.3	Comparison with PbX literature measurements.	270
10.1	Batch parameters extracted from absorption spectra.	286
10.2	Parameters used in fitting probe slices.	292
10.3	Parameters extracted by global fits.	296
10.4	Homogeneous and inhomogeneous linewidths.	299
12.1	Fitted parameters.	350

Acknowledgments

To John, you have truly been better than the best adviser I could have hoped for. Your mentorship has helped me realize what truly matters, and you have helped me to find myself professionally and personally. Thank you so much.

To my colleagues, you have each been at times mentor and at times friend. I am so proud of the work we have done together. I am excited to see what each of you accomplishes next.

To Tyler, thank you for showing me how to try hard and have fun.

To Claire, thank you for being my confidant and foundation for these last nine years.

To Sam, thank you for walking beside me through this crazy life.

To my parents, thank you for being endlessly supportive, despite my obstinance.

Finally, thank you to all humans who have and continue to undertake the ongoing free and responsible search for truth and meaning. Thanks to free software / free culture / open science advocates who have worked to create and share foundational tools and ideas, often at great personal opportunity cost. Thanks to thought leaders who have shown me what it means to have a good life without abandoning moral principles. And thank you to those who bravely speak truth to power. This universe is stranger, more terrible, and more fantastic than we want to believe. We must find ways to describe it's complexity without falling victim to the sometimes-overwhelming power of simple, "useful" narratives.

The explanatory stories that people find compelling are simple; are concrete rather than abstract; assign a larger role to talent, stupidity and intentions than to luck; and focus on a few striking events that happened rather than on the countless events that failed to happen.

The ultimate test of an explanation is whether it would have made the event predictable in advance.

Paradoxically, it is easier to construct a coherent story when you know little, when there are fewer pieces to fit into the puzzle. Our comforting conviction that the world makes sense rests on a secure foundation: our almost unlimited ability to ignore our ignorance.

– Daniel Kahneman [1]

Abstract

Coherent multidimensional spectroscopy (CMDS) encompasses a family of experimental strategies involving the nonlinear interaction between electric fields and a material under investigation. This approach has several unique capabilities:

1. resolving congested states [2, 3]
2. extracting spectra that would otherwise be selection-rule disallowed [4, 5],
3. resolving fully coherent dynamics [6],
4. measuring coupling [7], and
5. resolving ultrafast dynamics.

CMDS can be collected in the frequency or the time domain, and each approach has advantages and disadvantages. [8] Frequency domain “Multi-resonant” CMDS (MR-CMDS) requires pulsed ultrafast light sources with tunable output frequencies. These pulses are directed into a material under investigation. The pulses interact with the material, and due to the specific interference between the multiple fields the material is driven to emit a new pulse: the MR-CMDS signal. This signal may have a different frequency and/or direction than the input pulses, depending on the exact experiment being performed. The MR-CMDS experiment involves tracking the intensity of this output signal as a function of different properties of the excitation pulses. These properties include 1. frequency 2. relative arrival time and separation (delay) 3. fluence [9, 10], and 4. polarization [11], among others. Thus MR-CMDS can be thought of as a multidimensional experimental space, where experiments typically involve explorations in one to four of the properties above.

Because MR-CMDS is a family of related-but-separate experiments, each of them a multidimensional space, there are special challenges that must be addressed when designing a general-purpose MR-CMDS instrument. These issues require development of software, hardware, and theory. Part I: Background introduces relevant literature which informs on this development work. Part II: Development presents five strategies used to improve MR-CMDS: 1. processing software (Chapter 4), 2. acquisition software (Chapter 5) 3. active artifact correction (Chapter 6), 4. automated OPA calibration (Chapter 7), and 5. finite pulse accountancy (Chapter 8). Finally, Part III: Applications presents four examples where these instruments, with these improvements, have been used to address chemical questions in semiconductor systems.

Chapter 1

Introduction

Experience has shown that the experimental implementation of three laser FWM is a formidable task which is far from complete. Characterization and improvement of existing apparatus must be an integral part of future research if FWM is to become a viable spectroscopic technique. To this end, it would be helpful to stimulate an atmosphere within the scientific community in which the method of measurement is considered to be as important as the result of the measurement.

– Roger Carlson (1988) [12]

Coherent multidimensional spectroscopy (CMDS) is a family of experimental strategies capable of providing unique insights into microscopic material physics. It is similar to more familiar multidimensional NMR experiments [13, 14, 15], although the implementation is different due to differences between the behavior of nuclear spin states (probed by NMR) and electronic and vibrational states (probed by CMDS). CMDS can resolve couplings between states, and can decongest spectra by taking advantage of dimensionality and selection rules. With the advent of ultrafast lasers, CMDS can resolve dynamics in excited states and the coupling between them. [16]

CMDS is most often performed in the time domain, where multiple broadband pulses are scanned in time (phase) to collect a multidimensional interferogram. [17, 18] This technique is fast and robust—it has even been performed on a single shot. [19] However time-domain CMDS has some fundamental limitations:

- The frequency bandwidth must be contained within the excitation pulse—and ultrabroadband pulses are hard to make and control. [20]
- A phase-locked local oscillator is required, and preparing a local oscillator for experiments with unique output colors is challenging.

Scientists in the time-domain CMDS community are taking both of these challenges head-on, pushing the envelope in excitation pulse bandwidth [21] and performing two-stage experiments in which excitation pulses are used to generate a local oscillator in non-resonant media [22].

An alternative strategy is frequency domain “multi-resonant” CMDS (MR-CMDS). Rather than using a single broadband excitation pulse, MR-CMDS employs a relatively narrow-band source with a tunable frequency. Motorized optical parametric amplifiers (OPAs) are typically used to provide this tunability. [23] In MR-CMDS, frequency axes are resolved directly by scanning these motorized OPAs. This process is time intensive, and it can be challenging to ensure that the OPAs are well calibrated and that the experiment is not affected by the motion of crystals and other optics inside these automated OPAs. Despite these challenges, MR-CMDS is an incredibly flexible strategy that could become a powerful analytical tool. [6] Because MR-CMDS does not require that all frequencies be contained within one broadband source, there is no theoretical limit to the frequency range that can be resolved in this way. MR-CMDS can be homodyne-detected, so experiments with unique output colors are much more accessible. Finally, because the components are more self-contained, MR-CMDS instruments tend to

be more flexible in the kinds of experiments that they can perform.

This dissertation contains several projects undertaken to improve the reliability and accessibility of MR-CMDS. While MR-CMDS will never be a single-shot experiment, there are many improvements that can improve data collection speed. Necessary calibration, especially OPA calibration, can be made robust and fully automatic. Common artifacts can be addressed through relatively simple modifications in hardware and software. Finally, the complexity that arises from finite pulses with “marginal” resolution in frequency and time can be understood and accounted for through numerical simulation. Taken together, these improvements represent a significant improvement in the accessibility of frequency-domain coherent multidimensional spectroscopy.

Due to its diversity and dimensionality, MR-CMDS data is challenging to process and represent. The data processing tools that a scientist develops to process one experiment may not work when she attempts to process an experiment where different experimental variables are explored. Historically, this processing strategy has resulted in MR-CMDS practitioners have using custom, one-off data processing workflows that need to be changed for each particular experiment. These changes take time to implement, and can become stumbling blocks or opportunities for error. Even worse, the challenge of designing a new processing workflow may dissuade a scientist from creatively modifying their experimental strategy, or comparing their data with data taken from another group. This limit to creativity and flexibility defeats one of the main advantages of the MR-CMDS strategy. Chapter 4 describes a new software package, WrightTools, that greatly simplifies CMDS data processing. WrightTools defines a *universal format* that is capable of representing any CMDS dataset, regardless of dimensionality or the axes scanned. A set of simple functions are used to convert raw data into this universal format. Once converted, the data can be manipulated with a set of powerful methods that encompass the majority of operations needed to process such data. Finally, simple tools are defined to quickly and beautifully represent the datasets. WrightTools is made to be extended, so it will continue to evolve along with its users.

From an instrumental perspective, MR-CMDS is a problem of calibration and coordination. Within the Wright Group, each of our two main instruments are composed of roughly ten actively moving component hardwares. Many of these components are purchased directly from commercial vendors, while others are created or heavily modified by graduate students. The Wright Group has always maintained custom

acquisition software packages which control the complex, many-stepped dance that these components must perform to acquire MR-CMDS spectra.

When I joined the Wright Group, I saw that acquisition software was a barrier to experimental progress and flexibility. Graduate students had ideas for instrumental enhancements that were infeasible because of the challenge of incorporating the new components into the existing software ecosystem. At the same time, students were spending much of their time in lab repeatedly calibrating optical parametric amplifiers by hand, a process that sometimes took days. I chose to spend a significant portion of my graduate career focusing on solving these problems through software development. At first, I focused on improving the existing LabVIEW code. Eventually, I developed a vision for a deeply modular acquisition software that could not be practically created with LabVIEW. Using Python and Qt, I created a brand new acquisition software PyCMDS: built from the ground up to fundamentally solve historical challenges in the Group. PyCMDS offers a modular hardware model that can “re-configure” itself to flexibly control a variety of component hardware configurations. This has enabled graduate students to add and remove hardware whenever necessary, without worrying about a heavy additional programming burden. PyCMDS is now used to drive both MR-CMDS instruments in the Group, allowing for easy sharing of component hardware and lessening the total amount of software that the Group needs to maintain. Besides being more flexible than prior software, PyCMDS solves a number of other problems. It offers fully automated strategies for calibrating component hardware, making calibration less arduous and more reproducible. It offers more fine-grained control of data acquisition and timing, enabling more complex algorithms to quickly acquire artifact-free results. In conjunction with other algorithmic and hardware improvements that I have made, PyCMDS has decreased acquisition times by up to two orders of magnitude.

Like any analytical technique, MR-CMDS is subject to artifacts: features of the data that are caused by instrumental imperfections or limitations, and do not reflect the intrinsic material response that is of interest. For example, consider well-known artifacts such as absorptive effects [24], pulse effects [20], and window contributions [25]. Since MR-CMDS is a very active experiment, with many moving motors, an active approach to artifact correction is particularly appropriate. Chapter 6 describes strategies for implementing such corrections. Spectral delay correction can be applied to account for the fact that not all output colors arrive at the same time. Dual chopping can correct for scatter and other unwanted processes, ensuring that the observed signal depends on all of the excitation beams. Fibrillation can wash

out interference between desired and undesired processes, and is complementary to chopping. Automated poynting correction and power correction can account for non-idealities in OPA performance.

MR-CMDS instruments rely on OPAs as tunable light sources. OPAs are very sensitive to changes in upstream lasers and lab conditions, so OPA tuning is regularly required. Manual OPA tuning can easily take a full day of human effort. Furthermore, manual tuning typically results in inferior tuning curves, since it is difficult for humans to consider all available information simultaneously. Automated OPA tuning makes OPA upkeep easier, faster and more reproducible, facilitating higher throughput, higher quality frequency domain experiments. Chapter 7 describes fully automated tuning algorithms which I have developed.

The theory that is used to describe CMDS is typically derived in one of two limits. In the impulsive limit, pulses are broad in frequency and short in time compared to material resonances. Resonant responses are impulsive, like a hammer hitting a bell. The impulsive limit is particularly well suited for describing time domain experiments. In the driven limit, pulses are narrow in frequency and long in time compared to material response. Resonant responses are driven, like jiggling jello dessert sitting on a washing machine. The expected spectrum in both of these limits can be computed analytically. Things get more complicated in the mixed domain, where pulses have similar bandwidth as the material response. Experiments in this domain are a practical necessity as CMDS addresses systems with very fast dephasing times. [26, 27] At the same time, the marginal resolution in frequency *and* time that the mixed domain possess promises huge potential in pathway resolution and decongestion. [6] Chapter 8 describes the pitfalls and opportunities contained in the mixed domain approach. An intuitive description of mixed-domain experiments is given. False signatures of material correlation are discussed, and strategies for resolving true material correlation are defined.

In Part III: Applications, four projects in which MR-CMDS was used to answer chemical questions in materials systems are described. These chapters do not directly address improvements to the MR-CMDS methodology, but instead serve as case studies in the potential of MR-CMDS and the utility of the improvements described in Part II.

In Chapter 9, we employ transient grating MR-CMDS to interrogate the photophysics of lead selenide (PbSe) quantum dots (QDs). PbSe QDs are an interesting semiconductor system with many appealing

properties for basic method development work. They are easy to synthesize, store and prepare in the solution phase, and they have bright and relatively narrow band-edge excitons which are easy to interrogate using MR-CMDS. In Chapter 9, we describe a simple approach to extracting the quantitative third-order susceptibility of PbSe quantum dots using MR-CMDS. Using standard dilutions, we define this susceptibility in ratio to the known well-quantified susceptibility of our solvent and cuvette windows. A few-parameter model is employed to extract this ratio. We are optimistic that this approach will be generally applicable, making it simple to perform quantitative solution-phase MR-CMDS.

In Chapter 10 we continue to investigate PbSe QDs. Here we combine transient grating and transient absorption MR-CMDS to learn more about the nonlinear spectrum near the band edge, around the 1S exciton. By combining both methods with the information from Chapter 9, we are able to extract the complete amplitude and phase of the non-linear susceptibility. We develop a simulation that relates the microscopic physics of PbSe electronic states to transient grating and transient absorption spectra, and fit our model to both spectra simultaneously. Our model reveals the presence of continuum transitions, mostly invisible in typical transient absorption experiments. We show that our model is able to describe spectra from two different syntheses with two different sizes of quantum dot.

In Chapter 11 we report the first MR-CMDS study performed on a molybdenum disulfide thin film. MoS₂ is a member of a class of materials called transition metal dichalcogenides which have recently attracted a large amount of attention for their unique photophysical properties. [28, 29, 30] These thin films have relatively low optical density and are highly scattering, making them particularly challenging for MR-CMDS experiments. We employed several strategies to overcome these challenges, and performed three dimensional frequency-frequency-delay transient grating spectroscopy to understand the basic coupling and dynamics of MoS₂. We show that the band-edge excitons of MoS₂ are not easily resolved, and the dynamics of MoS₂ are fast. We describe a picture of MoS₂ electronic states that is consistent with our results.

In Chapter 12 we use MR-CMDS to interrogate the dynamics of electronic states of PEDOT:PSS. PEDOT:PSS is a transparent, electrically conductive polymer. The exact origin of the conductivity is not well understood, so it is unclear how to improve the conductivity or synthesize other conductive polymers. We performed photon echo experiments on PEDOT:PSS, directly interrogating the electronic

states that are speculated to participate in conductivity. Using a sophisticated model extended from the work in Chapter 8, we constrain the pure and ensemble dephasing lifetimes of PEDOT:PSS. These lifetimes can be directly related to the homogeneous and inhomogeneous broadening parameters in PEDOT:PSS. Amazingly, we find that PEDOT:PSS has very broad homogeneous *and* inhomogeneous linewidths. We cannot fully determine either quantity, but we can put lower limits on both. This basic information is complementary to other experiments in the ongoing effort to fully understand PEDOT:PSS.

Despite challenges in software, hardware, and theory MR-CMDS is a crucial tool in the hands of scientists. This dissertation describes several ways to make MR-CMDS more accessible through software and hardware development. PyCMDS has enabled new experiments, and has made data collection faster and more artifact-free. WrightTools has trivialized data processing, tightening the loop between idea and execution. Theory can be used to guide experimental insight in the promising, if challenging, mixed domain. Applications of these ideas in three material systems are presented.

Part I

Background

Chapter 2

Spectroscopy

A hundred years ago, Auguste Comte, . . . a great philosopher, said that humans will never be able to visit the stars, that we will never know what stars are made out of, that that's the one thing that science will never ever understand, because they're so far away. And then, just a few years later, scientists took starlight, ran it through a prism, looked at the rainbow coming from the starlight, and said: "Hydrogen!" Just a few years after this very rational, very reasonable, very scientific prediction was made, that we'll never know what stars are made of.

– Michio Kaku

2.1 Light-matter interaction

As scientists, light is perhaps the most useful tool we have for interrogating materials. Light is relatively easy to create and control, and light-matter interaction tells us a lot about the microscopic physics of the material under investigation. Spectroscopists use light-matter interaction as an analytical tool. For the purposes of this document, light can be treated as a classical electromagnetic wave and matter can be treated in the quantum mechanical density matrix formalism. More complete treatments which also take the quantum-mechanical nature of light into account are possible (see: “quantum optics”, “quantum electrodynamics”), but beyond the scope of this dissertation. This classical treatment still captures the full richness of the wave-nature of light, including interference effects. [31] It merely ignores the quantization of the electric field—a valid assumption in the limit of many photons.

The basics of light matter interaction have been covered in many texts. For a beginners introduction I recommend “How a Photon is Created or Absorbed” by Henderson et al. [32]. Here I present a very minimal overview.

Consider a two state system: “a” and “b”. These two states might be the initial and final states in a transition. The wavefunction for this system can be written as a sum of the stationary states (eigenstates) with appropriate scaling coefficients:

$$\Psi(r, t) = c_a(t)\psi_a(r) + c_b(t)\psi_b(r) \quad (2.1)$$

The time dependence lies in the c_a and c_b coefficients, and the spatial dependence lies in the ψ_a and ψ_b eigenstates.

Now we will expose this two-state system to an electric field:

$$E = E^0 \left[e^{i(kz-\omega t)} + e^{-i(kz-\omega t)} \right] \quad (2.2)$$

For simplicity, we consider a single transition dipole, μ .

The Hamiltonian which controls the coupling of our simple system to the electric field described in

Equation 2.2 can be written.

$$H = H_0 + \mu \cdot E \quad (2.3)$$

$$= H_0 + \mu \cdot \frac{E^0}{2} \left[e^{i(kz-\omega t)} + e^{-i(kz-\omega t)} \right] \quad (2.4)$$

Solving for the time-dependent coefficients, then:

$$c_a(t) = \cos \frac{\Omega t}{2} e^{-i\omega_a t} \quad (2.5)$$

$$c_b(t) = \sin \frac{\Omega t}{2} e^{-i\omega_b t} \quad (2.6)$$

Where ω_a and ω_b are the fast (and familiar) Bohr frequencies and Ω is the *Rabi frequency*:

$$\Omega \equiv \frac{\mu E^0}{\hbar} \quad (2.7)$$

In Dirac notation [33], an observable (such as $\mu(t)$) can be written simply:

$$\mu(t) = |c_a(t)|^2 \langle \phi_a | \mu | \phi_a \rangle + |c_b(t)|^2 \langle \phi_b | \mu | \phi_b \rangle + c_a(t) c_b^*(t) \langle \phi_b | \mu | \phi_a \rangle + c_b(t) c_a^*(t) \langle \phi_a | \mu | \phi_b \rangle \quad (2.8)$$

The complex wavefunction is called a *ket*, represented $|b\rangle$. The complex conjugate is called a *bra*, represented $\langle a|$. The first two terms are populations and the final two terms are coherences. The coherent terms will evolve with the rapid Bohr oscillations, coupling the dipole observable with the time-dependent electric field.

We commonly represent quantum mechanical systems using density matrices, where diagonal elements are populations and off-diagonal elements are coherences. Each density matrix element has the form ρ_{kb} , where k is the ket and b is the bra. A more complete discussion of the formalism we use to describe light-matter interaction is presented in Chapter 8.

Many strategies have been introduced for diagrammatically representing the interaction of multiple electric fields in an experiment. Spectroscopists have used diagrams to represent nonlinear optical phenomena since 1965. [34] Several competing strategies have been defined over the years. In 1978, Yee

and Gustafson [35] defined the “circle diagram” convention. Since then, the more popular “closed-time path-loop” [36, 37] and “double-sided Feynman” diagrams [38] (also known as Mukamel diagrams) were introduced. Biggs, Voll, and Mukamel [39] have written a paper which does an excellent job defining and comparing these two strategies. In their seminal 1985 work, *A Unified View of Raman, Resonance Raman, and Fluorescence Spectroscopy*, Lee and Albrecht [40] defined the conventions for a “wave-mixing energy level” (WMEL) diagram. Today, double-sided Feynman diagrams are probably most popular, but WMELs will be used in this document due to author preference.

WMEL diagrams are drawn using the following rules.

1. The energy ladder is represented with horizontal lines - solid for real states and dashed for virtual states.
2. Individual electric field interactions are represented as vertical arrows. The arrows span the distance between the initial and final state in the energy ladder.
3. The time ordering of the interactions is represented by the ordering of arrows, from left to right.
4. Ket-side interactions are represented with solid arrows.
5. Bra-side interactions are represented with dashed arrows.
6. Output is represented as a solid wavy line.

WMELs can be found throughout this dissertation.

2.2 Types of spectroscopy

Scientists have come up with many ways of exploiting light-matter interaction for measurement purposes. This section discusses several of these strategies. I take a compare-and-contrast approach, rather than getting too caught up in the infinite diversity of possible spectroscopic strategies.

2.2.1 Linear vs multidimensional

Most familiar spectroscopic experiments are linear. That is to say, they have just one frequency axis, and they interrogate just one resonance condition. These are workhorse experiments, like absorbance, reflectance, FTIR, UV-Vis, and common old ordinary Raman spectroscopy (COORS). These experiments are incredibly robust, and are typically performed using easy to use commercial desktop instruments. There are now even handheld Raman spectrometers for use in industrial settings. [41]

Multidimensional spectroscopy contains a lot more information about the material under investigation. In this work, by “multidimensional” I mean higher-order spectroscopy. I ignore “correlation spectroscopy” (otherwise known as covariance spectroscopy), which tracks linear spectral features against non-spectral dimensions like lab time, pressure, and temperature. So, in the context of this dissertation, multidimensional spectroscopy is synonymous with nonlinear spectroscopy.

Nonlinear spectroscopy relies upon higher-order terms in the light-matter interaction. In a generic system, a rule of thumb states that each term is roughly ten times smaller than the last. This means that nonlinear spectroscopy is typically very weak. Still, nonlinear signals are fairly easy to isolate and measure using modern instrumentation, as this dissertation describes.

The most obvious advantage of multidimensional spectroscopy comes directly from the dimensionality itself. Multidimensional spectroscopy can *decongest* spectra with overlapping peaks by isolating peaks in a multidimensional resonance landscape. This decongestion arises directly from the multiple resonances the multidimensional spectroscopy demands of a material.

2.2.2 Frequency vs time domain

Broadly, there are two ways to collect nonlinear spectroscopic signals: frequency and time domain. Both techniques involve exciting a sample with multiple pulses of light and measuring the output signal. The techniques differ in how they resolve the multiple frequency axes of interest.

Frequency domain is probably the more intuitive strategy: frequency axes are resolved directly by iteratively tuning the frequency of excitation pulses against each-other. This relies on pulsed light sources with tunable frequencies.

Time domain experiments use an interferometric technique to resolve frequency axes. Broadband excitation pulses which contain all of the necessary frequencies are used to excite the sample. The delay (phase) between pulses is scanned, and the resonances along that axis are resolved through Fourier transform of the resulting interferogram. In modern experiments, pulse shapers are used to control the delay between pulses in a very precise, fast, and reproducible way. The time domain strategy is by-far the most popular technique in multidimensional spectroscopy because these technologies allow for rapid, robust data collection.

This dissertation focuses on less-popular frequency domain strategies, so some discussion of the advantages of frequency domain when compared to time domain are warranted.

One of the biggest instrumental limitations of multidimensional spectroscopy is bandwidth. It is easy to get absorbance spectra over the entire visible spectrum, and even into the ultraviolet and near infrared. Not so for multidimensional spectroscopy. Multidimensional spectroscopy is limited by the bandwidth of our (tunable) light sources. For frequency domain techniques, this limitation is incidental: sources with greater tunability will be easy to incorporate into these instruments, and creating such sources is only a matter of more optomechanical engineering—more moving parts. Time domain techniques, on the other hand, have a more fundamental issue with bandwidth. Time domain requires that all of the desired frequencies be present within the single excitation pulse, and pulses with very large frequency bandwidth (very short in time) become very hard to use and control. With short, broad pulses:

- Non-resonant signal becomes brighter relative to resonant signal (the resonance advantage is lost).
[42]

→ Pulse distortions become essentially unavoidable. [20]

Time domain experiments require a phase-locked, independently controlled local oscillator in order to collect the interferogram at the heart of such techniques. This local oscillator enhances the information-gathering power of time domain because it allows the experiment to explicitly collect nonlinear spectra with full phase information. At the same time, the local oscillator requirement limits the flexibility of the time-domain because it essentially requires that the output frequency must be the same as one of the inputs. Novel, often fully coherent, experiments cannot be accomplished under this limitation.

2.2.3 Homodyne vs heterodyne

Within frequency domain multidimensional spectroscopy, one is free to use or forgo a local oscillator. That is to say, frequency domain spectroscopy can be collected in a heterodyne or homodyne technique. As discussed in the previous section, use of a local oscillator means that more useful phase information can be extracted from the spectrum. At the same time, generation of a phase locked, controllable local oscillator can be cumbersome, limiting the flexibility of possible experiments.

Note that heterodyne techniques may be self heterodyned (as in transient absorption) or “explicitly” heterodyned with a local oscillator.

Besides the aforementioned phase information, probably the biggest difference between heterodyne and homodyne-detected experiments is their scaling with oscillator number density, N . In all heterodyne spectroscopies, signal goes linearly, as N . If the number of oscillators is doubled, the signal doubles. In all homodyne spectroscopies, signal goes as N^2 . If the number of oscillators is doubled, the signal goes up by four times. This is what we mean when we say that homodyne signals are “intensity level” and heterodyne signals are “amplitude level”.

Recently we have been taking to representing homodyne-detected multidimensional experiments on the “amplitude level” by plotting the square root of the collected signal. Many of the figures in this dissertation are plotted in this way. In my opinion, this strategy makes interpretation of spectra easier. Certainly it eases comparison with other experiments, like absorbance and COORS, which go as N .

One easy-to-miss consequence of homodyne collected experiments is the behavior of signals in delay space. Since signal goes as N^2 , signal decays much faster in homodyne-collected experiments. If signal decays as a single exponential, the extracted decay is twice as fast for homodyne vs heterodyne-detected data. This fact is often forgotten, and papers have been corrected for forgetting this factor of two. [43]

Chapter 3

Software

The following guidelines are to be used in the documentation of all software developed in the Wright group for the IBM 9000 computer. These rules have arisen as a necessary consequence of the group's programming philosophy of writing software in the form of units which can be readily shared among a number of programmers. The approach outlined here should help to avoid some of the confusion otherwise produced by several persons simultaneously developing and modifying shared software.

– Roger Carlson, "Software Development Guidelines" (1988) [12]

3.1 Science needs software

Cutting-edge science increasingly relies on custom software. Scientific software enables scientists to collect, analyze, and model results in ways that would otherwise be wholly impossible.

How does scientific software get made? Who makes it, and what is the quality of that product? Much has been written about these questions. To my knowledge, there are at least 8 case studies and surveys dedicated to how scientists develop and use scientific software. [44, 45, 46, 47, 48, 49, 50, 51] Although they focus on different disciplines, and were published at different times, these articles present a remarkably consistent perspective on what challenges tend to arise when developing software “by and for” scientists.

Scientists do more than just use software: they develop it. In their 2008 survey, Hannay et al. [50] showed just how much of the work of science comes down to software development:

- 84.3% of surveyed scientists state that developing scientific software is important or very important for their own research.
- 91.2% of surveyed scientists state that using scientific software is important or very important for their own research.
- On average, scientists spend approximately 40% of their work time using scientific software.
- On average, scientists spend approximately 30% of their work time developing scientific software.

Prabhu et al. [52] had similar results in their 2011 survey, finding that 35% of research time is spent in programming and developing software. Most of that time (57%) is spent “*finding and fixing errors in their programs*”. The amount of software work done for each scientific project is very heterogeneous, with projects ranging between 5% and 95% software development time. To me, the averages reported by Hannay et al. [50] and Prabhu et al. [52] seem roughly correct for the average Wright Group member.

Despite the importance of software to science and scientists, most scientists are not familiar with basic software engineering concepts. This is in part due to their general lack of formal training in programming and software development. Hannay et al. [50] found that over 90% of scientists learn software development through ‘informal self study’, while Segal [47] mentions that “[*scientists*] do not describe themselves as software developers and have little formal education or training in software development”.

This lack of training is not in-and-of-itself a problem. After all, academic scientists are required to be “do-it-yourself” ers in many contexts for which they receive no formal training: everything from plumbing and electrical engineering to human resources and project management. So why pay particular attention to software development practices and skills?

One reason to pay special attention to software is that software mistakes can have particularly dramatic consequences. As experimentalists in the physical sciences, we are often tempted by the intuition that small mistakes lead to small errors. These intuitions do not typically apply to software—software is “brittle” and small bugs have huge consequences. In his 2015 opinion article “Rampant software errors may undermine scientific results”, David A. W. Soergel attempts to estimate how many errors there might be in scientific software, and how far reaching the consequences might be. Quoting Soergel:

...software is profoundly brittle: “small” bugs commonly have unbounded error propagation. A sign error, a missing semicolon, an off-by-one error in matching up two columns of data, etc. will render the results complete noise. It is rare that a software bug would alter a small proportion of the data by a small amount. More likely, it systematically alters every data point, or occurs in some downstream aggregate step with effectively global consequences. In general, software errors produce outcomes that are inaccurate, not merely imprecise.

On a more positive note, better software development practices may be “low-hanging-fruit” that can greatly improve researcher’s lives without huge amounts of investment. Great software makes science easier, faster, and often of higher quality. And making great software isn’t necessarily harder than the development practices that scientists are following today—indeed sometimes it is easier to follow best practices.

In the United States, funding agencies have recognized the crucial role that software plays in science. The National Science Foundation has a long-running “Software Infrastructure for Sustained Innovation” (SI²) program, which endeavors to take a “*leadership role in providing software as enabling infrastructure for science and engineering research*”. [53] Other funding agencies have similar projects.

3.2 Challenges in scientific software development

Software development “by-and-for” scientists poses unique challenges. In this section, I attempt to summarize the literature about these challenges, with a focus on those that I have found most relevant.

“End-user developers.” [48, 50, 54] Typically the developers of scientific software are not trained software developers. This is perfectly appropriate, because scientific software development typically requires a large amount of domain-specific knowledge that only “end-users” possess. Software development practices may not be valued in a scientific environment. End-users may lack the skill and knowledge required to develop high quality, maintainable software. They may not be aware of best practices in software development. They focus on feature additions and neglect documentation and maintenance.

Shifting goals. [48, 49, 50, 52] Traditional software development paradigms typically demand an upfront articulation of goals and requirements. This allows the developers to carefully design their software, even before a single line of code is written. In her seminal 2005 case study Segal [48] describes a collaboration between a team of researchers and a contracted team of software engineers.

Unlike traditional commercial software developers, but very much like developers in open source projects or startups, scientific programmers usually don't get their requirements from customers, and their requirements are rarely frozen. In fact, scientists often can't know what their programs should do next until the current version has produced some results.

Scientific software is *explorative*, and it needs to be flexible and extendable. Scientific software developers cannot know what will be required before they set out to try. This is probably the most fundamental challenge in such projects, and a big part of why science cannot simply “contract out” a large part of its software development needs. Sometimes, a scientific problem is worked out through the iterative process of developing software to solve it.

Maintenance. [49, 52] Scientific software is famously hard to maintain. Graduate students graduate, and institutional knowledge about the internal workings of software projects is diminished over time. This problem is compounded by the long lifetime of some software, the poorly defined requirements, and lack of documentation and testing. Often times, scientific software ends up being a mess of layer upon layer of incongruent pieces written by generation upon generation of student. Worse, software

is sometimes abandoned or left untouched to become a crucial but arcane component of a scientific research project.

Lack of testing. [55, 52, 54] Testing is a huge part of software development practices, but many researchers do not engage in sufficient testing of their software. Without testing, even small software projects can rapidly “get out of hand”—they can become unsustainable and unmaintainable. Especially for domain-specific computational software, determining the “correct outcome” to test against is often infeasible. Software is not typically peer reviewed, so a lack of software testing is often a weak link in the loop of critical self assessment that science depends upon. On the positive side, testing can be an easy-to-add development practice with huge rewards. Well written tests can be a programmers best friend: helping her to ensure that her code has met all of the given requirements. This allows programmers to optimize without worrying about breaking crucial components of their software.

Struggles with optimization. [52] Sometimes, a scientific application requires performant code. Scientists typically struggle to write such code. They may struggle with parallelization paradigms, or they may not understand what is limiting the speed of their software. They may not have good intuitions about how long certain operations should take, or what patterns could be used to speed up execution. Scientists typically do not use profiling tools which help them see which parts of their program would benefit most from optimization.

3.3 Good-enough practices

In their 2017 perspective, “Good enough practices in scientific computing”, (from which this section gets its name) Wilson et al. [56] describe a set of techniques that, in their words, “*every researcher can and should consider adopting*”. In this section, I attempt to very quickly summarize my personal perspective on what makes good software development good—with citations to literature that supports each idea. These practices are not, generally, *extra work*. In fact, many of them save massive amounts of time and effort in the long *and* short run, when properly applied. [57]

Do not reinvent. [56] Before you sit down and implement a piece of software, stop! First you should try hard to find a library that already has what you need. You’ll often surprise yourself with what you can find. Search the package repository for your language, such as PyPI [58], MATLAB File Exchange [59] or CRAN [60]. Even if there is not a full solution to your problem out there, there is almost certainly a solution to some part of it. Much better to have a dependency than a custom implementation. Make your dependencies explicit, in machine readable ways where possible.

Do not duplicate. [56] If you do need to write some software, make sure that you do not duplicate code within your own work. Instead of writing the same few lines of code again and again with small tweaks, write a function that accepts a set of arguments. If you are doing the same operation in many different contexts, consider defining a library to that operation that can be imported and shared between your different projects. If your software package grows to contain multiple files, make those files modular. As a general rule, once you have two classes you need multiple files.

Choose good data formats. [61, 56] Choose a non-proprietary format if at all possible—remember: you yourself might not have access to the proprietary software in 10 years. Choose plain text if you can. Consider conforming to specifications, such as Tidy Data [62]. If you must, use open binary formats such as HDF5 [63]. Put as much metadata as you can into the file. Any piece of metadata that can automatically be added by the computer is essentially free—you might as well do it. Make sure that it is clear what each piece of data means. For tabular data, use headers. Don’t forget units.

Use version control. [61, 57] Version control systems allow programmers to save a software package such that they can always return to that save point. All of the files in the package are saved together.

These systems also allow programmers to see exactly what has changed between each save point, and since the last save point. This is indispensable when trying to diagnose software problems. In order to use version control as effectively as possible, try to save the package after every change (feature addition, bugfix, etc). Typically version control is coupled with uploading to a remote server, for example using git with GitHub [64], GitLab [65] or git.chem.wisc.edu [66], but version control need not be synonymous with uploading and distribution. Tools like git have a lot of fantastic features beyond simply saving, but those are beyond the scope of these “good enough” recommendations. Also consider defining a version for the software package as a whole. Use semantic versioning (MAJOR.MINOR.PATCH) [67], unless there is a strong reason not to. If the language you are using has a convention for representing the version programmatically, such as a `__version__` attribute in Python, comply with that convention.

Test. [61, 57, 56] As the old saying goes, “if it’s not tested, it’s broken”. If you rely on a piece of functionality in your software, consider writing a test that defines that functionality. In this way, as you make changes you can run your tests to ensure that those changes do not accidentally break important functionality. Testing sounds difficult, but it’s really just about writing simple functions that use your software to do something, and then asking if the result is correct. If you add tests when you add features or fix bugs, you’ll quickly find that you have a lot of tests that do a good job of defining the expected behavior of your software. Software engineers tend to be dogmatic about testing, but don’t worry too much about test coverage unless your project becomes very important. Distribute test datasets, when appropriate. Remember, your tests can serve double duty as simple minimal examples.

Collaborate and share. [61, 56, 68] If you are part of a team, consider sharing software and collaborating to create it. Try using practices like code review and issue tracking, but don’t feel obligated to use them if it doesn’t make sense for your project. When working as part of a team, making incremental changes and using version control become even more important. Earlier we mentioned “do not reinvent”. The other side of that coin is “if you make something, consider sharing it”. Put your software on an open platform, like GitHub [64], and mint a DOI. Cite your software, and ask other people who are using your software to do the same. Choose a license early, and choose permissive and commercially compatible unless you 1. know what you are doing and 2. plan to enforce. Afraid to share because your code needs more polish? If your software is good enough to be used in active scientific research, it’s worth sharing. As Nick Barnes says, “*Publish your computer code: it is good enough*”. [68]

Write human readable code, and document it well. [56] Let the computer do the work, but write the source code to be read by a human. Give classes, functions, attributes and variables meaningful names. Don't be afraid to be verbose, most programming environments have tab completion so long names are not all that hard to type. Try to follow the recommended style for your language, but don't obsess about it.

Avoid premature optimization. [69, 56] Don't get pulled into the trap of trying to make things perfect the first time. Software design is typically a very iterative process, and for good reason. This is particularly true in a scientific context, where goals may evolve during the development process. Write for correctness first, and if it works, consider optimization. If you do need to make your software faster, use profiling tools like cProfile [70] and SnakeViz [71] to empirically determine what operations are taking the longest, rather than trying to guess or use intuition. Only optimize speed-limiting operations, and stop optimizing once the code runs as quickly as needed.

3.4 Object oriented programming

The work in this dissertation makes heavy use of object oriented programming, so some very basic introduction to the concept seems warranted. Object oriented programming (OOP) is a *programming paradigm*. Other popular paradigms are procedural programming and functional programming. Python is a popular programming language which allows for OOP. This section will discuss OOP in the context of a Python implementation.

The basic idea of OOP is defining object types (classes) that are self-contained. These classes define pieces of associated data (attributes) and associated procedures (functions) within themselves. Once the class is defined, instances of that class can be created. Instances, as the name implies, are just specific “concrete occurrences” of a given class. The classic example: `Dog` is a class, `fido`, `spot`, and `duke` are three dogs—three instances of the dog class.

OOP is easier to demonstrate than explain, so let’s have some fun with some working Python examples. First, we will define a class.

```
class Person():

    def __init__(self, name, favorite_food=None, hated_food=None):
        self.name = name
        self.favorite_food = favorite_food
        self.hated_food = hated_food

    def react_to(self, food):
        if food == self.favorite_food:
            return 'yum! my favorite'
        elif food == self.hated_food:
            return 'gross---no thank you'
        else:
            return 'meh'
```

(3.1)

Now I can make some instances of that class, and access their attributes and methods.

```

>>> mary = Person(name='Mary', favorite_food='pizza', hated_food='falafel')
>>> jane = Person(name='Jane', favorite_food='salad')
>>> mary.react_to('falafel')
'gross---no thank you'
>>> jane.react_to('salad')
'yum! my favorite'
>>> mary.favorite_food
'pizza'
>>> jane.react_to(mary.favorite_food)
'meh'

```

(3.2)

We can already begin to see how powerful this approach is. Instances of `Person` contain their own attributes and methods. Instances can be interacted with in complex or simple ways. The attributes `favorite_food` and `hated_food` are fully accessible, but need not be directly dealt with when using the `react_to` method. When using OOP, one can hide complexity while still being able to access everything.

One of the most powerful patterns within OOP is *inheritance*. Inheritance is a special relationship between classes. When a class (the child) is made to inherit from another class (the parent), all of the attributes and methods of the parent come automatically. The child class, then, can benefit from all of the behaviors enabled by its parent while still maintaining its own identity where needed. The inheritance pattern makes it very easy to cleanly define expectations and shared structure throughout a large piece of software without repeating functionality. As an example, let's create a child of or `Person` class, defined in 3.1:

```

class GradStudent(Person):
    def react_to(self, food):
        if food == self.hated_food:
            return 'thanks!'
        else:
            return super().react_to(food)

```

(3.3)

Again, let's make an instance and see how it behaves:

```

>>> joe = GradStudent(name='Joe', favorite_food='pizza', hated_food='falafel')
>>> joe.react_to('falafel')
'thanks!'
>>> joe.react_to('pizza')
'yum! my favorite'

```

(3.4)

`joe` has the same preferences as `mary`, but we were able to *overload* the behavior of `Person` to give `joe` a different reaction when faced with his `hated_food` (the joke being that graduate students will eat anything). The wonderful thing is that all of the other behaviors—the `__init__` method, the reaction to `favorite_food`—were inherited from `Person`. We could even add new functionality to our `Person` class, and that functionality would immediately be available to `GradStudent`. In complex programs with trees of inheritance being able to edit one class to change the behavior of entire sections of the software is a very useful capability. You can even have inheritance between different packages, allowing programmers to customize or extend the behavior of existing tools for their specific needs.

OOP is a deep subject with many patterns and concepts behind it. There are many places to read further. I recommend *The Quarks of Object-Oriented Development*, by Armstrong [72].

3.5 Hierarchical data format

One of the particularly important challenges in CMDS is data storage. CMDS datasets are multi-dimensional, and the particular dimensions are different from experiment to experiment. Historically, the Wright Group has stored data as “flattened” arrays in plain text, where each column corresponds to one of the scannable hardware or one of the sensors in the experiment. The simplicity and portability of these formats is fantastic, but they do not scale well with increasingly large and higher-dimensional data.

Hierarchical data files are an alternative strategy that scales much better with large and high-dimensional data. These are binary files that store the arrays directly, not in a flattened way. They can contain multiple arrays, with different data types, in the same file under a well-defined organizational system. They support arbitrary metadata, integrated into the same hierarchy as the arrays, so making them self-describing is trivial. While in general plain text is preferred for its simplicity, these file-types are simply superior for storing CMDS data.

To this author's best knowledge, the Common Data Format (CDF) was the first general purpose self-describing multidimensional array data format. [73] The engineers at the National Space Science Data Center (a division of NASA) created the CDF. Using this construct, *“scientific softwares at NSSDC ... do not need specific knowledge of the data with which they are working. This permits users of such systems to apply the same functions to different sets of data.”* These are exactly the capabilities that CMDS requires.

A second-order challenge in CMDS data storage is the size of the arrays. While by no-means “big data”, CMDS data is often awkwardly large: large enough to fill up the memory of an average modern laptop or desktop computer. CDF also has a unique solution to this problem: use a block structure to allow access to parts of the array without reading the entire data into memory.

Slightly later, NetCDF was introduced [74]. Very similar to CDF, NetCDF focused on enhancements to portability. Certain metadata conventions were also introduced, including named dimensions. NetCDF remains popular in the aerospace and geoscience communities.

The Flexible Image Transform System (FITS) is a similar format with a focus on visualization and backwards compatibility. [75, 76] FITS is still popular in the astronomy community.

Today, these hierarchical data formats have gathered under the umbrella of the HDF5 format, built and maintained by the HDF Group. [63] This format has all of the advantages of FITS, CDF, and NetCDF. It can support arbitrary datatypes and is optimized to quickly process large and complex datasets. In Python, HDF5 is supported primarily through the h5py package. [77]

Many scientific disciplines have built custom file formats on top of the HDF5 standard. These include biological imaging [78], scanning transmission X-ray microscopy [79], scalable nucleotide tallies [80], and even ultrasonic concrete data [81].

3.6 Scientific Python

SciPy is a collection of *“open-source software for mathematics, science, and engineering.”* [82, 83] SciPy was an absolutely essential component of this dissertation and the work it describes. There are many packages under the SciPy umbrella. NumPy is a very powerful and fast package for working with multidimensional arrays. [84] The SciPy library contains a vast number of scientific computing tools, including many mathematical operations that this work depends on. [85] Matplotlib is a beautiful visualization package for 1, 2, and 3D plotting. [86]

Part II

Development

Chapter 4

Processing

What we have is data glut. What we really want is the ability to manipulate the information and to reach conclusions from it. I think we are at the point where that is slipping beyond unaided humans' abilities. So the real thing to be looking for is processing schemes. One way is automatic processing: for instance, the sort of analysis that we saw with the IBM Watson on Jeopardy. Putting that in service to humankind in fields that are suffering from data glut at least gives people who are in charge the ability to keep some sort of track of what is going on.

The other great thing that we have going for us is that we have billions of very intelligent people out there in the world. With the networking that we have now, we're beginning to see that those large populations, coordinating amongst themselves, are an intellectual resource that trumps all institutional intellectual resources and has a real possibility, if it's supported by the proper automation, of creating solutions to problems, including the problem of the data glut.

– Vernor Vinge [87]

CMDS takes a somewhat unique approach to instrumental science. How, then could one go about making a data processing software package for CMDS? The package has to be flexible enough to accommodate the diversity of experiments, but still solid enough to be a foundational tool.

When creating a toolkit for CMDS, there are several challenges worth considering:

- Dimensionality of datasets can typically be greater than two, complicating representation.
- Shape and dimensionality change, and relevant axes can be different from the scanned dimensions.
- Data can be awkwardly large-ish (several million pixels), to legitimately large—it is not always possible to store entire arrays in memory.
- There are no agreed-upon file formats for CMDS dataset storage.

The biggest challenge is to find a really good definition for what constitutes a CMDS dataset. Once understood, this common denominator can be enshrined into software and built upon. This chapter describes WrightTools, a software package that I created to process CMDS datasets.

WrightTools is a software package written in Python, built using the excellent tools provided by the scientific Python collection of packages, especially Scipy [85] and Numpy [84]. WrightTools defines a universal file-format that is flexible enough to encompass the diversity of CMDS while still being entirely self-describing. This file format is based on the popular binary format “HDF5” [63], as interfaced by the h5py python library [77]. This format allows WrightTools to interact with the arrays piece-by-piece in a very fast and reliable way, without loading the entire array in and out of memory. This allows users to interact with legitimately large CMDS datasets without worrying about memory overflow. WrightTools takes a unique approach to representing CMDS data in array format, what I call “semi-structure”, that allows for greater flexibility in representing CMDS in different coordinate spaces.

WrightTools is written to be used in scripts and in the command line. It does not have any graphical components built in, except for the ability to generate plots using matplotlib [86]. Being built in this way gives WrightTools users maximum flexibility, and allows for rapid collaborative development. It also allows other software packages to use WrightTools as a “back-end” foundational software, as has already been done in simulation and acquisition software created in the Wright Group.

4.1 Introduction to WrightTools

WrightTools is a moderately complex piece of software (~10,000 source lines of code), so it is important to keep the package internally organized so that users are able to use the pieces they need without feeling overwhelmed by the full complexity. For organizational purposes, WrightTools is designed in a nested, hierarchical manner through heavy use of object oriented programming (see Section 3.4). In this introductory section, I wish to describe the overall structure of WrightTools, without going into too much detail. In this way the reader can have some context in the sections below, where I describe some crucial pieces of WrightTools in greater detail.

WrightTools is written in Python, and endeavors to have a “pythonic”, explicit and “natural” application programming interface (API). To use WrightTools, simply import:

```
>>> import WrightTools as wt
>>> wt.__version__
3.0.0
```

(4.1)

I’ll discuss more about how exactly WrightTools packaging, distribution, and installation works in Section 4.10.4.

Table 4.1 contains a description of each top-level component within the WrightTools package. Within an interactive python session, we could see these components using the built in `dir` function: `dir(wt)`. There are several types of component: functions, attributes, classes, modules, and subclasses. Functions are simple objects that take some input(s), do something, and return something. For example, the function `wt.open`, which accepts a path to a WrightTools dataset file and returns an opened version of that file. Attributes are not interactive, they are simply pieces of attached information that can be accessed. For example, `wt.__version__` as used in the code fragment above. Classes are instructions for construction of particular custom object types, and can be instantiated (see Section 3.4). We’ll talk extensively about the five main WrightTools classes: `Axis`, `Collection`, `Channel`, `Data`, and `Variable`, in the coming pages. Modules are literally `.py` files within WrightTools, and they themselves contain attributes, functions, and classes. Finally, subpackages are literally folders that contain several `.py` files: several modules.

All spectra are stored within wt5 files as multidimensional arrays. Arrays are containers that store many instances of the same data type, typically numerical datatypes. These arrays have some `shape`, `size`, and `dtype`. In the context of WrightTools, they can contain floats, integers, complex numbers and NaNs.

WrightTools is designed around a universal “wt5” file format. wt5 files are simply extensions of the hdf5 format, with some additional requirements applied to their internal structure. This puts wt5 files in the same category as other domain-specific heirarchical data formats (see Section 3.5). One of the most important features of the HDF5 paradigm is the ability to access portions of the multidimensional arrays at a time. WrightTools takes full advantage of this, such that the WrightTools package is simply an *interface* to the data contained with the wt5 file, and arrays are not stored in memory until needed.

There are two classes which are top-level components of the WrightTools package: `Collection` and `Data`.

`Data` is arguably the most important class, as it provides the crucial function of interfacing to the stored multidimensional arrays that constitute the CMDS datasets. `Data` can be instantiated directly, but typically is instantiated by helper functions within the `data` subpackage, or by the `open` function. See Section 4.2 for more information.

`Collection` is a container class, charged with storing groups of data objects and other collection objects—empowering users to organize their datasets into clearly structured and well labeled hierarchies within the wt5 file. See Section 4.3 for more information about `Collection`.

The `artists` subpackage contains all of the tools needed to plot `Data` objects. There are “quick” artist functions made primarily for use in interactive plotting, and a larger, more flexible set of classes and functions that can be used to construct more elaborate figures. See Section 4.4 for more information.

The `fit` subpackage is an interface which endeavors to make fitting multidimensional `Data` objects as easy as possible. Towards this end, the `fit` subpackage takes a unique approach of dimensionality reduction via fitting. See Section 4.9 for more information.

The `datasets` subpackage is simply a python interface to the set of raw data that is distributed within

WrightTools. `datasets` is not imported by default, so “from” syntax must be used. `datasets` allows users to access full filepaths to the raw data, rather than returning instances of `Data` or `Collection`.

```
>>> from WrightTools import datasets
>>> datasets.COLORS.v0p2_d1_d2_diagonal
'../WrightTools/datasets/COLORS/v0.2/d1_d2_diagonal.dat'
```

(4.2)

This strategy is more flexible and allows the developers of WrightTools to write tests and examples using datasets that are guaranteed to be on every machine.

The `diagrams` subpackage is a small set of tools used for drawing diagrams, with a focus on diagrams commonly required by CMDS practitioners. Currently `diagrams` can draw WMELs [40] and delay space labels (see Figure 12.4 for an example). `diagrams` interfaces well with artists since they both are built on top of matplotlib, so it is easy for WrightTools users to draw diagrams in the same figure as other elements.

The `units` module handles all unit information, and conversion between values in different unit systems.

The `exceptions` module defines the unique exceptions and warnings that WrightTools raises. All exceptions are children of the `WrightToolsException` class, and all warnings are children of the `WrightToolsWarning` class. In this way, users of WrightTools can easily intercept all exceptions/warnings coming from WrightTools itself (as opposed to packages that WrightTools relies upon) when debugging their application.

Finally, the `kit` subpackage is a small menagerie of classes and functions that are useful, but have no other place within WrightTools. Many of these are used internally throughout the rest of the program, and others are distributed to be used by WrightTools users. As examples:

- The `TimeStamp` class represents a moment in time, and handles conversion between different popular representations of time.
- The `INI` class is a very simple python interface to `.ini` configuration files.
- The `fft` function is a friendly user interface for N-dimensional fft operations.
- The `closest_pair` function finds the pair(s) of indices corresponding to the closest elements in an array.

	type	description
<code>artists</code>	subpackage	Plotting and figure generation.
<code>Collection</code>	class	Nestable container for data objects.
<code>collection</code>	subpackage	Collection creation functions.
<code>Data</code>	class	Central multidimensional dataset class.
<code>data</code>	subpackage	Data creation functions.
<code>datasets</code>	subpackage	Raw data for examples, testing.
<code>diagrams</code>	subpackage	Draw diagrams.
<code>exceptions</code>	module	WrightTools exception and warning types.
<code>fit</code>	subpackage	Fit data objects.
<code>kit</code>	subpackage	Small useful functions and classes.
<code>open</code>	module	Universal wt5 file open function.
<code>units</code>	module	Unit handling and conversion.

Table 4.1: Key components of WrightTools, lexicographically listed.

I now focus on the `Data` class. Table 4.2 contains a description of each key component of `Data`.

`Data` can be thought of as a container class that contains everything needed to define a single multidimensional spectra. To do this, each data object contains several multidimensional arrays (typically 2 to 50 arrays, depending on the kind of data). There are two kinds of arrays, instances of `Variable` and `Channel`. Variables are coordinate arrays that define the position of each pixel in the multidimensional spectrum, and channels are each a particular kind of signal within that spectrum. Typical variables might be `[labtime, w1, w2, w3, d1, d2]`, and typical channels `[pmt, pyro1, pyro2, pyro3]`. The data object contains attributes `Data.variables` and `Data.channels` which are tuples of the instances of `Variable`, `Channel` contained within that instance of `Data`. The data object also has convenience attributes `variable_names` and `channel_names`; creation methods `create_channel` and `create_variable`; and basic manipulation methods `remove_channel`, `remove_variable`, and `rename_channels`. More information about channels and variables will come on the next pages.

Variables contain all of the information about where every piece of hardware was at each coordinate in the multidimensional dataset, but most of the time users only want to work with data as parameterized by a few key variables. Crucially, the exact choice of parameterization may be context dependent, or multiple parameterizations may be desirable. [88] Axes, instances of the WrightTools `Axis` class, are easy to use parameterized interfaces to the variable arrays. Axes do not contain any *new* information, they simply contain expressions which describe how the variable arrays are accessed when manipulating or displaying the data. The `transform` method allows users to change these expressions. Convenience attribute `axis_expressions` allows for quick inspection. See section ... for more information.

Besides merely allowing users to access variables and channels, the `Data` class allows for manipulation and processing. Many simple data processing tools are methods of `Channel` and `Variable`, and are discussed further later. The data manipulation methods that `Data` contains are more holistic—they are manipulations that involve multiple variable and channel arrays. `heal` attempts to “fill” holes via multidimensional interpolation. `chop`, `collapse`, `pythonsplit`, `map_axis`, and `zoom` change the shape of the data object, by slicing, interpolation, or both.

	type	description
<code>collapse</code>	method	Collapse along one dimension in a well-defined way.
<code>convert</code>	method	Convert all axes of a certain kind.
<code>create_channel</code>	method	Create a new channel.
<code>create_variable</code>	method	Create a new variable.
<code>fullpath</code>	attribute	External and internal path to data.
<code>get_nadir</code>	method	Get the coordinates, in units, of the minimum of a channel.
<code>get_zenith</code>	method	Get the coordinates, in units, of the maximum of a channel
<code>heal</code>	method	Remove nans from channel using interpolation.
<code>kind</code>	attribute	Instrumental origin of data.
<code>level</code>	method	Subtract the average value of npts at the edge of a given axis.
<code>map_variable</code>	method	Map points of a variable to new points using linear interpolation.
<code>natural_name</code>	attribute	Natural name.
<code>ndim</code>	attribute	Number of dimensions.
<code>offset</code>	method	Offset one variable based on another variables values.
<code>print_tree</code>	method	Print a pretty tree including all contents.
<code>remove_channel</code>	method	Remove a channel.
<code>remove_variable</code>	method	Remove a variable.
<code>rename_channels</code>	method	Rename (multiple) channel(s).
<code>shape</code>	attribute	Shape of data.
<code>share_nans</code>	method	Share not-a-numbers between all channels.
<code>size</code>	attribute	Number of pixels in entire data shape.
<code>smooth</code>	method	Smooth a channel using an n-dimensional Kaiser window.
<code>source</code>	attribute	File of origin.
<code>split</code>	method	Split the data along a given axis, in units.
<code>transform</code>	method	Transform the data.
<code>units</code>	attribute	Tuple of units for each axis.
<code>variable_names</code>	attribute	Variable names.
<code>variables</code>	attribute	Variable objects.
<code>zoom</code>	method	Zoom the data using spline interpolation of the requested order.

Table 4.2: Key attributes and methods of data, lexicographically listed.

I now focus on the `Channel` and `Variable` classes. These are the principle multidimensional array containers, and each instance of these classes corresponds to exactly one multidimensional array. These two classes share a large amount of functionality, and they both inherit from the parent `WrightTools Dataset` class, which itself is a child of `h5py.Dataset`. See Section 3.4 to understand the concept of inheritance.

Table 4.3 contains a description of each key component of the `Channel` and `Variable` classes. For each component the column “of” indicates if it is a shared feature (inherited from `Dataset`), or unique to one or the other class. Many of these are attributes which describe the contents or behavior of these arrays. `argmax`, `argmin`, `max`, and `min` are methods that make it easy to inspect the most basic features of the array. The concept of `null` as different from zero is unique to channels, and the components `signed`, `mag`, `major_extent`, and `minor_extent` come in association with the null idea.

These classes also have basic mathematical manipulation methods, such as `log`, `normalize`, and `symmetric_sqrt`. Other operations are supported by in-place operations, as described in Section 4.7.1.

Channels and variables inherit from `h5py`, so they support partial access through slicing (`__getitem__` syntax). This means that, in principle, very large datasets can be processed piece-by-piece without loading the entire array into memory simultaneously. This is trivial for “blind” operations like taking a logarithm or normalizing, and becomes more complex for operations like smoothing and interpolation. `WrightTools` offers several methods that try to make it easier to process arrays piecewise. `slices` returns a generator which yields tuples of slice objects for each chunk of the array. `chunkwise` accepts a function and executes it on each chunk returned by `slices`.

	type	of	description
argmax	method	both	Index of maximum, ignoring nans.
argmin	method	both	Index of minimum, ignoring nans.
chunkwise	method	both	Execute a function for each chunk in the dataset.
clip	method	both	Clip values outside of a desired range.
convert	method	both	Convert units, writing to disk.
full	attribute	both	Content array expanded to full shape.
fullpath	attribute	both	External and internal path to dataset.
label	attribute	variable	Label to appear in subscript.
log	method	both	Take the log of the entire dataset, with choice of base.
log10	method	both	Take the base 10 log of the entire dataset.
log2	method	both	Take the base 2 log of the entire dataset.
mag	method	channel	Alias for major_extent.
major_extent	attribute	channel	Maximum deviation from null.
max	method	both	Maximum, ignoring nans.
min	method	both	Minimum, ignoring nans.
minor_extent	attribute	channel	Minimum deviation from null.
natural_name	attribute	both	Natural name.
normalize	method	channel	Zero out znull, divide such that mag is 1.
null	attribute	channel	Null value.
parent	attribute	both	Parent data object.
points	attribute	both	Squeezed content array.
signed	attribute	channel	Flag to indicate data is signed.
slices	method	both	Returns a generator yielding tuple of slice objects.
symmetric_root	method	both	Take the root, propagating sign.
trim	method	channel	Remove outliers using statistical tests.

Table 4.3: Key attributes and methods of channel and variable, lexicographically listed

I now focus on the `Axis` class. Table 4.4 contains a description of each key component of the `Axis` class.

Axes are ways to organize data as functional of particular variables (and combinations thereof). The `Axis` class does not directly contain the respective arrays—it merely refers to the associated variables. The flexibility of this association is one of the main new features in WrightTools 3.

Axis expressions are simple human-friendly strings made up of numbers and variable `natural_name`s. Given 5 variables with names `['w1', 'w2', 'wm', 'd1', 'd2']`, example valid expressions include `'w1'`, `'w1=wm'`, `'w1+w2'`, `'2*w1'`, `'d1-d2'`, and `'wm-w1+w2'`. Axes can be directly indexed / sliced into using `__getitem__`, and they support many of the “numpy-like” attributes.

Axes need not be one-dimensional. In fact, axes must have the same dimensionality as their parent `Data`. The loosening of the one-dimensional axis requirement is what makes WrightTools data not fully structured, but “semi-structured”.

Section 4.6 describes the `Axis` class in greater detail.

	type	description
<code>full</code>	attribute	Content array expanded to full shape.
<code>label</code>	attribute	Label to appear in subscript.
<code>natural_name</code>	attribute	Natural name.
<code>ndim</code>	attribute	Number of dimensions.
<code>points</code>	attribute	Squeezed content array.
<code>shape</code>	attribute	Shape of axis.
<code>size</code>	attribute	Number of pixels in axis.
<code>units</code>	attribute	Units of axis.
<code>variables</code>	attribute	List of variables owned by axis.
<code>convert</code>	method	Convert the axis to a different set of units.
<code>min</code>	method	Get the axis minimum.
<code>max</code>	method	Get the axis maximum.

Table 4.4: Key attributes and methods of axis, lexicographically listed

4.2 Creating a data object

WrightTools data objects are capable of storing arbitrary multidimensional spectra, but how can we actually get data into WrightTools? If you start with a wt5 file, the answer is easy: `wt.open(<filepath>)`. But what if you have data that was written using some other software? WrightTools offers data conversion functions (“from” functions) that do the hard work of creating data objects from other files. These from-functions are as parameter free as possible, which means they recognize details like shape and units from each specific file format without manual user intervention.

The most important thing about from-functions is that they are extensible: that is, that more from-functions can be easily added as needed. This modular approach to data creation means that individuals who want to use WrightTools for new data sources can simply add one function to unlock the capabilities of the entire package as applied to their data.

Table 4.5 contains the currently supported from functions in WrightTools.

Discover dimensions

Certain older Wright Group file types (COLORS and KENT) are particularly difficult to import using a parameter-free from-function. There are two problems:

- Dimensionality limitation to individual files (1D for KENT, 2D for COLORS).
- Lack of self-describing metadata (headers).

The way that WrightTools handles data creation for these file-types deserves special discussion.

Firstly, WrightTools contains hardcoded column information for each filetype. Data from Kent Meyer’s “picosecond control” software had consistent columns over the lifetime of the software, so only one dictionary is needed to store these correspondences. Schuyler Kain’s “COLORS” software used at least 7 different formats, and unfortunately these format types were not fully documented. [89] WrightTools attempts to guess the COLORS data format by counting the number of columns.

Because these file-types are dimensionality limited, there are many acquisitions that span over multiple

files. COLORS offered an explicit queue manager which allowed users to repeat the same 2D scan (often a Wigner scan) many times at different coordinates in non-scanned dimensions. ps_control scans were done more manually. To account for this problem of multiple files spanning a single acquisition, the functions `from_COLORS` and `from_KENT` optionally accept *lists* of filepaths. Inside the function, WrightTools simply appends the arrays from all given files into one long array with many more rows.

The final and most challenging problem of parameter-free importing for these filetypes is *dimensionality recognition*. Because the files contain no metadata, the shape and coordinates of the original acquisition must be guessed by simply inspecting the columnar arrays. In general, this problem can become very hard. Luckily, each of these previous instrumental software packages was only used on one instrument with limited flexibility in acquisition type, so it is possible to make educated guesses for almost all acquisitions.

The function `wt.kit.discover_dimensions` handles the work of dimensionality recognition for both COLORS and ps_control arrays. This function may be used for more filetypes in the future. Roughly, the function does the following:

1. Remove dimensions containing nan(s).
2. Find which dimensions are equal (within tolerance), condense into single dimensions.
3. Find which dimensions are scanned (move beyond tolerance).
4. For each scanned dimension, find how many unique (outside of tolerance) points were taken.
5. Linearize each scanned dimension between smallest and largest unique point.
6. Return scanned dimension names, column indices and points.

The `from_COLORS` and `from_KENT` functions then linearly interpolate each row in the channels onto the grid defined by `discover_dimensions`. This interpolation uses `scipy.interpolate.griddata`, which in turn relies upon the C++ library Qhull.

This strategy can be copied in the future if other non-self-describing data sources are added into WrightTools.

function	data source
<code>wt.collection.from_CARY</code>	Cary brand spectrometers.
<code>wt.data.from_COLORS</code>	COLORS. [89]
<code>wt.data.from_JASCO</code>	JASCO brand spectrometers
<code>wt.data.from_KENT</code>	ps_control. [90]
<code>wt.data.from_PyCMDS</code>	PyCMDS (Chapter 5).
<code>wt.data.from_OceanOptics</code>	Ocean Optics brand spectrometers.
<code>wt.data.from_shimadzu</code>	Shimadzu brand spectrometers.
<code>wt.data.from_Tensor27</code>	Tensor 27 FT-IR.

Table 4.5: WrightTools data import functions.

4.3 Collections

The WrightTools `Collection` class is a container class meant to organize the contents of the wt5 file. It can contain other collection instances and data objects. Conceptually, it behaves like a folder in a traditional file-system. `wt.Collection` is a child of `h5py.Group` [91].

The primary attributes and methods of `Collection` are

- attribute `item_names`
- attribute `fullpath`

Collections are useful because they allow WrightTools users to “carry around” several associated data objects in the same file. For example, a publication might contain several experiments on the same sample. Collections allow such experiments to be organized in a hierarchical way. The hierarchy of contents that a collection contains can be easily visualized using the `print_tree` method. As an example, consider the following collection instance which contains some experiments accomplished on neat carbon tetrachloride.

```
>>> import WrightTools as wt
>>> root = wt.open('CC14.wt5')
>>> root.print_tree()
CC14 (/tmp/0tze7b8a.wt5)
├── 0: delay (111,)
│   ├── axes: d1 (fs)
│   └── channels: ai0, ai1, ai2, ai3
└── 1: frequency
    ├── 0: delay_0 (51, 51)
    │   ├── axes: w2 (eV), w1=wm (eV)
    │   └── channels: ai0, ai1, ai2, ai3, ai4, mc
    └── 1: delay_200 (18, 20)
        ├── axes: w1=wm (eV), w2 (eV)
        └── channels: ai0, ai1, ai2, ai3
```

(4.3)

Looking at the output of `print_tree`, we can see that this collection (named `CC14`) contains the following:

1. A data object “`delay`”, shape `(111,)`.
2. A collection object “`frequency`”, containing two 2D data objects.
 - (a) A data object “`delay_0`”, shape `(51, 51)`.
 - (b) A data object “`delay_200`”, shape `18, 20`.

Since this is all contained in one file, a user of WrightTools can easily manage all three associated datasets. Upon simple inspection it is obvious that two of the datasets are 2D frequency-frequency scans while one is a 1D delay slice.

Like `Channel`, `Data` and `Variable`, `Collection` supports adding arbitrary metadata through the `attrs` dictionary.

4.3.1 From directory

The `wt.collection.from_directory` function can be used to automatically import all of the data sources in an entire directory tree. It returns a WrightTools collection with the same internal structure of the directory tree, but with WrightTools data objects in the place of raw data source files. Users can configure which files are routed to which from-function.

4.4 Visualizing a data object

After importing and manipulating data, one typically wants to create a plot. The artists sub-package contains everything users need to plot their data objects. This includes both “quick” artists, which generate simple plots as quickly as possible, and a full figure layout toolkit that allows users to generate full publication quality figures. It also includes “specialty” artists which are made to perform certain common plotting operations.

Currently the artists sub-package is built on-top of the wonderful matplotlib library [86]. In the future, other libraries (e.g. Mayavi [92]), may be incorporated.

4.4.1 Strategies for 2D visualization

Representing two-dimensional data is an important capability for WrightTools, so some special discussion about how such representations work is warranted. WrightTools data is typically very structured, with values recorded at a grid of positions. To represent two-dimensional data, then, WrightTools needs to map the values onto a color axis. When doing this mapping, choosing an appropriate colormap and interpolation type is crucial.

Colormap

For the purposes of this discussion, there are two types of colormaps: 1. qualitative 2. perceptual. Qualitative colormaps have random orderings of color. They are best used to represent unordered things, and they typically have high dynamic range. Perceptual colormaps are monotonic in lightness, and are best at representing ordered information (like signal levels in MR-CMDS). [93]

Historically the Wright Group has used a qualitative colormap for all plotting. Figure 4.1 shows the red, green, and blue components of four different colormaps. The black line is the net lightness of each color (larger value means lighter color). Below each figure is a gray-scale representation of the corresponding colormap. The r, g, and b components are scaled according to human perception. The traditional Wright Group colormap (derived from jet) is shown first. Following are two perceptual colormaps,

cubehelix from GreenDA2009a, and viridis, the new matplotlib default. WrightTools uses the algorithm from Green [94] to define a custom cubehelix colormap with good perceptual properties and familiar Wright Group coloration. The lightness is not perfectly straight as in viridis and cubehelix, but it is monotonic at least.

Figure 4.2 compares the new (top row) and old (bottom row) colormaps in full color and greyscale. While the old colormap shows a greater dynamic range, the new colormap is more perceptually uniform. In my opinion, users should use logarithmic scaling of an appropriate base if more dynamic range is required, rather than using a non-perceptual colormap.

Interpolation type

WrightTools data is defined at discrete points, but an entire 2D surface must be defined in order to make a full colored surface. Defining this surface requires *interpolation*, and there are various strategies that have different advantages and disadvantages. Choosing the wrong type of interpolation can be misleading.

Figure 4.3 represents different kinds of plot-type interpolation. Each is labeled according to the corresponding matplotlib method call. In the multidimensional spectroscopy community, the most popular form of interpolation is “contourf” and “contour”, both based on Delaunay triangulation. In Figure 4.3 the edges of the Delaunay triangles are drawn for clarity. Such interpolation methods result in *smoother* looking spectra, but they can look strange and cause visual artifacts. “pcolor” is a much more direct approach that results in *blocky* but honest two-dimensional plots.

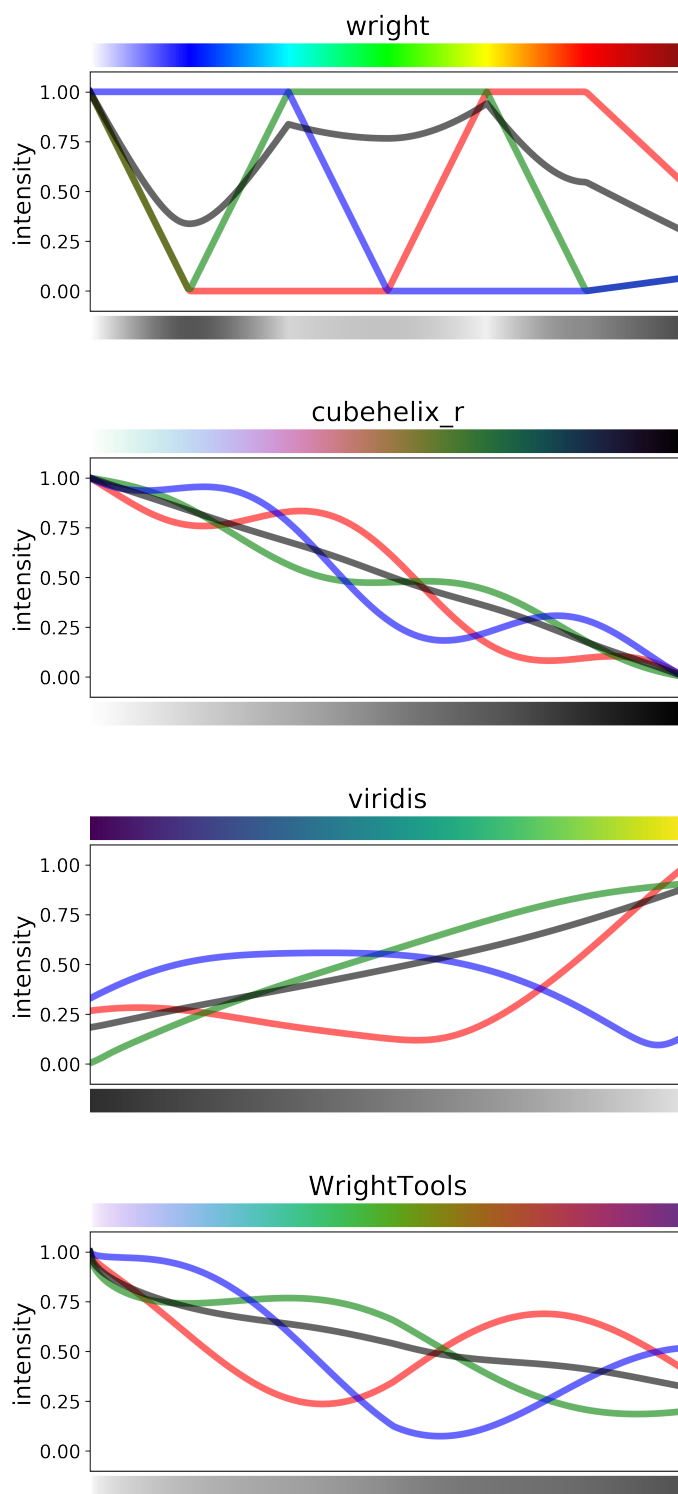


Figure 4.1: Comparison of four different colormaps. R, G, and B components are each shown. Black line is human-perceived lightness.

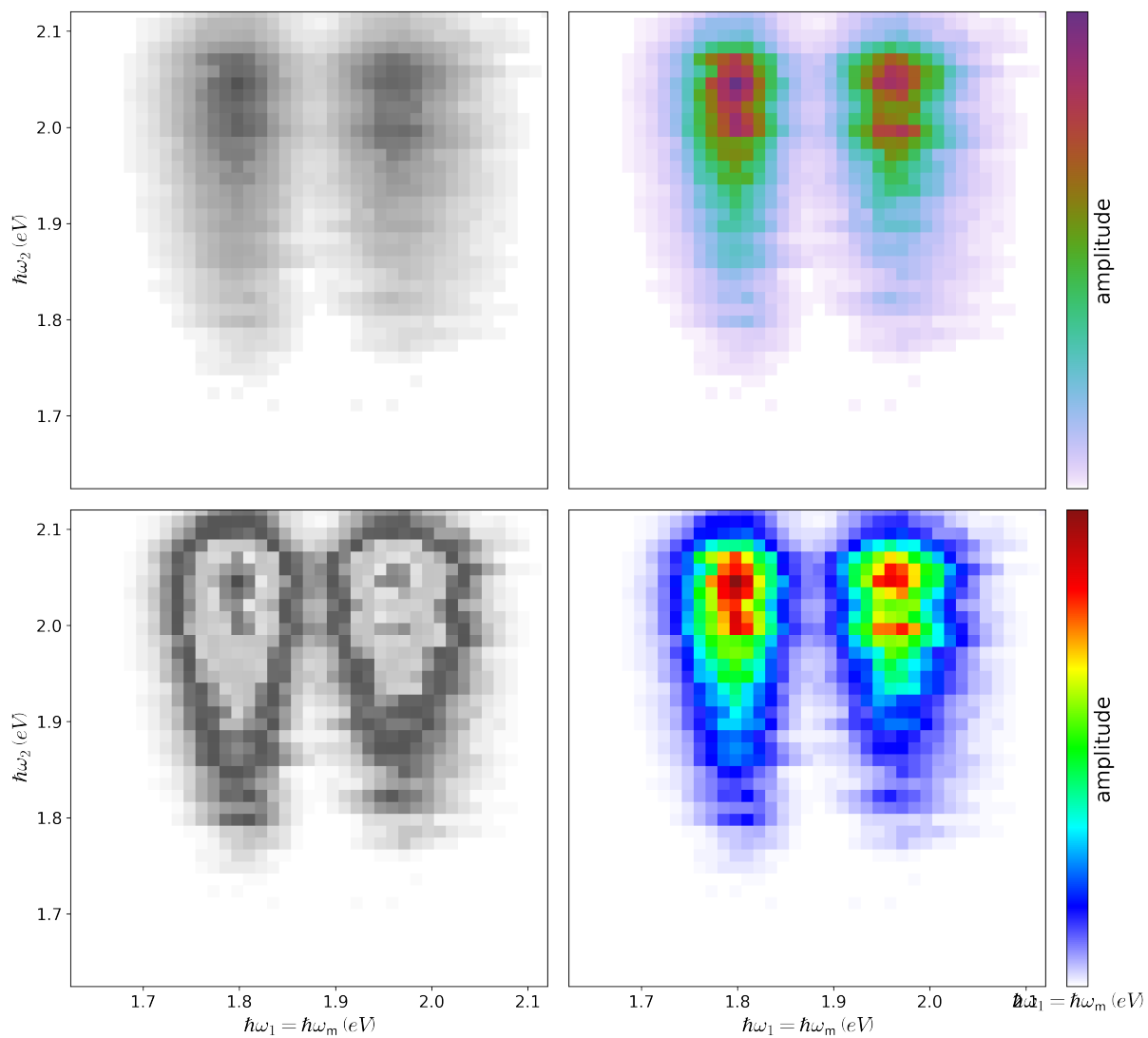


Figure 4.2: Comparison between colormaps in plotting two-dimensional data. Right column is full-color, and left column is lightness. The “new” (top) colormap is evenly spaced in terms of lightness, while the “old” colormap has dramatic, misleading changes at the transition in and out of green.

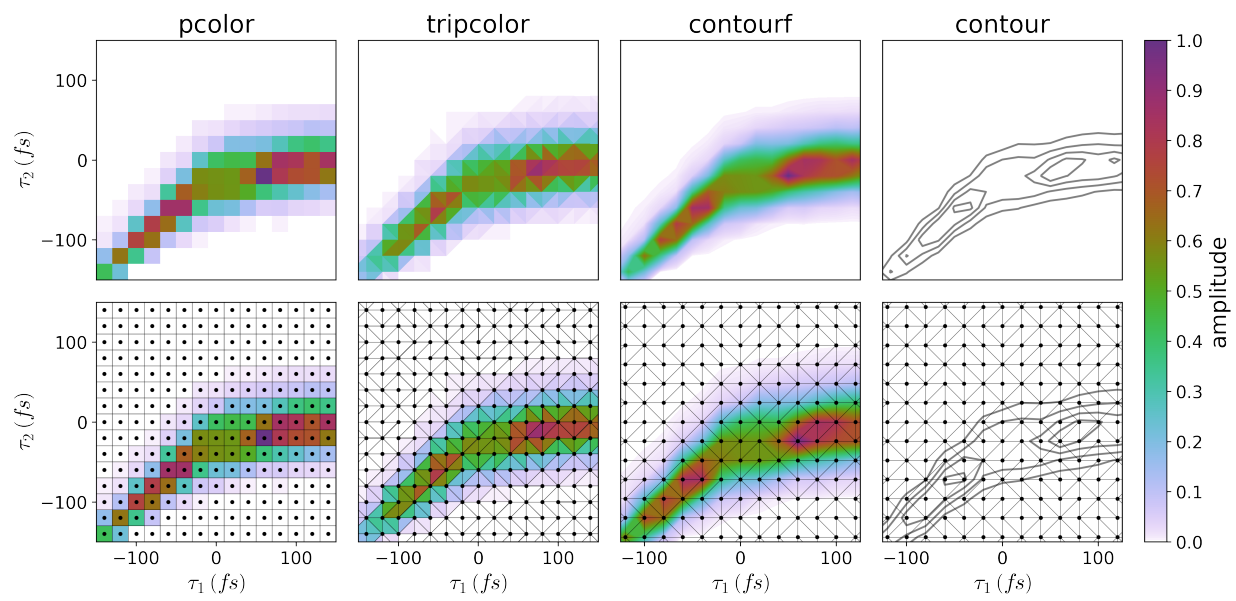


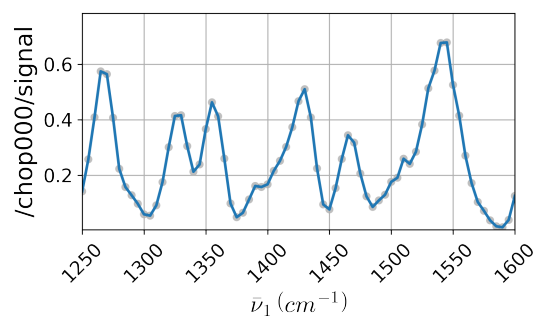
Figure 4.3: Interpolation types in 2D plotting.

4.4.2 Quick

To facilitate easy visualization of data, WrightTools offers “quick” artist functions which quickly generate 1D or 2D representations. These functions are made to make good representations by default, but they do have certain keyword arguments to make popular customization easy. These are particular useful functions within the context of repls and auto-generated plots in acquisition software.

Default outputs of `wt.artists.quick1D` and `wt.artists.quick2D` are shown in Figure 4.4 and Figure 4.5, respectively. The full script used to create each image is included in the Figures. Note that the actual quick functions are each one-liners, and that the supplied keyword arguments are necessary only because the images are being saved (not typical for users in interactive mode).

Perhaps the most powerful feature of `quick1D` and `quick2D` are their ability to treat higher-dimensional datasets by automatically generating multiple figures. When handing a dataset of higher dimensionality to these artists, the user may choose which axes will be plotted against using keyword arguments. Any axis not plotted against will be iterated over such that an image will be generated at each coordinate in that axis. Users may also provide a dictionary with entries of the form `{axis_name: [position, units]}` to choose a single coordinate along non-plotted axes. These functionalities are derived from `wt.Data.chop`, discussed further in Section 4.8.1.



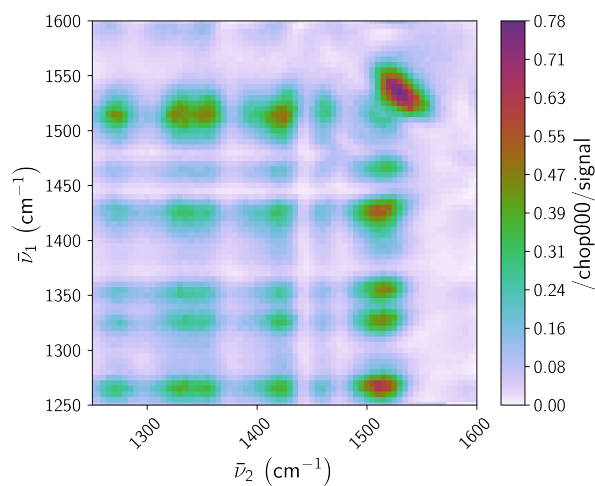
```
import os
import WrightTools as wt
from WrightTools import datasets

here = os.path.abspath(os.path.dirname(__file__))

ps = datasets.KENT.LDS821_TRSF
data = wt.data.from_KENT(ps, ignore=['d1', 'd2', 'wm'], verbose=False)

wt.artists.quick1D(data, 'w1', at={'w2': [1520, 'wn']}), autosave=True,
    save_directory=here, fname='quick1D')
```

Figure 4.4: Minimal example of the `quick1D` function.



```

import os
import WrightTools as wt
from WrightTools import datasets

here = os.path.abspath(os.path.dirname(__file__))

ps = datasets.KENT.LDS821_TRSF
data = wt.data.from_KENT(ps, ignore=['d1', 'd2', 'wm'], verbose=False)

wt.artists.apply_rcparams('publication')
wt.artists.quick2D(data, autosave=True, save_directory=here, fname='quick2D')

```

Figure 4.5: Minimal example of the `quick2D` function.

4.5 Variables and channels

Data objects are made up of many component channels and variables, each array having the same dimensionality of its parent data. This strategy allows for maximal flexibility in data representation, but it can be overly expensive if certain arrays do not actually change against all of the dimensions. This is often especially true with variables, which typically correspond to scannable hardware that may not have even moved across some (or any) dimensions. To avoid unnecessarily large arrays, WrightTools allows Channels and Variables to have different sizes than the parent data. As an example, consider the following object.

```
>>> import WrightTools as wt; from WrightTools import datasets
>>> data = wt.data.from_COLORS(datasets.COLORS.v2p1_MoS2_TrEE_movie)
>>> data.print_tree()
MoS2 (/tmp/qhg_1b3l.wt5)
├── axes
│   ├── 0: w2 (nm) (41, 1, 1)
│   ├── 1: w1=wm (nm) (1, 41, 1)
│   └── 2: d2 (fs) (1, 1, 23)
├── variables
│   ├── 0: w2 (nm) (41, 1, 1)
│   ├── 1: w1 (nm) (1, 41, 1)
│   ├── 2: wm (nm) (1, 41, 1)
│   ├── 3: d2 (fs) (1, 1, 23)
│   ├── 4: w3 (nm) (1, 1, 1)
│   ├── 5: d0 (fs) (1, 1, 1)
│   └── 6: d1 (fs) (1, 1, 1)
└── channels
    ├── 0: ai0 (41, 41, 23)
    ├── 1: ai1 (41, 41, 23)
    ├── 2: ai2 (41, 41, 23)
    ├── 3: ai3 (41, 41, 23)
    ├── 4: ai4 (41, 41, 23)
    └── 5: mc (41, 41, 23)
```

(4.4)

Note that this is the primary dataset discussed in Chapter 11. The shape of this data object is `(41, 41, 23)`, but none of the variables have that full shape. From a quick inspection, one can see that `w1` and `wm` were scanned together, while `w2` and `d2` were the other two dimensions. `w3`, `d0`, and `d1` were not moved at all, yet their coordinates are still propagated.

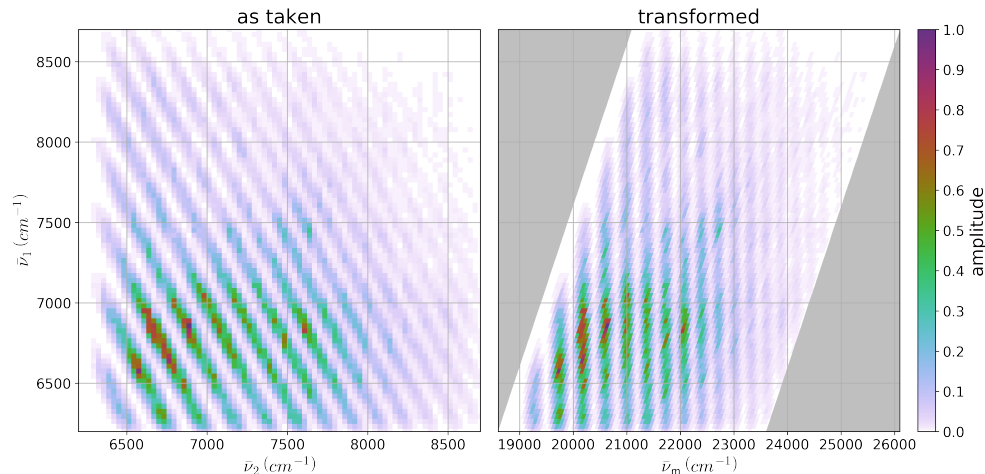
4.6 Axes

Axes are the primary interface to coordinate positions in WrightTools. Axes are *not* arrays, although they do behave like arrays. They are merely *interfaces* into the information stored in one or more variables.

Each axis has an expression, like `'w1'`, `'d1=d2'`, `'2*w3'` or `'w1+w2-wm'`. These expressions describe an unambiguous mathematical operation involving one or more variables. The axis computes these expressions on the fly when needed, giving users real flexibility over how they would like to represent and process their results.

Axes, variables, and channels are array-likes, so they support slicing operations. The axes have the joint shape of their component variables. In addition, all three classes have `points` and `full` attributes that return the squeezed and broadcasted array, respectively.

One of the coolest features enabled by this approach to axes is *transformation*. Figure 4.6 shows how easy it is to transform data from one coordinate system into another. Transforming is essentially free, since axes compute coordinates on the fly.



```

import os
import matplotlib.pyplot as plt
import WrightTools as wt; from WrightTools import datasets

here = os.path.abspath(os.path.dirname(__file__))

data = wt.data.from_PyCMDS(datasets.PyCMDS.w2_w1_000)
data.signal_mean.symmetric_root(0.5) # to amplitude level
data.convert('wn')

fig, gs = wt.artists.create_figure(width='double', cols=[1, 1, 'cbar'])
# as taken
ax = plt.subplot(gs[0, 0])
ax.pcolor(data)
wt.artists.set_ax_labels(xlabel=data.w2.label, ylabel=data.w1.label)
ax.grid(); ax.set_title('as taken', fontsize=20)
# transformed
ax = plt.subplot(gs[0, 1])
data.transform('wm', 'w1')
data.convert('wn')
ax.pcolor(data)
wt.artists.set_ax_labels(xlabel=data.wm.label, yticks=False)
ax.grid(); ax.set_title('transformed', fontsize=20)
# colorbar
wt.artists.plot_colorbar(plt.subplot(gs[0, -1]), label='amplitude')
# save
wt.artists.savefig(os.path.join(here, 'fringes_transform.png'))

```

Figure 4.6: Example using transform.

4.7 Math

Now that we know the basics of how the WrightTools `Data` class stores data, it's time to do some data manipulation. Let's start with some elementary algebra.

4.7.1 In-place operators

In Python, operators are symbols that carry out some computation. Consider the following:

```
>>> import numpy as np
>>> a = np.array([4, 5, 6])
>>> b = np.array([-1, -2, -3])
>>> c = a + b
>>> c
array([3, 3, 3])
```

(4.5)

Here, `a` and `b` are operands and `+` is an operator. When used in this simple way, operators typically create and return a *new* object in the computers memory. We can verify this by using Python's built-in `id` function on the objects created in 4.5.

```
>>> id(a), id(b), id(c)
(139712529580400, 139712333712320, 139712333713040)
```

(4.6)

This is usually fine, but sometimes the operands are unwieldy large objects that take a lot of memory to store. In other cases operators are used millions of times such that, used as above, millions of new arrays will be created.

One way to avoid these problems is to use *in-place* operators. Using a slightly different syntax, one can tell Python to overwrite one of the operands with the new value. Continuing from 4.5:

```
>>> a += b
>>> a
array([3, 3, 3])
```

(4.7)

No output `c` array was created, so no additional memory footprint is needed in 4.7. Since WrightTools channels and variables are typically large arrays, and since these arrays are stored on disk inside of a larger file, WrightTools requires the use of in-place operators for all normal math. Currently WrightTools

supports addition (`+=`), multiplication (`*=`), power (`**=`), subtraction (`-=`), and division (`/=`). As an example, consider dividing a channel by a specific factor:

```
>>> import WrightTools as wt; from WrightTools import datasets
>>> data = wt.data.from_JASCO(datasets.JASCO.PbSe_batch_1)
data.created at /tmp/tdyvfxu8.wt5:/
  range: 2500.0 to 700.0 (nm)
  size: 1801
>>> data.signal
<WrightTools.Channel 'signal' at /tmp/tdyvfxu8.wt5:/signal>
>>> data.signal.min(), data.signal.max()
(0.10755, 1.58144)
>>> data.signal /= 2
>>> data.signal.max(), data.signal.min()
(0.053775, 0.79072)
```

(4.8)

Variables also support in-place operators.

4.7.2 Clip

Clip allows users to exclude values outside of a certain range. Excluded values can be replaced with a user-specified value or with not-a-number. This can be particularly useful in cases like fitting. See Section 4.9 for an example.

It's also useful for when noise in a certain region of a spectrum obscures useful data. Particularly true for normalized and signed data.

4.7.3 Symmetric root

Homodyne vs heterodyne-detected data need to be scaled appropriately for comparison. Much of the data that we collect in the Wright Group is homodyne detected, so it goes as N^2 . To compare with the majority of other experiments, including basic linear experiments like absorption and Raman spectroscopy, need to plot on “amplitude level”, that is $\text{amplitude} = \sqrt{|\text{signal}|}$.

Due to things like leveling, chopping, baseline subtraction, and simple noise even homodyne detected data typically include negative numbers. Symmetric root treats these values as cleanly as possible by applying the same relative scaling to positive and negative values, and keeping the sign of each pixel,

as the following pseudocode shows.

```
def symmetric_root(value):
    return sign(value) * sqrt(abs(value))
```

(4.9)

For generality, `wt.Channel.symmetric_root` accepts any root as an argument. The default is 2, for the common case of going from intensity scaling to amplitude scaling.

Any other power can be applied to a channel using the in-place `**=` syntax.

4.7.4 Log

The method `wt.Channel.log` applies logarithmic scaling to a channel. The base of the log is settable by keyword argument, with a default of e. There are also methods `wt.Channel.log10` and `wt.Channel.log2`, which accept no keyword arguments. These may be slightly faster than `channel.log(base=10)` and `channel.log(base=2)`.

4.7.5 Trim

Trim uses statistical treatment to find and remove outliers from a dataset. It is useful in cases where the naive strategy employed by `wt.Channel.clip` is not sufficient, and when preparing for fitting. See Figure 4.7 for an example of outlier detection.

Currently `trim` only supports one statistical treatment: the z-test. Z-testing compares each pixel to its multidimensional neighborhood of pixels. If the pixel is more than n standard deviations outside of the neighborhood mean (using the neighborhood standard deviation) it is either masked, replaced with `np.nan`, or replaced with the neighborhood mean. All outliers are found before any outliers are modified, so the algorithm is not directional.

`wt.Channel.trim` can easily be enhanced with other statistical methods as needed.

4.7.6 Smooth

`wt.Channel.smooth` essentially passes the channel through a low-pass filter. It does this by convolving the channel with an n-dimensional Kaiser–Bessel window.

Smoothing is a highly destructive process, and can be very dangerous if used unthinkingly. However it can be useful when noisy data is collected in high resolution. By taking many more pixels than required to capture the relevant spectral or temporal features, one can confidently smooth collected data in post to achieve clean results. This strategy is similar to that accomplished in time domain CMDS where a low-pass filter is applied on the very high resolution raw data.

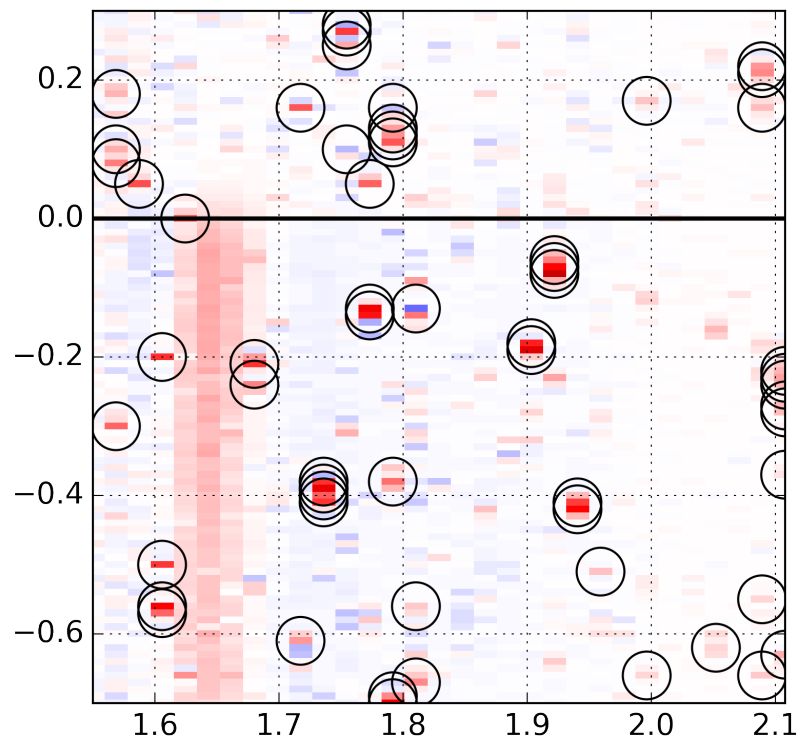


Figure 4.7: Outlier detection using a z-test.

4.8 Dimensionality manipulation

WrightTools offers several strategies for reducing the dimensionality of a data object. Also consider using the `fit` sub-package described in Section 4.9.

4.8.1 Chop

Chop is one of the most important methods of data, although it is typically not called directly by users of WrightTools. Chop takes n-dimensional data and “chops” it into all of its lower dimensional components. Consider a 3D dataset in `('wm', 'w2'', 'w1'''')`. This dataset can be chopped to its component 2D `('wm'', 'w1')` spectra.

```
>>> import WrightTools as wt; from WrightTools import datasets
>>> data = wt.data.from_PyCMDS(datasets.PyCMDS.wm_w2_w1_000)
data created at /tmp/lzyjg4au.wt5:./
  axes ('wm', 'w2', 'w1')
  shape (35, 11, 11)
>>> chopped = data.chop('wm', 'w1')
chopped data into 11 piece(s) in ('wm', 'w1')
>>> chopped.chop000
<WrightTools.Data 'chop000' ('wm', 'w1') at /tmp/935c2v5a.wt5:./chop000>
```

(4.10)

`chopped` is a collection containing 11 data objects: `chop000, chop001 ... chop010`. Note that, by default, the collection is made at the root level of a new tempfile. An optional keyword argument `parent` allows users to specify the destination for this new collection. These lower dimensional data objects can then be used in plotting routines, fitting routines etc.

By default, chop returns *all* of the lower dimensional slices. Considering the same data object from 4.10, we can choose to get all 1D `wm` slices.

```
>>> chopped = data.chop('wm')
chopped data into 121 piece(s) in ('wm',)
>>> chopped.chop000
<WrightTools.Data 'chop000' ('wm',) at /tmp/pqkbc0qr.wt5:./chop000>
```

(4.11)

If desired, users may use the `at` keyword argument to specify a particular coordinate in the un-retained dimensions. For example, suppose that you want to plot the data from 4.10 as an `wm, w1` plot at `w2`

= 1580 wn.

```
>>> chopped = data.chop('wm', 'w1', at={'w2': [1580, 'wn']})[0]
chopped data into 1 piece(s) in ('wm', 'w1')
>>> chopped
<WrightTools.Data 'chop000' ('wm', 'w1') at /tmp/_yhrdprp.wt5:/chop000>
>>> chopped.w2.points
array([1580.0])
```

(4.12)

4.8.2 Collapse

`wt.Data.collapse` reduces the dimensionality of the data object by exactly 1 using some mathematical operation. Currently supported methods are integrate, average, sum, max, and min, with integrate as default. Collapsing a dataset is a very simple and powerful method of dimensionality reduction. It allows users to inspect the net dependency along a set of axes, without being opinionated about the coordinate in other dimensions. It can also be used as a method of noise reduction.

4.8.3 Split

`wt.Data.split` is not a proper method of dimensionality reduction, but it is a crucial tool for interacting with the dimensionality of a data object. `split` allows users to access a portion of the dataset. The most common use-case is certainly in fitting operations. In population spectroscopies like transient absorption and transient grating it has become typical to take three-dimensional “movies” in `('w1', 'w2', 'd2')`, where `w1` is a probe, `w2` is a pump, and `d2` is a population delay. It can be informative to fit each `d2` trace to a model (often single exponential), but such a fit will not do well to describe the signal through zero delay and for positive `d2` values (into the coherence pathways). `data.split(d2=0.)` will return two data objects, one for the positive delays and one for negative. You can then pass the data object with only population response into your fitting routine.

4.8.4 Join

Like `split`, `wt.data.join` is not a method of dimensionality reduction. It is also not a method of the `Data` class, it is a bare function. `Join` accepts multiple data objects and attempts to join them together. To do this, the variable and channel names must agree.

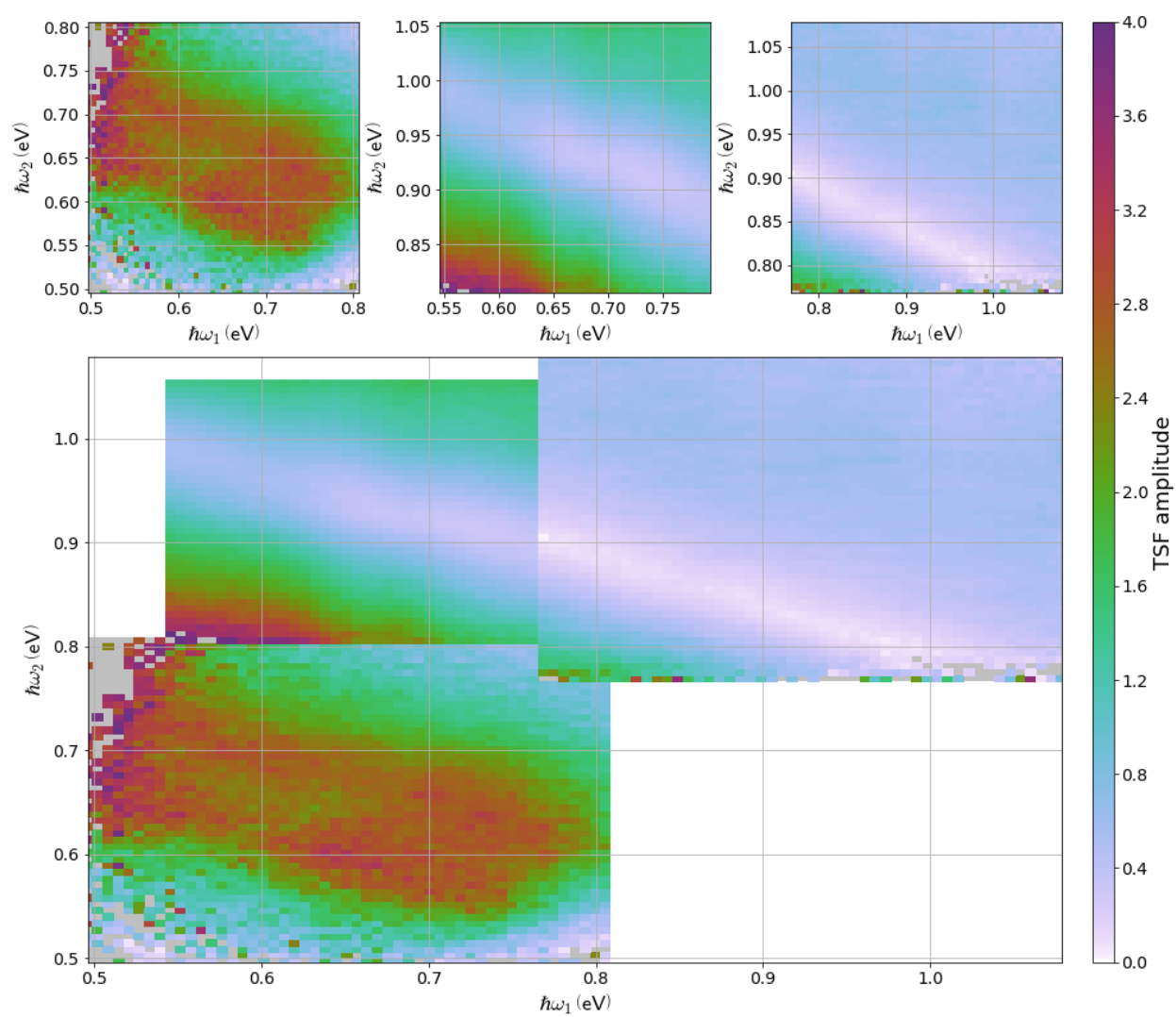


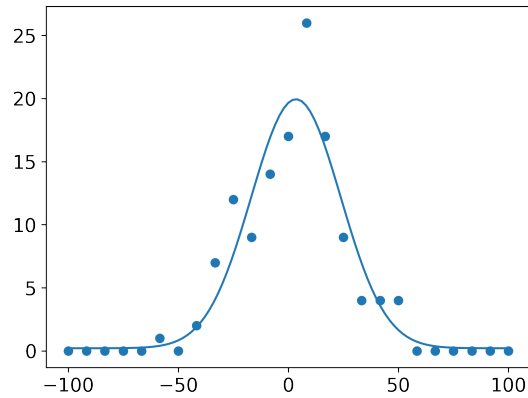
Figure 4.8: Join example.

4.9 Fitting

Like the rest of WrightTools, the `fit` sub-package is made to play as nicely as possible with high-dimensional data. WrightTools uses fitting as a method of dimensionality reduction. For example, consider a three-dimensional `('w1', 'w2', 'd2')` “movie”, where `d2` is a population delay that can be well approximated by a single exponential decay with offset. Rather than attempt to visualize `w1, w2` at some specific value of `d2`, it can be powerful to instead consider the parameters (amplitude, offset, and time constant) of an exponential fit at each `w1, w2` coordinate. On a more practical note, this kind of slice-by-slice dimensionality reduction via fitting can greatly simplify automated instrumental calibration (see Chapter 7). WrightTools employs some simple tricks to enable these kind of fit operations, described here.

4.9.1 Function objects

One challenge of slice-by-slice fitting is making a good initial guess to optimize from. It is not tractable to ask the user to provide a guess for each slice, so some kind of reasonable automated guessing must be used. WrightTools “function” objects are self contained describers of a particular function. As an example, consider the `wt.fit.Exponential` class. It can be used directly, as shown in Figure 4.9



```

# import
import os
import numpy as np
import matplotlib.pyplot as plt
import WrightTools as wt
from WrightTools import fit
# define
here = os.path.abspath(os.path.dirname(__file__))
# noisy gaussian
xi = np.linspace(-100, 100, 25)
yi = 20*np.exp(-0.5*((xi-5)/20)**2)
yi = np.random.poisson(yi)
plt.scatter(xi, yi)
# fitted
g = wt.fit.Gaussian()
ps = g.fit(yi, xi)
xi = np.linspace(-100, 100, 101)
model = g.evaluate(ps, xi)
# plot
plt.plot(xi, model)
wt.artists.savefig(os.path.join(here, 'fit_function.png'))

```

Figure 4.9: Fitting a Gaussian.

4.9.2 Fitter

The Fitter class is specially made to work seamlessly with data objects.

WrightTools is especially good at dimensionality reduction through fitting. This concept is best demonstrated through an example.

Let's load in some test data.

```
#import
import WrightTools as wt
from WrightTools import datasets
# create
ps = datasets.COLORS.v2p1_MoS2_Tree_movie
data = wt.data.from_COLORS(ps)
# cleanup
data.level('ai0', 'd2', -3)
data.scale()
data.convert('eV')
data.name = 'MoS2'
```

(4.13)

This is a three-dimensional dataset:

```
>>> data.axis_names
['w2', 'w1', 'd2']
>>> data.shape
(41, 41, 23)
```

(4.14)

We could create an animation to see every single pixel, but we can't see everything at once that way. Instead we could imagine fitting every decay (τ_{21} trace) to an exponential. Then we could plot the amplitude and time constant of that exponential decay. This helps us get at subtle questions about the data. Do the lineshapes narrow with time? Does the redder feature decay slower than the bluer feature? Faster?

Using the `Fitter` class, it is easy to perform an exponential fit along each TAU21 trace at each OMEGA1, OMEGA2 coordinate.

```
# isolate only relevant data
data = data.split('w1', 1.75)[1].split('d2', 0)[0]
# prepare a function
function = wt.fit.Exponential()
function.limits['amplitude'] = [0, 1]
function.limits['offset'] = [0, 0]
function.limits['tau'] = [0, 2000]
# do the fit
fitter = wt.fit.Fitter(function, data, 'd2')
outs = fitter.run()
```

(4.15)

When we call `fitter.run()`, every slice of the data object will be fit according to the given function object. Fitter automatically creates two new data objects when this happens. `outs` contains the fit parameters, in this case amplitude, tau, and offset. Accordingly, `outs` is lower-dimensional than the original data object. `model` contains the fit evaluated at each coordinate of the original dataset—it's really useful for inspecting the quality of your fit procedure.

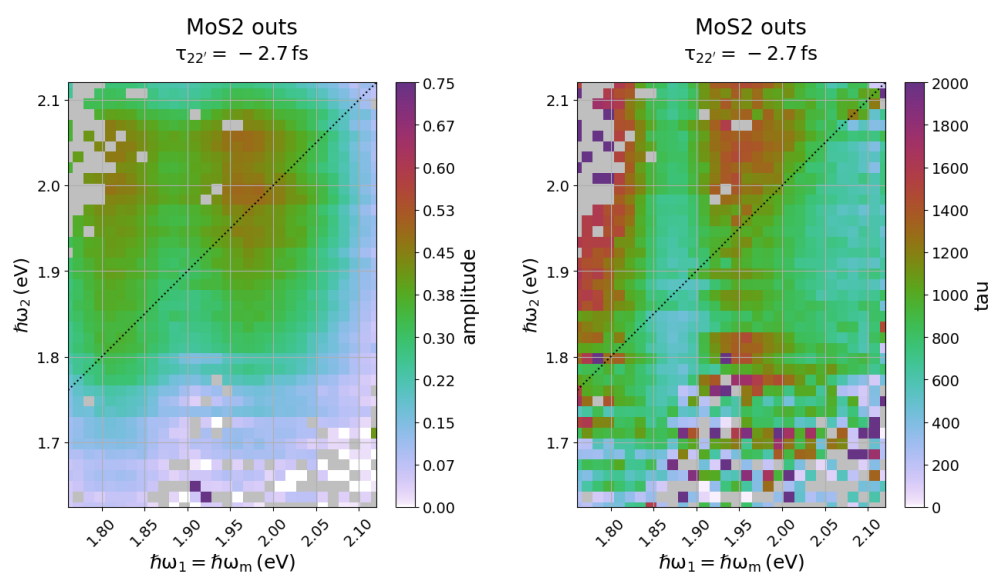


Figure 4.10: Fitting as dimensionality reduction.

4.10 Construction, maintenance, and distribution

While WrightTools has already been useful to the work done in the WrightGroup over the last 3 years, the true potential of the package lies in its future. WrightTools is designed in a modular way so that it can be continuously enhanced to serve an ever-wider set of users and spectroscopies. Despite its name, WrightTools is built to be used even by those outside of the Wright Group. Currently WrightTools may be only 75% of what a typical multidimensional spectroscopist needs, but if those scientists work to enhance the package with what *they* need, they may also solve problems for others such that the usefulness of the software gradually increases.

In order for this dream to come true, WrightTools must be constructed and maintained by collaborative tools such that users feel comfortable contributing to future enhancements. All of the challenges to collaboration discussed in Chapter 3 certainly apply to WrightTools, so it is important that we follow best practices now in order to make WrightTools as maintainable and future-proof as possible. To this end, this section discusses strategies that I have employed in the construction, maintenance, and distribution of WrightTools.

4.10.1 Licensing

As discussed in Chapter 3, open source licenses are an important part of scientific software development. Those writing software must explicitly license their project in order to ensure that users have basic rights to copy edit and distribute code. WrightTools is licensed under the hugely popular Expat/MIT license. This license is incredibly permissive and puts as few restrictions as possible on the end users. The following is the WrightTools license, reproduced in its entirety.

The MIT License (MIT)

Copyright (c) 2016-2018 WrightTools Developers.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

As an aside, since Python is an interpreted language the source code of a library *must* be distributed for that library to work. In this way, “closed-source Python” is a kind of oxymoron. However many Python libraries end up being interfaces to compiled code that could in theory be closed-source. The Scientific Python Stack have MIT-compatible licenses, including BSD-like licenses.

4.10.2 Version control

As mentioned several times in Chapter 3, having software under source control is probably the most important recommendation in scientific software development. Source control allows developers to create “checkpoints” for their software package that can be returned to again and again. Developers can collaborate together to edit the software by making incremental changes that are easy to review.

WrightTools uses git for source control, and the package is hosted on GitHub [64]. As of 2018-04-08, WrightTools has 1,018 commits from seven developers, as shown in Table 4.6. In addition to simply hosting the git repository, GitHub gives us issue tracking, continuous integration, and Zenodo support.

The WrightTools package has a developer controlled version as well, following the semantic versioning convention [67]. The current distributed version of WrightTools is 3.0.1, with 3.0.2 under active development. The wt5 file also has a semantic version, currently 1.0.0. These attributes can be accessed through python: `wt.__version__` and `wt.__wt5_version__`.

person	number of commits	lines added	lines removed
Blaise Thompson	478	621,918	507,938
Kyle Sunden	208	19,293	9,218
Darien Morrow	29	1,589	127
Nathan Neff-Mallon	20	2,880	910
Kyle Czech	5	1,150	25
Daniel Kohler	3	113	29
Rachel Swedin	1	5,197	0

Table 4.6: Commits to WrightTools by individual, ordered by number of commits. Note that datasets are included, so lines added and removed are massively inflated.

4.10.3 Unit tests

Maintainable code must be tested, so that future developers can use tests to ensure that they do not break necessary functionality. Unit testing is a very simple testing paradigm in which small, separate tests are written to address each “unit” of the software package. As an example, the following is one of WrightTools’ tests:

```
# part of WrightTools/tests/data/convert_data.py
def test_wigner():
    p = datasets.COLORS.v2p2_WL_wigner
    data = wt.data.from_COLORS(p)
    assert data.d1.units == 'fs'
    data.convert('ns')
    assert data.d1.units == 'ns'
    assert data.wm.units == 'nm'
    data.close()
```

 (4.16)

This test loads one of the distributed COLORS datasets and makes sure that the `convert` method works as intended. To do this, it uses the `assert` statement which raises an exception when a condition is `False`. This particular test is pretty humble, but there is strength in numbers: as of 2018-04-08 there are 224 unit tests within WrightTools. Using the built in `pytest` library (`python setup.py test`), a programmer can run all of the tests and receive a report on what failed and why. If a future programmer unintentionally breaks `convert`, the above test will fail and alert her to the unexpected side effect of her modification.

WrightTools uses continuous integration services to run unit tests every time a pull request is made on GitHub.

4.10.4 Distribution

WrightTools is on GitHub, which is a fantastic way for developers to get software onto their computers. But how does software get onto everybody else’s machine? Developers call this process “distribution”. Luckily for us, distribution is fairly simple within the Python ecosystem. The same tools that are used to distribute hugely popular packages like `numpy` are also available for anyone else, including WrightTools.

The Python Package Index (PyPI) is the official third party software repository for Python. It is community maintained, and supported by the Python Software Foundation and The Python Packaging Group. As of 2018-04-08 PyPI hosts 134,758 Python packages, all for free. WrightTools is also hosted on PyPI. Every time we change our version, we “release” by uploading the newest version to PyPI. [58] pip (“pip installs packages”, “pip installs python”, or “preferred installer program”) can be used to install packages directly from PyPI:

```
pip install WrightTools (4.17)
```

Conda is a multilingual package manager that handles virtual environments and dependencies, even binary dependencies, in a hassle-free way. Since the scientific Python ecosystem has so many non-Python binary dependencies, Conda is a popular choice—especially on Windows where the necessary compilers are not typically pre-installed. Unlike pip, conda is not tied to a single repository. There is the official repository, maintained by Anaconda, the company that develops conda. There is also the popular conda-forge repository, which is maintained by the community via GitHub. WrightTools is on conda-forge:

```
conda config --add channels conda-forge  
conda install WrightTools (4.18)
```

4.10.5 Documentation

WrightTools is a piece of scripted software, and many spectroscopists many not be comfortable with using such a thing immediately. To this end, it is important to have easy to use, searchable documentation with end-users in mind.

I have built a website for WrightTools documentation at <http://wright.tools/>. The documentation is built and hosted by Read the Docs. [95]. Both master and development versions are built, so users of both can get documentation that is up to date for them.

Chapter 5

Acquisition

The question of software correctness ultimately boils down to, "Does it do what we have in our minds, even the things we have not gotten around to thinking about yet?"

– Alistair Cockburn

5.1 Introduction

At its core, MR-CMDS is about delivering multiple pulses of light to a sample. The frequency and relative arrival time (delay) of each pulse must be scanned in the context of a basic multidimensional experiment. Scanning frequency requires using motors to change crystal angles and other optics within Optical Parametric Amplifiers (OPAs). Scanning delay typically involves moving mirrors very small distances such that the optical path length of certain beam-lines changes. In addition to these “first-order” controls, the power of MR-CMDS is enhanced with additional control of pulse intensity and polarization. An automated monochromator is typically used to spectrally resolve or isolate output signal. Each of these features is an optomechanical device: a piece of hardware that must be controlled in the context of an MR-CMDS experiment.

A scan in MR-CMDS typically means sending hardware to a whole series of different positions. As an example, a three-dimensional “movie” might be collected in the following way:

```
w1_points = [14300, 14400, 14500, 14600, 14700] # wn
w2_points = [14100, 14200, 14300, 14400, 14500, 14600, 14700] # wn
d2_points = [100, 75, 50, 25, 0, 50, 75, 100, 125, 150, 175, 200] # fs
for w2 in w2_points:
    set_w2(w2)
    for w1 in w1_points:
        set_w1(w1)
        for d2 in d2_points:
            set_d2(d2)
            measure_signal()
```

(5.1)

In this simple example, there are 5 `w1` destinations, 7 `w2` destinations, and 12 `d2` destinations, so there are a total of $5 \times 7 \times 12 = 420$ pixels in the three-dimensional scan. The acquisition software must set the hardware to each of these points and acquire data at each of them. Each hardware motion and signal measurement takes roughly one second, so this acquisition would take roughly 7 minutes to complete. In practice, real scans are composed of ~ 100 to $\sim 100,000$ pixels, and take between 1 minute and one day to acquire.

Because of the highly specialized nature of these experiments, MR-CMDS instruments typically require custom software to address all of the simultaneous, repeated motor motion that a scan requires. Because MR-CMDS is really a family of techniques which require different kinds of motor motion, this acquisition

software should be flexible enough to meet the creativity of its users. Furthermore, because MR-CMDS is a rapidly evolving technique the instrumental software must be built in an extendable way to accommodate the ever-changing hardware configurations. In this chapter I describe how I built such an acquisition software, PyCMDS.

For context, some description of the acquisition software that PyCMDS replaced is warranted.

On the “ps table” (focus on mixed vibrational-electronic spectroscopy of molecular systems), PyCMDS replaces ‘ps_control’, an older acquisition software first developed prior to 2004 by Kent Meyer [96]. ps_control was very-much not modular, designed by generations of graduate students each getting to “minimum viable” as quickly as possible. When I joined the group, ps_control had become unsustainable. The ps table was being revamped with new hardware, and the old motherboard finally died, so new software was desperately needed.

On the “fs table” (focus on semiconductor photophysics), PyCMDS replaces ‘Control for Lots of Research in Spectroscopy’ (COLORS), developed by Schuyler Kain [89]. PyCMDS is, in many ways, inspired by COLORS. Kain’s design was modular in many ways. Still, there were fundamental problems that COLORS could not address. Chief among them was Kain’s approach to the National Instruments DAQ card, which did not allow for the flexibility required to introduce more interesting chopping schemes (see Section 6.4).

PyCMDS is a unified software for controlling hardware and collecting data in the Wright Group. It is written almost entirely in Python, with a graphical user interface (GUI) made using Qt. It is cross-platform, with a core capable of running on Linux, Windows and macOS. It is open source, developed on GitHub. [64] Today PyCMDS is used to drive both the ps and fs tables.

PyCMDS is best thought of as a core program with three kinds of modular “plugins” that can be extended as needed. The three plugin kinds are

- Hardware: things that can be set to a position (Section 5.4).
- Sensors: things that can be used to measure a signal (Section 5.5).
- Acquisition modules: things that can be used to define and carry out an acquisition, and associated post-processing (Section 5.7).

The first design rule for PyCMDS is that these three things should be easy for the average (motivated) user to add by herself. Modularity and extensibility is important for all scientific software projects (see Chapter 3), but it is of paramount importance for acquisition software simply because the diversity of hardware and experimental configurations is so great. It is conceivable to imagine 90% overlap between the data processing and simulation needs of one spectroscopist to the next, but there is almost no overlap between hardware configurations even in the two primary instruments maintained by the Wright Group. Besides the extendable modular pieces, the rest of PyCMDS is a mostly-static code-base that accepts modules and does the necessary things to handle display of information from, and communication between, them.

PyCMDS offers many ways to interact with component hardware. Hardware can be set directly, or it can be moved in the context of a scan. Less obviously, hardware must also move in the context of “active correction”, such as spectral delay correction. I love to use the analogy of the nervous system when thinking of these two kinds of instructions. I have borrowed the terms “autonomic” and “somatic”:

The autonomic nervous system, which innervates primarily the smooth musculature of all organs, the heart and the glands, mediates the neuronal regulation of the internal milieu. The actions of this system, as its name implies, are in general not under direct voluntary control. These characteristics distinguish the autonomic nervous system from the somatic nervous system, which mediates afferent and efferent communication with the environment and, for the most part, is subject to voluntary control and accessible to consciousness.

The autonomic and somatic systems operate hand in hand...

The functions of the autonomic nervous system are to keep the internal milieu of the body constant (homeostasis...) or adjust it as required by changing circumstances (e.g., mechanical work, food intake, water deprivation, heat or cold).

– W. Jänig, Autonomic Nervous System (1989) [97]

Within PyCMDS, the autonomic system (Section 5.6) handles the “reflex” motion that is part of active correction. The somatic system (Section 5.7), on the other hand, handles the “voluntary” motion in the context of acquiring a multidimensional scan.

5.2 Graphical user interface

In this section I introduce the GUI of PyCMDS, with the goal of introducing the basic structure as experienced by a first time user of the software. When PyCMDS starts up, the GUI is constructed out of modules depending on which hardware and sensors the user has instructed the program to address. A screenshot of the PyCMDS GUI, running on the fs table, is shown in Figure 5.1.

On the left hand there is a single column displaying the current positions for all loaded hardware. Users may enter new destinations and hit the “SET” button. Positions have units, which are changeable through the pull-down menu next to the control and display. Each hardware knows its own limits, displayed in a tool tip when hovering over the control. Users cannot type values outside of hardware limits into the controls. Each hardware also has an “ADVANCED” button, which takes the user to a more extensive GUI to control lots more features (Section 5.4). At the very top, on the left hand side, is the “SHUT DOWN” button.

On the right hand side there is a extensive set of nested tabs. The top level tabs are “Program”, “Hardware”, “Devices”, “Autonomic”, “Somatic” and “Plot”. Under each of these tabs is an entire separate set of display and control elements. Some of these elements are themselves tabbed, like the “Somatic” tab (active in Figure 5.1), which has “Queue” and “Scan” sub-tabs. At the top of the right hand side there is a progress bar and queue status display, which I will discuss further in future sections.

When PyCMDS opens (Figure 5.1), the user is first greeted with the “Somatic/Queue” tab on the right hand side of the GUI. This is where she may instruct PyCMDS to do acquisitions. In PyCMDS, all acquisitions are done in a queue system. A queue is just a list of acquisitions, where the acquisitions are carried out one by one in order.

To instruct PyCMDS to do an acquisition, the user must first create a queue by entering in a chosen name (default is “queue”), and pressing the “MAKE NEW QUEUE” button. Alternatively the user may open an existing queue using “OPEN QUEUE”. Next, the user must add the desired acquisition(s) to the queue. There are several acquisition modules, each with a different purpose (Section 5.7). Choose an acquisition module using the drop down menu. Enter a name for your acquisition, and any additional info that you might want to keep track of. Finally, fill out any additional information that acquisition

module might require, and press “APPEND TO QUEUE”. Once there are acquisitions in the queue, the user can press “RUN QUEUE”.

Figure 5.2 is a screenshot of the “Somatic/Queue” tab while a queue is in progress. In this screen shot there are seven enqueued items, and PyCMDS is in the process of acquiring the final one (index 6). For each item the status, start time, exit time, and description are shown. Users may enqueue new items at any time, even while the queue is running. They may delete items which are enqueued but not yet started. Using the green “LOAD” button in each row, users may populate the right hand input menu with the parameters from that acquisition. This is useful when repeating an acquisition with slightly changed parameters, or simply when inspecting what parameters were used in a given item.

Figure 5.3 is a screenshot of PyCMDS during a representative acquisition. This time, the “Somatic/Scan” tab is chosen. On the left hand side, we can see that the “SHUT DOWN” and “SET” buttons are grayed out and unclickable, since the somatic system is in control of PyCMDS. At the moment when this screen shot was taken, PyCMDS is waiting for w3 (OPA-800CG) to finish moving, as the “BUSY” indicator shows. The progress bar, near the top of the GUI, is partially green to indicate the portion of the current scan that is finished. On the left hand side of the progress bar the time elapsed (fifteen minutes, thirty-seven seconds) is shown, and on the right hand shows the time remaining (three hours, eight minutes, sixteen seconds). In the center of the progress bar are the current scan parameters.

The “Somatic/Scan” tab is built as an active display of currently acquiring data. A large graph and number show the current signal levels. On the far right-hand side, the device and channel to display can be chosen via drop-down menu. Under “Status”, the loop time and scan index help users gauge the progress of their scan. In this case, the displayed pixel (index 6, 40) took 2.448 seconds to acquire.

The other tabs are for more advanced interactions, and will be discussed further in future sections. The “Program” tab contains various rarely-needed displays and inputs to configure PyCMDS. The “Hardware” tab is where the advanced menu for each hardware appears (Section 5.4). The “Devices” tab is where sensor settings live (Section 5.5). The “Autonomic” tab is where users configure the autonomic system for active correction (Section 5.6). The “Somatic” tab has already been described in this section. Finally, the “Plot” tab was intended to be a interactive, graphical post processing environment. This functionality has not yet been implemented.

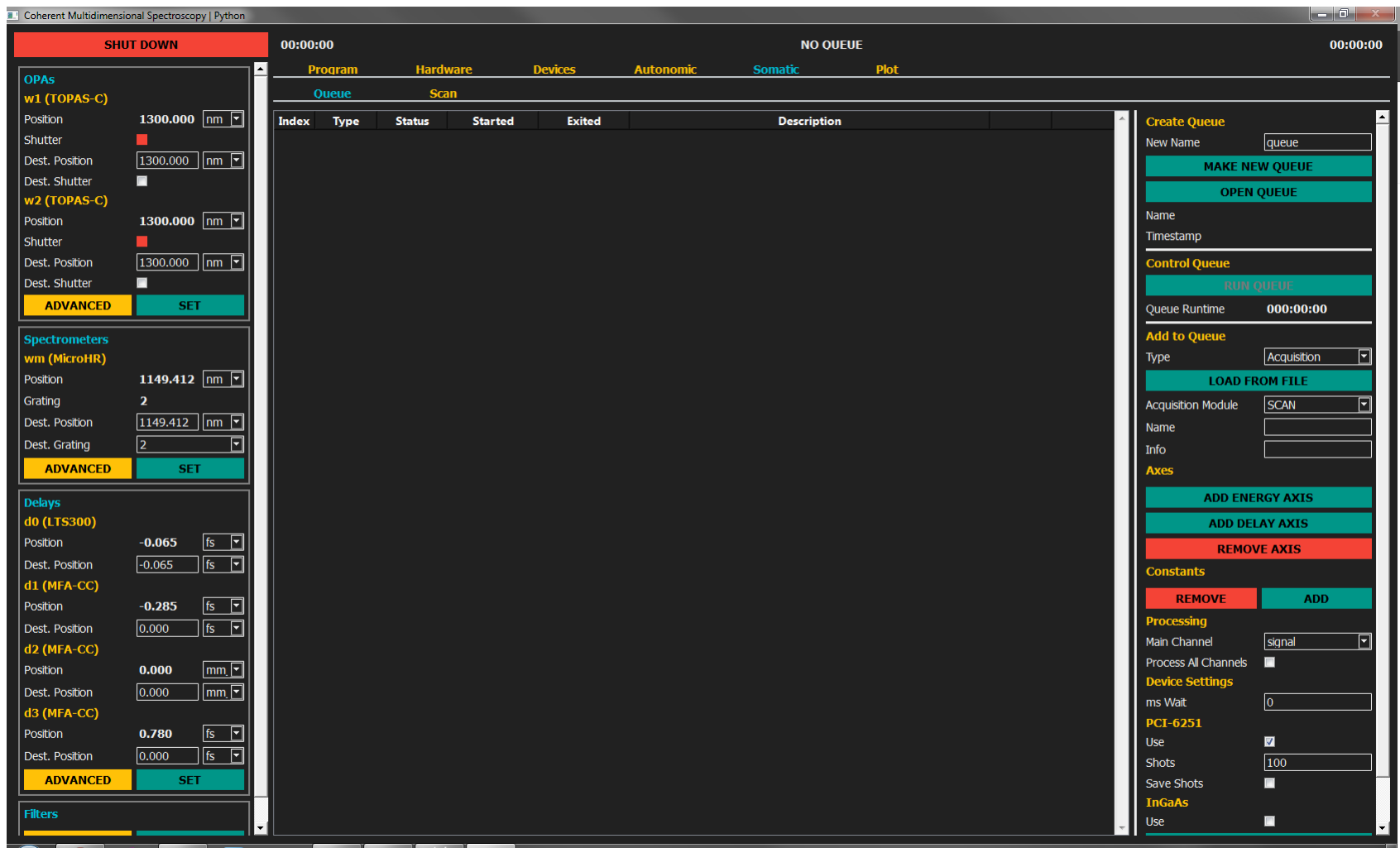


Figure 5.1: PyCMDS at startup, on the fs system.

Coherent Multidimensional Spectroscopy | Python

SHOT: 000001 00:21:25 SCAN: [W2, W3] 03:01:37

Program Hardware Devices Autonomic Somatic Plot

Queue Scan

Index	Type	Status	Started	Exited	Description	REMOVE	LOAD
0	acquisition	FAILED	15:08:08	15:11:30	SCAN: [d1, d2]	REMOVE	LOAD
1	acquisition	FAILED	15:11:48	15:16:28	SCAN: [d1, d2]	REMOVE	LOAD
2	acquisition	COMPLETE	15:18:12	15:24:26	SCAN: [w3]	REMOVE	LOAD
3	acquisition	COMPLETE	15:36:13	15:38:22	SCAN: [w2]	REMOVE	LOAD
4	acquisition	COMPLETE	15:42:13	15:44:15	SCAN: [w1]	REMOVE	LOAD
5	acquisition	COMPLETE	15:49:01	17:20:41	SCAN: [w2, w1]	REMOVE	LOAD
6	acquisition	RUNNING	17:20:41		SCAN: [w2, w3]	REMOVE	LOAD

OPAs

w1 (TOPAS-800)
 Position: 3040.000 wn
 Dest. Position: 2790.000 wn

w2 (OPA-800)
 Position: 1520.000 wn
 Dest. Position: 1270.000 wn

w3 (OPA-800CG) BUSY
 Position: 16400.000 wn
 Dest. Position: 16500.000 wn

Spectrometers

wm (MicroHR)
 Position: 17919.785 wn
 Grating: 1
 Dest. Position: 18020.000 wn
 Dest. Grating: 1

Delays

d1 (PMC)
 Position: 0.600 ps
 Dest. Position: 0.600 ps

d2 (PMC)
 Position: -1.800 ps
 Dest. Position: -1.800 ps

Filters

0 (energy)
 Initial: 1550.000 wn
 Final: 1250.000 wn
 Number: 61
 w1:
 w2:
 w3:
 wm:

1 (energy)
 Initial: 3100.000 wn
 Final: 2500.000 wn
 Number: 121
 w1:
 w2:
 w3:
 wm:

ADD ENERGY AXIS
 ADD DELAY AXIS
 REMOVE AXIS

Constants

Constant
 Hardware: wm
 Expression: w1-w2+w3
 REMOVE ADD

Processing
 Main Channel: signal_diff
 Process All Channels:

Device Settings
 ms Wait: 0

PCI-6251
 Use:
 Shots: 200
 Save Shots:

SAVE FILE
 APPEND TO QUEUE

Figure 5.2: PyCMDS queue while acquiring data, on the ps system.

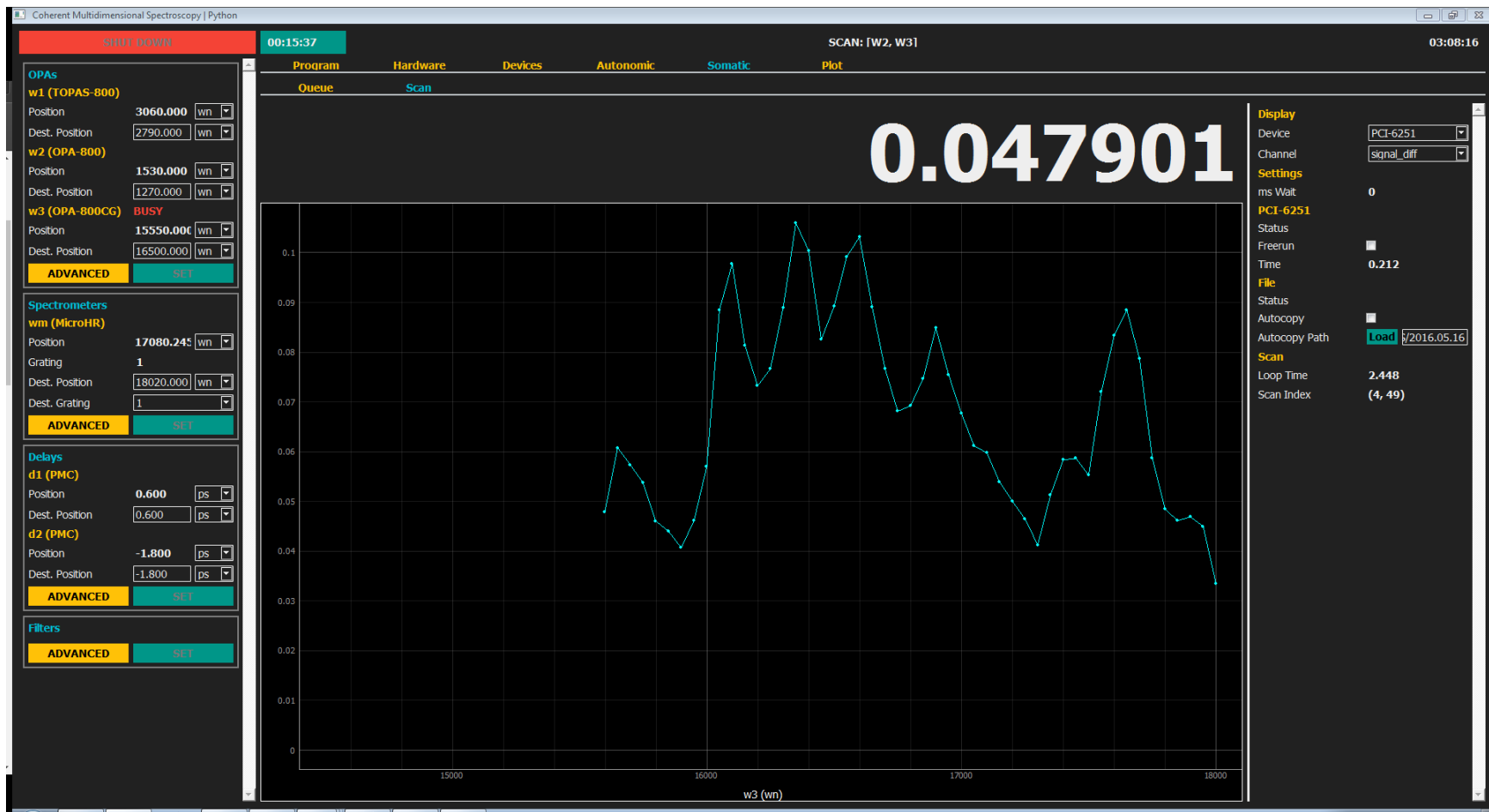


Figure 5.3: PyCMDS scan tab while acquiring data, on the ps system.

5.3 Internal structure

In this section I describe the internal structure of PyCMDS. While there are a huge number of details not worthy of discussion (at time of writing, PyCMDS consists of 16,582 lines of source code), PyCMDS is made to be maintained and extended by future graduate students, so some insight into the internal structure is warranted.

5.3.1 Multithreading

PyCMDS spends the vast majority of its runtime waiting—waiting for user input through mouse clicks or keyboard presses, waiting for hardware to finish moving or for sensors to finish reading and return signals. Despite all of this downtime, it is crucial that the software respond very quickly when instructions or signals are received. To achieve this, PyCMDS is designed using *multithreading*. The main thread handles the graphical user interface (GUI) and other “top level” things. Everything else happens in child threads. Each hardware instance (e.g. a delay stage) lives in its own thread, as does each sensor. Since only one scan happens at a time, all acquisition modules share a single thread that handles the orchestration of hardware motion, sensor operation, and data processing that the chosen acquisition requires.

Threads are powerful because they allow for “semi-synchronous” operation. Imagine PyCMDS is in the middle of a 2D delay-delay scan, and the scan thread has just told each of the two delay stages to head to their destinations. PyCMDS must sit in a tight loop to keep track of the position as closely as possible during motor motion. In a single-threaded configuration, this tight loop would only run for one delay at a time, such that PyCMDS would have to finish shepherding one delay stage before turning its attention to the second. In a multi-threaded configuration, each thread will run simultaneously, switching off CPU cycles quickly at a low level. This switching is handled in an OS and hardware specific way—luckily it is all abstracted through platform-agnostic Qt threads.

It is hard to pass information between threads. Without any special protection, two threads have no reason not to simultaneously edit and read the same location in memory. If a delay stage is writing its position to memory as a 64-bit double at the same time as the acquisition thread reads that memory

address, the acquisition thread will read in nonsense (or worse, PyCMDS will crash). So some strategy is needed to ensure that threads respect each other. The Mutex design allows threads to “lock” an object such that it cannot be modified by a different thread. This lock is like the “talking stick” employed by many early child educators. When the talking stick is being used, only the child that holds the stick is allowed to speak. The stick must be passed to another child (as directed by the teacher) before they can share their thought. PyCMDS makes heavy use of Mutexes, in particular the `QMutex` class [98].

Mutexes handle basic information transfer (two threads can both safely modify and read a particular object), but what about sending instructions between threads? Here the problem is deciding what happens when multiple instructions are given simultaneously, or an instruction is given while another instruction is being carried out. Again, this is a classic problem in computer science, and the classic answer is the queue. Queues are just like lines at the coffee shop—each person (instruction) is served (carried out) in the order that they joined the line. Queues are commonly referred to as FIFO (First In First Out) for this reason. PyCMDS uses queues for almost all instructions.

Finally, PyCMDS makes extensive use of the “signals and slots” construct, which is somewhat unique (and certainly original) to Qt. Signals and slots are powerful because they allow threads without instruction to go completely silent, making them essentially free in terms of CPU usage. Normally, a thread needs to sit in a loop to listen for instructions. Within the Qt framework, a thread can be “woken” by a signal without needing that thread to explicitly “listen”. These concepts fit within the broader umbrella of “event-driven programming”, a concept that has been used in many languages and frameworks (notably high level LabVIEW tends to be very event-driven). The Qt signals and slots system massively simplifies programming within PyCMDS.

Note that multithreading is very different from multiprocessing.

5.3.2 Abstraction and inheritance

Towards the goal of stability and extensibility, PyCMDS makes heavy use of abstraction and inheritance.

Abstraction means that complex implementation details are hidden by simple interfaces. For example, consider the simple case of setting an OPA to a particular color. For OPAs in the Wright Group, this

operation requires the following:

- Load the OPA tuning curve, find which motors must move.
- Interpolate the discrete tuning curve, and evaluate that interpolated curve at the desired destination.
- Send each motor towards their new destination.
- Wait for the motors to arrive, check that nothing has gone wrong.

This is not even to mention the complexity of spawning and sending information between the main thread and working threads. Through abstraction, PyCMDS is able to wrap all this complexity into the `OPA` class and its `set_position` method—so doing all of the operations above is as simple as `opa.set_position(1300, 'nm')`. Importantly, abstraction does not magically get rid of the complexity. It simply *hides* the complexity so that it becomes tractable to write simple interfaces to accomplish complex things.

PyCMDS implements abstraction through inheritance. In object oriented programming, inheritance is when one class is based on another. The *child* class acquires all of the properties of the *parent* class. The child class, then, can modify or extend the properties that it needs, without needing to re-implement the properties that it shares with the parent. Let's consider PyCMDS hardware again. Every single unique hardware in PyCMDS lives in its own worker thread, so the basic problem of information transfer through queues and Mutexes is shared between them. A `Hardware` class which is parent to *all* hardwares can define the methods and attributes necessary to abstract this basic thread communication issue. Every type of delay stage addressed by PyCMDS needs to handle the basic task of translating between “natural” units (femtosecond, picosecond, nanosecond) and “native” units (typically mm). They all need an attribute `zero_position`, in native units, and a method to do the conversion. A class common to all delay stages can abstract away all of these conversion details. In summary, an inheritance-based system for implementing delay stages might look like this:

```
Hardware # implements basic thread control
├── Delay # implements conversion between natural and native units
│   ├── Homemade
│   ├── Thorlabs
│   └── Newport
```

(5.2)

The powerful thing about this strategy is that the three driver-specific classes (`Homemade`, `Thorlabs`, and `Newport`) need only implement minimal driver-specific code, typically `start`, `set` and `close`.

This means that code is more maintainable and less repeated. For example, when I added the autonomic system to PyCMDS, I edited the parent `Delay` class to respect a new method `set_offset`. I did *not* need to modify any of the child classes, because nothing about communicating with the particular delay stages, in native units, had changed. This allowed me to implement the core of the autonomic system in just one weekend, something that probably would have taken weeks to do without inheritance.

5.3.3 Core classes of PyCMDS

Now we can see that PyCMDS is going to use multi-threading, inheritance and abstraction as much as possible, so let's get into some details about the *actual* internal structure of the software.

For those that want to dig deeper, most of these top level classes are defined in

`PyCMDS/project/classes.py`.

Data types

PyCMDS is made to be enhanced and extended by chemistry graduate students who may not have time, energy or enthusiasm to learn about Mutexes, signals, slots, and threads. They probably also don't have time to read Qt documentation and learn the details of GUI design and layout. PyCMDS does its very best to abstract these details away from developers by offering a set of basic *data type* classes which work seamlessly with every part of the program. These are simple classes, each meant to represent one kind of data:

- Bool
- Combo
- Filepath
- Number
- String

All are children of the parent `PyCMDS_object` class, which defines much of their shared functionality.

By using these classes to store and pass around bits of information within PyCMDS, users get the following advantages:

- Thread safety. These classes *are* Mutexes.
- Optional integration with the GUI (see section ...)
- Optional storage of state within INI files, including through restart.
- Special conveniences, like limits, units, and labels.

In short, developers should use these classes whenever possible for a worry-free development experience. The only downside is that the value has to be accessed with `read` and `write` methods, rather than directly. This is typical behavior for Mutexes, however.

Hardware and driver

Now that we have basic data types to work with, let's actually communicate with hardware. Every hardware and sensor are made of two classes: a driver class, which lives in the worker thread and handles direct communication, and a hardware class which lives in the main thread and “represents” that device to the rest of PyCMDS. The idea is that all of PyCMDS communicates to that device *only* via its hardware class, and that hardware class only talks to that driver class. Designing it in this simple way keeps everything clean and easy to understand.

All hardware and driver classes are children of the same parent `Hardware` and `Driver` classes. These parent classes know how to communicate in a thread safe way, and they know the specific attributes (like `name`), and signals (like `update_ui`) that all hardware and sensors must have. Figure 5.4 shows the parent hardware class, and Figure 5.5 shows the parent driver class.

Communication between the hardware and the driver goes via a queue, as mentioned previously. The hardware class has an attribute `q`, which is an instance of the `Q` class. To enqueue an operation, use `q.push(<method>, <arguments>)` where `<method>` is a string corresponding to the name of the method you wish to run in the worker thread, and `<arguments>` is a list of arguments passed to that method. The `q` instance will hold the instruction until the `Driver` instance is ready, at which point `Driver.dequeue` will be called in the worker thread. There is no way for the driver to command hardware to do something in the main thread, but the driver can trigger signals like `update_ui` and modify Mutexes.

```

class Hardware(QtCore.QObject):
    update_ui = QtCore.pyqtSignal()
    initialized_signal = QtCore.pyqtSignal()

    def __init__(self, driver_class, driver_arguments, gui_class,
                 name, model, serial=None):
        """
        Hardware representation object living in the main thread.

        Parameters
        driver_class : Driver class
        driver_arguments : dictionary
        name : string
        model : string
        serial : string or None (optional)
        """
        QtCore.QObject.__init__(self)
        self.name = name
        self.model = model
        self.serial = serial
        # create objects
        self.thread = QtCore.QThread()
        self.enqueue = Enqueued()
        self.busy = Busy()
        self.driver = driver_class(self, **driver_arguments)
        self.initialized = self.driver.initialized
        self.gui = gui_class(self)
        self.q = Q(self.enqueue, self.busy, self.driver)
        # start thread
        self.driver.moveToThread(self.thread)
        self.thread.start()
        # connect to address object signals
        self.driver.update_ui.connect(self.update)
        self.busy.update_signal = self.driver.update_ui
        # initialize drivers
        self.q.push('initialize')
        # integrate close into PyCMDS shutdown
        self.shutdown_timeout = 30 # seconds
        g.shutdown.add_method(self.close)
        g.hardware_waits.add(self.wait_until_still)

```

Figure 5.4: Parent class of all hardware and sensors. For brevity, methods `close`, `update` and `wait_until_still` have been omitted.

```

class Driver(QtCore.QObject):
    update_ui = QtCore.pyqtSignal()
    queue_emptyed = QtCore.pyqtSignal()
    initialized = Bool()

    def check_busy(self):
        """
        Handles writing of busy to False.

        Must always write to busy.
        """
        if self.is_busy():
            time.sleep(0.01) # don't loop like crazy
            self.busy.write(True)
        elif self.queued.read():
            time.sleep(0.1) # don't loop like crazy
            self.busy.write(True)
        else:
            self.busy.write(False)
            self.update_ui.emit()

    @QtCore.pyqtSlot(str, list)
    def dequeue(self, method, inputs):
        """
        Slot to accept enqueued commands from main thread.

        Method passed as qstring, inputs as list of [args, kwargs].

        Calls own method with arguments from inputs.
        """
        self.update_ui.emit()
        method = str(method) # method passed as qstring
        args, kwargs = inputs
        if g.debug.read():
            print(self.name, ' dequeue:', method, inputs, self.busy.read())
        self.queued.pop()
        getattr(self, method)(*args, **kwargs)
        if not self.queued.read():
            self.queue_emptyed.emit()
            self.check_busy()

    def is_busy(self):
        return False

```

Figure 5.5: Parent class of all drivers.

GUI components

The PyCMDS GUI must change depending on which exact hardware, sensors, and acquisition modules are being used on a given instrument and given day. Internally, the GUI components are made to be modular and flexible to accommodate this requirement. Rather than having each piece placed “by hand”, the PyCMDS GUI is defined programmatically and, as such, the full power of abstraction and inheritance is available to the GUI-defining code.

To keep things simple and easy to extend, PyCMDS is made up of only a few minimalist GUI elements. Probably the most important graphical element is the fixed-width vertical scroll area. As seen in [PYCMDS SCREENSHOTS], these vertical scroll areas contain almost all of the interactive elements within PyCMDS, with the only exceptions being the “SHUT DOWN” button and the interactive graphs. The left-hand scroll area is always present, and it contains the principle display and control for each hardware. There are also scroll areas inside the tabbed menus, typically only one per tab. Because the scroll areas can expand downwards infinitely, they are great at accommodating the changing contents of the PyCMDS GUI.

Ignoring small decorative items, vertical scroll areas contain only two kinds of widgets: instances of `Button` and `InputTable`. Buttons are fairly self-explanatory. Internally they work through signals and slots (the `clicked` signal), and they can have different behaviors including changing their label and color when clicked and being disabled or enabled.

Input tables are the two column GUI elements that are everywhere in PyCMDS. The great thing about input tables is that they accept PyCMDS “data type” objects directly. Given an instance of `Number` called `destination`, adding to the input table is as easy as

```
input_table.add('Destination', destination) (5.3)
```

Internally, PyCMDS will do all of the work to make sure that `destination` is displayed in the GUI. The `destination.updated` signal will fire whenever a user manually interacts with the display. Attributes `display` and `disabled` change the behavior of the GUI element.

Vertical scroll areas contain the input tables, and (on the right hand side), tabs contain the vertical scroll

areas. Like the input tables, this tabbed structure is designed to be extended as needed. For example, in the autonomic system, there needs to be a tab for each and every hardware currently loaded by PyCMDS. To accommodate this, the somatic system simply builds a tab for each hardware at PyCMDS startup.

In addition to the parent `Hardware` and `Driver` classes mentioned in the previous section, each hardware type has a `GUI` class, itself a child of the parent `GUI` class. The `GUI` class defines the “ADVANCED” interface that is unique to each hardware (see HARDWARE SECTION). Like everywhere else, inheritance and abstraction are used to minimize unnecessary replication of code. To a first approximation, every delay stage needs the same “ADVANCED” settings as every other delay stage.

PyCMDS uses `pyqtgraph` [99] for interactive plotting. `pyqtgraph` is great because it is optimized for speed and interactivity. Currently only line plots are supported (through PyCMDS’ `Plot1D` class), but 2D plots are supported by `pyqtgraph` and could be added in future versions.

For those wanting to learn more, all GUI components are defined in `PyCMDS/project/widgets.py`.

5.4 Hardware

Hardware are things that 1. have a position, and 2. can be set to a destination. Typically they also have associated units and limits. They sometimes have an offset, as specified by the autonomic system (Section 5.6). Each hardware can be thought of as a dimension of the MR-CMDS experiment, and scans include a specific traversal through this multidimensional space.

In this section I briefly discuss PyCMDS' implementation for each type of hardware.

5.4.1 Hardware inheritance

All hardware classes are children of the parent `Hardware` class (Figure 5.6), which is itself a child of the the global `Hardware` class shown in Figure 5.4. By inspecting Figure 5.6, we can see that all hardware require the following methods:

```
→ close
→ get_destination
→ get_position
→ on_address_initialized
→ poll
→ set_offset
→ set_position
→ @property units
```

Figure 5.7 shows the full inheritance tree, including all nine types of hardware currently supported by PyCMDS. In general the nesting is type/model, although there can be additional levels of nesting when required, as can be seen in the case of OPA/TOPAS/TOPAS-C and OPA/TOPAS/TOPAS-800.

The “kind-defining” classes (`Delay`, `Filter`, `OPA`, `Spectrometer`) are particularly important. Each of these defines *what it means* to be that kind of hardware, as far as PyCMDS is concerned. To put it another way, these classes are minimum viable versions of their hardware kind. Each of these classes can be directly instantiated as a minimum-viable instance, which allows for “virtual” hardware and offline development.

```

class Hardware(pc.Hardware):

    def __init__(self, *args, **kwargs):
        pc.Hardware.__init__(self, *args, **kwargs)
        self.exposed = self.driver.exposed
        for obj in self.exposed:
            obj.updated.connect(self.update)
        self.recorded = self.driver.recorded
        self.offset = self.driver.offset
        self.position = self.exposed[0]
        self.native_units = self.driver.native_units
        self.destination = pc.Number(units=self.native_units, display=True)
        self.destination.write(self.position.read(self.native_units),
                               self.native_units)
        self.limits = self.driver.limits
        self.driver.initialized_signal.connect(self.on_address_initialized)
        hardwarees.append(self)

    def set_offset(self, offset, input_units=None):
        if input_units is None:
            pass
        else:
            offset = wt.units.converter(offset, input_units,
                                         self.native_units)
            # do nothing if new offset is same as current offset
            if offset == self.offset.read(self.native_units):
                return
            self.q.push('set_offset', offset)

    def set_position(self, destination, input_units=None, force_send=False):
        if input_units is None:
            pass
        else:
            destination = wt.units.converter(destination, input_units,
                                              self.native_units)
            # do nothing if new destination is same as current destination
            if destination == self.destination.read(self.native_units):
                if not force_send:
                    return
            self.destination.write(destination, self.native_units)
            self.q.push('set_position', destination)

```

Figure 5.6: Parent class of all hardware. For brevity, methods `close`, `get_destination`, `get_position`, `is_valid`, `on_address_initialized`, `poll`, and `@property` units have been omitted.

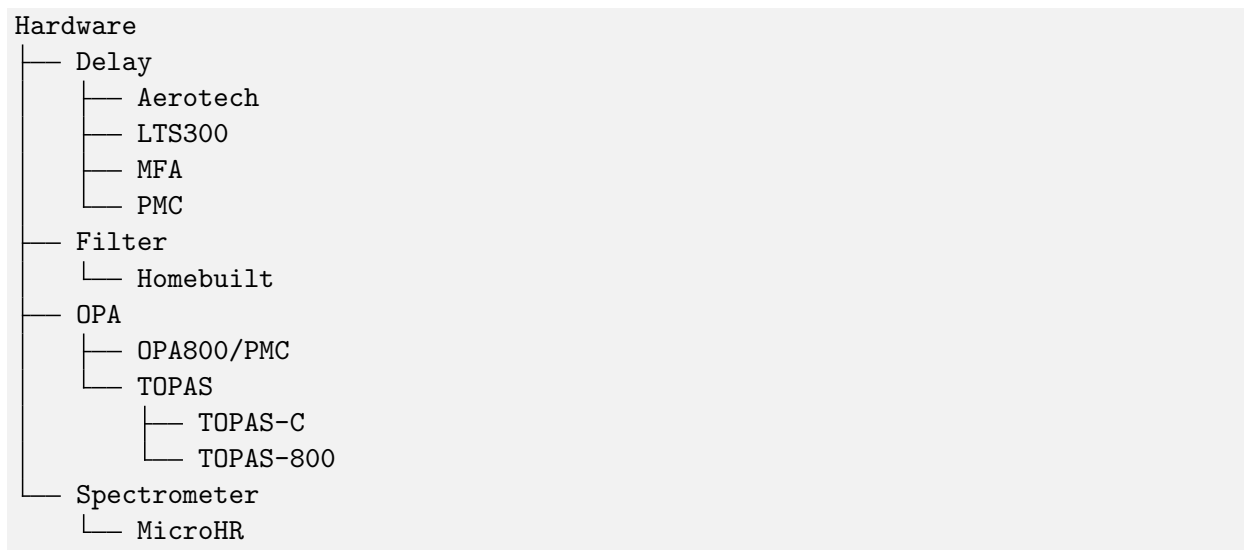


Figure 5.7: Full PyCMDS hardware inheritance tree.

5.4.2 Delays

Delays are the kind of thing that have a position in absolute units, a zero position, and a relative position measured in fs or ps. They can be offset by the autonomic system to accommodate spectral delay correction.

In addition to the inherited methods, delay hardware objects require the following:

→ `set_motor_position`

Delay driver objects require the following:

→ `set_motor_position`

The delay GUI, by default, offers

→ `on_home`

→ `on_set_motor`

→ `on_set_zero`

Figure 5.8 is a screenshot of the advanced panel for one of the Newport MFA [100] delay stages. This advanced panel is pretty typical. Users may directly set the motor position or zero position. They may also change the label. There is a “factor” argument, which can be any non-zero integer (positive or negative). At its simplest this factor can accommodate different directions that the stage is placed, but it can also account for double-passes and similar configurations.

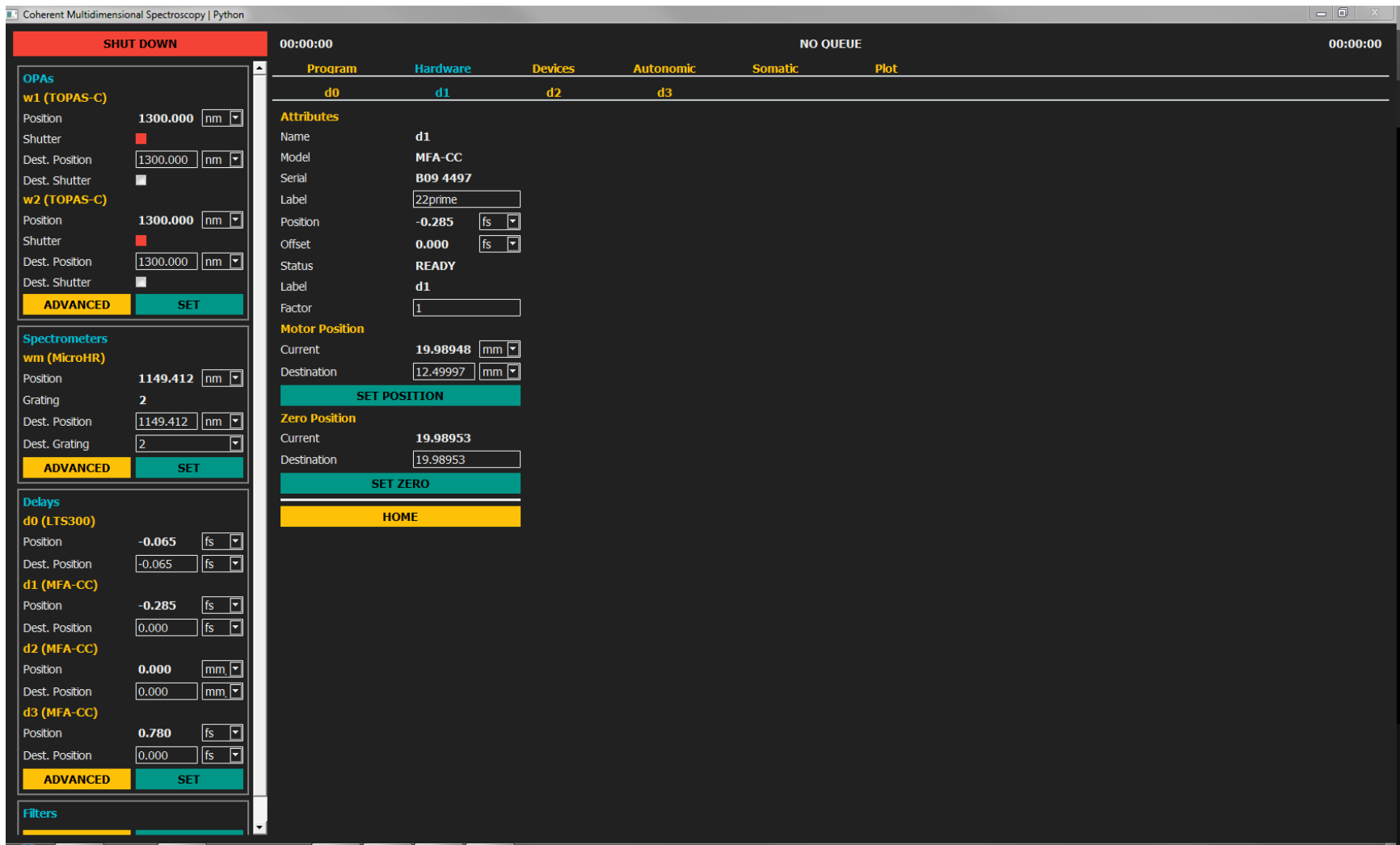


Figure 5.8: Advanced menu for one of the MFA-CC (SMC-100) delay stages, on the fs system.

5.4.3 Spectrometers

Spectrometers are the kind of thing that can be set to a single color (typically native units are in nm). They often have turrets that allow for switching between gratings. Other features are not currently supported as PyCMDS has only needed to drive one monochromator at this time.

Spectrometer hardware instances do not need any features beyond the default hardware object.

Spectrometer driver instances require the following:

- `get_grating_details`
- `set_turret`

Figure 5.9 is a screenshot of the MicroHR advanced menu on the fs system. Nothing can be changed in the advanced menu except the label, and the offset can be seen.

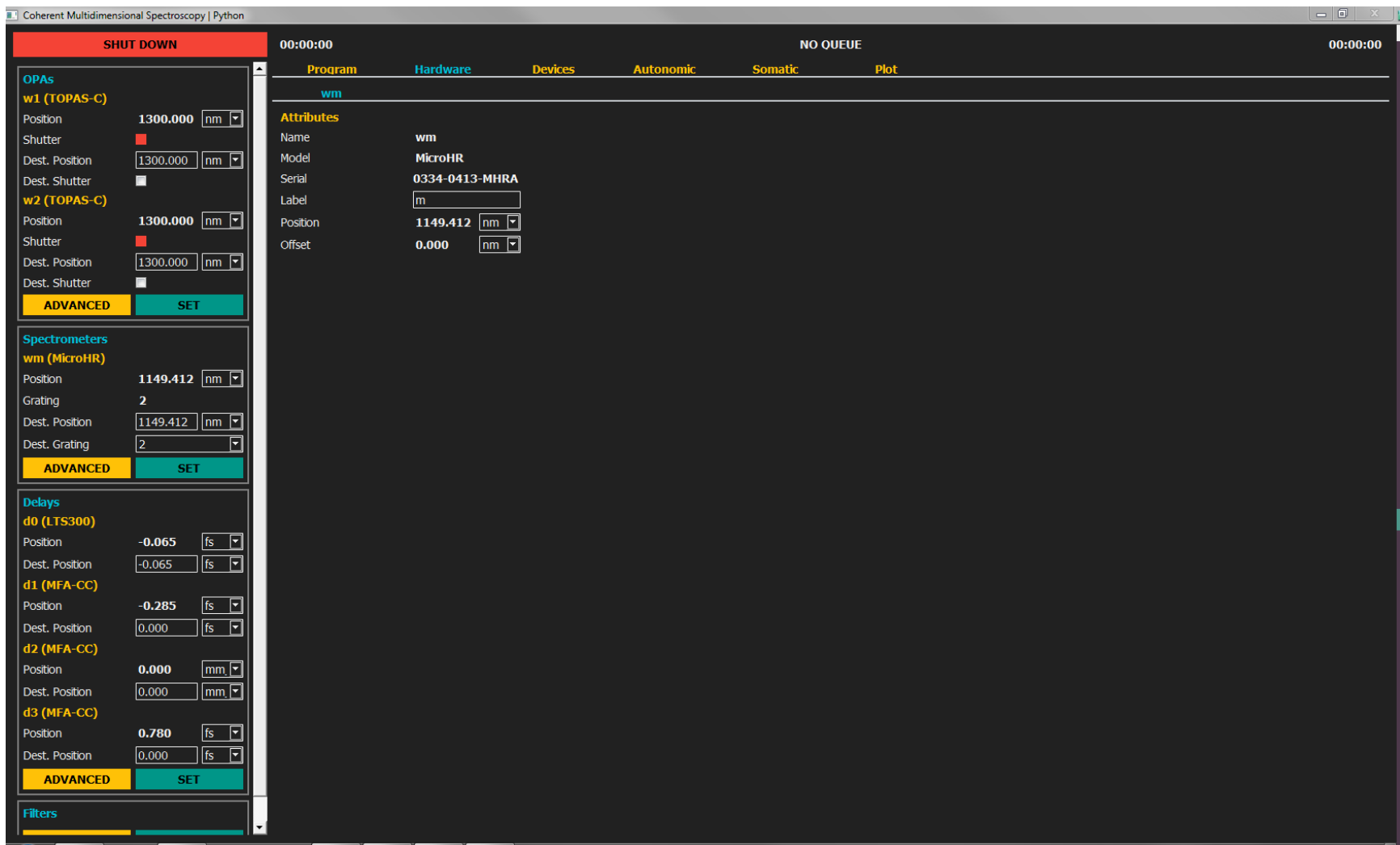


Figure 5.9: Representative spectrometer advanced menu.

5.4.4 OPAs

OPAs are the most complicated piece of hardware addressed by PyCMDS. They range between 3 and 6 motors.

In addition to what's inherited from the parent hardware class, OPA hardware instances require the following:

- `@property curve`
- `@property curve_paths`
- `get_tune_points`
- `home_motor`
- `load_curve`
- `@property motor names`
- `run_auto_tune`
- `set_motor`

In addition to what's inherited from the parent driver class, OPA driver instances require the following:

- `bool poynting_correction`
- `home_motor`
- `home_all`
- `curve`
- `motor_names`
- `set_motor`
- `set_motors`
- `interaction`
- `load_curve`
- `set_position_except`

Many of these additional features have to do with the tuning curve, a crucial feature of OPAs. The tuning curve contains motor positions needed to achieve each valid output color. Read more about my implementation of tuning curves in Chapter 7.

Figure 5.10 is a screenshot of the advanced menu for one of the TOPAS-C OPAs on the fs table. A large central plot displays the currently loaded tuning curve for one of the motors (chosen in the pull down menu). Multiple filepath menus allow the user to specify each tuning curve. Each motor can be independently set and homed.

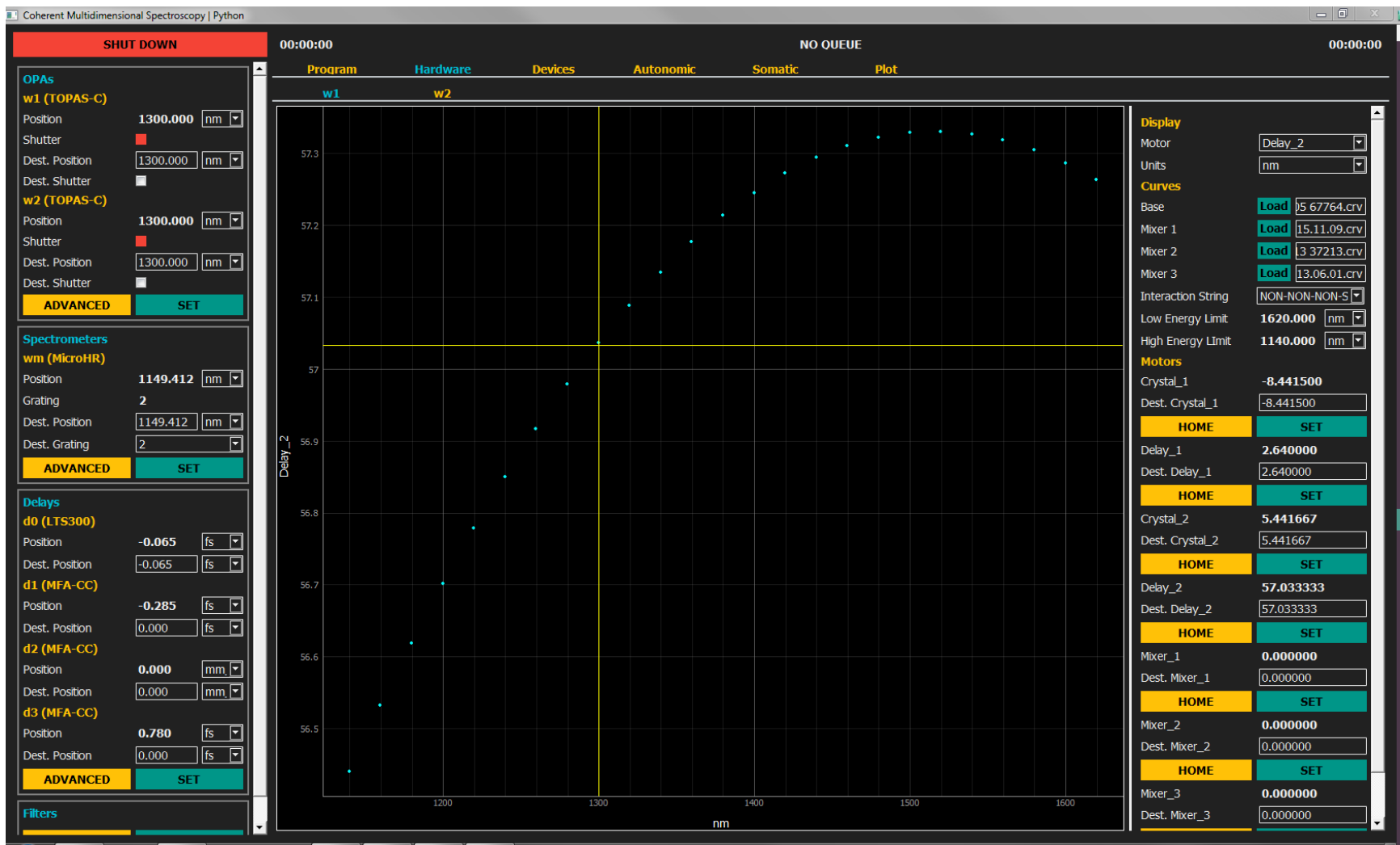


Figure 5.10: Representative OPA advanced menu.

5.5 Sensors (devices)

Sensors are the kind of things that actually measure data. In spectroscopy these can be photodiodes, array detectors, photo-multiplier tubes etc. In this section I describe the strategy that PyCMDS uses to represent sensors.

5.5.1 The DAQ card

The National Instruments PCI-6251 card is capable of eight analog inputs and 1 million samples per second (one sample per microsecond). [101] Both instruments operate at 1 KHz, so that leaves 1000 samples per shot. It is important to be able to configure all of the timings within each shot.

Figure 5.11 shows the GUI designed to control the sample level timing of the NI PCI-6251 card [101]. All 1000 samples for the current shot are displayed in the large central plot. Users may allocate regions of samples to be assigned to particular channels. A second region may be allocated for explicit baseline subtraction. For things like PMTs, make sure these two regions are the same size. For each region, a simple processing method is applied before the baseline is (optionally) subtracted and the signal is (optionally) inverted. All of this results in a single processed value for each channel on each shot. Users may also allocate exactly one sample for chopper monitoring.

Figure 5.12 is a screenshot of the shots GUI. Only one channel is shown, chosen by the pull down menu on the right hand side. Each point is one processed value from all of the samples designated for that particular shot. Because this instrument is using dual chopping at this time, only one out of four shots has a large amount of light, so most shots are near zero in Figure 5.12.

Finally, the shot-level information is passed through a “shots processing” script that is entirely user defined. Figure 5.13 shows a typical shots processing script for a dual chopping experiment. Because the script is entirely user defined, exciting statistical treatments are possible moving forward.

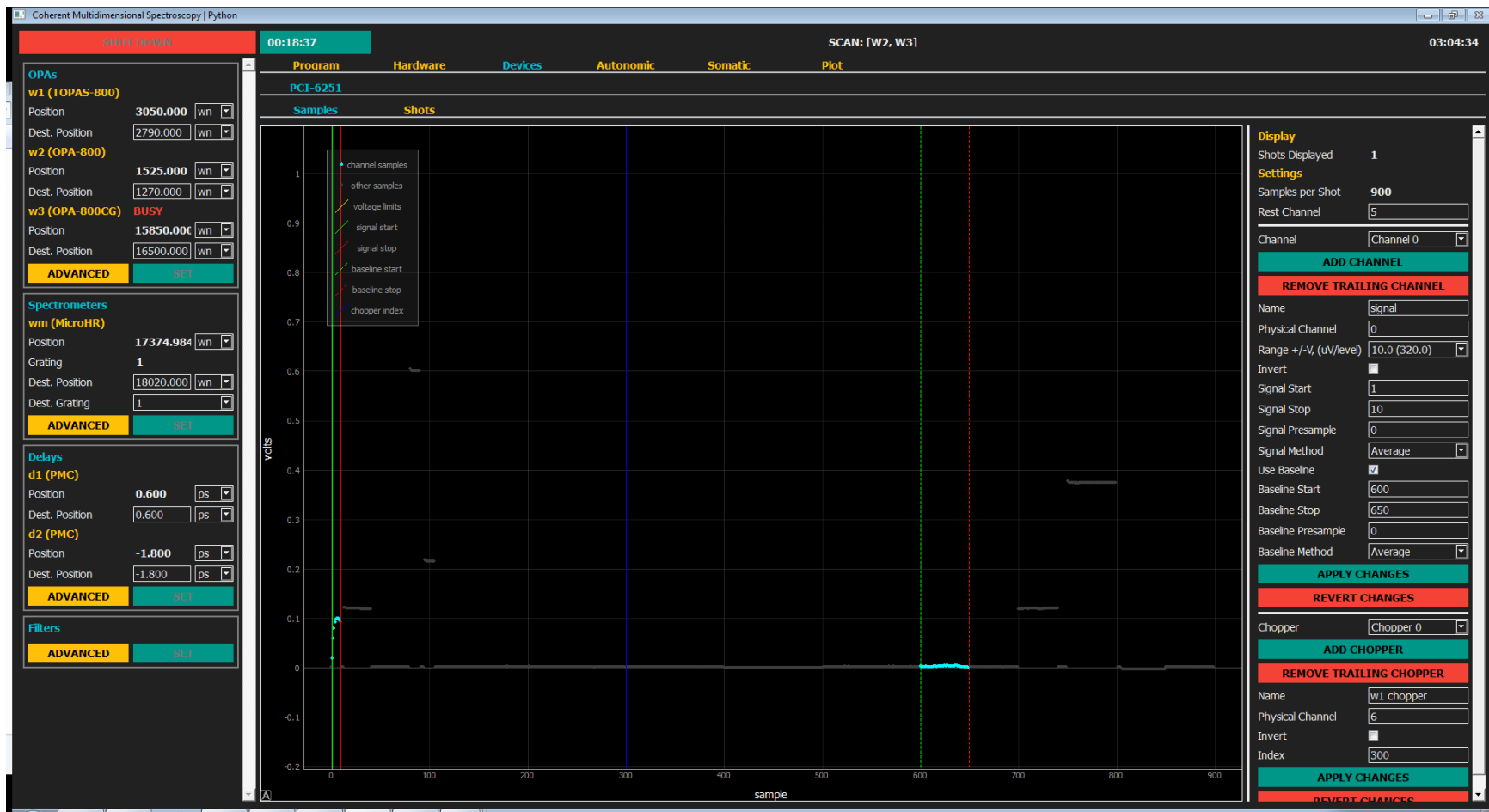


Figure 5.11: Screenshot of PCI-6251 samples tab.

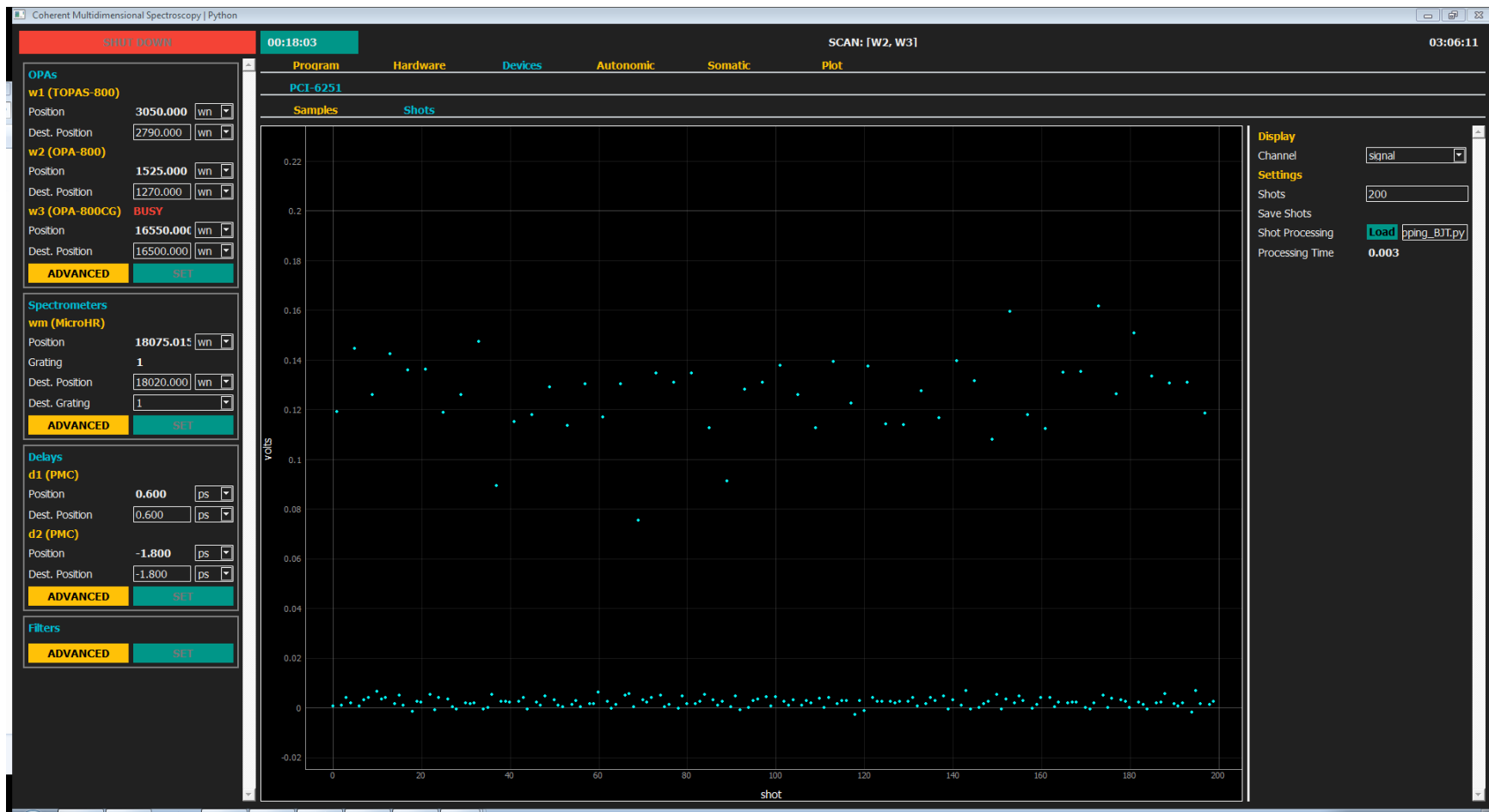


Figure 5.12: PCI-6251 shots tab.

```

import numpy as np

def process(shots, names, kinds):
    channel_indicies = [i for i, x in enumerate(kinds) if x == 'channel']
    out = np.full(len(channel_indicies)+1, np.nan)
    channel_indicies.pop(0)
    out_names = []
    # signal diff
    #           A B C D
    # chopper 1: - + + -
    # chopper 2: - - + +
    # we want A-B+C-D
    c1 = shots[-2]
    c2 = shots[-1]
    a = np.mean(shots[0, (c1==-1)*(c2==-1)])
    b = np.mean(shots[0, (c1==+1)*(c2==-1)])
    c = np.mean(shots[0, (c1==+1)*(c2==+1)])
    d = np.mean(shots[0, (c1==-1)*(c2==+1)])
    out[0] = a-b+c-d
    out_names.append('signal_diff')
    # signal mean
    out[1] = np.mean(shots[0])
    out_names.append('signal_mean')
    # others
    for i in channel_indicies:
        out[i+1] = np.mean(shots[i])
        out_names.append(names[i])
    # finish
    return [out, out_names]

```

Figure 5.13: Simple shots processing script for dual chopping.

5.5.2 Multidimensional sensors

Sensors need not be single valued at each hardware coordinate. Sometimes, a sensor returns an entire array of information. In cases like these, PyCMDS expands the dimensionality of the scan to accommodate the many-valued sensor.

For example, consider Figure 5.14. This simple tune test was taken with an array detector, rather than using a scanning monochromator. Although only one piece of hardware was scanned, the data is considered to be *two-dimensional*, with the second dimension being $\bar{\nu}_a$, the differential color axis for the array vs the OPA setpoint.

The data in Figure 5.14 does not occupy a rectangular region in this parameterization. This is because the range of colors covered at each monochromator setpoint is different, with a smaller dispersion (more colors across the finite array) at higher energies. This relationship is somewhat complex, and requires terms like angle of deviation, focal length, and focal plane tilt to solve. It has been derived previously by Kain [89].

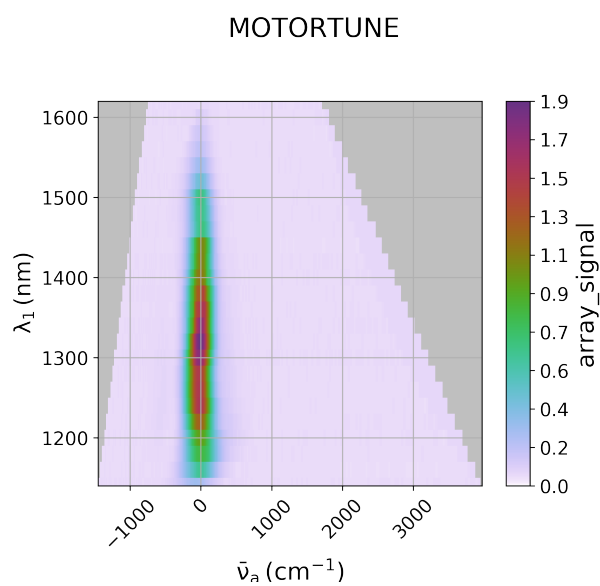


Figure 5.14: Array detector serving as an axis. Data collected 2017-11-06.

5.6 Autonomic

The autonomic system is used to define “reflexes” for PyCMDS—operations that are automatically applied when certain conditions are met. Currently the autonomic system has only functionality: *offset* certain hardware as functional of other hardware’s positions.

The classic example of autonomic offsets comes from spectral delay correction (see section [ACT]). Spectral delay refers to the small delay changes that occur when OPAs change output frequency. To correct for spectral delay, the appropriate delay stages can simply be *offset*.

In PyCMDS, the autonomic system is fully general—that is to say, any hardware can be offset according to any other hardware. This means that spectral delay correction is possible for arbitrarily complex laser setups, and it means that PyCMDS is prepared for corrections that have not yet been fully implemented, such as automated power correction.

Simple plain-text `.coset` files define the offset arrays. They are automatically generated using processing scripts in `attune`. Their headers prevent them from being loaded in the wrong spot. These files are internally represented as instances of the `CoSet` class, which is capable of linear interpolation and extrapolation at the edges.

A single hardware can be offset by multiple other hardwares. In such cases, *offsets always add*.

Figure 5.15 displays the GUI of the autonomic system, in this case for “d1” on the picosecond system. Note the triple-nested tab structure—there are one of these tabs for each hardware loaded by PyCMDS. A large display occupies the middle of the program, displaying the offsets as applied by “w1” in this case. On the right-hand side users can add as many `.coset` files as desired.

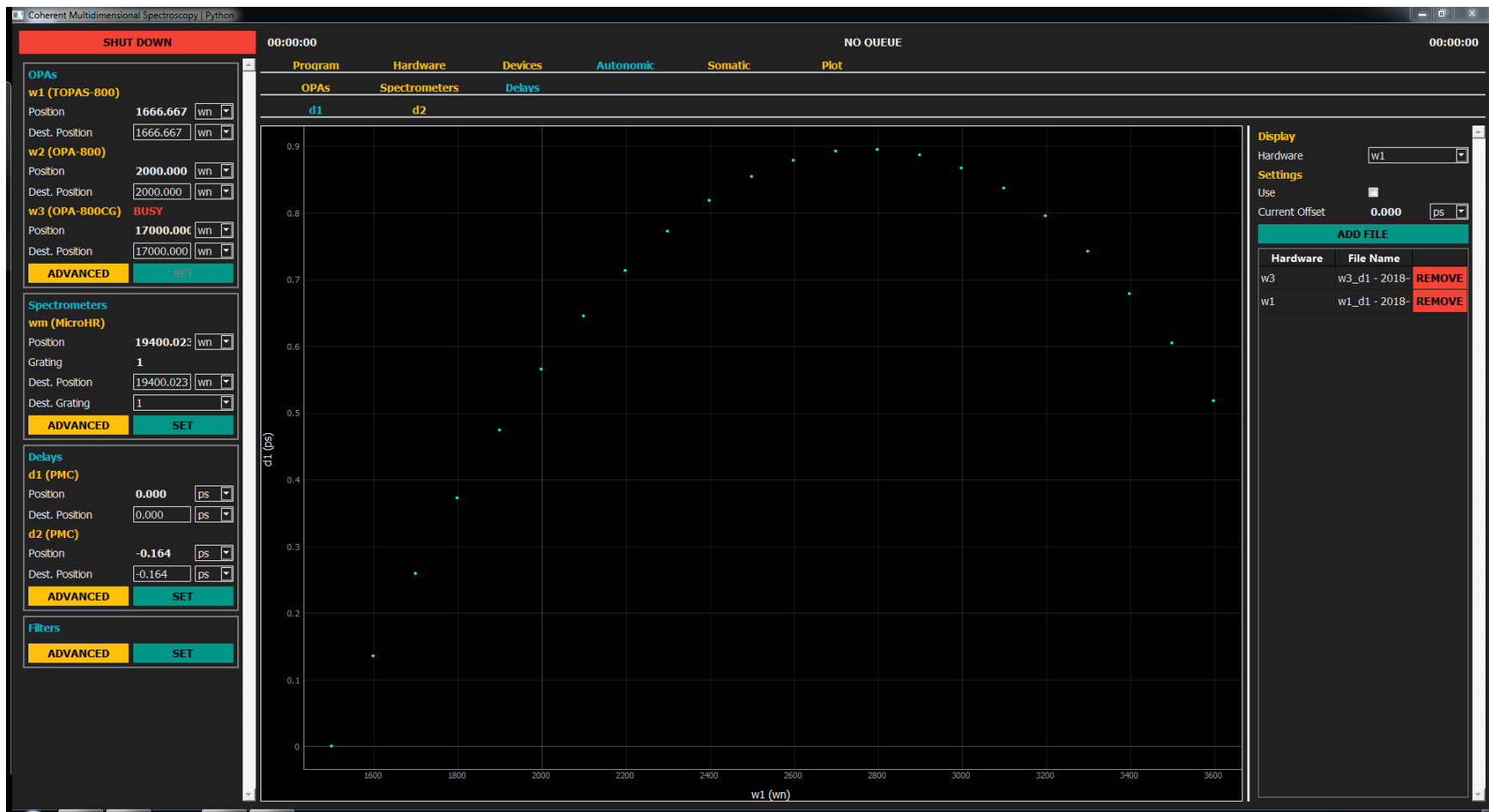


Figure 5.15: Autonomic tab.

5.7 Somatic

In contrast with the autonomic system (Section 5.6), the somatic system is all about voluntary, user specified motion. This is where the fun stuff happens—the acquisitions!

PyCMDS uses the words “scan” and “acquisition” in very careful ways.

- An acquisition is a single user-defined, enqueueable, instruction.
- A scan is a single traversal in the multidimensional hardware space.

Each scan corresponds to one `.data` file, and one WrightTools `Data` instance. There can be many scans within a single acquisition. And there can be many acquisitions in a queue.

PyCMDS saves the data that it is collecting within a nested folder structure `queue/acquisition/scan`. A `.queue` file holds everything needed to recreate the queue, `.aqn` files define each acquisition in plain text and `.data` files hold the multidimensional data itself.

In this section I describe each component of the somatic system in greater detail.

5.7.1 Queue manager

The queue manager keeps track of all enqueued acquisitions, and tells each acquisition when to begin. A singleton `Queue` class lives in the main thread and handles interfacing to the `.queue` plain-text file. When a user appends new acquisitions, or changes their order, the `queue` instance makes sure that those changes are reflected in the GUI and the file.

A special singleton `QueueStatus` keeps track of, well, the queue status: a series of booleans `go`, `going`, `pause`, `paused`, `stop`, and `stopped`. The verb booleans (`go`, `pause`, `paused`), are control flags, to be written by the main thread. The present participle booleans (`going`, `paused`, `stopped`) are flags to be written by the worker thread to indicate status of the current acquisition.

A singleton `Worker` class lives in the acquisition worker thread and carries out the actual operation. The `queue` instance pushes operations to the `worker`, and the `worker` sends a signal that causes `queue.on_action_complete` to be called. If there are more enqueued acquisitions, and

`queue_status.go` is true, the `queue` instance pushes the next operation to the `worker` and the process starts over again.

The queue manager is capable of more than just acquisitions. For example, users can enqueue a wait operation that simply pauses PyCMDS for a certain amount of time or until some condition is met.

Those who wish to learn more can refer to `PyCMDS/somatic/queue.py`.

5.7.2 Scans

Every single scan within PyCMDS is handled by one method of one class, `Worker.scan` from the file `PyCMDS/somatic/acquisition.py`. This method understands how to any scan that PyCMDS can do: any stepwise multidimensional scan with separate hardware and sensors.

The central scan method requires acquisition modules to provide all of the axes (instances of `PyCMDS.somatic.acquisition.Axis`) that need to be broadcast across each-other. Then something very simple happens. For each hardware that will move during the scan, a destinations object (instance of `PyCMDS.somatic.acquisition.Destinations`) is created. This object has the same shape as the full multidimensional scan, and contains the destination for that hardware for that pixel in the scan. Then, a `.data` file is created to accept the data that is about to be collected. A bunch of signals go off, telling PyCMDS that it needs to yield to somatic control. Then PyCMDS simply sits in a loop generated by `numpy.ndindex` [102] and visits each pixel in turn. `ndindex` is a n-dimensional iterator over a given array shape. Written simply, it does the following:

```
def ndindex(shape):
    rs = [range(s) for s in shape]
    pools = map(tuple, rs)
    result = [[]]
    for pool in pools:
        result = [x+[y] for x in result for y in pool]
    for prod in result:
        yield tuple(prod)
```

(5.4)

So that when evaluated:

```
>>> for idx in ndindex((4, 3, 2)): print idx
(0, 0, 0)
(0, 0, 1)
(0, 1, 0)
(0, 1, 1)
(0, 2, 0)
(0, 2, 1)
(1, 0, 0)
...
(3, 2, 0)
(3, 2, 1)
```

(5.5)

This simple algorithm is used to visit each pixel in the entire PyCMDS scan space. At each pixel, something much like the following happens.

```
for idx in ndindex(shape):
    for hardware in hardwares::
        hardware.set(idx)
    for hardware in hardwares:
        hardware.wait_until_still()
    for sensor in sensors:
        sensor.read()
    for sensor in sensors:
        sensor.wait_until_done()
```

(5.6)

The pattern is simple: launch hardware, wait for still, launch sensors, wait for done. The simplicity of this central loop truly shows the power of abstraction in PyCMDS.

5.7.3 Acquisition modules

Acquisition modules are defined interfaces which know how to assemble a scan.

Figure 5.16 shows an aqn file for an acquisition using SCAN.

```
[info]
pymcnds version = '0.8.0-py3'
created = '2018-03-30T20:02:59-05:00'
module = 'SCAN'
name = 'example'
info = 'just an example'
description = 'SCAN: [w1, w2]'

[scan]
axis names = ['w1', 'w2']
constant names = ['wm']

[w1]
start = 1500
stop = 1200
number = 51
units = 'wn'
hardware = ['w1']

[w2]
start = 1500
stop = 1200
number = 35.0
units = 'wn'
hardware = ['w2']

[wm]
hardware = 'wm'
expression = 'w1+2*w2'

[processing]
main channel = 'signal'
process all channels = False

[device settings]
ms wait = 0

[PCI-6251]
use = True
shots = 100
save shots = False

[InGaAs]
use = False
```

Figure 5.16: Representative aqn file.

SCAN

SCAN is by far the most important acquisition module in PyCMDS—it handles acquisitions for almost all of the CMDS experiments that PyCMDS has ever accomplished. SCAN is capable of acquisitions of arbitrary dimensionality. Users simply append as many axes as they want. Acquisitions are done with the trailing (highest index) axis as the innermost loop. Arbitrary expressions (PyCMDS calls them “constants”) are also possible, as can be seen in Figure 5.16.

TUNE TEST

The TUNE TEST module does a simple thing: it sets a chosen OPA to each of the points in its tuning curve and does a monochromator scan of set width about that setpoint. In this way the tune (output color) agreement between the curve and the OPA can be determined. A new point curve with remapped colors is automatically created. See Chapter 7 for more information.

MOTORTUNE

MOTORTUNE is capable of arbitrary acquisitions in OPA motor space. Users are free to choose any set of positions for any set of motors, and PyCMDS will broadcast all of those positions against each-other to form a multidimensional acquisition. Optionally, a leading dimension of *setpoint* may be added, with motor positions being scanned about the points in the old curve.

AUTOTUNE

AUTOTUNE does multiple scans to tune OPAs as automatically as possible. The interface and procedure is different for each model of OPA. The procedures are like those described in Chapter 7.

POYNTING TUNE

POYNTING TUNE is much like AUTOTUNE, except that it deals with Poynting correction instead of regular tuning. See Section 6.3 for more information.

5.8 Conditional validity

The central requirement of the PyCMDS modular hardware abstraction is that experiments can be boiled down to a set of orthogonal axes that can be set separately from other hardware and sensor status. This requirement is loosened by the autonomic and expression systems, such that any experiment could *probably* be forced into PyCMDS, but still the requirement stands—PyCMDS is probably *not* the correct framework if your experiment cannot be reduced in this way. From this we can see that it is useful to talk about the conditional validity of the modular hardware abstraction.

The important axis is hardware complexity vs measurement complexity.

For hardware-complex problems, the challenge is coordination. MR-CMDS is the perfect example of a hardware-complex problem. MR-CMDS is composed of a collection (typically 5 to 10 members) of relatively simple hardware components. The challenge is that experiments involve complex “dances”, including expressions, of the component hardwares.

For measurement-complex problems, the challenge is, well, the measurement. These are experiments where every little piece of the instrument is tied together into a complex network of inseparable parts. These are often time-domain or “single shot” measurements. Such instruments are typically much faster at data acquisition and more reliable. This comes at a price of flexibility: often such instruments cannot be modified or enhanced without touching everything.

From an acquisition software perspective, measurement-complex problems are not amenable to general purpose modular software design. The instrument is so custom that it certainly requires entirely custom software.

Measurements can be neither hardware-complex nor software-complex (simple) or both (expensive). Thus PyCMDS can be proud to try and generalize the hardware-complex part of acquisition software because indeed that is all that can be generalized.

5.9 Integrations

PyCMDS has several *integrations* to assist users.

PyCMDS automatically uploads all collected data to Google Drive. This provides an easy way for remote users to access the newest data, and it provides a data backup in case something goes wrong.

PyCMDS interacts with users through a custom slack bot, the Wright Instrument Technology Communication Handler (WITCH). WITCH automatically posts when scans are done, and responds to simple commands like “status”. This allows users to be aware of what PyCMDS is doing without physically being in the laser lab. Figure 5.17 is a screenshot of a typical interaction with WITCH. Note Kyle Sunden asking for the status of the queue.

These integrations are particularly useful because MR-CMDS scans take so long. When the lasers are working and everything is tuned, PyCMDS is typically *constantly* scanning to get as much use as possible out of the available time.

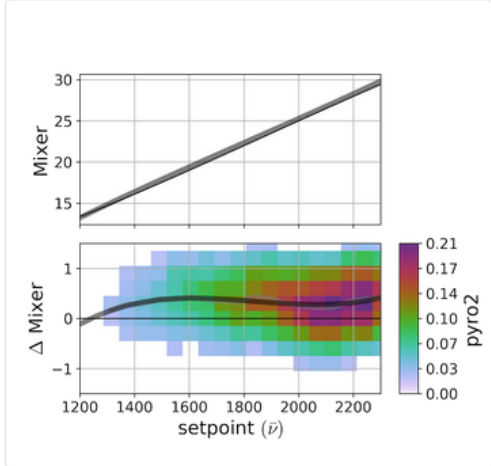
#ps-witch

☆ | 👤 12 | 🗨️ 0 | ➕ Add a topic

Today

witch 7:49 PM
🔥 scan complete - 00:13:21 elapsed

000 [w2, w2_Mixer] (126 kB) ▾



acquisition complete - 005 AUTOTUNE - <https://drive.google.com/open?id=1qDxwOfIaaqfy-9aFAaUp7cmBwamiyoUs>

🔔 queue emptied - total runtime 006:15:00

ksunden 10:21 PM
@witch status

witch 10:21 PM
👤 QUEUE WAITING

name	created
atw3	2018-03-27 18:44:57
runtime	done/total
008:47:18	6/6
url	https://drive.google.com/open?id=1Ild3ijU6WZUQ4rvSG01G01588VO-aAT1
000 - acquisition - AUTOTUNE w3 [BBO]	
started	exited
2018-03-27 18:45:32	2018-03-27 18:55:06

Figure 5.17: Slack screenshot.

5.10 Future directions

5.10.1 Spectral delay correction module

Currently spectral delay correction is done “manually”. Relevant Wigner scans are taken using the SCAN module, processed using WrightTools, and generated coset files are manually applied in the autonomic menu. Ideally, “check” Wigners are then taken to verify the veracity of the corrections. In the future, it would be preferable to have a dedicated SDC module that did all of these things automatically.

It would start from existing corrections (for that table geometry) and iterate until the applied Wigners are all correct.

5.10.2 “Headless” hardware, sensors

The abstraction of hardware complexity that PyCMDS offers is really convenient, but currently these convenient classes can only be used within PyCMDS itself. Since code in PyCMDS is inseparable from the GUI, it is not possible to `import PyCMDS` and use the abstract tools in other programs or scripts. This design is not necessary, and doing the work to free hardware and sensor code from the GUI code would allow for all kinds of creative experiments that are not currently possible within the central conceit of PyCMDS.

5.10.3 Ideal Axis Positions

Frequency domain multidimensional spectroscopy is a time-intensive process. A typical pixel takes between one-half second and three seconds to acquire. Depending on the exact hardware being scanned and signal being detected, this time may be mostly due to hardware motion or signal collection. Due to the curse of dimensionality, a typical three-dimensional CMDS experiment contains roughly 100,000 pixels. CMDS hardware is transiently-reliable, so speeding up experiments is a crucial component of unlocking ever larger dimensionalities and higher resolutions.

One obvious way to decrease the scan-time is to take fewer pixels. Traditionally, multidimensional scans

are done with linearly arranged points in each axis—this is the simplest configuration to program into the acquisition software. Because signal features are often sparse or slowly varying (especially so in high-dimensional scans) linear stepping means that *most of the collected pixels* are duplicates or simply noise. A more intelligent choice of axis points can capture the same nonlinear spectrum in a fraction of the total pixel count.

An ideal distribution of pixels is linearized in *signal*, not coordinate. This means that every signal level (think of a contour in the N-dimensional case) has roughly the same number of pixels defining it. If some generic multidimensional signal goes between 0 and 1, one would want roughly 10% of the pixels to be between 0.9 and 1.0, 10% between 0.8 and 0.9 and so on. If the signal is sparse in the space explored (imagine a narrow two-dimensional Lorentzian in the center of a large 2D-Frequency scan) this would place the majority of the pixels near the narrow peak feature(s), with only a few of them defining the large (in axis space) low-signal floor. In contrast linear stepping would allocate the vast majority of the pixels in the low-signal 0.0 to 0.1 region, with only a few being used to capture the narrow peak feature. Of course, linearizing pixels in signal requires prior expectations about the shape of the multidimensional signal—linear stepping is still an appropriate choice for low-resolution “survey” scans.

CMDS scans often contain correlated features in the multidimensional space. In order to capture such features as cheaply as possible, one would want to define regions of increased pixel density along the correlated (diagonal) lineshape. As a concession to reasonable simplicity, our acquisition software (PyCMDS) assumes that all scans constitute a regular array with-respect-to the scanned axes. We can acquire arbitrary points along each axis, but not for the multidimensional scan. This means that we cannot achieve strictly ideal pixel distributions for arbitrary datasets. Still, we can do much better than linear spacing.

Almost all CMDS lineshapes (in frequency and delay) can be described using just a few lineshape functions:

- exponential
- Gaussian
- Lorentzian
- bimolecular

Exponential and bimolecular dynamics fall out of simple first and second-order kinetics (I will ignore higher-order kinetics here). Gaussians come from our Gaussian pulse envelopes or from normally-distributed inhomogeneous broadening. The measured line-shapes are actually convolutions of the above. I will ignore the convolution except for a few illustrative special cases. More exotic lineshapes are possible in CMDS—quantum beating and breathing modes, for example—I will also ignore these. Derivation of the ideal pixel positions for a single exponential appear below.

Simple exponential decays are typically used to describe population and coherence-level dynamics in CMDS. For some generic exponential signal S with time constant τ ,

$$S(t) = e^{-\frac{t}{\tau}}. \quad (5.7)$$

We can write the conjugate equation to 5.7, asking “what t do I need to get a certain signal level?”:

$$\log(S) = -\frac{t}{\tau} \quad (5.8)$$

$$t = -\tau \log(S). \quad (5.9)$$

So to step linearly in t , my step size has to go as $-\tau \log(S)$.

We want to go linearly in signal, meaning that we want to divide S into even sections. If S goes from 0 to 1 and we choose to acquire N points,

$$t_n = -\tau \log\left(\frac{n}{N}\right). \quad (5.10)$$

Note that t_n starts at long times and approaches zero delay. So the first t_1 is the smallest signal and t_N is the largest.

Now we can start to consider realistic cases, like where τ is not quite known and where some other longer dynamics persist (manifested as a static offset). Since these values are not separable in a general

system, I'll keep S normalized between 0 and 1.

$$S = (1 - c) e^{-\frac{t}{\tau_{\text{actual}}}} + c \quad (5.11)$$

$$S_n = (1 - c) e^{-\frac{-\tau_{\text{step}} \log\left(\frac{n}{N}\right)}{\tau_{\text{actual}}}} + c \quad (5.12)$$

$$S_n = (1 - c) e^{-\frac{\tau_{\text{step}} \log\left(\frac{N}{n}\right)}{\tau_{\text{actual}}}} + c \quad (5.13)$$

$$S_n = (1 - c) \left(\frac{N}{n}\right)^{-\frac{\tau_{\text{step}}}{\tau_{\text{actual}}}} + c \quad (5.14)$$

$$S_n = (1 - c) \left(\frac{n}{N}\right)^{\frac{\tau_{\text{step}}}{\tau_{\text{actual}}}} + c \quad (5.15)$$

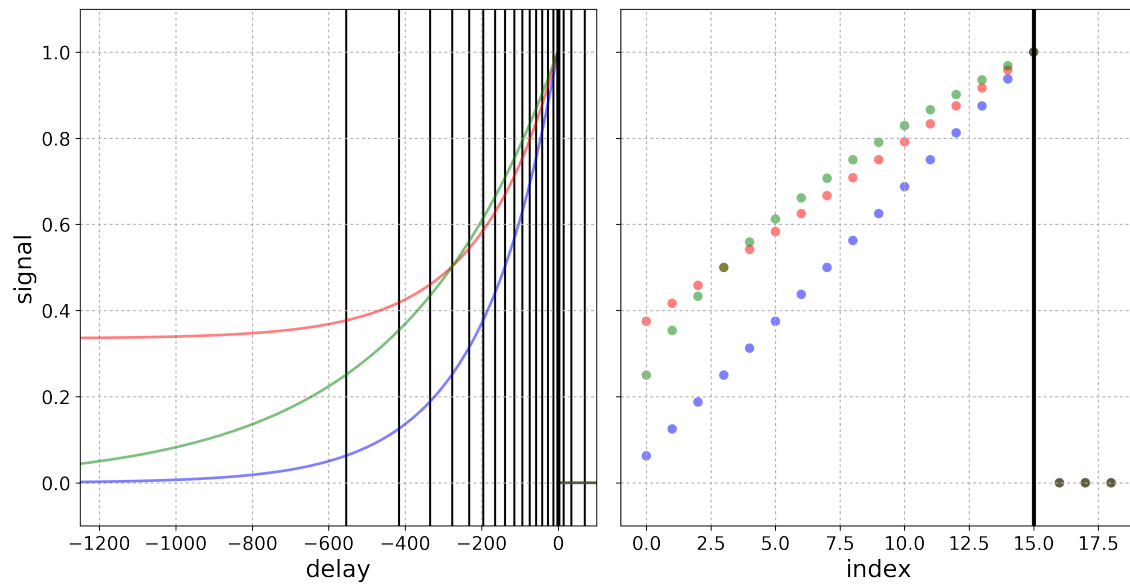


Figure 5.18: Ideal axis positions, exponential decay.

5.10.4 Simultaneous acquisitions

Sometimes PyCMDS needs to do multiple scans that are completely orthogonal. For example, certain OPA tuning operations only require signals from pyroelectric detectors that are each permanently installed to monitor a single OPA. In such cases, theoretically multiple OPAs could be tuned simultaneously. The coordination of such operations would be more difficult than what currently exists, but could be possible within PyCMDS' multithreaded design.

5.10.5 Hotswappable hardware

One of the most important basic capabilities of PyCMDS is the ability to reconfigure itself depending on what hardware is loaded. However controlling the hardware configuration is currently more difficult than it should be. Currently users need to shut down PyCMDS, manually edit INI files to configure which hardware are included, and restart PyCMDS. A better solution would allow users to “hotswap”—load, or drop hardware and sensors without shutting down PyCMDS. The work of allowing this would also allow users to reload hardware, a great fallback option when there are communication issues.

5.10.6 wt5 savefile

The wt5 save format, introduced in WrightTools 3, is a huge improvement over the simple, plaintext data format currently used by PyCMDS. wt5 files are smaller on disk, have richer metadata and self-descriptive properties, and multiple scans can be stored in the same file using `Collections`. Because wt5 files are stored on disk but fully accessible through slicing, PyCMDS could have full access to the scan arrays without risk of memory overflow, which promises to allow for new potential visualizations during acquisition. See Chapter 4 for more information about WrightTools.

Even more exciting, wt5 files can be instantiated as empty and *then* filled. This means that the wt5 file can be a self-describing set of destinations that actually defines which pixels PyCMDS should visit.

Chapter 6

Active correction

*Calibrate,
calibrate,
calibrate,
calibrate,
calibrate,
run.*

– Long-hanging poster in Wright Group laser lab.

6.1 Introduction

MR-CMDS is subject to a number of possible artifacts, many of them stemming from the imperfect nature of the frequency-tunable light sources we currently have. It is self-evidently desirable to correct these artifacts, when possible. Indeed many of these artifacts, such as OPA power, phase mismatch and absorption effects have regularly been corrected for. These corrections are applied after measurement, typically including information from other sources (such as absorption spectra, in the case of absorption effect corrections).

A more interesting class of corrections are “active” corrections—that is, corrections that must be actively applied during acquisition and cannot be applied in post processing. These corrections are more insidious, as they are often neglected because the hardware and/or software does not allow for them.

In this chapter I explore some of these active correction strategies that are useful in the context of MR-CMDS. Some of these strategies have already been implemented, others are partially implemented, and others are still just ideas. I hope to show that active correction is a particularly useful strategy in MR-CMDS.

Section 6.2 addresses spectral delay correction, where automated delay stages are used to explicitly correct for small changes in optical path length at different pulse frequencies.

Section 6.3 addresses poynting correction, where mirrors with motorized pitch and yaw control are used to actively correct for small changes in OPA output poynting.

Section 6.4 addresses (dual) chopping, used to actively subtract artifacts such as scatter and unwanted nonlinear outputs. Chopping can only account for intensity level (additive) artifacts. Fibrillation is the opposite of chopping, as it can only account for amplitude level *interference* effects. Section 6.4 also addresses fibrillation.

6.2 Spectral delay correction

As a frequency domain technique, MR-CMDS requires automated tuning of multiple OPAs. These OPAs generate pulses, which are then manipulated and directed into a sample of interest. Crucially, the relative arrival time of each pulse must be carefully controlled in the context of the MR-CMDS experiment. Unfortunately, changing the output frequency also changes the optical path length, meaning that there is some unavoidable coupling between delay and frequency axes. Because we have full control over delay with our automated stages, we can correct for this phenomenon by choosing a different zero delay *offset* for each OPA output color. This strategy has been dubbed “spectral delay correction”.

Spectral delay correction (SDC) is certainly the oldest active correction strategy employed within the Wright Group. SDC was first implemented by Schuyler Kain within his COLORS acquisition software. [89] COLORS’ implementation was hardcoded for one particular OPA / delay configuration—it wasn’t until PyCMDS that fully arbitrary SDC became possible through the autonomic system (see Section 5.6). Erin Boyle “backported” similar functionality into `ps_control`, although her implementation allowed only for a simple, first order linear correction.

Sometimes, spectral delay can be corrected for after-the-fact. Post processed spectral delay correction can only be done if all relevant delays are scanned during the experiment. A common example is transient absorption, where the single delay axis is typically scanned at each color coordinate. A special method of `Data`, `Data.offset` is designed to do the necessary interpolation for *post hoc* SDC.

In many experiments spectral delay *must* be actively corrected for. Fully coherent experiments are typically performed by scanning OPA frequencies while attempting to keep delays constant. In such experiments, the dataset does not in-and-of-itself contain the information needed to offset in post processing. Indeed it can easily become time-prohibitive to collect the full response. For a three-beam experiment, an entire two dimensional delay-delay collection would be required at each pixel to allow for post-correction.

It has been found that SDC is necessary for each individual scanned OPA, and for each separate path when pulses from a single OPA are split. The difference between different paths is typically small, but enough to move pulses a noticeable amount relative to each-other. For this reason, SDC for split OPAs

is a multidimensional problem, which in principle requires a multi-dimensional acquisition to fully record. In practice, however, these corrections are typically recorded iteratively.

White light sources are also interesting to consider in the context of spectral delay correction. White light is typically quite chirped, with lower frequency (redder) colors traveling ahead of high frequency (bluer) colors. Typically, white light might serve as a “probe” that interacts last with the sample. If a scanning monochromator is used, spectral delay correction could be employed such that each color within the white light pulse arrives at the same delay. If a fully coherent experiment is performed with chirped white light as one of the excitation pulses, the other pulses will provide a gating effect in time that isolates interaction from one frequency in the white pulse. This is similar to the gating that is accomplished using “delay 1” in the TOPAS-C OPAs (see Chapter 7). By gating in this way, a *frequency* axis along the white light dimension could be scanned using a delay stage. COLORS’ has taken this idea to its logical conclusion, with support for “OPAs” that are actually controlled by delay stages, although the idea has not yet been realized in practice.

Figure 6.1 contains two plots that were automatically generated by PyCMDS in the context of an experiment. In this case, the user used a sapphire plate as a nonresonant medium to record the spectral delay dependence. It was a three beam $\omega_1, \omega_2, \omega_2'$ experiment, so three corrections were necessary: D2 vs OPA1, D1 vs OPA2, and D2 vs OPA2. Here we focus only on D2 (τ_{21}) vs OPA1, the simplest of the corrections.

In the left-hand subplot of Figure 6.1 we see the original experiment. Without corrections applied, the user scanned OPA1 vs D2. The curvature in the plot is due entirely to SDC, as sapphire is entirely nonresonant (driven). Using WrightTools, PyCMDS fits each slice to find the delay that gives maximum signal. It then passes those separate fits through a spline to guess the ultimate SDC dependence. PyCMDS makes a best guess in regions where there is not enough signal to determine the appropriate delay, like at 800 nm in the left hand plot. The magnitude of the corrections are roughly 30 fs in this particular experiment: not large, but enough to change signal levels by roughly a factor of 2. In other cases SDC is as much as 200 fs.

In the right-hand subplot the user has taken the same scan again, this time after corrections were applied. The delay traces (horizontal) peaks at the same value for every OPA1 position (vertical).

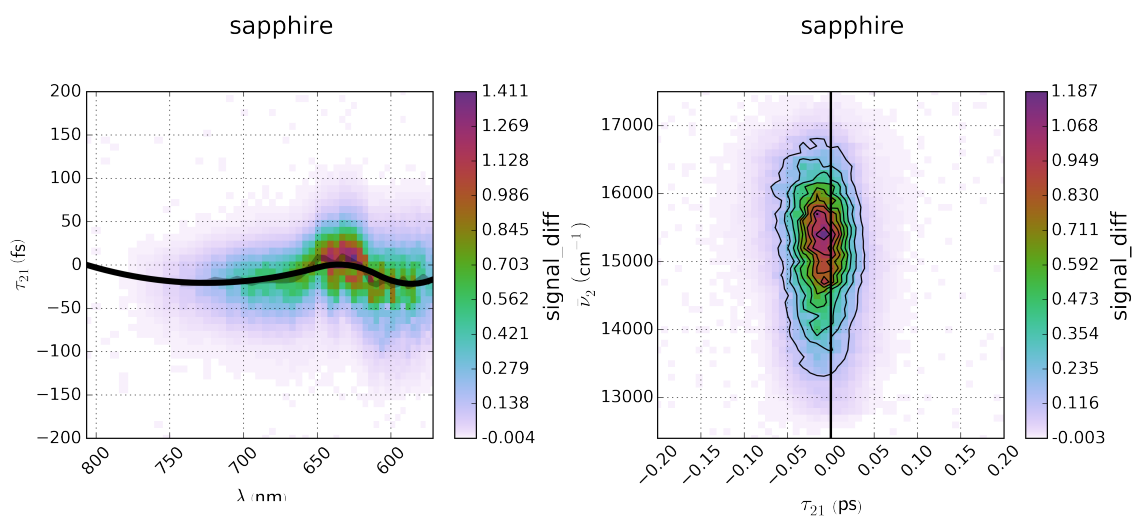


Figure 6.1: Spectral delay correction.

6.3 Poynting correction

With scanning OPAs, output Poynting can change along with optical path length. We now correct for such changes actively using mirrors with motorized pitch and yaw controls. These Poynting corrections are implemented as part of the recursive OPA tuning curve system (see Chapter 7), and not through the autonomic system.

To determine the appropriate positions for a Poynting correction, a pitch vs setpoint and yaw vs setpoint scan is performed. A pinhole is placed at the sample location, and a detector is placed *immediately* after the pinhole.

Figure 6.2 is an autogenerated figure from PyCMDS. It is a typical Poynting correction scan. Like in spectral delay correction, each slice is fit and then a spline is fit to all slices simultaneously. The top plot of Figure 6.2 show the initial Poynting correction in comparison with the new one. In this case the change in the correction is mostly an offset, although there is a fairly dramatic shape change at the lowest energy setpoints.

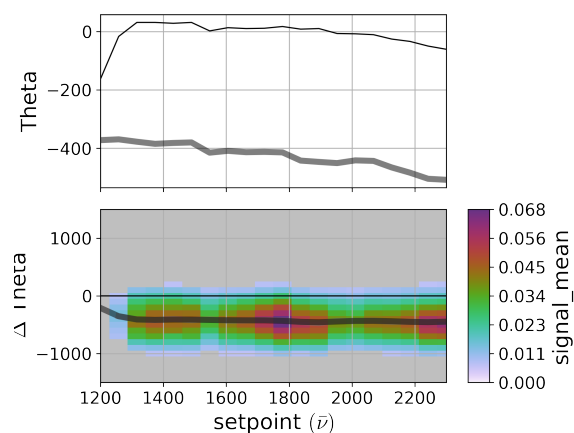


Figure 6.2: A typical example of the output of the Poynting Tune module. The top figure shows the curve for the axis designated Theta. The thin, solid line is the previous curve, the thick transparent line is the new tuning curve. The y-axis is in units of motor steps, the x-axis is in units of wavenumbers. The bottom plot shows the intensity of signal measured for the change in Theta vs. setpoint. Center of mass of each vertical slice are extracted and linearly interpolated to obtain the new calibration curve. This figure was automatically generated by PyCMDS on March 28, 2018 using an OPA-800 generating 1 ps infrared light.

6.4 Chopping

6.4.1 Scatter

Scatter is a complex microscopic process whereby light traveling through a material elastically changes its propagation direction. In CMDS we use propagation direction to isolate signal. Scattering samples defeat this isolation step and allow some amount of excitation light to reach the detector. In homodyne-detected 4WM experiments,

$$I_{\text{detected}} = |E_{4\text{WM}} + E_1 + E_2 + E_{2'}|^2 \quad (6.1)$$

Where E is the entire time-dependent complex electromagnetic field. When expanded, the intensity will be composed of diagonal and cross terms:

$$\begin{aligned} I_{\text{detected}} = & \overline{(E_1 + E_2)E_{2'}} + (E_1 + E_2)\overline{E_{2'}} + |E_1 + E_2|^2 + (E_1 + E_2)\overline{E_{4\text{WM}}} \\ & + (E_1 + E_2)\overline{E_{4\text{WM}}} + \overline{E_{2'}}E_{4\text{WM}} + E_{2'}\overline{E_{4\text{WM}}} + |E_{4\text{WM}}|^2 \end{aligned} \quad (6.2)$$

A similar expression in the case of heterodyne-detected 4WM is derived by Brixner et al. [103]. The goal of any 'scatter rejection' processing procedure is to isolate $|E_{4\text{WM}}|^2$ from the other terms.

Interference Patterns in TrEE

TrEE is implicitly homodyne-detected. Scatter from excitation fields will interfere on the amplitude level with TrEE signal, causing interference patterns that beat in delay and frequency space. The pattern of beating will depend on which excitation field(s) reach(es) the detector, and the parameterization of delay space chosen.

First I focus on the interference patterns in 2D delay space where all excitation fields and the detection field are at the same frequency.

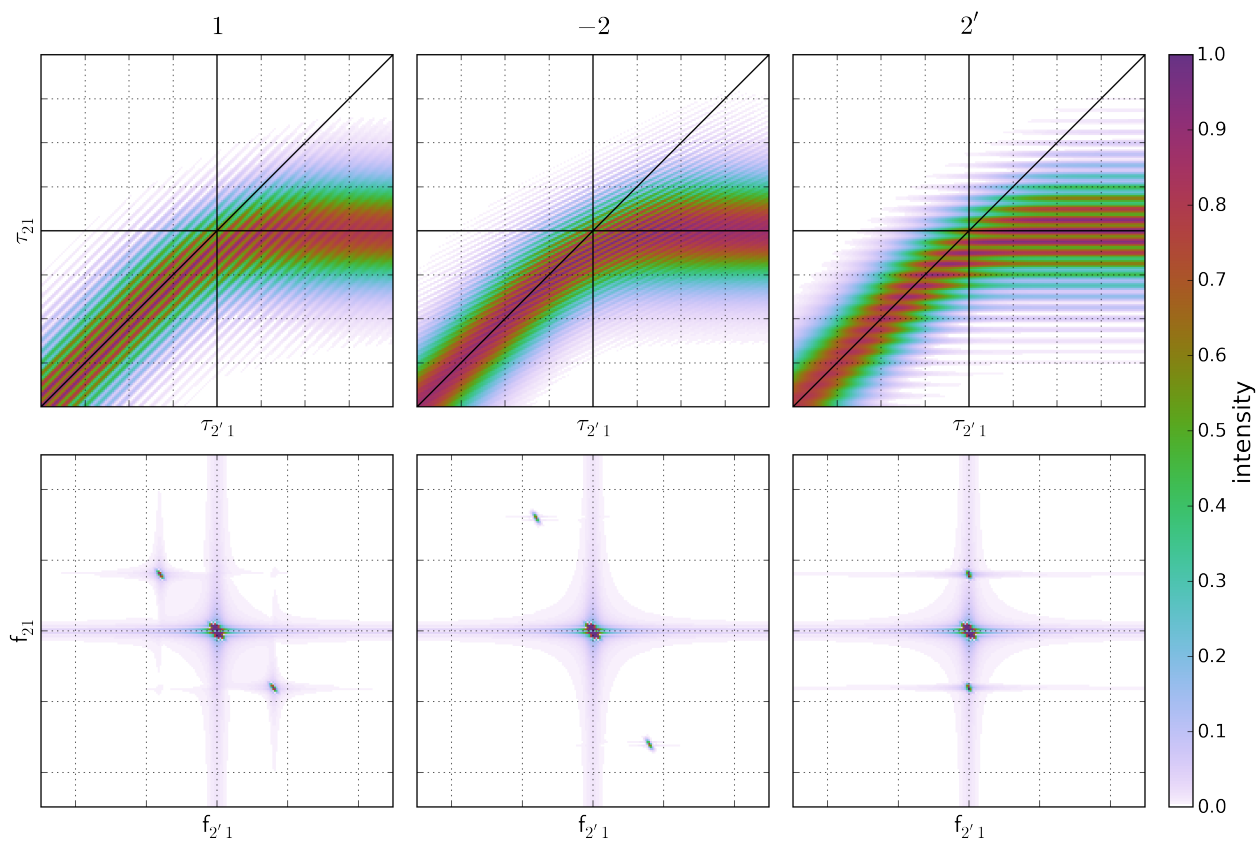


Figure 6.3: Numerically simulated interference patterns between scatter and TrEE for the old delay parametrization. Each column has scatter from a single excitation field. The top row shows the measured intensities, the bottom row shows the 2D Fourier transform, with the colorbar's dynamic range chosen to show the cross peaks.

Here I derive the slopes of constant phase for the old delay space, where $d1 = \tau_{2'1}$ and $d2 = \tau_{21}$. For simplicity, I take τ_1 to be 0, so that $\tau_{21} \rightarrow \tau_2$ and $\tau_{2'1} \rightarrow \tau_{2'}$. The phase of signal is then

$$\Phi_{\text{sig}} = e^{-((\tau_{2'} - \tau_2)\omega)} \quad (6.3)$$

The phase of each excitation field can also be written:

$$\Phi_1 = e^0 \quad (6.4)$$

$$\Phi_2 = e^{-\tau_2\omega} \quad (6.5)$$

$$\Phi_{2'} = e^{-\tau_{2'}\omega} \quad (6.6)$$

The cross term between scatter and signal is the product of Φ_{sig} and Φ_{scatter} . The cross terms are:

$$\Delta_1 = \Phi_{\text{sig}} = e^{-((\tau_{2'} - \tau_2)\omega)} \quad (6.7)$$

$$\Delta_2 = \Phi_{\text{sig}} e^{-\tau_2\omega} = e^{-((\tau_{2'} - 2\tau_2)\omega)} \quad (6.8)$$

$$\Delta_{2'} = \Phi_{\text{sig}} e^{-\tau_{2'}\omega} = e^{-\tau_2\omega} \quad (6.9)$$

Figure 6.3 presents numerical simulations of scatter interference as a visual aid. See Yurs 2011 [104].

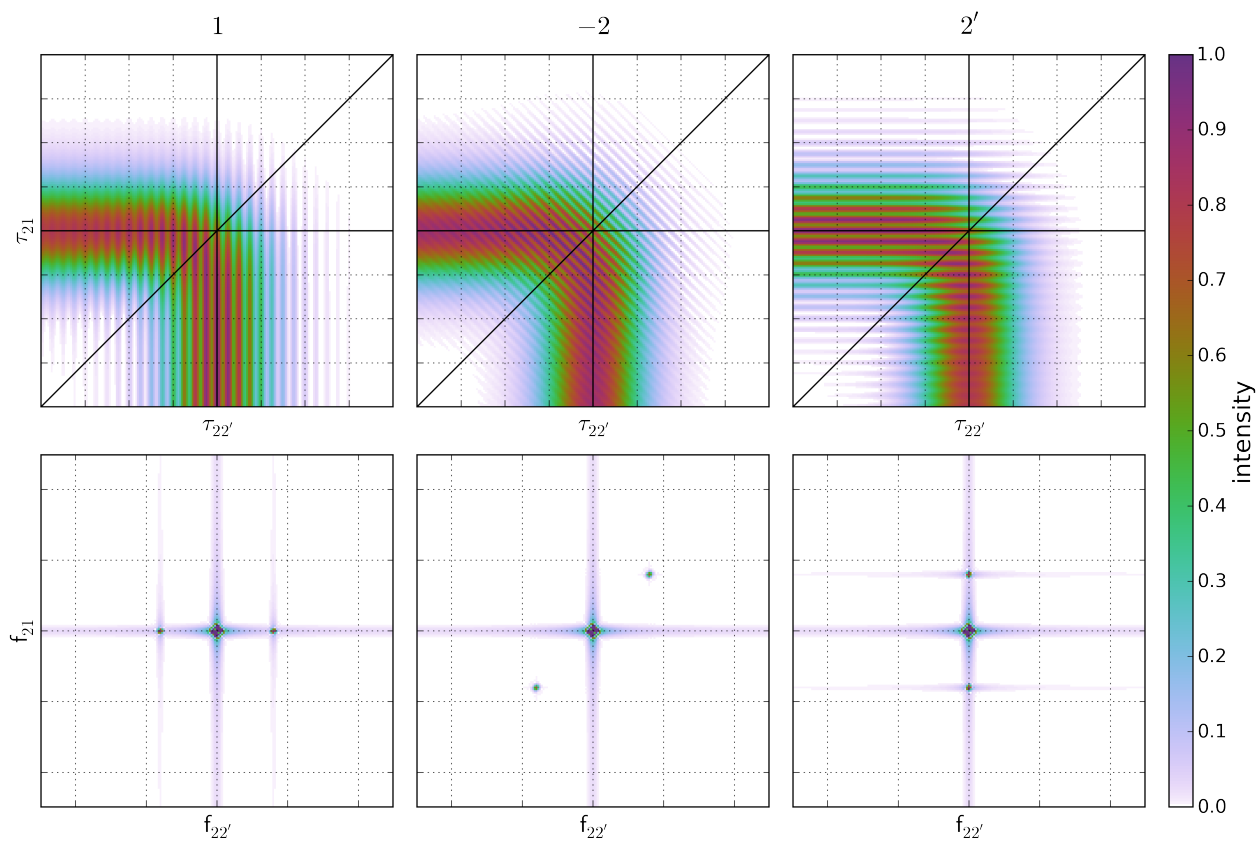


Figure 6.4: Numerically simulated interference patterns between scatter and TrEE for the current delay parametrization. Each column has scatter from a single excitation field. The top row shows the measured intensities, the bottom row shows the 2D Fourier transform, with the colorbar's dynamic range chosen to show the cross peaks.

Here I derive the slopes of constant phase for the current delay space, where $d1 = \tau_{22'}$ and $d2 = \tau_{21}$. I take τ_2 to be 0, so that $\tau_{22'} \rightarrow \tau_{2'}$ and $\tau_{21} \rightarrow \tau_1$. The phase of the signal is then

$$\Phi_{\text{sig}} = e^{-((\tau_{2'} + \tau_1)\omega)} \quad (6.10)$$

The phase of each excitation field can also be written:

$$\Phi_1 = e^{-\tau_1\omega} \quad (6.11)$$

$$\Phi_2 = e^0 \quad (6.12)$$

$$\Phi_{2'} = e^{-\tau_{2'}\omega} \quad (6.13)$$

The cross term between scatter and signal is the product of Φ_{sig} and Φ_{scatter} . The cross terms are:

$$\Delta_1 = \Phi_{\text{sig}} e^{-\tau_1\omega} = e^{-\tau_{2'}\omega} \quad (6.14)$$

$$\Delta_2 = \Phi_{\text{sig}} = e^{-((\tau_2 + \tau_1)\omega)} \quad (6.15)$$

$$\Delta_{2'} = \Phi_{\text{sig}} e^{-\tau_{2'}\omega} = e^{-\tau_1\omega} \quad (6.16)$$

Figure 6.4 presents numerical simulations of scatter interference for the current delay parameterization.

Instrumental Removal of Scatter

The effects of scatter can be entirely removed from CMDS signal by combining two relatively straightforward instrumental techniques: *chopping* and *fibrillation*. [105] Conceptually, chopping removes intensity-level offset terms and fibrillation removes amplitude-level interference terms. Both techniques work by modulating signal and scatter terms differently so that they may be separated after light collection.

	A	B	C	D
signal			✓	
scatter 1		✓	✓	
scatter 2			✓	✓
other	✓	✓	✓	✓

Table 6.1: Four shot-types in a general phase shifted parallel modulation scheme. The 'other' category represents anything that doesn't depend on either chopper, including scatter from other excitation sources, background light, detector voltage offsets, etc.

We use the dual chopping scheme developed by Furuta, Fuyuki, and Wada [106] called ‘phase shifted parallel modulation’. In this scheme, two excitation sources are chopped at 1/4 of the laser repetition rate (two pulses on, two pulses off). Very similar schemes are discussed by Augulis and Zigmantas [107] and Heisler et al. [108] for two-dimensional electronic spectroscopy. The two chop patterns are phase-shifted to make the four-pulse pattern represented in Table 6.1. In principle this chopping scheme can be achieved with a single judiciously placed mechanical chopper - this is one of the advantages of Furuta’s scheme. Due to practical considerations we have generally used two choppers, one on each OPA. The key to phase shifted parallel modulation is that signal only appears when both of your chopped beams are passed. It is simple to show how signal can be separated through simple addition and subtraction of the A, B, C, and D phases shown in Table 6.1. First, the components of each phase:

$$A = I_{\text{other}} \quad (6.17)$$

$$B = I_1 + I_{\text{other}} \quad (6.18)$$

$$C = I_{\text{signal}} + I_1 + I_2 + I_{\text{other}} \quad (6.19)$$

$$D = I_2 + I_{\text{other}} \quad (6.20)$$

Grouping into difference pairs,

$$A - B = -I_1 \quad (6.21)$$

$$C - D = I_{\text{signal}} + I_1 \quad (6.22)$$

So:

$$A - B + C - D = I_{\text{signal}} \quad (6.23)$$

I have ignored amplitude-level interference terms in this treatment because they cannot be removed via any chopping strategy. Interference between signal and an excitation beam will only appear in ‘C’-type shots, so it will not be removed in Equation 6.23. To remove such interference terms, you must *fibrillate* your excitation fields.

An alternative to dual chopping is single-chopping and ‘leveling’. This technique was used prior to May 2016 in the Wright Group. ‘Leveling’ and single-chopping is also used in some early 2DES work. [103].

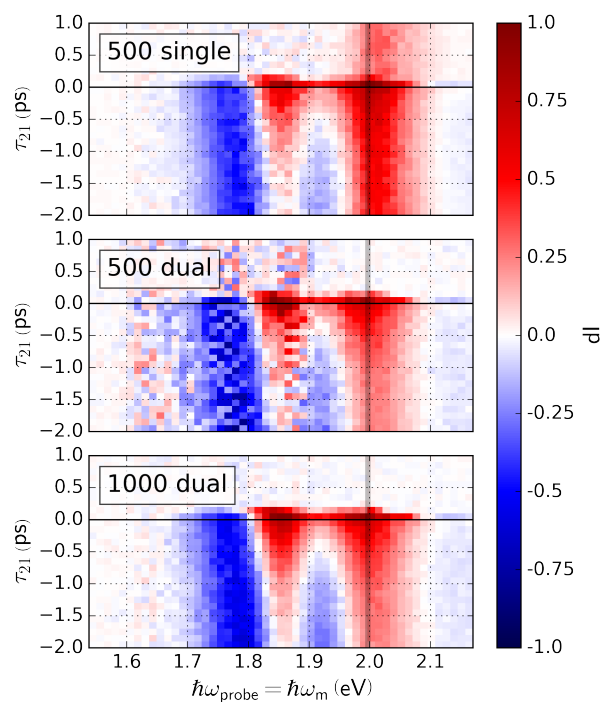


Figure 6.5: Comparison of single and dual chopping in a MoS₂ transient absorption experiment. Note that this data has not been processed in any way - the colorbar represents changes in intensity seen by the detector. The grey line near 2 eV represents the pump energy. The inset labels are the number of laser shots taken and the chopping strategy used.

Figure 6.5 shows the effects of dual chopping for some representative MoS₂ TA data. Each subplot is a probe wigner, with the vertical grey line representing the pump energy. Note that the single chopper passes pump scatter, visible as a time-invariant increase in intensity when the probe and monochromator are near the pump energy. Dual chopping efficiently removes pump scatter, but at the cost of signal to noise for the same number of laser shots. Taking twice as many laser shots when dual chopping brings the signal to noise to at least as good as the original single chopping.

6.4.2 Fibrillation

Fibrillation is the intentional randomization of excitation phase during an experiment. Because the interference term depends on the phase of the excitation field relative to the signal, averaging over many shots with random phase will cause the interference term to approach zero. This is a well known strategy for removing unwanted interference terms [109, 110].

Chapter 7

A robust, automated strategy to collect high quality OPA tuning curves

Principle design features of the new EVV 2DIR optical delivery system include the following:

- Pairs of motorized gimbal mount mirrors on each OPA to compensate beam pointing changes.*
- Automated calibration of OPAs, delay stages and motorized mounts.*

– Paul Donaldson, “Improving ... EVV 2DIR Spectroscopy” (2007) [111]

7.1 Introduction

In frequency-domain Multi-Resonant Coherent Multidimensional Spectroscopy (MR-CMDS), automated Optical Parametric Amplifiers (OPAs) are used to actively scan excitation color axes. [23] To accomplish these experiments, exquisite OPA performance is required. During the experiment, motors inside the OPA move to pre-recorded positions to optimize output at the desired color. Parametric conversion (“mixing”) strategies are now readily available, extending the 800 nm pumped OPA tuning range into the visible, near-infrared, and mid-infrared.

OPAs are very sensitive to changes in upstream lasers and lab conditions, so OPA tuning is regularly required. Manual OPA tuning can easily take a full day. Furthermore, manual tuning typically results in inferior tuning curves, since it is difficult for humans to consider all available information simultaneously. Automated OPA tuning makes OPA upkeep easier, faster and more reproducible, facilitating higher throughput, higher quality frequency domain experiments. The major challenges in automated OPA tuning are:

1. Expensive to take high resolution data.
2. Need smooth curves for interpolation, especially at edges where output is low.
3. Optimization metrics are not necessarily separable along motor dimensions.

In this chapter I describe my strategy for automatically collecting high resolution OPA tuning curves. While I have strategies for all four kinds of OPAs used in the Wright Group, I focus on the femtosecond TOPAS-C models because they are by far the most challenging model to calibrate.

7.2 Curves

OPA tuning curves are the functional correspondence between desired output color and motor positions. In theory, these could be recorded as analytical functions derived in an *ab initio* way from known phase matching and dispersion relations. In practice, ideal tuning curves are determined empirically by simply monitoring OPA output at a series of given motor combinations. This practice of seeing how OPA output depends on motor positions is called “tuning” the OPA.

I have defined a Python class `Curve` which acts as a interface to OPA tuning curves. Within the class, a series of discrete OPA output colors (“setpoints”) are defined, and the motor positions are defined for each setpoint. Since it is important that OPAs be settable to *any* position within their output range, a one-dimensional interpolator is used to determine the correct motor positions for *any* valid color. There are three kinds of interpolators, linear, spline and polynomial. The particular interpolator used depends on the model of OPA and the complexity of its tuning curve. The method `Curve.get_motor_positions` abstracts away this complexity, simply returning a list of motor positions for the desired color(s).

OPAs often use multiple “stages” of interaction to create the desired output. For example, an OPA might generate signal and idler in a first stage, then send that signal on to be doubled in a second “second harmonic signal” (SHS) stage. Depending on the experiment being performed, different stages of the OPA will be used. One could approach this complexity by simply creating an entirely separate curve for each combination of stages, but this would result in the same information being duplicated in many different curves. Instead, I have chosen to use a nested approach that directly reflects the approach that the hardware uses. Curve objects can have “subcurves” which define the behavior of the proceeding stage. In the example above, the parent curve would control the second harmonic signal stage. For each SHS position, the parent would define a desired signal color for the first stage to create. This is passed to the subcurve, which defines the motor positions needed in the first stage to achieve optimal conversion at the desired *signal* color. In this way, each stage can be tuned separately and the tuning of an upstream stage is immediately propagated to all downstream stages.

7.3 TOPAS-C

The TOPAS-C is a popular commercially available motorized OPA. It consists of a large initial stage where signal and idler are generated, and a series of optional mixing stages where further up- or down-conversion can occur to widen the total range of output frequencies. Figure 7.1 shows all of the possible output ranges of the TOPAS. It ranges from the mid infrared (accessible through difference frequency generation) to the ultraviolet (accessible through multiple second harmonic upconversion).

Figure 7.2 diagrams the internals of the TOPAS-C initial stage, where signal and idler are generated. Upon entering the OPA, roughly 98% of pump light is split off immediately (BS1). The remaining 2% goes on to be split again (BS2). After being attenuated further and passed through an aperture, part of this 800 nm light is sent into a sapphire plate to generate white light. This white light is then intentionally chirped, and mixed with the other small portion of the pump `non-collinearly` in NC1. The angle of the crystal is tuned, as is the relative arrival time of chirped white light and the small pump portion. These two degrees of freedom control the efficiency of conversion at a given color in C1, and together they make up the “preamp”. I describe my strategy for preamp tuning in Section 7.4.

The signal portion from the preamp is picked off and meets the gigantic 98% portion of pump split off at the very beginning in NC2. Again, the relative arrival time and crystal angle are motorized internally. Together these degrees of freedom make up the “poweramp”. I describe my strategy for poweramp tuning in Section 7.5.

After the poweramp, the output signal and idler can be sent through appropriate filters and, optionally, mixed further in three subsequent mixing stages to create all of the ranges seen in Figure 7.1. Each of these mixing stages has only crystal angle tunability. I describe my strategy for mixer tuning in Section 7.6.

It is important to realize that the total conversion efficiency for each output color varies wildly over all of the different mixing strategies. Figure 7.3 shows the empirical-best output energy achievable for each setpoint.

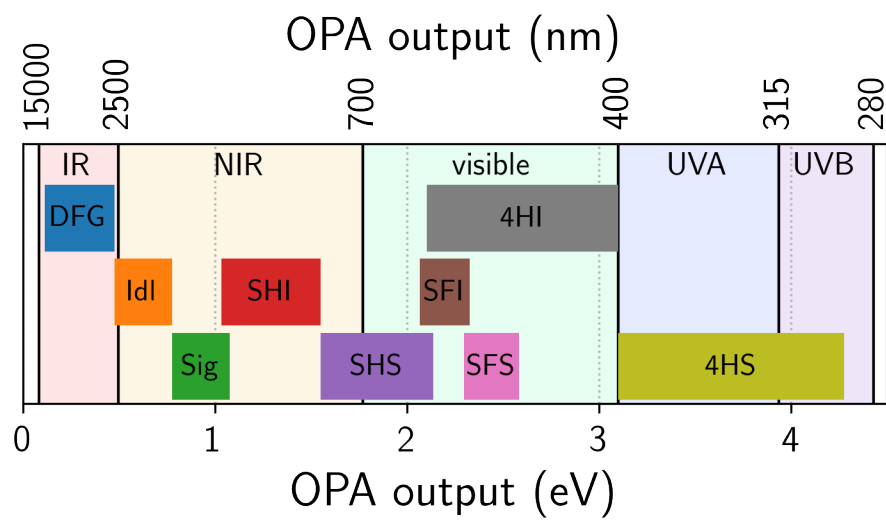


Figure 7.1: TOPAS-C interaction ranges.

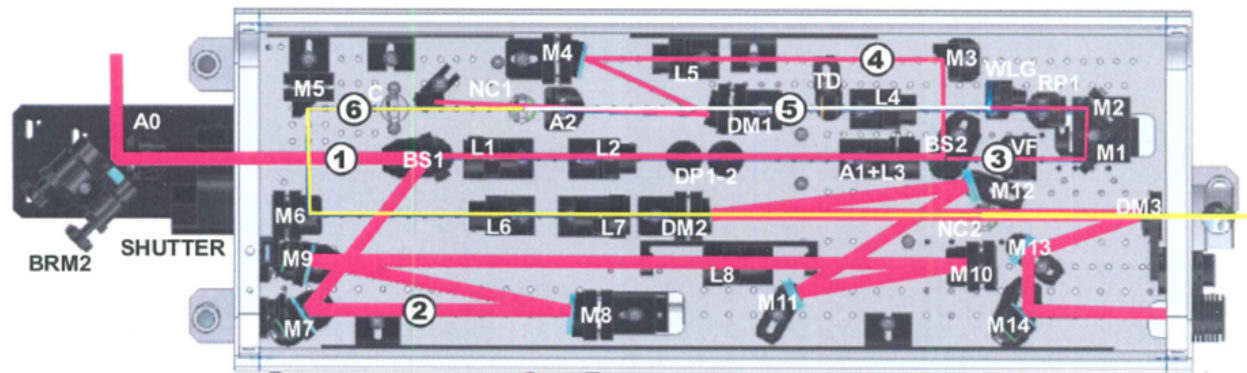


Figure 7.2: TOPAS-C internal optics and beam path. Image taken from manual, originally generated by Light Conversion.

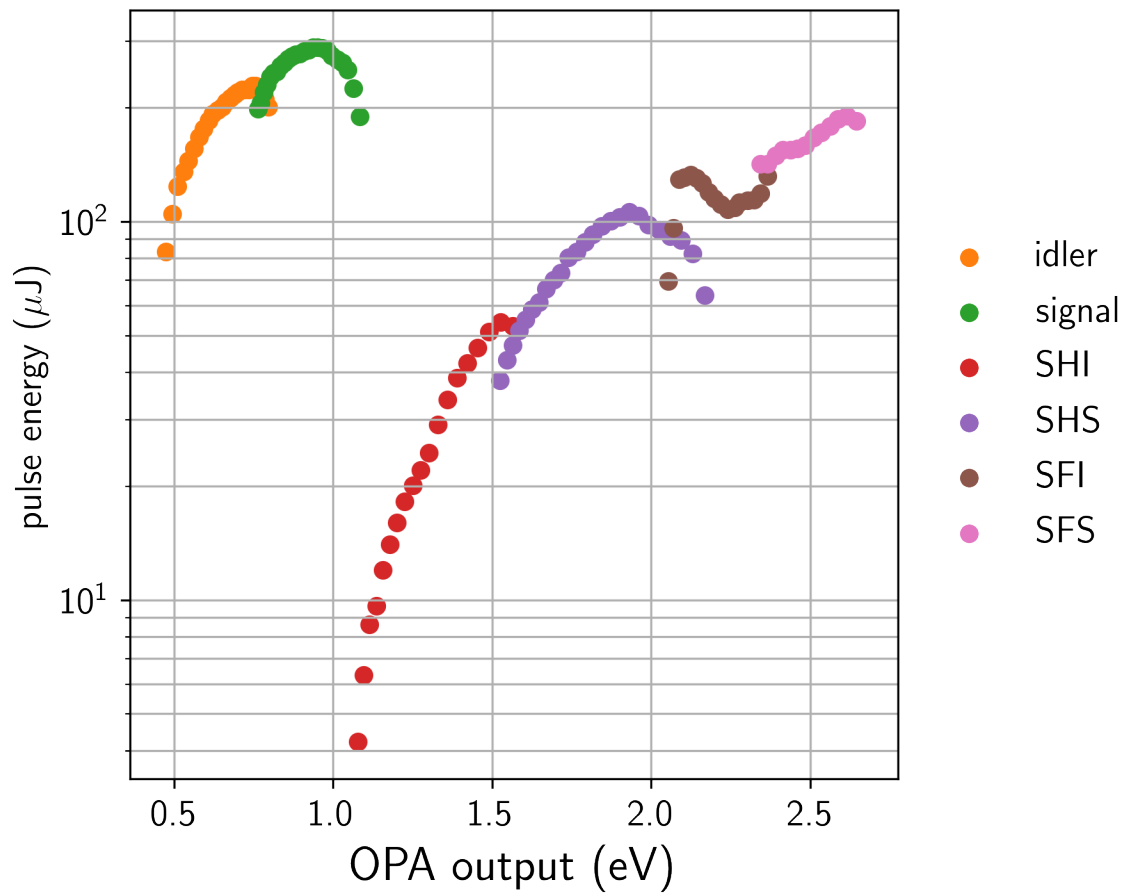


Figure 7.3: TOPAS-C interaction range output powers.

7.4 Preamp

In TOPAS-C OPAs, a small portion of input light is used to generate a signal seed in a BBO crystal “C1”. A motorized delay stage “D1” is used to temporally overlap a particular color in chirped white light with 800 nm pump. C1 angle is tuned to optimize phase matching.

Measured seed intensity and color for all combinations of C1 and D1 position are shown in Figure 7.4. Crucially, output color and intensity are not separable along the preamp motor axes. We are obligated to use a multidimensional fitting strategy to find the best preamp motor positions at each setpoint.

Luckily we have an InGaAs near-infrared array detector, so it is very quick to capture the entire output spectrum at each motor position. PyCMDS visits an entire series of (C1, D1) positions, scanning D1 about the prior best position for each C1 in the curve.

Figure 7.5 diagrams the preamp processing procedure in its entirety. The original dataset is three-dimensional in C1, D1, color. In the first step, the dimensionality is reduced by fitting each array slice to extract a center, amplitude and width. These fits are interpolated to find contours of constant output color. I then search along that contour in *intensity* space to find the motor positions that give maximum intensity for that color. Finally I fit a smooth spline through those chosen values to generate the output curve.

A representative preamp tune procedure output image is shown in Figure 7.6. This is an automatically generated image from PyCMDS. The thick black line is the final output curve. The dark grey lines are the contours of constant color. Each contour of constant color is marked with the output color in nanometers. The colorbar shows the Delaunay-interpolated intensity values for each motor position.

Preamp tuning takes less than 20 minutes, in large part due to a NIR array detector which collects the full spectrum at each motor position.

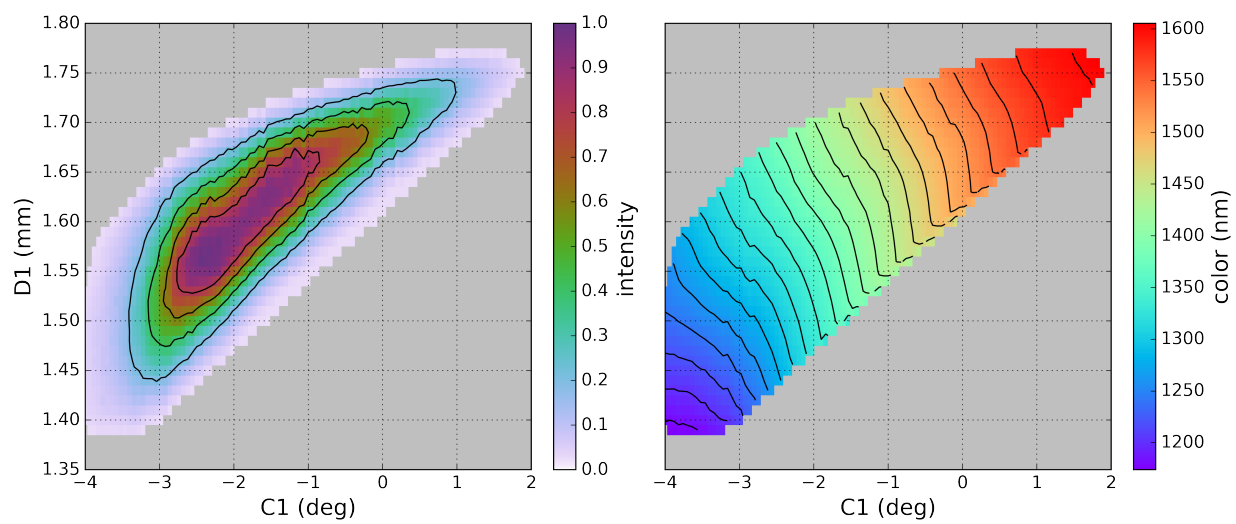


Figure 7.4: TOPAS-C preamp motortune.

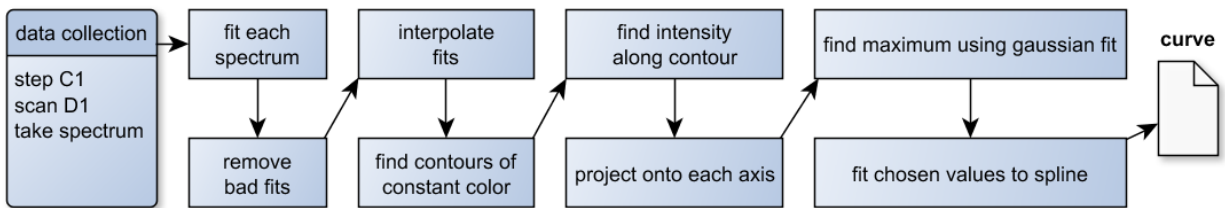


Figure 7.5: Preamp tune procedure flowchart.

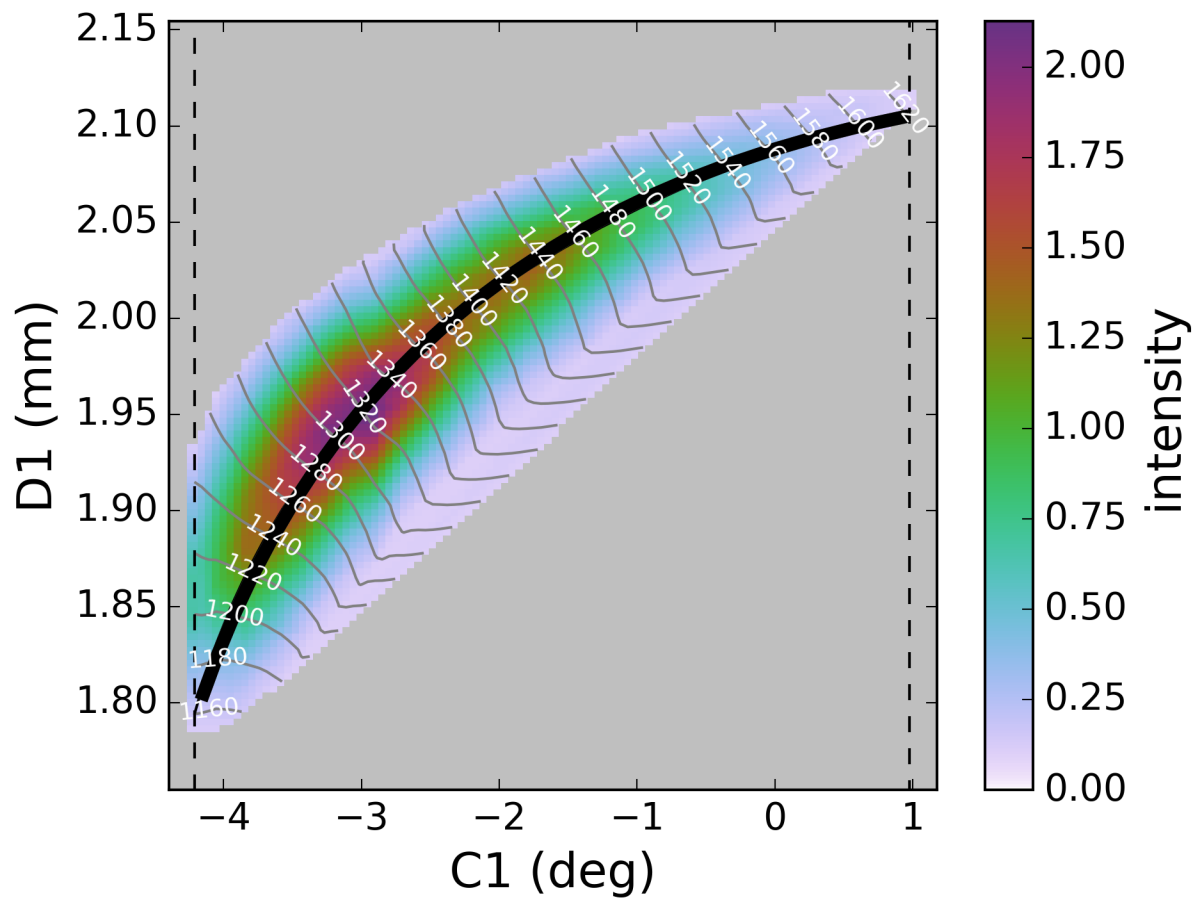


Figure 7.6: Preamp tuning output.

7.5 Poweramp

Once generated, the seed goes on to be amplified in a second BBO crystal “C2” with the rest of the 800 nm pump. Optimizing this amplification step is primarily a matter of setting C2 angle. A small delay correction “D2” is necessary to account for dispersion in the seed optics. To fully explore poweramp behavior, we need to take a C2-D23 scan for each seed color. Measured output intensity and color in this 3D space is represented in Figure 7.7. Note that the motor axes are scans about the previously recorded tuning curve value.

The best position (zero displacement along both axes) is chosen to maximize output intensity while keeping the output color identical to the seed color. Optimizing for zero detuning rather than simply for output intensity has led to better OPA performance and stability. Like in the preamp case, color and intensity are not fully separable along the poweramp motor dimensions (this is especially true at the edge output colors). In the poweramp, the increased dimensionality makes it too expensive to do a full multidimensional tuning procedure. Instead we employ an iterative procedure as diagrammed below. Figure 7.8 diagrams this iterative procedure.

We always end the iteration(s) with C2 so that the OPA's color calibration is as good as possible. Typically only one iteration is required but multiple iterations may be necessary if dramatic OPA realignment has occurred. In total, poweramp tuning typically takes less than 1 hour. Representative procedure output images for D2 (Figure 7.9) and C2 (Figure 7.10) are shown.

For the D2 figure, the lower panel shows the intensity of the data taken. Note the thick grey line, which represents the chosen points before the final spline step. The top panel compares the old tuning curve (thin) with the output tuning curve (thick). For the C2 image, the bottom panel represents the color of each fit mapped onto detuning. Each separate marker color represents a different setpoint. As with D2, the C2 upper panel compares the old tuning curve (thin black) with the output tuning curve (colored X's).

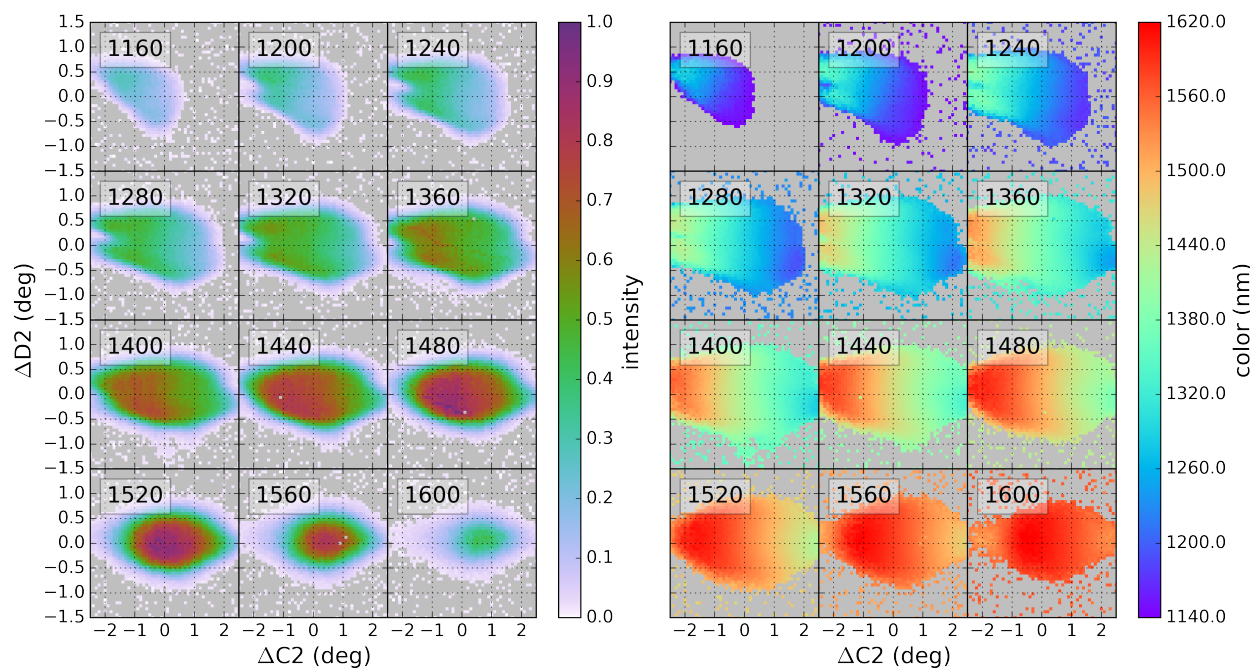


Figure 7.7: TOPAS-C poweramp motortune.

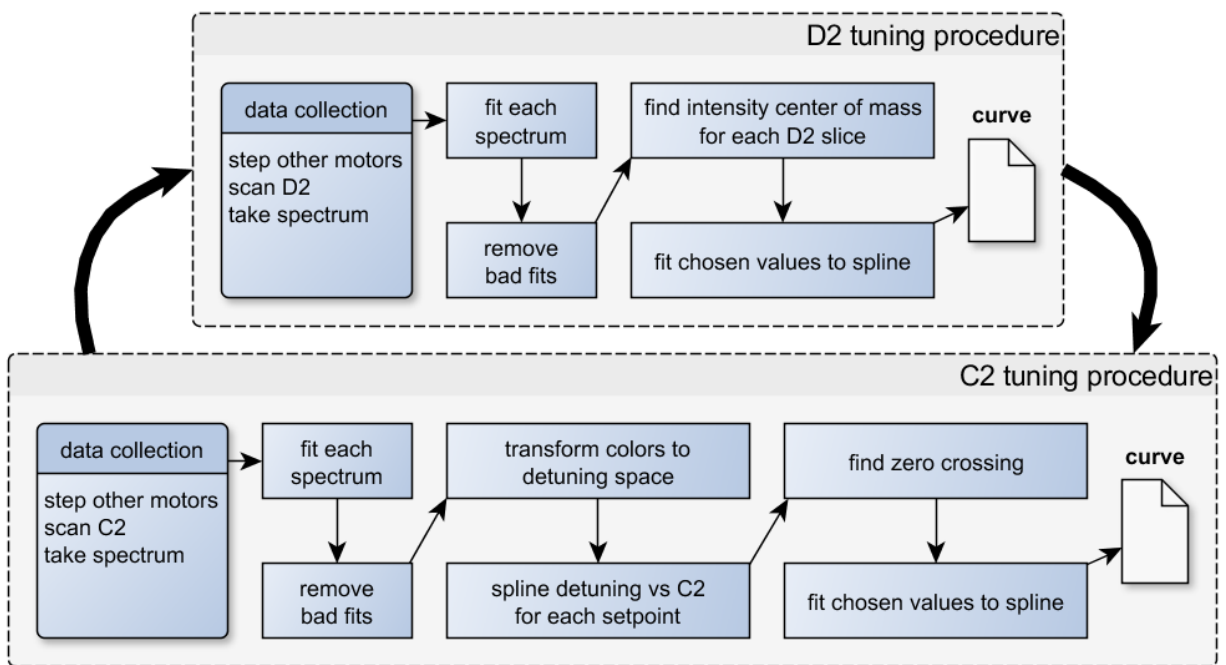


Figure 7.8: Poweramp tune procedure flowchart.

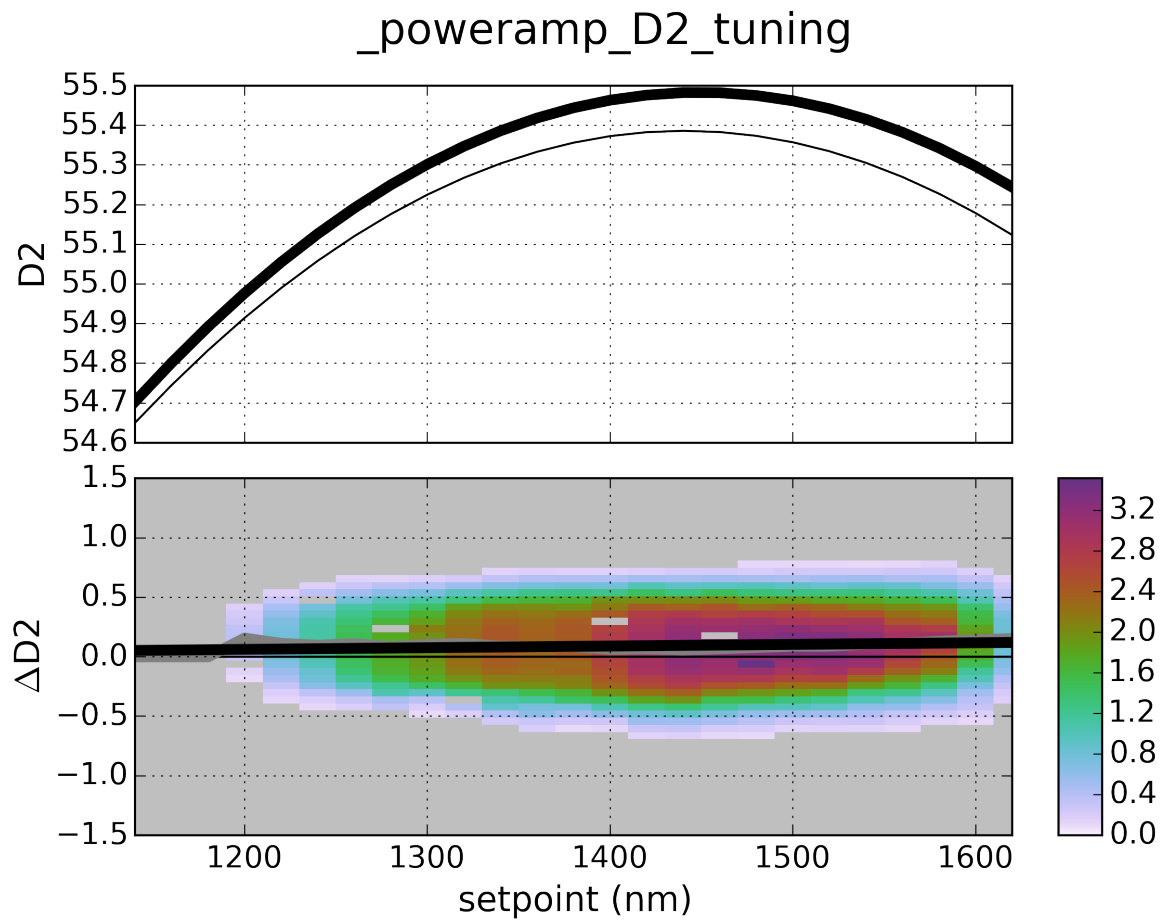


Figure 7.9: Poweramp D2 tuning output.

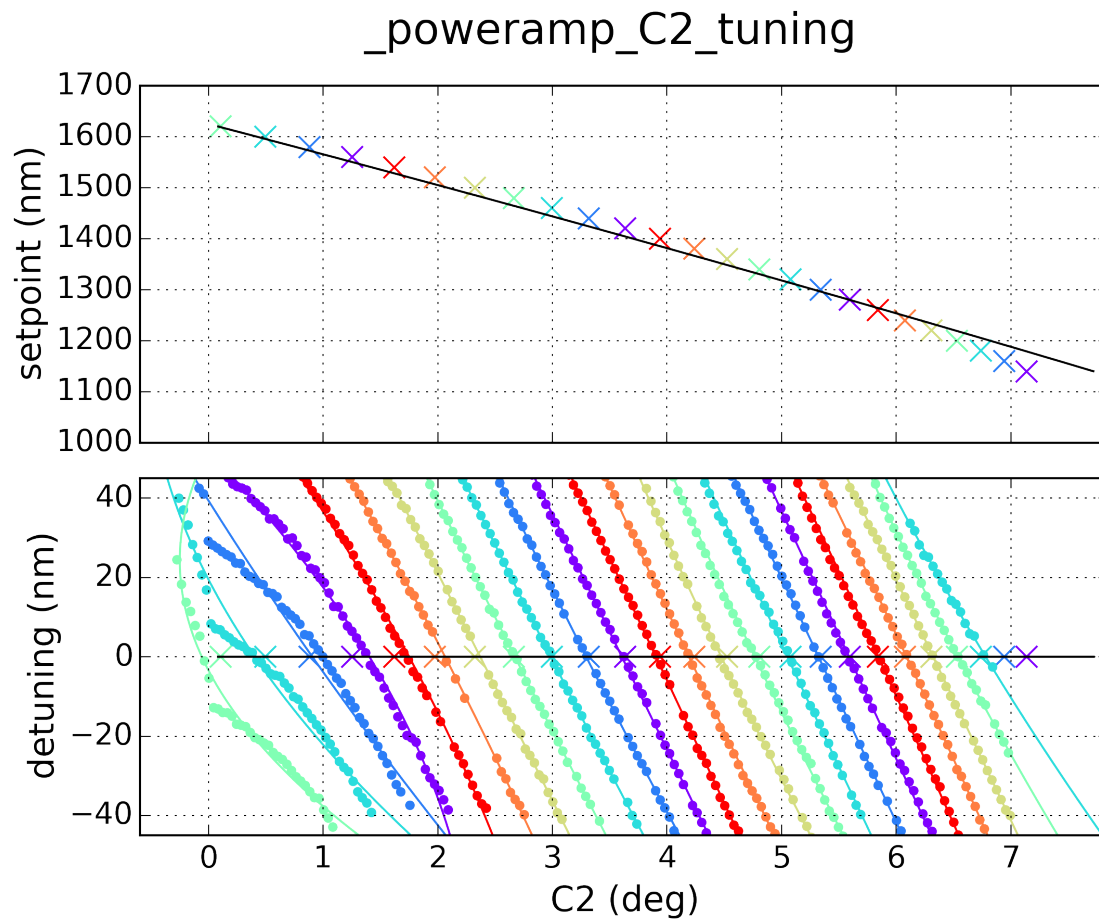


Figure 7.10: Poweramp C2 tuning output.

7.6 Mixers

Because mixers only have one degree of freedom each (crystal angle), there is really not that much ambiguity about what the ideal motor positions are. In fact, the best motor positions can be chosen simply by taking excursions relative to the old points (as in Figure 7.9) and picking the points with the highest intensity. After choosing motor positions, a simple correction for actual output frequencies can be applied using the monochromator.

I have prepared two functions: `process_intensity` and `process_tune` which accomplish each of these goals. They are general, capable of being used for *any* mixer or tune test.

PyCMDS can also explicitly take a spectrum at each motor position. This information takes longer to collect, but less human intervention—so it is a valid strategy that is sometimes employed.

7.7 Generalizability

This chapter has considered the automated procedures used in tuning the TOPAS-C, just one of the four models of OPA owned by the Wright Group. Simply put, this is because the other three OPA models (all picosecond OPAs) are easy to tune.

Figure 7.11 displays the entire tuning space for generation of signal and idler in one of the picosecond OPAs. In contrast to the TOPAS behavior, where neither motor axis constrains the output very well, *both* motors have very narrow features in this picosecond OPA. This means that it is at all times *unambiguous* whether a given motor position is ideal.

Much like the mixers, these OPAs can be readily tuned using a combination of the general functions `process_intensity` and `process_tune`.

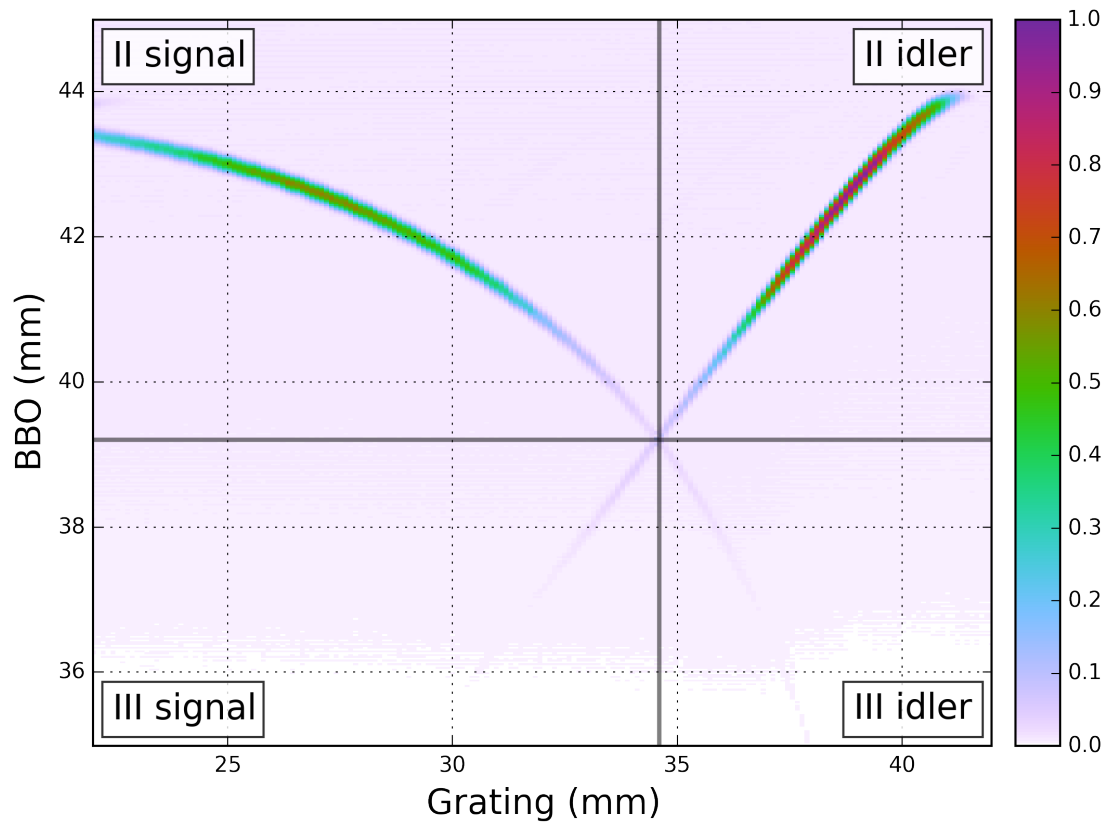


Figure 7.11: Motortune for picosecond OPA, monitored using a single pyroelectric detector.

Chapter 8

Disentangling material and instrument response

This Chapter borrows extensively from Kohler, Thompson, and Wright [112].

Ultrafast spectroscopy is often collected in the mixed frequency/time domain, where pulse durations are similar to system dephasing times. In these experiments, expectations derived from the familiar driven and impulsive limits are not valid. This work simulates the mixed-domain Four Wave Mixing response of a model system to develop expectations for this more complex field-matter interaction. We explore frequency and delay axes. We show that these line shapes are exquisitely sensitive to excitation pulse widths and delays. Near pulse overlap, the excitation pulses induce correlations which resemble signatures of dynamic inhomogeneity. We describe these line shapes using an intuitive picture that connects to familiar field-matter expressions. We develop strategies for distinguishing pulse-induced correlations from true system inhomogeneity. These simulations provide a foundation for interpretation of ultrafast experiments in the mixed domain.

8.1 Introduction

Ultrafast spectroscopy is based on using nonlinear interactions, created by multiple ultrashort (10^{-9} – 10^{-15} s) pulses, to resolve spectral information on timescales as short as the pulses themselves. [16, 113] The ultrafast spectra can be collected in the time domain or the frequency domain. [8]

Time-domain methods scan the pulse delays to resolve the free induction decay (FID). [18] The Fourier Transform of the FID gives the ultrafast spectrum. Ideally, these experiments are performed in the impulsive limit where FID dominates the measurement. FID occurs at the frequency of the transition that has been excited by a well-defined, time-ordered sequence of pulses. Time-domain methods are compromised when the dynamics occur on faster time scales than the ultrafast excitation pulses. As the pulses temporally overlap, FID from other pulse time-orderings and emission driven by the excitation pulses both become important. These factors are responsible for the complex “coherent artifacts” that are often ignored in pump-probe and related methods. [114, 115, 116, 117] Dynamics faster than the pulse envelopes are best measured using line shapes in frequency domain methods.

Frequency-domain methods scan pulse frequencies to resolve the ultrafast spectrum directly. [118, 119] Ideally, these experiments are performed in the driven limit where the steady state dominates the measurement. In the driven limit, all time-orderings of the pulse interactions are equally important and FID decay is negligible. The output signal is driven at the excitation pulse frequencies during the excitation pulse width. Frequency-domain methods are compromised when the spectral line shape is narrower than the frequency bandwidth of the excitation pulses. Dynamics that are slower than the pulse envelopes can be measured in the time domain by resolving the phase oscillations of the output signal during the entire FID decay.

There is also the hybrid mixed-time/frequency-domain approach, where pulse delays and pulse frequencies are both scanned to measure the system response. This approach is uniquely suited for experiments where the dephasing time is comparable to the pulse durations, so that neither frequency-domain nor time-domain approaches excel on their own. [119, 120, 121] In this regime, both FID and driven processes are important. [15] Their relative importance depends on pulse frequencies and delays. Extracting the correct spectrum from the measurement then requires a more complex analysis that explicitly treats

the excitation pulses and the different time-orderings. [122, 123, 124] Despite these complications, mixed-domain methods have a practical advantage: the dual frequency- and delay-scanning capabilities allow these methods to address a wide variety of dephasing rates.

The relative importance of FID and driven processes and the changing importance of different coherence pathways are important factors for understanding spectral features in all ultrafast methods. These methods include partially-coherent methods involving intermediate populations such as pump-probe [125], transient grating [126, 127, 128], transient absorption/reflection [129, 130], photon echo [131, 132, 133], two dimensional-infrared spectroscopy (2D-IR) [134, 135, 136], 2D-electronic spectroscopy (2D-ES) [137, 103], and three pulse photon echo peak shift (3PEPS) [138, 131, 139, 140, 141] spectroscopies. These methods also include fully-coherent methods involving only coherences such as Stimulated Raman Spectroscopy (SRS) [142, 143], Doubly Vibrationally Enhanced (DOVE) [144, 2, 145, 146, 147, 3, 148], Triply Resonant Sum Frequency (TRSF) [149, 4, 5], Sum Frequency Generation (SFG) [150], Coherent Anti-Stokes Raman Spectroscopy (CARS) [151, 152, 153], and other coherent Raman methods [154].

This paper focuses on understanding the nature of the spectral changes that occur in Coherent Multi-dimensional Spectroscopy (CMDS) as experiments transition between the two limits of frequency- and time-domain methods. CMDS is a family of spectroscopies that use multiple delay and/or frequency axes to extract homogeneous and inhomogeneous broadening, as well as detailed information about spectral diffusion and chemical changes. [155, 156] For time-domain CMDS (2D-IR, 2D-ES), the complications that occur when the impulsive approximation does not strictly hold has only recently been addressed. [27, 26]

Frequency-domain CMDS methods, referred to herein as multi-resonant CMDS (MR-CMDS), have similar capabilities for measuring homogeneous and inhomogeneous broadening. Although these experiments are typically described in the driven limit, [18, 127, 128] many of the experiments involve pulse widths that are comparable to the widths of the system. [146, 147, 6, 144, 157, 123] MR-CMDS then becomes a mixed-domain experiment whereby resonances are characterized with marginal resolution in both frequency and time. For example, DOVE spectroscopy involves three different pathways [158] whose relative importance depends on the relative importance of FID and driven responses. [159] In the

driven limit, the DOVE line shape depends on the difference between the first two pulse frequencies so the line shape has a diagonal character that mimics the effects of inhomogeneous broadening. In the FID limit where the coherence frequencies are defined instead by the transition, the diagonal character is lost. Understanding these effects is crucial for interpreting experiments, yet these effects have not been characterized for MR-CMDS.

This work considers the third-order MR-CMDS response of a 3-level model system using three ultrafast excitation beams with the commonly used four-wave mixing (FWM) phase-matching condition, $\vec{k}_{\text{out}} = \vec{k}_1 - \vec{k}_2 + \vec{k}_2'$. Here, the subscripts represent the excitation pulse frequencies, ω_1 and $\omega_2 = \omega_2'$. These experimental conditions were recently used to explore line shapes of excitonic systems, [123, 157] and have been developed on vibrational states as well [90]. Although MR-CMDS forms the context of this model, the treatment is quite general because the phase matching condition can describe any of the spectroscopies mentioned above with the exception of SFG and TRSF, for which the model can be easily extended. We numerically simulate the MR-CMDS response with pulse durations at, above, and below the system coherence time. To highlight the role of pulse effects, we build an interpretation of the full MR-CMDS response by first showing how finite pulses affect the evolution of a coherence, and then how finite pulses affect an isolated third-order pathway. When considering the full MR-CMDS response, we show that spectral features change dramatically as a function of delay, even for a homogeneous system with elementary dynamics. Importantly, the line shape can exhibit correlations that mimic inhomogeneity, and the temporal evolution of this line shape can mimic spectral diffusion. We identify key signatures that can help differentiate true inhomogeneity and spectral diffusion from these measurement artifacts.

8.2 Theory

We consider a simple three-level system (states $n = 0, 1, 2$) that highlights the multidimensional line shape changes resulting from choices of the relative dephasing and detuning of the system and the temporal and spectral widths of the excitation pulses. For simplicity, we will ignore population relaxation effects: $\Gamma_{11} = \Gamma_{00} = 0$.

The electric field pulses, $\{E_I\}$, are given by:

$$E_I(t; \omega_I, \tau_I, \vec{k}_I \cdot z) = \frac{1}{2} \left[c_I(t - \tau_I) e^{i\vec{k}_I \cdot z} e^{-i\omega_I(t - \tau_I)} + c.c. \right], \quad (8.1)$$

where ω_I is the field carrier frequency, \vec{k}_I is the wavevector, τ_I is the pulse delay, and c_I is a slowly varying envelope. In this work, we assume normalized (real-valued) Gaussian envelopes:

$$c_I(t) = \frac{1}{\Delta_t} \sqrt{\frac{2 \ln 2}{2\pi}} \exp \left(-\ln 2 \left[\frac{t}{\Delta_t} \right]^2 \right), \quad (8.2)$$

where Δ_t is the temporal FWHM of the envelope intensity. We neglect non-linear phase effects such as chirp so the FWHM of the frequency bandwidth is transform limited: $\Delta_\omega \Delta_t = 4 \ln 2 \approx 2.77$, where Δ_ω is the spectral FWHM (intensity scale).

The Liouville-von Neumann Equation propagates the density matrix, ρ :

$$\frac{d\rho}{dt} = -\frac{i}{\hbar} \left[\mathbf{H}_0 + \boldsymbol{\mu} \cdot \sum_{I=1,2,2'} E_I(t), \rho \right] + \boldsymbol{\Gamma} \rho. \quad (8.3)$$

Here \mathbf{H}_0 is the time-independent Hamiltonian, $\boldsymbol{\mu}$ is the dipole superoperator, and $\boldsymbol{\Gamma}$ contains the pure dephasing rate of the system. We perform the standard perturbative expansion of Equation 8.3 to third order in the electric field interaction [38, 35, 119, 160, 161] and restrict ourselves only to the terms that have the correct spatial wave vector $\vec{k}_{\text{out}} = \vec{k}_1 - \vec{k}_2 + \vec{k}_{2'}$. This approximation narrows the scope to sets of three interactions, one from each field, that result in the correct spatial dependence. The set of three interactions have $3! = 6$ unique time-ordered sequences, and each time-ordering produces either two or three unique system-field interactions for our system, for a total of sixteen unique system-field

interaction sequences, or Liouville pathways, to consider. Figure 8.1 shows these sixteen pathways as Wave Mixing Energy Level (WMEL) diagrams [40].

We first focus on a single interaction in these sequences, where an excitation pulse, x , forms ρ_{ij} from ρ_{ik} or ρ_{kj} . For brevity, we use $\hbar = 1$ and abbreviate the initial and final density matrix elements as ρ_i and ρ_f , respectively. Using the natural frequency rotating frame, $\tilde{\rho}_{ij} = \rho_{ij}e^{-i\omega_{ij}t}$, the formation of ρ_f using pulse x is written as

$$\begin{aligned} \frac{d\tilde{\rho}_f}{dt} = & -\Gamma_f \tilde{\rho}_f \\ & + \frac{i}{2} \lambda_f \mu_f c_x(t - \tau_x) e^{i\kappa_f(\vec{k}_x \cdot \vec{z} + \omega_x \tau_x)} e^{i\kappa_f \Omega_{fx} t} \tilde{\rho}_i(t), \end{aligned} \quad (8.4)$$

where $\Omega_{fx} = \kappa_f^{-1} \omega_f - \omega_x (= |\omega_f| - \omega_x)$ is the detuning, ω_f is the transition frequency of the i^{th} transition, μ_f is the transition dipole, and Γ_f is the dephasing/relaxation rate for ρ_f . The λ_f and κ_f parameters describe the phases of the interaction: $\lambda_f = +1$ for ket-side transitions and -1 for bra-side transitions, and κ_f depends on whether ρ_f is formed via absorption ($\kappa_f = \lambda_f$) or emission ($\kappa_f = -\lambda_f$). κ_f also has a direct relationship to the phase matching relationship: for transitions with E_2 , $\kappa_f = 1$, and for E_1 or $E_{2'}$, $\kappa_f = -1$. In the following equations we neglect spatial dependence ($z = 0$).

Equation 8.4 forms the basis for our simulations. It provides a general expression for arbitrary values of the dephasing rate and excitation pulse bandwidth. The integral solution is

$$\begin{aligned} \tilde{\rho}_f(t) = & \frac{i}{2} \lambda_f \mu_f e^{i\kappa_f \omega_x \tau_x} e^{i\kappa_f \Omega_{fx} t} \\ & \times \int_{-\infty}^{\infty} c_x(t - u - \tau_x) \tilde{\rho}_i(t - u) \Theta(u) \\ & \times e^{-(\Gamma_f + i\kappa_f \Omega_{fx})u} du, \end{aligned} \quad (8.5)$$

where Θ is the Heaviside step function. Equation 8.5 becomes the steady state limit expression when $\Delta_t |\Gamma_f + i\kappa_f \Omega_{fx}| \gg 1$, and the impulsive limit expression results when $\Delta_t |\Gamma_f + i\kappa_f \Omega_{fx}| \ll 1$. Both limits are important for understanding the multidimensional line shape changes discussed in this paper.

Figure 8.2 gives an overview of the simulations done in this work. Figure 8.2a shows an excitation pulse (gray-shaded) and examples of a coherent transient for three different dephasing rates. The color bindings to dephasing rates introduced in Figure 8.2a will be used consistently throughout this work. Our

simulations use systems with dephasing rates quantified relative to the pulse duration: $\Gamma_{10}\Delta_t = 0.5, 1,$ or 2 . The temporal axes are normalized to the pulse duration, Δ_t . The $\Gamma_{10}\Delta_t = 2$ transient is mostly driven by the excitation pulse while $\Gamma_{10}\Delta_t = 0.5$ has a substantial free induction decay (FID) component at late times. Figure 8.2b shows a pulse sequence (pulses are shaded orange and pink) and the resulting system evolution of pathway $V\gamma$ ($00 \xrightarrow{2} 01 \xrightarrow{2'} 11 \xrightarrow{1} 10 \xrightarrow{\text{out}} 00$) with $\Gamma_{10}\Delta_t = 1$. The final polarization (teal) is responsible for the emitted signal, which is then passed through a frequency bandpass filter to emulate monochromator detection (Figure 8.2c). The resulting signal is explored in 2D delay space (Figure 8.2d), 2D frequency space (Figure 8.2f), and hybrid delay-frequency space (Figure 8.2e). The detuning frequency axes are also normalized by the pulse bandwidth, Δ_ω .

We now consider the generalized Liouville pathway $L : \rho_0 \xrightarrow{x} \rho_1 \xrightarrow{y} \rho_2 \xrightarrow{z} \rho_3 \xrightarrow{\text{out}} \rho_4$, where $x, y,$ and z denote properties of the first, second, and third pulse, respectively, and indices 0, 1, 2, 3, and 4 define the properties of the ground state, first, second, third, and fourth density matrix elements, respectively. Figure 8.2b demonstrates the correspondence between x, y, z notation and 1, 2, 2' notation for the laser pulses with pathway $V\gamma$.

The electric field emitted from a Liouville pathway is proportional to the polarization created by the third-order coherence:

$$E_L(t) = i\mu_4\rho_3(t). \quad (8.6)$$

Equation 8.6 assumes perfect phase-matching and no pulse distortions through propagation. Equation 8.5 shows that the output field for this Liouville pathway is

$$E_L(t) = \frac{i}{8}\lambda_1\lambda_2\lambda_3\mu_1\mu_2\mu_3\mu_4 e^{i(\kappa_1\omega_x\tau_x + \kappa_2\omega_y\tau_y + \kappa_3\omega_z\tau_z)} e^{-i(\kappa_3\omega_z + \kappa_2\omega_y + \kappa_1\omega_x)t} \\ \times \iiint_{-\infty}^{\infty} c_z(t-u-\tau_z)c_y(t-u-v-\tau_y)c_x(t-u-v-w-\tau_x)R_L(u,v,w)dw dv du, \quad (8.7)$$

$$R_L(u,v,w) = \Theta(w)e^{-(\Gamma_1 + i\kappa_1\Omega_{1x})w}\Theta(v)e^{-(\Gamma_2 + i[\kappa_1\Omega_{1x} + \kappa_2\Omega_{2y}])v}\Theta(u)e^{-(\Gamma_3 + i[\kappa_1\Omega_{1x} + \kappa_2\Omega_{2y} + \kappa_3\Omega_{3z}])u}, \quad (8.8)$$

where R_L is the third-order response function for the Liouville pathway. The total electric field will be

the superposition of all the Liouville pathways:

$$E_{\text{tot}} = \sum_L E_L(t). \quad (8.9)$$

For the superposition of Equation 8.9 to be non-canceling, certain symmetries between the pathways must be broken. In general, this requires one or more of the following inequalities: $\Gamma_{10} \neq \Gamma_{21}$, $\omega_{10} \neq \omega_{21}$, and/or $\sqrt{2}\mu_{10} \neq \mu_{21}$. Our simulations use the last inequality, which is important in two-level systems ($\mu_{21} = 0$) and in systems where state-filling dominates the non-linear response, such as in semiconductor excitons. The exact ratio between μ_{10} and μ_{21} affects the absolute amplitude of the field, but does not affect the multidimensional line shape. Importantly, the dipole inequality does not break the symmetry of double quantum coherence pathways (time-orderings II and IV), so such pathways are not present in our analysis.

In MR-CMDS, a monochromator resolves the driven output frequency from other nonlinear output frequencies, which in our case is $\omega_m = \omega_1 - \omega_2 + \omega_{2'} = \omega_1$. The monochromator can also enhance spectral resolution, as we show in Section 8.4.1. In this simulation, the detection is emulated by transforming $E_{\text{tot}}(t)$ into the frequency domain, applying a narrow bandpass filter, $M(\omega)$, about ω_1 , and applying amplitude-scaled detection:

$$S_{\text{tot}}(\omega_1, \omega_2, \tau_{21}, \tau_{22'}) = \sqrt{\int |M(\omega - \omega_1)E_{\text{tot}}(\omega)|^2 d\omega}, \quad (8.10)$$

where $E_{\text{tot}}(\omega)$ denotes the Fourier transform of $E_{\text{tot}}(t)$ (see Figure 8.2c). For M we used a rectangular function of width $0.408\Delta_\omega$. The arguments of S_{tot} refer to the *experimental* degrees of freedom. The signal delay dependence is parameterized with the relative delays τ_{21} and $\tau_{22'}$, where $\tau_{nm} = \tau_n - \tau_m$ (see Figure 8.2b). Table S1 summarizes the arguments for each Liouville pathway. Figure 8.2f shows the 2D (ω_1, ω_2) S_{tot} spectrum resulting from the pulse delay times represented in Figure 8.2b.

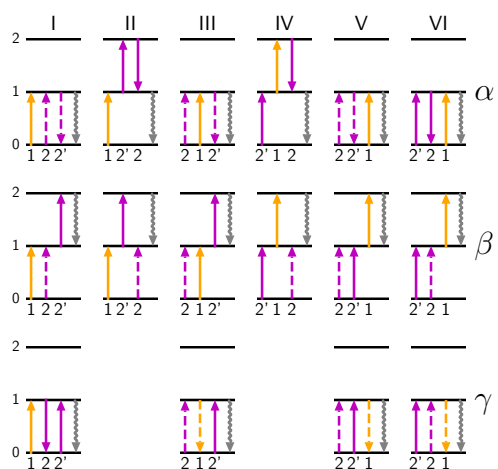


Figure 8.1: The sixteen triply-resonant Liouville pathways for the third-order response of the system used here. Time flows from left to right. Each excitation is labeled by the pulse stimulating the transition; excitations with ω_1 are yellow, excitations with $\omega_2 = \omega_{2'}$ are purple, and the final emission is gray.

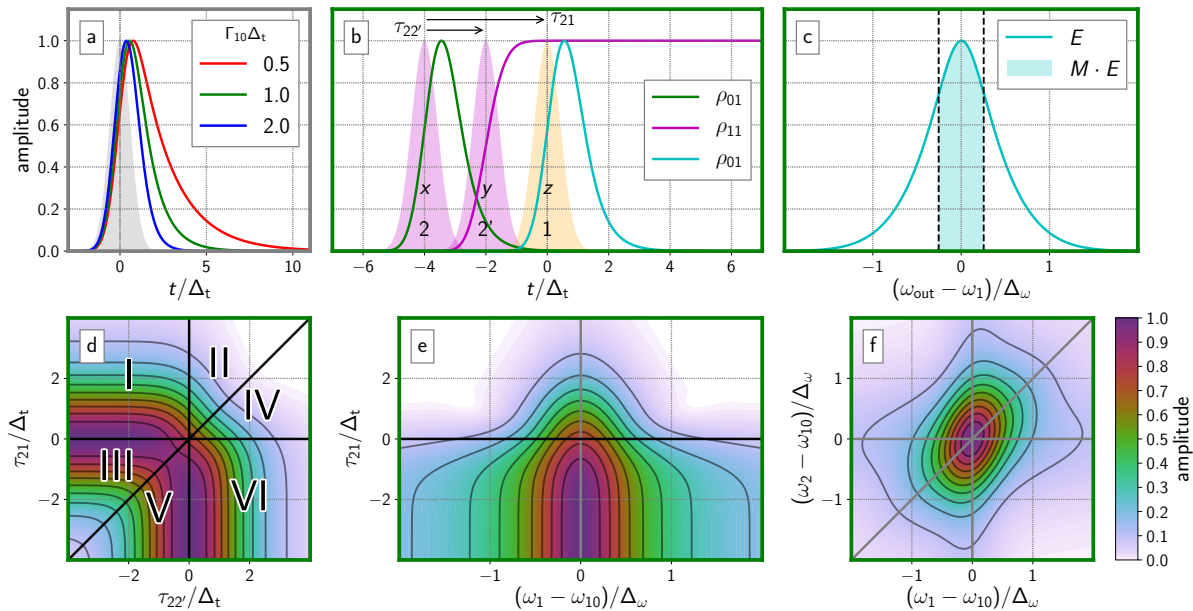


Figure 8.2: Overview of the MR-CMDS simulation. (a) The temporal profile of a coherence under pulsed excitation depends on how quickly the coherence dephases. In all subsequent panes, the relative dephasing rate is kept constant at $\Gamma_{10}\Delta_t = 1$. (b) Simulated evolution of the density matrix elements of a third-order Liouville pathway $V\gamma$ under fully resonant excitation. Pulses can be labeled both by their time of arrival (x, y, z) and by the lab lasers used to stimulate the transitions ($2, 2', 1$). The final coherence (teal) creates the output electric field. (c) The frequency profile of [Eccentricity] the output electric field is filtered by a monochromator gating function, $M(\omega)$, and the passed components (shaded) are measured. (d-f) Signal is viewed against two laser parameters, either as 2D delay (d), mixed delay-frequency (e), or 2D frequency plots (f). The six time-orderings are labeled in (d) to help introduce our delay convention.

8.2.1 Characteristics of driven and impulsive response

The changes in the spectral line shapes described in this work are best understood by examining the driven/continuous wave (CW) and impulsive limits of Equation 8.5 and 8.7. The driven limit is achieved when pulse durations are much longer than the response function dynamics: $\Delta_t |\Gamma_f + i\kappa_f \Omega_{fx}| \gg 1$. In this limit, the system will adopt a steady state over excitation: $d\rho/dt \approx 0$. Neglecting phase factors, the driven solution to Equation 8.5 will be

$$\tilde{\rho}_f(t) = \frac{\lambda_f \mu_f c_x(t - \tau_x) e^{i\kappa_f \Omega_{fx} t}}{2 \kappa_f \Omega_{fx}} \tilde{\rho}_i(t). \quad (8.11)$$

The frequency and temporal envelope of the excitation pulse controls the coherence time evolution, and the relative amplitude and phase of the coherence is directly related to detuning from resonance.

The impulsive limit is achieved when the excitation pulses are much shorter than response function dynamics: $\Delta_t |\Gamma_f + i\kappa_f \Omega_{fx}| \ll 1$. The full description of the temporal evolution has two separate expressions: one for times when the pulse is interacting with the system, and one for times after pulse interaction. Both expressions are important when describing CMDS experiments.

For times after the pulse interaction, $t \gtrsim \tau_x + \Delta_t$, the field-matter coupling is negligible. The evolution for these times, on resonance, is given by

$$\tilde{\rho}_f(t) = \frac{i\lambda_f \mu_f}{2} \tilde{\rho}_i(\tau_x) \int c_x(u) du e^{-\Gamma_f(t-\tau_x)}. \quad (8.12)$$

This is classic free induction decay (FID) evolution: the system evolves at its natural frequency and decays at rate Γ_f . It is important to note that, while this expression is explicitly derived from the impulsive limit, FID behavior is not exclusive to impulsive excitation, as we have defined it. A latent FID will form if the pulse vanishes at a fast rate relative to the system dynamics.

For evaluating times near pulse excitation, $t \lesssim \tau_x + \Delta_t$, we implement a Taylor expansion in the response function about zero: $e^{-(\Gamma_f + i\kappa_f \Omega_{fx})u} = 1 - (\Gamma_f + i\kappa_f \Omega_{fx})u + \dots$. Our impulsive criterion requires that a low order expansion will suffice; it is instructive to consider the result of the first order expansion of

Equation 8.5:

$$\begin{aligned} \tilde{\rho}_f(t) = & \frac{i\lambda_f\mu_f}{2} e^{-i\kappa_f\omega_x\tau_x} e^{-i\kappa_f\Omega_{fx}t} \tilde{\rho}_i(\tau_x) \\ & \times \left[(1 - (\Gamma_f + i\kappa_f\Omega_{fx})(t - \tau_x)) \int_{-\infty}^{t-\tau_x} c_x(u) du \right. \\ & \left. + (\Gamma_f + i\kappa_f\Omega_{fx}) \int_{-\infty}^{t-\tau_x} c_x(u) u du \right]. \end{aligned} \quad (8.13)$$

During this time $\tilde{\rho}_f$ builds up roughly according to the integration of the pulse envelope. The build-up is integrated because the pulse transfers energy before appreciable dephasing or detuning occurs. Contrary to the expectation of impulsive evolution, the evolution of $\tilde{\rho}_f$ is explicitly affected by the pulse frequency, and the temporal profile evolves according to the pulse.

It is important to recognize that the impulsive limit is defined not only by having slow relaxation relative to the pulse duration, but also by small detuning relative to the pulse bandwidth (as is stated in the inequality). As detuning increases, the higher orders of the response function Taylor expansion will be needed to describe the rise time, and the driven limit of Equation 8.11 will become valid. The details of this build-up time can often be neglected in impulsive approximations because build-up contributions are often negligible in analysis; the period over which the initial excitation occurs is small in comparison to the free evolution of the system. The build-up behavior can be emphasized by the measurement, which makes Equation 8.13 important.

We now consider full Liouville pathways in the impulsive and driven limits of Equation 8.7. For the driven limit, Equation 8.7 can be reduced to

$$\begin{aligned} E_L(t) = & \frac{1}{8} \lambda_1 \lambda_2 \lambda_3 \mu_1 \mu_2 \mu_3 \mu_4 e^{-i(\kappa_1\omega_x\tau_x + \kappa_2\omega_y\tau_y + \kappa_3\omega_z\tau_z)} \\ & \times e^{i(\kappa_3\omega_z + \kappa_2\omega_y + \kappa_1\omega_x)t} \\ & \times c_z(t - \tau_z) c_y(t - \tau_y) c_x(t - \tau_x) \\ & \times \frac{1}{\kappa_1\Omega_{1x} - i\Gamma_1} \frac{1}{\kappa_1\Omega_{1x} + \kappa_2\Omega_{2y} - i\Gamma_2} \\ & \times \frac{1}{\kappa_1\Omega_{1x} + \kappa_2\Omega_{2y} + \kappa_3\Omega_{3z} - i\Gamma_3}. \end{aligned} \quad (8.14)$$

It is important to note that the signal depends on the multiplication of all the fields; pathway discrimina-

tion based on pulse time-ordering is not achievable because polarizations exists only when all pulses are overlapped. This limit is the basis for frequency-domain techniques. Frequency axes, however, are not independent because the system is forced to the laser frequency and influences the resonance criterion for subsequent excitations. As an example, observe that the first two resonant terms in Equation 8.14 are maximized when $\omega_x = |\omega_1|$ and $\omega_y = |\omega_2|$. If ω_x is detuned by some value ε , however, the occurrence of the second resonance shifts to $\omega_y = |\omega_2| + \varepsilon$, effectively compensating for the ω_x detuning. This shifting of the resonance results in 2D line shape correlations.

If the pulses do not temporally overlap ($\tau_x + \Delta_t \lesssim \tau_y + \Delta_t \lesssim \tau_z + \Delta_t \lesssim t$), then the impulsive solution to the full Liouville pathway of Equation 8.7 is

$$\begin{aligned}
 E_L(t) &= \frac{i}{8} \lambda_1 \lambda_2 \lambda_3 \mu_1 \mu_2 \mu_3 \mu_4 e^{i(\omega_1 + \omega_2 + \omega_3)t} \\
 &\quad \times \int c_x(w) dw \int c_y(v) dv \int c_z(u) du \\
 &\quad \times e^{-\Gamma_1(\tau_y - \tau_x)} e^{-\Gamma_2(\tau_z - \tau_y)} e^{-\Gamma(t - \tau_z)}.
 \end{aligned} \tag{8.15}$$

Pathway discrimination is demonstrated here because the signal is sensitive to the time-ordering of the pulses. This limit is suited for delay scanning techniques. The emitted signal frequency is determined by the system and can be resolved by scanning a monochromator.

The driven and impulsive limits can qualitatively describe our simulated signals at certain frequency and delay combinations. Of the three expressions, the FID limit most resembles signal when pulses are near resonance and well-separated in time (so that build-up behavior is negligible). The build-up limit approximates well when pulses are near-resonant and arrive together (so that build-up behavior is emphasized). The driven limit holds for large detunings, regardless of delay.

8.2.2 Inhomogeneity

Inhomogeneity is isolated in CMDS through both spectral signatures, such as line-narrowing [162, 119, 163, 164, 154], and temporal signatures, such as photon echoes [165, 166]. We simulate the effects of static inhomogeneous broadening by convolving the homogeneous response with a Gaussian distribution function. Dynamic broadening effects such as spectral diffusion are beyond the scope of this work.

Here we describe how to transform the data of a single reference oscillator signal to that of an inhomogeneous distribution. The oscillators in the distribution are allowed have arbitrary energies for their states, which will cause frequency shifts in the resonances. To show this, we start with a modified, but equivalent, form of Equation 8.4:

$$\begin{aligned} \frac{d\tilde{\rho}_f}{dt} = & -\Gamma_f \tilde{\rho}_f + \frac{i}{2} \lambda_f \mu_f c_x (t - \tau_x) \\ & \times e^{i\kappa_f(\vec{k} \cdot \mathbf{z} + \omega_x \tau_x)} e^{-i\kappa_f(\omega_x - |\omega_f|)t} \tilde{\rho}_i(t). \end{aligned} \quad (8.16)$$

We consider two oscillators with transition frequencies ω_f and $\omega'_f = \omega_f + \delta$. So long as $|\delta| \leq \omega_f$ (so that $|\omega_f + \delta| = |\omega_f| + \delta$ and thus the rotating wave approximation does not change), Equation 8.16 shows that the two are related by

$$\frac{d\tilde{\rho}'_f}{dt}(t; \omega_x) = \frac{d\tilde{\rho}_f}{dt}(t; \omega_x - \delta) e^{i\kappa_f \delta \tau_x}. \quad (8.17)$$

Because both coherences are assumed to have the same initial conditions ($\rho_0(-\infty) = \rho'_0(-\infty) = 0$), the equality also holds when both sides of the equation are integrated. The phase factor $e^{i\kappa_f \delta \tau_x}$ in the substitution arises from Equation 8.1, where the pulse carrier frequency maintains its phase within the pulse envelope for all delays.

The resonance translation can be extended to higher order signals as well. For a third-order signal, we compare systems with transition frequencies $\omega'_{10} = \omega_{10} + a$ and $\omega'_{21} = \omega_{21} + b$. The extension of Equation 8.17 to pathway $V\beta$ gives

$$\begin{aligned} \tilde{\rho}'_3(t; \omega_2, \omega'_2, \omega_1) = & \tilde{\rho}_3(t; \omega_2 - a, \omega_2' - a, \omega_1 - b) \\ & \times e^{i\kappa_2 a \tau_2} e^{i\kappa_2' a \tau_2'} e^{i\kappa_1 b \tau_1}. \end{aligned} \quad (8.18)$$

The translation of each laser coordinate depends on which transition is made (e.g. a for transitions between $|0\rangle$ and $|1\rangle$ or b for transitions between $|1\rangle$ and $|2\rangle$), so the exact translation relation differs between pathways. We can now compute the ensemble average of signal for pathway $V\beta$ as a convolution

between the distribution function of the system, $K(a, b)$, and the single oscillator response:

$$\begin{aligned} \langle \tilde{\rho}_3(t; \omega_2, \omega_{2'}, \omega_1) \rangle &= \iint K(a, b) \\ &\times \tilde{\rho}_3(t; \omega_2 + a, \omega_{2'} + a, \omega_1 + b) \\ &\times e^{i\kappa_2 a \tau_2} e^{i\kappa_{2'} a \tau_{2'}} e^{i\kappa_1 b \tau_1} da db. \end{aligned} \quad (8.19)$$

For this work, we restrict ourselves to a simpler ensemble where all oscillators have equally spaced levels (i.e. $a = b$). This makes the translation identical for all pathways and reduces the dimensionality of the convolution. Since pathways follow the same convolution we may also perform the convolution on the total signal field:

$$\begin{aligned} \langle E_{\text{tot}}(t) \rangle &= \sum_L \mu_{4,L} \int K(a, a) \\ &\times \tilde{\rho}_{3,L}(t; \omega_x - a, \omega_y - a, \omega_z - a) \\ &\times e^{ia(\kappa_x \tau_x + \kappa_y \tau_y + \kappa_z \tau_z)} da. \end{aligned} \quad (8.20)$$

Furthermore, since $\kappa = -1$ for E_1 and $E_{2'}$, while $\kappa = 1$ for E_2 , we have $e^{ia(\kappa_x \tau_x + \kappa_y \tau_y + \kappa_z \tau_z)} = e^{-ia(\tau_1 - \tau_2 + \tau_{2'})}$ for all pathways. Equivalently, if the electric field is parameterized in terms of laser coordinates ω_1 and ω_2 , the ensemble field can be calculated as

$$\begin{aligned} \langle E_{\text{tot}}(t; \omega_1, \omega_2) \rangle &= \int K(a, a) E_{\text{tot}}(t; \omega_1 - a, \omega_2 - a) \\ &\times e^{-ia(\tau_1 - \tau_2 + \tau_{2'})} da. \end{aligned} \quad (8.21)$$

which is a 1D convolution along the diagonal axis in frequency space. Figure 8.3 demonstrates the use of Equation 8.21 on a homogeneous line shape.

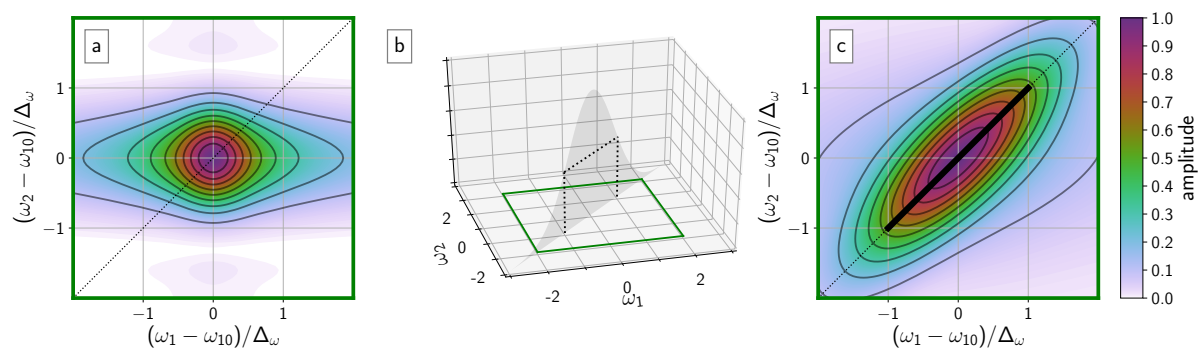


Figure 8.3: Overview of the convolution. (a) The homogeneous line shape. (b) The distribution function, K , mapped onto laser coordinates. (c) The resulting ensemble line shape computed from the convolution. The thick black line represents the FWHM of the distribution function.

8.3 Methods

A matrix representation of differential equations of the type in Equation 8.7 was numerically integrated for parallel computation of Liouville elements (see SI for details). [167, 168] The lower bound of integration was $2\Delta_t$ before the first pulse, and the upper bound was $5\Gamma_{10}^{-1}$ after the last pulse, with step sizes much shorter than the pulse durations. Integration was performed in the FID rotating frame; the time steps were chosen so that both the system-pulse difference frequencies and the pulse envelope were well-sampled.

The following simulations explore the four-dimensional $(\omega_1, \omega_2, \tau_{21}, \tau_{22'})$ variable space. Both frequencies are scanned about the resonance, and both delays are scanned about pulse overlap. We explored the role of sample dephasing rate by calculating signal for systems with dephasing rates such that $\Gamma_{10}\Delta_t = 0.5, 1, \text{ and } 2$. Inhomogeneous broadening used a spectral FWHM, Δ_{inhom} , that satisfied $\Delta_{\text{inhom}}/\Delta\omega = 0, 0.5, 1, \text{ and } 2$ for the three dephasing rates. For all these dimensions, both $\rho_3(t; \omega_1, \omega_2, \tau_{21}, \tau_{22'})$ and $S_{\text{tot}}(\omega_1, \omega_2, \tau_{21}, \tau_{22'})$ are recorded for each unique Liouville pathway. Our simulations were done using the open-source SciPy library. [82]

There are three details of our simulation strategy that deserve more exposition:

1. Liouville pathway parameters
2. Matrix formulation of Liouville pathways
3. Efficient integration of the Liouville Equation using the Heun method

8.3.1 Liouville pathway parameters

Table 8.1 describes the relationship between our notation and the parameters that make up the 16 Liouville pathways.

L	Liouville Pathway	x	y	z	λ_1	λ_2	λ_3	κ_1	κ_2	κ_3	μ_1	μ_2	μ_3	μ_4
$I\alpha$	$00 \rightarrow 10 \rightarrow 11 \rightarrow 10 \rightarrow 00$	3^*1	3^*2	$3^*2'$	1	-1	-1	3^*-1	3^*1	3^*-1	μ_{10}	μ_{10}	μ_{10}^*	μ_{10}^*
$I\beta$	$00 \rightarrow 10 \rightarrow 11 \rightarrow 21 \rightarrow 11$				1	-1	1				μ_{10}	μ_{10}	μ_{21}	μ_{21}^*
$I\gamma$	$00 \rightarrow 10 \rightarrow 00 \rightarrow 10 \rightarrow 00$				1	1	1				μ_{10}	μ_{10}^*	μ_{10}	μ_{10}^*
$II\alpha$	$00 \rightarrow 10 \rightarrow 20 \rightarrow 10 \rightarrow 00$	2^*1	$2^*2'$	2^*2	1	1	1	2^*-1	2^*-1	2^*1	μ_{10}	μ_{21}	μ_{21}^*	μ_{10}^*
$II\beta$	$00 \rightarrow 10 \rightarrow 20 \rightarrow 21 \rightarrow 11$				1	1	-1				μ_{10}	μ_{21}	μ_{10}	μ_{21}^*
$III\alpha$	$00 \rightarrow 01 \rightarrow 11 \rightarrow 10 \rightarrow 00$	3^*2	3^*1	$3^*2'$	-1	1	-1	3^*1	3^*-1	3^*-1	μ_{10}	μ_{10}	μ_{10}^*	μ_{10}^*
$III\beta$	$00 \rightarrow 01 \rightarrow 11 \rightarrow 21 \rightarrow 11$				-1	1	1				μ_{10}	μ_{10}	μ_{21}	μ_{21}^*
$III\gamma$	$00 \rightarrow 01 \rightarrow 00 \rightarrow 10 \rightarrow 00$				-1	-1	1				μ_{10}	μ_{10}^*	μ_{10}	μ_{10}^*
$IV\alpha$	$00 \rightarrow 10 \rightarrow 20 \rightarrow 10 \rightarrow 00$	$2^*2'$	2^*1	2^*2	1	1	1	2^*-1	2^*-1	2^*1	μ_{10}	μ_{21}	μ_{21}^*	μ_{10}^*
$IV\beta$	$00 \rightarrow 10 \rightarrow 20 \rightarrow 21 \rightarrow 11$				1	1	-1				μ_{10}	μ_{21}	μ_{10}	μ_{21}^*
$V\alpha$	$00 \rightarrow 01 \rightarrow 00 \rightarrow 10 \rightarrow 00$	3^*2	$3^*2'$	3^*1	-1	-1	1	3^*1	3^*-1	3^*-1	μ_{10}	μ_{10}^*	μ_{10}	μ_{10}^*
$V\beta$	$00 \rightarrow 01 \rightarrow 11 \rightarrow 21 \rightarrow 11$				-1	1	1				μ_{10}	μ_{10}	μ_{21}	μ_{21}^*
$V\gamma$	$00 \rightarrow 01 \rightarrow 11 \rightarrow 10 \rightarrow 00$				-1	1	-1				μ_{10}	μ_{10}	μ_{10}^*	μ_{10}^*
$VI\alpha$	$00 \rightarrow 10 \rightarrow 00 \rightarrow 10 \rightarrow 00$	$3^*2'$	3^*2	3^*1	1	1	1	3^*-1	3^*1	3^*-1	μ_{10}	μ_{10}^*	μ_{10}	μ_{10}^*
$VI\beta$	$00 \rightarrow 10 \rightarrow 11 \rightarrow 21 \rightarrow 00$				1	-1	1				μ_{10}	μ_{10}	μ_{21}	μ_{21}^*
$VI\gamma$	$00 \rightarrow 10 \rightarrow 11 \rightarrow 10 \rightarrow 00$				1	-1	-1				μ_{10}	μ_{10}	μ_{10}^*	μ_{10}^*

Table 8.1: Parameters of each Liouville Pathway.

8.3.2 Matrix formulation

Generic single pathway

In this work we explicitly incorporate our time-dependent electric fields into the Liouville equations as done by Gelin, Egorova, and Domcke [169].

Our third order experiment involves three successive light-matter interactions. In a generic Liouville pathway $\rho_0 \xrightarrow{x} \rho_1 \xrightarrow{y} \rho_2 \xrightarrow{z} \rho_3 \xrightarrow{\text{out}} \rho_4$, the third order polarization can be written as three coupled differential equations:

$$\frac{d\tilde{\rho}_1}{dt} = -\Gamma_1\tilde{\rho}_1 + S_1(t)\tilde{\rho}_0(t) \quad (8.22)$$

$$\frac{d\tilde{\rho}_2}{dt} = -\Gamma_2\tilde{\rho}_2 + S_2(t)\tilde{\rho}_1(t) \quad (8.23)$$

$$\frac{d\tilde{\rho}_3}{dt} = -\Gamma_3\tilde{\rho}_3 + S_3(t)\tilde{\rho}_2(t) \quad (8.24)$$

Where S contains all light-matter interactions:

$$S_j(t) \equiv \frac{i}{2}\lambda_j\mu_j e^{-i\kappa_j\omega_n\tau_n} c_n(t - \tau_n) e^{i(\kappa_j\omega_n - \omega_j)t} \quad (8.25)$$

Here, j denotes the transition and n denotes the driving pulse. See discussion of Equation 4 for identification of each term. The emitted electric field is proportional to $\tilde{\rho}_3$ (see Equation 6), so there is no need to explicitly consider $\rho_3 \xrightarrow{\text{out}} \rho_4$. Note that we do not include depletion due to light-matter interaction. 0 is the ground state, so $\rho_0 = 1$ and $\Gamma_0 = 0$.

Our three coupled differential equations can be cast into a matrix form:

$$\frac{d\bar{\rho}}{dt} = \bar{Q}(t)\bar{\rho} \quad (8.26)$$

$$\bar{\rho} \equiv \begin{bmatrix} \tilde{\rho}_0 \\ \tilde{\rho}_1 \\ \tilde{\rho}_2 \\ \tilde{\rho}_3 \end{bmatrix} \quad (8.27)$$

$$\bar{Q}(t) \equiv \begin{bmatrix} -\Gamma_0 & 0 & 0 & 0 \\ S_1(t) & -\Gamma_1 & 0 & 0 \\ 0 & S_2(t) & -\Gamma_2 & 0 \\ 0 & 0 & S_3(t) & -\Gamma_3 \end{bmatrix} \quad (8.28)$$

Equation 8.26 could be numerically integrated to determine $\tilde{\rho}_3$ for this pathway.

Full formulation

As discussed throughout this chapter and shown in Figure 8.1, there are 6 unique time-orderings and 16 unique pathways to consider in this work. One could write a separate differential equation for each pathway—this would require tabulation of 48 density matrix elements. Instead, we capitalize on the symmetry of our pathways to create a single differential equation that is computationally cheaper.

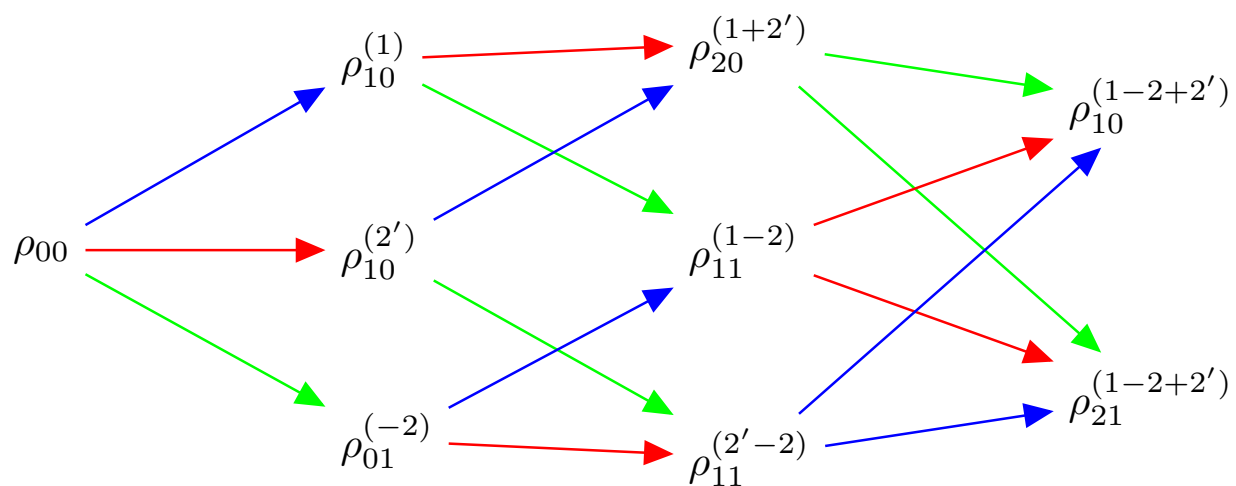


Figure 8.4: Network of Liouville pathways in this work. Superscripts denote the field interactions that have occurred to make the density matrix element. Colors denote the pulse that is used for each transition (blue is E_1 , red is $E_{2'}$ and green is E_2).

Figure 8.4 diagrams our 16 Liouville pathways as a network of interconnected steps. Some density matrix elements, such as $\tilde{\rho}_{10}$, appear multiple times because there are several distinguishable pathways that involve that element. We do not include $\tilde{\rho}_{00}^{(1-2)}$ and $\tilde{\rho}_{00}^{(2'-2)}$ because they have exactly the same amplitude as $\tilde{\rho}_{11}^{(1-2)}$ and $\tilde{\rho}_{11}^{(2'-2)}$ within our simulation. Pathways where ρ_3 is fed by $\tilde{\rho}_{00}$ are sign-flipped to account for this.

First we define a state vector containing all nine elements in Figure 8.4:

$$\bar{\rho} \equiv \begin{bmatrix} \tilde{\rho}_{00} \\ \tilde{\rho}_{01}^{(-2)} \\ \tilde{\rho}_{10}^{(2')} \\ \tilde{\rho}_{10}^{(1)} \\ \tilde{\rho}_{20}^{(1+2')} \\ \tilde{\rho}_{11}^{(1-2)} \\ \tilde{\rho}_{11}^{(2'-2)} \\ \tilde{\rho}_{10}^{(1-2+2')} \\ \tilde{\rho}_{21}^{(1-2+2')} \end{bmatrix} \quad (8.29)$$

To write \bar{Q} , we pull most of the time dependence into a series of six variables, one for each combination of pulse (subscript) and material resonance (A for 10, B for 21).

$$A_1 \equiv \frac{i}{2} \mu_{10} e^{-i\omega_1 \tau_1} c_1(t - \tau_1) e^{i(\omega_1 - \omega_{10})t} \quad (8.30)$$

$$A_2 \equiv \frac{i}{2} \mu_{10} e^{i\omega_2 \tau_2} c_2(t - \tau_2) e^{-i(\omega_2 - \omega_{10})t} \quad (8.31)$$

$$A_{2'} \equiv \frac{i}{2} \mu_{10} e^{-i\omega_{2'} \tau_{2'}} c_{2'}(t - \tau_{2'}) e^{i(\omega_{2'} - \omega_{10})t} \quad (8.32)$$

$$B_1 \equiv \frac{i}{2} \mu_{21} e^{-i\omega_1 \tau_1} c_1(t - \tau_1) e^{i(\omega_1 - \omega_{21})t} \quad (8.33)$$

$$B_2 \equiv \frac{i}{2} \mu_{21} e^{i\omega_2 \tau_2} c_2(t - \tau_2) e^{-i(\omega_2 - \omega_{21})t} \quad (8.34)$$

$$B_{2'} \equiv \frac{i}{2} \mu_{21} e^{-i\omega_{2'} \tau_{2'}} c_{2'}(t - \tau_{2'}) e^{i(\omega_{2'} - \omega_{21})t} \quad (8.35)$$

Each off-diagonal element in $\overline{\overline{Q}}$ has two influences determining its sign: the $\tilde{\rho}_{00}$ feeding effect discussed above and λ , + for ket-side and – for bra-side transitions. $\tilde{\rho}_{11}^{(1-2)} \xrightarrow{2'} \tilde{\rho}_{10}^{(1-2+2')}$ and $\tilde{\rho}_{11}^{(2'-2)} \xrightarrow{1} \tilde{\rho}_{10}^{(1-2+2')}$ are each doubled to account for the α and γ pathways that are overlapped due to our combination of $\tilde{\rho}_{00}$ and $\tilde{\rho}_{11}$.

$$\overline{\overline{Q}} \equiv \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -A_2 & -\Gamma_{10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ A_{2'} & 0 & -\Gamma_{10} & 0 & 0 & 0 & 0 & 0 & 0 \\ A_1 & 0 & 0 & -\Gamma_{10} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & B_1 & B_{2'} & -\Gamma_{20} & 0 & 0 & 0 & 0 \\ 0 & A_1 & 0 & -A_2 & 0 & -\Gamma_{11} & 0 & 0 & 0 \\ 0 & A_{2'} & -A_2 & 0 & 0 & 0 & -\Gamma_{11} & 0 & 0 \\ 0 & 0 & 0 & 0 & B_2 & -2A_{2'} & -2A_1 & -\Gamma_{10} & 0 \\ 0 & 0 & 0 & 0 & -A_2 & B_{2'} & B_1 & 0 & -\Gamma_{21} \end{bmatrix} \quad (8.36)$$

Note that this approach implicitly enforces our phase matching conditions and pathways. To simulate single time-orderings we simply remove elements from 8.36. For example, to isolate time ordering I:

$$\overline{\overline{Q}}_I \equiv \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\Gamma_{10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\Gamma_{10} & 0 & 0 & 0 & 0 & 0 & 0 \\ A_1 & 0 & 0 & -\Gamma_{10} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\Gamma_{20} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -A_2 & 0 & -\Gamma_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\Gamma_{11} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2A_{2'} & 0 & -\Gamma_{10} & 0 \\ 0 & 0 & 0 & 0 & 0 & B_{2'} & 0 & 0 & -\Gamma_{21} \end{bmatrix} \quad (8.37)$$

These equations are somewhat general to four wave mixing of systems like the one considered in this work. We have not taken all the simplifications that are possible in our specific system, such as $\omega_{10} = \omega_{21}$

and $\Gamma_{11} = 0$. See Section 8.3.4 for more details about our simulation, including where to find the scripts and raw output behind this work.

8.3.3 Heun method

The equations defining evolution of a density matrix element under the influence of a perturbative time-dependent coupling source can be expressed as a first order linear ordinary differential equation with the form

$$\frac{dy}{dt} = -Py + Q(t) \quad (8.38)$$

For finite time steps, Δ , numerical integration can be achieved by Taylor expansion about the current time point, t_0 :

$$y(t_0 + \Delta) = y(t_0) + \Delta \frac{d}{dt}y(t_0) + \frac{\Delta^2}{2} \frac{d^2}{dt^2}y(t_0) + \dots \quad (8.39)$$

The Taylor expansion must be truncated to compute. A first order expansion (*i.e.* the Euler method) will suffice for sufficiently small time steps. Equation 8.38 makes the first-order expansion easily evaluated:

$$y(t_0 + \Delta) \approx y(t_0) + \Delta \frac{d}{dt}y(t_0) \quad (8.40)$$

$$= y(t_0) + \Delta f(t_0, y(t_0)) \quad (8.41)$$

We have used the operator $f(t_0, y(t_0)) \equiv -Py(t_0) + Q(t_0)$ for simplicity.

We used the Heun method (also known as the improved Euler method [**BlanchardP2006.000**]), which includes the second order of Equation 8.39. By doing this we can increase the time step while maintaining the same error tolerance. For this case, a computationally cheap way to increase the order of the Taylor

expansion without explicitly evaluating the second derivative is to approximate the second derivative in terms of the first derivative:

$$\frac{d^2}{dt^2} \approx \frac{1}{\Delta} \left[\frac{d}{dt} y(t_0 + \Delta) - \frac{d}{dt} y(t_0) \right] \quad (8.42)$$

$$= \frac{1}{\Delta} [f(t_0 + \Delta, y(t_0 + \Delta)) - f(t_0, y(t_0))] \quad (8.43)$$

This substitution is problematic, however, because the value of $y(t_0 + \Delta)$ is not known—indeed it is the desired quantity. For an approximation of the second derivative we approximate $y(t_0 + \Delta)$ using the first-order expansion of y about t_0 :

$$y(t_0 + \Delta) \approx [y(t_0) + \Delta f(t_0, y(t_0))] \quad (8.44)$$

Note that since $y(t_0 + \Delta) = y(t_0) + \Delta f(t_0, y(t_0)) + O(\Delta^2)$, the second derivative term will be

$$\frac{\Delta^2}{2} \frac{d^2}{dt^2} y(t_0) = \frac{\Delta}{2} f(t_0, y(t_0) + \Delta f(t_0, y(t_0))) + O(\Delta^3) \quad (8.45)$$

which has the same error scaling as truncating the Taylor series at second order. With our second derivative substitution, the second-order Taylor expansion becomes

$$y(t_0 + \Delta) \approx y(t_0) + \frac{\Delta}{2} [f(t_0, y(t_0)) + f(t_0 + \Delta, y(t_0) + \Delta f(t_0, y(t_0)))] \quad (8.46)$$

$$= y(t_0) + \frac{\Delta}{2} [-Py(t_0) + Q(t_0) - P(y(t_0) + \Delta(-Py(t_0) + Q(t_0))) + Q(t_0 + \Delta)] \quad (8.47)$$

$$= y(t_0) + \frac{\Delta}{2} [Q(t_0) + Q(t_0 + \Delta) - 2Py(t_0) + P^2 \Delta y(t_0)] \quad (8.48)$$

Which is now easily evaluated.

It is not clear *a priori* that the Heun method is more efficient than the Euler method. Efficiency is the product of the number of time steps used and the amount of time spent at each point. The Heun method would presumably be able to use fewer time steps, but each point sampled requires more calculation. Inevitably, efficiency is also a question of external factors, such as how efficiently the code executes each method.

We resorted to empirical tests to verify that the Heun methods helped our implementation of the integration. First we examined the convergence of the integration techniques. Figure 8.5 shows the integration of a Liouville pathway using our software with Euler and Heun methods. The mutual convergence to the true integral value of 1 is seen with both methods. For a given error tolerance, the Heun method requires roughly half as many points as the Euler method. With this in mind, we benchmarked computations where the Euler method simulations used roughly twice as many points as the Heun method simulations. We found the Heun method to be roughly 1.9 times faster, indicating the extra computation of the Heun method had minimal effect on the total computation time.

We note that there are more advanced methods suited for integrating ordinary differential equations, such as multi-stage Runge-Kutta methods. These methods may speed up computational times even further, but we have not investigated them at this time.

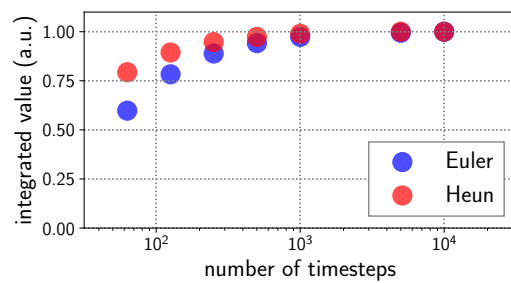


Figure 8.5: Comparison of numerical integration techniques for a Liouville pathway signal with different number of time steps.

8.3.4 Scripts & raw output

We have made the tools and raw output of this work publicly available, including our. . .

- custom simulation package
- raw output
- processing scripts
- figure creation scripts

This section contains details necessary to understand and use what we have shared. All supplementary information, including this PDF, can be found on the Open Science Framework, DOI: [10.17605/OSF.IO/EJ2XE](https://doi.org/10.17605/OSF.IO/EJ2XE).

The raw and measured simulation arrays are stored within this supplementary information as a series of Hierarchical Data Format (HDF5) files. HDF5 is an open source format supported by many programs and programming languages, see <https://www.hdfgroup.org> for more information.

To save space we compressed the arrays using the lossless compression format gzip, see <http://www.gzip.org> for more information.

If you are opening the arrays within Python we recommend using the fantastic h5py library (part of the Scipy stack) to easily decompress and import.

8.3.5 Simulation package

We have built an open source Python package for simulating CMDS. We call it Numerical Integration of Schrödinger's Equation (NISE). It is built on top of the open source SciPy library [[JonesEric2011.000](#)], and is compatible with all versions of Python 2.7 and newer.

We have included the NISE package in this supplementary information. To use NISE first install Python and SciPy if you haven't already, see <https://www.scipy.org/install.html>. Then place the NISE package into a directory within your PYTHONPATH. You should be able to `import NISE` to run and interact with simulations.

We have also built a general-purpose data analysis package called WrightTools. NISE does not require

WrightTools, but many of the processing scripts and the figure creation script used in this work do. For this reason we have included the WrightTools package in the supplementary information.

Our software packages are constantly being developed. The versions kept in this supplementary information are primarily for legacy purposes, to ensure that the processing scripts are kept in their full context. Please refer to our GitHub page, <https://github.com/wright-group>, for the most recent version of our open source software and hardware.

Raw

The raw simulation in this work was generated using NISE and the `NISE_iterator.py` script found in the raw folder.

In NISE, an `experiment` module is loaded to define the electric field variables and the experimental conditions, like the phase matching. In this case we used the `trive` module, which defines our normal, two-color four wave mixing experimental conditions.

Within this work we have represented our data in terms of dimensionless quantities like τ/Δ_t and $(\omega - \omega_{10})/\Delta_\omega$. The simulation within NISE was done with choices for each parameter, as tabulated below. These quantities are necessary to fully understand the unprocessed arrays generated by NISE.

ω_{10}	7000 cm ⁻¹
μ_{10}	1
μ_{21}	1
Γ_{11}	0 fs ⁻¹
Δ_t	50 fs

Table 8.2: Parameters used in large NISE simulation.

Γ_{10} , Γ_{21} and Γ_{20} were kept equal. Their exact value for a given run of the simulation depends on the $\Gamma_{10}\Delta_t$ quantity as discussed in the paper. We use the term `dpr` (dephasing pulse ratio) which is the inverse of $\Gamma_{10}\Delta_t$.

In NISE the system parameters are contained within the `hamiltonian` module, in this case `H0`. The `Omega` object contains all of the system attributes you would expect, including the $\bar{\rho}(t)$ (`dm_vector`) and $\bar{Q}(t)$ (`gen_matrix()`) matrices as in Equation 8.26. The matrix in the software does not account for relaxation and dephasing, that is accounted for directly during the integration. Within `H0` we actually define two \bar{Q} 'permutations', one for pathways in which E_1 arrives first (`w1_first == True`) and one for pathways in which E_2 arrives first (`w1_first == False`). This separate permutation approach is mathematically identical to the single matrix approach in Equation 8.36, just slightly more computationally expensive. `H0.Omega` allows you to define which time-orderings are included using the `T0s` keyword argument. This directly affects how the propagator is assembled, as discussed in Equation 8.37.

To generate the raw data we calculated the polarization at all coordinates within a four-dimensional experimental array:

Arrays containing these points were assembled and passed into the `trive` module. These axes and `H0` were passed into the `scan` module for numerical integration. A single output array was saved for each scan. To keep the output array sizes reasonable a separate simulation was done for each $\Gamma_{10}\Delta_t$, time-ordering, and `d1` value. For each of these simulations we saved one five-dimensional array to an HDF5 file:

The final index 'timestep' contains the dependence of the output polarization on lab time. It changes from simulation to simulation to help with computation speed. For each simulation the timestep array is defined by a starting position (always 100 fs before the first pulse arrives) an ending position ($5 \times \tau_{10}$ fs after the last pulse arrives), and a step (4 fs for our longest dephasing time, 2 fs otherwise).

The output polarization is kept as a complex array in the lab frame.

axis	center	full width	number of points
w1	7000	3000	41
w2	7000	3000	41
d1	0	400	21
d2	0	400	21

Table 8.3: Description of four-dimensional simulation coordinates.

index	name	size
0	w1	41
1	w2	41
2	d2	21
3	permutation	2
4	timestep	variable

Table 8.4: Simulation output format (polarization).

Measured

As mentioned in the appendix, we introduce inhomogeneity by convolving the output with a distribution function on the intensity level: 'smearing'. This is done in the measurement stage. We store a separate HDF5 file for each combination of d_{pr} , time ordering, and Δ_{inhom} . Each HDF5 file contains four arrays, shown in Table 8.5

The coordinate arrays are in their native units (fs, cm^{-1}). The signal array is purely real, stored on the intensity level.

We measured the entire simulation space twice, one with and one without the monochromator bandpass filter.

Representations

We present many representations of our simulated dataset in this work. The script used to create these figures is `figures.py`, found in this supplementary information. In some cases some small additional simulation or some pre-processing step was necessary, these can be found in the 'precalculated' folder.

alias	dimensions	shape
w1	w1	(41,)
w2	w2	(41,)
d1	d1	(21,)
d2	w2	(21,)
arr	w1, w2, d1, d2	(41, 41, 21, 21)

Table 8.5: Simulation output format (measured).

8.4 Results

We now present portions of our simulated data that highlight the dependence of the spectral line shapes and transients on excitation pulse width, the dephasing rate, detuning from resonance, the pulse delay times, and inhomogeneous broadening.

8.4.1 Evolution of single coherence

It is illustrative to first consider the evolution of single coherences, $\rho_0 \xrightarrow{x} \rho_1$, under various excitation conditions. Figure 8.6 shows the temporal evolution of ρ_1 with various dephasing rates under Gaussian excitation. The value of ρ_1 differs only by phase factors between various Liouville pathways (this can be verified by inspection of 8.5 under the various conditions in Table S1), so the profiles in Figure 8.6 apply for the first interaction of any pathway. The pulse frequency was detuned from resonance so that frequency changes could be visualized by the color bar, but the detuning was kept slight so that it did not appreciably change the dimensionless product, $\Delta_t (\Gamma_f + iK_f \Omega_{fx}) \approx \Gamma_{10} \Delta_t$. In this case, the evolution demonstrates the maximum impulsive character the transient can achieve. The instantaneous frequency, $d\varphi/dt$, is defined as

$$\frac{d\varphi}{dt} = \frac{d}{dt} \tan^{-1} \left(\frac{\text{Im}(\rho_1(t))}{\text{Re}(\rho_1(t))} \right). \quad (8.49)$$

The cases of $\Gamma_{10} \Delta_t = 0(\infty)$ agree with the impulsive (driven) expressions derived in Section 8.2.1. For $\Gamma_{10} \Delta_t = 0$, the signal rises as the integral of the pulse and has instantaneous frequency close to that of the pulse (Equation 8.13), but as the pulse vanishes, the signal adopts the natural system frequency and decay rate (Equation 8.12). For $\Gamma_{10} \Delta_t = \infty$, the signal follows the amplitude and frequency of the pulse for all times (the driven limit, Equation 8.11).

The other three cases show a smooth interpolation between limits. As $\Gamma_{10} \Delta_t$ increases from the impulsive limit, the coherence within the pulse region conforms less to a pulse integral profile and more to a pulse envelope profile. In accordance, the FID component after the pulse becomes less prominent, and the instantaneous frequency pins to the driving frequency more strongly through the course of evolution.

The trends can be understood by considering the differential form of evolution (Equation 8.4), and the time-dependent balance of optical coupling and system relaxation. We note that our choices of $\Gamma_{10}\Delta_t = 2.0, 1.0,$ and 0.5 give coherences that have mainly driven, roughly equal driven and FID parts, and mainly FID components, respectively. FID character is difficult to isolate when $\Gamma_{10}\Delta_t = 2.0$.

Figure 8.7a shows the temporal evolution of ρ_1 at several values of $\Omega_{1x}/\Delta_\omega$ with $\Gamma_{10}\Delta_t = 1$. Figure 8.8 shows the Fourier domain representation of Figure 8.7a. As detuning increases, total amplitude decreases, FID character vanishes, and ρ_1 assumes a more driven character, as expected. During the excitation, ρ_1 maintains a phase relationship with the input field (as seen by the instantaneous frequency in Figure 8.7a). The coherence will persist beyond the pulse duration only if the pulse transfers energy into the system; FID evolution equates to absorption. The FID is therefore sensitive to the absorptive (imaginary) line shape of a transition, while the driven response is the composite of both absorptive and dispersive components. If the experiment isolates the latent FID response, there is consequently a narrower spectral response. This spectral narrowing can be seen in Figure 8.7a by comparing the coherence amplitudes at $t = 0$ (driven) and at $t/\Delta_t = 2$ (FID); amplitudes for all $\Omega_{fx}/\Delta_\omega$ values shown are comparable at $t = 0$, but the lack of FID formation for $\Omega_{fx}/\Delta_\omega = \pm 2$ manifests as a visibly disproportionate amplitude decay (Figure 8.9 shows explicit plots of $\rho_1(\Gamma_{fx}/\Delta_\omega$ at discrete t/Δ_t values). Many ultrafast spectroscopies take advantage of the latent FID to suppress non-resonant background, improving signal to noise. [150, 170, 159, 3]

Figure 8.8 shows how a single quantum coherence evolves with detuning in the time and frequency domain. At large detunings (2, -2), the coherence is mostly driven, taking on the excitation pulse shape in time and frequency. On resonance the coherence has significant FID character, seen as narrowing in the frequency domain. For intermediate detuning (1, -1) the coherence is a complex mixture of driven and FID components, resulting in a skewed frequency domain lineshape. Time and frequency gating of this coherence results in complex multidimensional lineshapes in the full four wave mixing experiment.

Figure 8.9 shows how the time-gated amplitude of a single quantum coherence changes as the excitation pulse is detuned. At larger detunings the coherence decays faster, resulting in a lineshape narrowing with increasing time (red to yellow). At long times, the coherence lineshape approaches the convolution of the excitation pulse (grey) with the absorptive material response (narrow black).

In driven experiments, the output frequency and line shape are fully constrained by the excitation beams. In such experiments, there is no additional information to be resolved in the output spectrum. The situation changes in the mixed domain, where E_{tot} contains FID signal that lasts longer than the pulse duration. Figure 8.7a provides insight on how frequency-resolved detection of coherent output can enhance resolution when pulses are spectrally broad. Without frequency-resolved detection, mixed-domain resonance enhancement occurs in two ways: (1) the peak amplitude increases, and (2) the coherence duration increases due to the FID transient. Frequency-resolved detection can further discriminate against detuning by requiring that the driving frequency agrees with latent FID. The implications of discrimination are most easily seen in Figure 8.7a with $\Omega_{1x}/\Delta\omega = \pm 1$, where the system frequency moves from the driving frequency to the FID frequency. When the excitation pulse frequency is scanned, the resonance will be more sensitive to detuning by isolating the driven frequency (tracking the monochromator with the excitation source).

The functional form of the measured line shape can be deduced by considering the frequency domain form of Equation 8.5 (assume $\rho_i = 1$ and $\tau_x = 0$):

$$\tilde{\rho}_f(\omega) = \frac{i\lambda_f\mu_f}{2\sqrt{2\pi}} \cdot \frac{\mathcal{F}\{c_x\}(\omega - \kappa_f\Omega_{fx})}{\Gamma_f + i\omega}, \quad (8.50)$$

where $\mathcal{F}\{c_x\}(\omega)$ denotes the Fourier transform of c_x , which in our case gives

$$\mathcal{F}\{c_x\}(\omega) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(\Delta_f\omega)^2}{4\ln^2}}. \quad (8.51)$$

For squared-law detection of ρ_f , the importance of the tracking monochromator is highlighted by two limits of Equation 8.50:

- When the transient is not frequency resolved, $\text{sig} \approx \int |\tilde{\rho}_f(\omega)|^2 d\omega$ and the measured line shape will be the convolution of the pulse envelope and the intrinsic (Green's function) response (Figure 8.7b, magenta).
- When the driven frequency is isolated, $\text{sig} \approx |\tilde{\rho}_f(\kappa_f\Omega_{fx})|^2$ and the measured line shape will give the un-broadened Green's function (Figure 8.7b, teal).

Monochromatic detection can remove broadening effects due to the pulse bandwidth. For large $\Gamma_{10}\Delta_t$

values, FID evolution is negligible at all $\Omega_{f_x}/\Delta_\omega$ values and the monochromator is not useful. Figure 8.7b shows the various detection methods for the relative dephasing rate of $\Gamma_{10}\Delta_t = 1$.

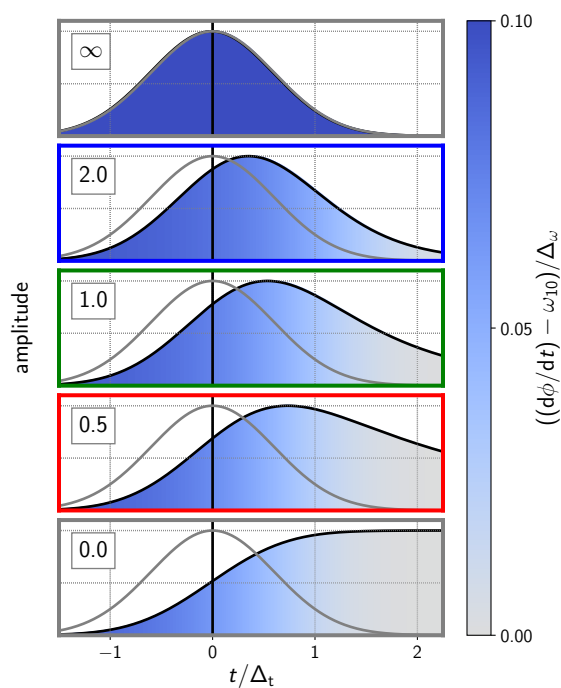


Figure 8.6: The relative importance of FID and driven response for a single quantum coherence as a function of the relative dephasing rate (values of $\Gamma_{10}\Delta_t$ are shown inset). The black line shows the coherence amplitude profile, while the shaded color indicates the instantaneous frequency (see colorbar). For all cases, the pulsed excitation field (gray line, shown as electric field amplitude) is slightly detuned (relative detuning, $\Omega_{fx}/\Delta\omega = 0.1$).

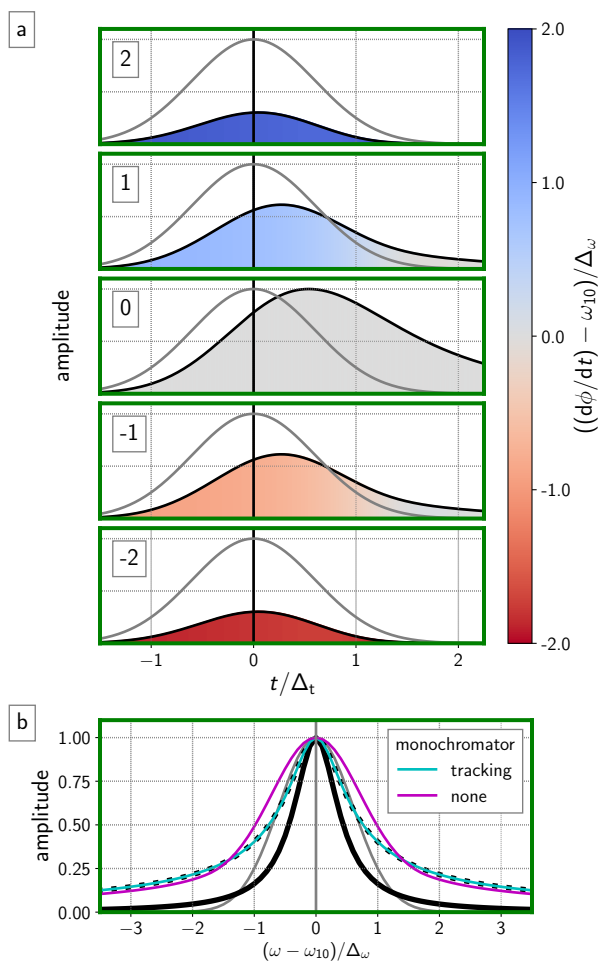


Figure 8.7: Pulsed excitation of a single quantum coherence and its dependence on the pulse detuning. In all cases, relative dephasing is kept at $\Gamma_{10}\Delta_t = 1$. (a) The relative importance of FID and driven response for a single quantum coherence as a function of the detuning (values of relative detuning, $\Omega_{\tilde{f}_X}/\Delta_\omega$, are shown inset). The color indicates the instantaneous frequency (scale bar on right), while the black line shows the amplitude profile. The gray line is the electric field amplitude. (b) The time-integrated coherence amplitude as a function of the detuning. The integrated amplitude is collected both with (teal) and without (magenta) a tracking monochromator that isolates the driven frequency components. For comparison, the Green's function of the single quantum coherence is also shown (amplitude is black, hashed; imaginary is black, solid). In all plots, the gray line is the electric field amplitude.

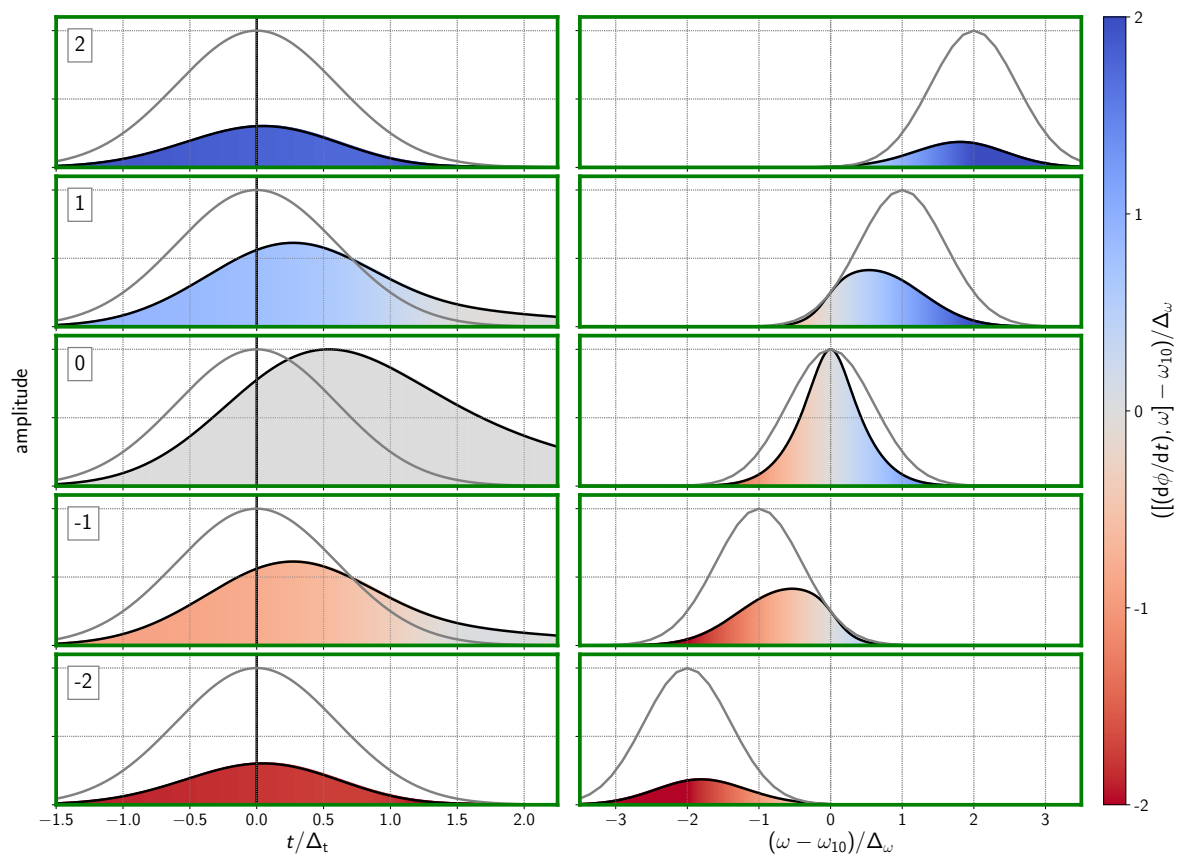


Figure 8.8: Numerical simulation of a single quantum coherence under pulsed excitation ($\Gamma_{10}\Delta_t = 1$) at different detunings (labelled inset). The coherence is shown in both the time (left column) and frequency (right column) domain. The color indicates the frequency (scale on right), while the black line shows the amplitude profile. The excitation profile is shown as a grey line.

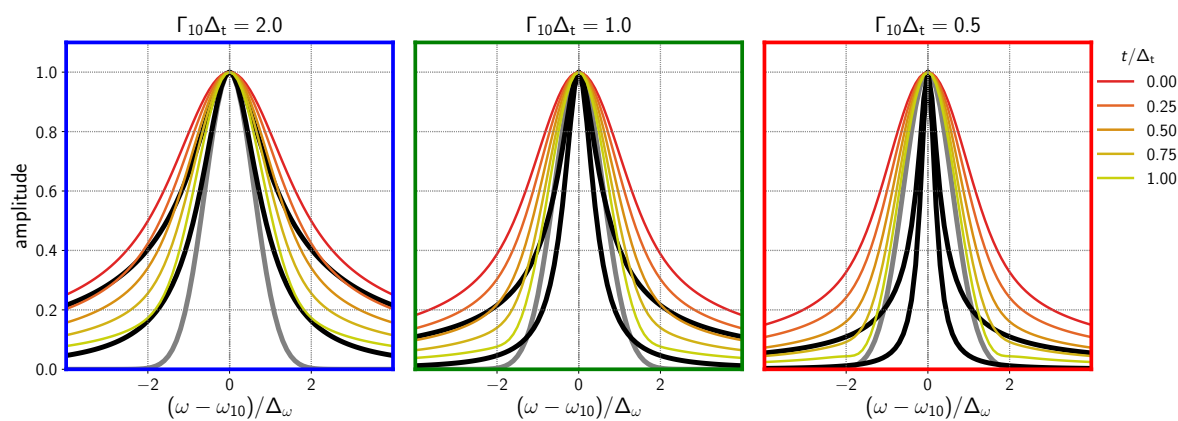


Figure 8.9: Amplitude of a single quantum coherence under pulsed excitation as a function of detuning (x axis) and delay after excitation (line color, scale on right) for the three $\Gamma_{10}\Delta_t$ values considered in this work. For comparison, the excitation pulse lineshape (grey), absorptive material response (black, narrow) and magnitude material response (black, wide) are also shown. Each peak is normalized to its own maximum amplitude.

8.4.2 Evolution of single Liouville pathway

We now consider the multidimensional response of a single Liouville pathway involving three pulse interactions. In a multi-pulse experiment, ρ_1 acts as a source term for ρ_2 (and subsequent excitations). The spectral and temporal features of ρ_1 that are transferred to ρ_2 depend on when the subsequent pulse arrives. Time-gating later in ρ_1 evolution will produce responses with FID behavior, while time-gating ρ_1 in the presence of the initial pulse will produce driven responses. An analogous relationship holds for ρ_3 with its source term ρ_2 . As discussed above, signal that time-gates FID evolution gives narrower spectra than driven-gated signal. As a result, the spectra of even single Liouville pathways will change based on pulse delays.

The final coherence will also be frequency-gated by the monochromator. The monochromator isolates signal at the fully driven frequency $\omega_{\text{out}} = \omega_1$. The monochromator will induce line-narrowing to the extent that FID takes place. It effectively enforces a frequency constraint that acts as an additional resonance condition, $\omega_{\text{out}} = \omega_1$. The driven frequency will be ω_1 if E_1 is the last pulse interaction (time-orderings V and VI), and the monochromator tracks the coherence frequency effectively. If E_1 is not the last interaction, the output frequency may not be equal to the driven frequency, and the monochromator plays a more complex role.

We demonstrate this delay dependence using the multidimensional response of the $l\gamma$ Liouville pathway as an example (see Figure 8.1). Figure 8.10 shows the resulting 2D delay profile of pathway $l\gamma$ signals for $\Gamma_{10}\Delta_t = 1$ (left) and the corresponding ω_1, ω_2 2D spectra at several pulse delay values (right). The spectral changes result from changes in the relative importance of driven and FID components. The prominence of FID signal can change the resonance conditions; Table 8.6 summarizes the changing resonance conditions for each of the four delay coordinates studied. Since E_1 is not the last pulse in pathway $l\gamma$, the tracking monochromator must also be considered. Figure 8.11 shows a simulation of Figure 8.10 without monochromator frequency filtering.

When the pulses are all overlapped ($\tau_{21} = \tau_{22'} = 0$, lower right, orange), all transitions in the Liouville pathway are simultaneously driven by the incident fields. This spectrum strongly resembles the driven limit spectrum. For this time-ordering, the first, second, and third density matrix elements have driven

resonance conditions of $\omega_1 = \omega_{10}$, $\omega_1 - \omega_2 = 0$, and $\omega_1 - \omega_2 + \omega_{2'} = \omega_{10}$, respectively. The second resonance condition causes elongation along the diagonal, and since $\omega_2 = \omega_{2'}$, the first and third resonance conditions are identical, effectively making ω_1 doubly resonant at ω_{10} and resulting in the vertical elongation along $\omega_1 = \omega_{10}$.

The other three spectra of Figure 8.10 separate the pulse sequence over time so that not all interactions are driven. At $\tau_{21} = 0$, $\tau_{22'} = -2.4\Delta_t$ (lower left, pink), the first two resonances remain the same as at pulse overlap (orange) but the last resonance is different. The final pulse, $E_{2'}$, is latent and probes ρ_2 during its FID evolution after memory of the driven frequency is lost. There are two important consequences. Firstly, the third driven resonance condition is now approximated by $\omega_{2'} = \omega_{10}$, which makes ω_1 only singly resonant at $\omega_1 = \omega_{10}$. Secondly, the driven portion of the signal frequency is determined only by the latent pulse: $\omega_{\text{out}} = \omega_{2'}$. Since our monochromator gates ω_1 , we have the detection-induced correlation $\omega_1 = \omega_{2'}$. The net result is double resonance along $\omega_1 = \omega_2$, and the vertical elongation of pulse overlap is strongly attenuated.

At $\tau_{21} = 2.4\Delta_t$, $\tau_{22'} = 0$ (upper right, purple), the first pulse E_1 precedes the latter two, which makes the two resonance conditions for the input fields $\omega_1 = \omega_{10}$ and $\omega_2 = \omega_{10}$. The signal depends on the FID conversion of ρ_1 , which gives vertical elongation at $\omega_1 = \omega_{10}$. Furthermore, ρ_1 has no memory of ω_1 when E_2 interacts, which has two important implications. First, this means the second resonance condition $\omega_1 = \omega_2$ and the associated diagonal elongation is now absent. Second, the final output polarization frequency content is no longer functional of ω_1 . Coupled with the fact that E_2 and $E_{2'}$ are coincident, so that the final coherence can be approximated as driven by these two, we can approximate the final frequency as $\omega_{\text{out}} = \omega_{10} - \omega_2 + \omega_{2'} = \omega_{10}$. Surprisingly, the frequency content of the output is strongly independent of all pulse frequencies. The monochromator narrows the $\omega_1 = \omega_{10}$ resonance. The $\omega_1 = \omega_{10}$ resonance condition now depends on the monochromator slit width, the FID propagation of ρ_1 , the spectral bandwidth of ρ_3 ; its spectral width is not easily related to material parameters. This resonance demonstrates the importance of the detection scheme for experiments and how the optimal detection can change depending on the pulse delay time.

Finally, when all pulses are well-separated ($\tau_{21} = -\tau_{22'} = 2.4\Delta_t$, upper left, cyan), each resonance condition is independent and both E_1 and E_2 require FID buildup to produce final output. The resulting

line shape is narrow in all directions. Again, the emitted frequency does not depend on ω_1 , yet the monochromator resolves the final coherence at frequency ω_1 . Since the driven part of the final interaction comes from $E_{2'}$, and since the monochromator track ω_1 , the output signal will increase when $\omega_1 = \omega_{2'}$. As a result, the line shape acquires a diagonal character.

The changes in line shape seen in Figure 8.10 have significant ramifications for the interpretations and strategies of MR-CMDS in the mixed domain. Time-gating has been used to isolate the 2D spectra of a certain time-ordering [90, 15, 147], but here we show that time-gating itself causes significant line shape changes to the isolated pathways. The phenomenon of time-gating can cause frequency and delay axes to become functional of each other in unexpected ways.

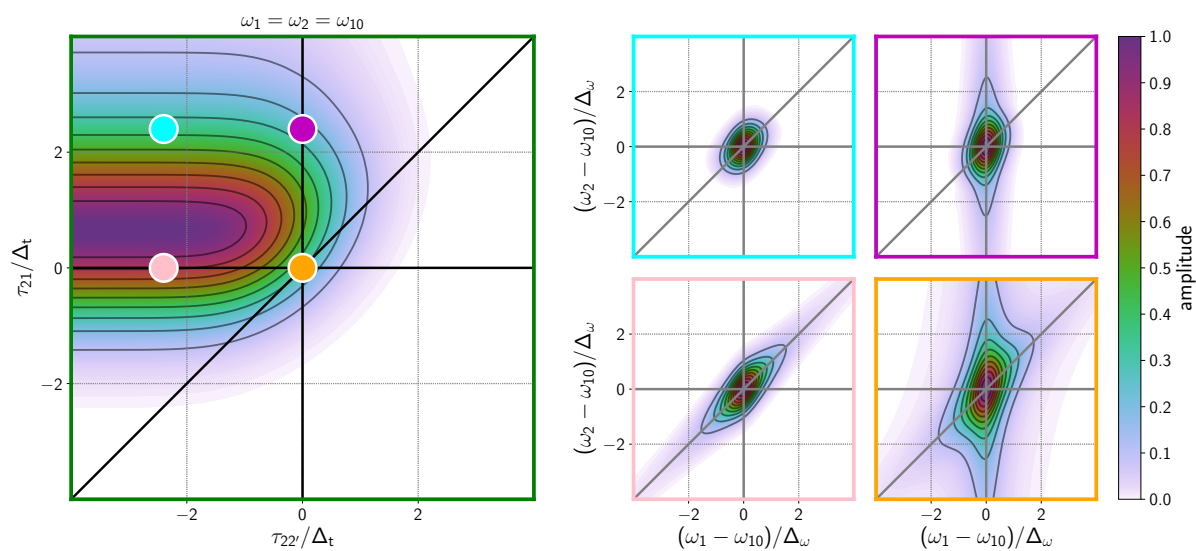


Figure 8.10: Changes to the 2D frequency response of a single Liouville pathway (I_γ) at different delay values. The normalized dephasing rate is $\Gamma_{10}\Delta_t = 1$. Left: The 2D delay response of pathway I_γ at triple resonance. Right: The 2D frequency response of pathway I_γ at different delay values. The delays at which the 2D frequency plots are collected are indicated on the delay plot; compare 2D spectrum frame color with dot color on 2D delay plot.

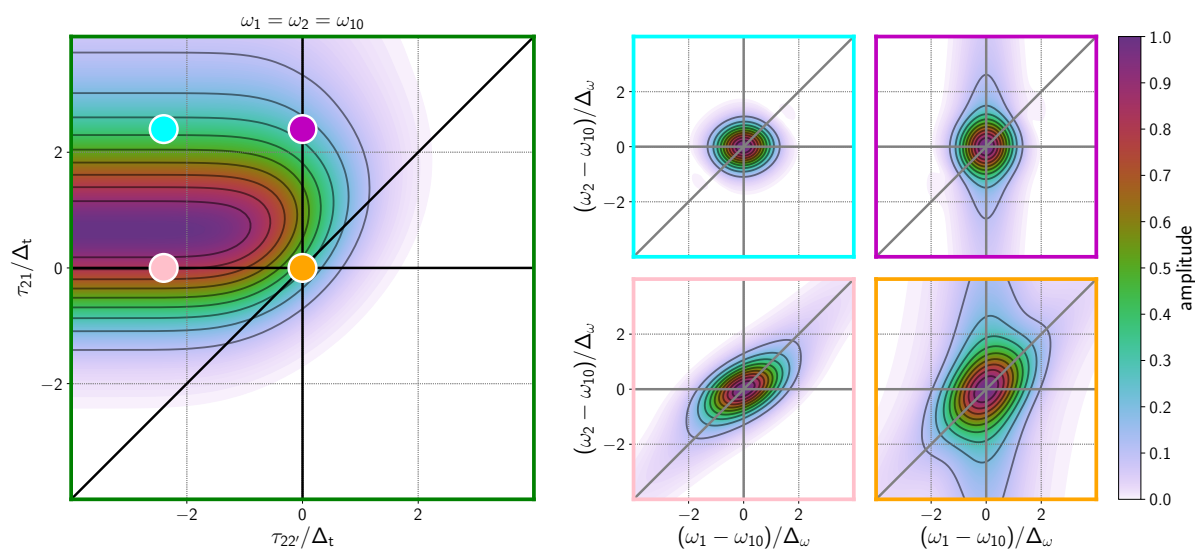


Figure 8.11: Pathway I_γ temporal response in the 2D pulse delay space at triple resonance (left) and the corresponding 2D frequency plots at different delay values. The delays at which the 2D frequency plots are collected are indicated on the delay plot; compare 2D spectrum frame color with dot color on 2D delay plot. Unlike elsewhere in this work, signal here was not filtered by a tracking monochromator.

Delay		Approximate Resonance Conditions			
τ_{21}/Δ_t	$\tau_{22'}/\Delta_t$	$\rho_0 \xrightarrow{1} \rho_1$	$\rho_1 \xrightarrow{2} \rho_2$	$\rho_2 \xrightarrow{2'} \rho_3$	$\rho_3 \rightarrow$ detection at $\omega_m = \omega_1$
0	0	$\omega_1 = \omega_{10}$	$\omega_1 = \omega_2$	$\omega_1 = \omega_{10}$	–
0	-2.4	$\omega_1 = \omega_{10}$	$\omega_1 = \omega_2$	$\omega_2 = \omega_{10}$	$\omega_1 = \omega_2$
2.4	0	$\omega_1 = \omega_{10}$	$\omega_2 = \omega_{10}$	–	$\omega_1 = \omega_{10}$
2.4	-2.4	$\omega_1 = \omega_{10}$	$\omega_2 = \omega_{10}$	$\omega_2 = \omega_{10}$	$\omega_1 = \omega_2$

Table 8.6: Conditions for peak intensity at different pulse delays for pathway I γ .

8.4.3 Temporal pathway discrimination

In the last section we showed how a single pathway's spectra can evolve with delay due to pulse effects and time gating. In real experiments, evolution with delay is further complicated by the six time-orderings/sixteen pathways present in our three-beam experiment (see Figure 8.1). Each time-ordering has different resonance conditions. When signal is collected near pulse overlap, multiple time-orderings contribute. To identify these effects, we start by considering how strongly time-orderings are isolated at each delay coordinate.

While the general idea of using time delays to enhance certain time-ordered regions is widely applied, quantitation of this discrimination is rarely explored. Because the temporal profile of the signal is dependent on both the excitation pulse profile and the decay dynamics of the coherence itself, quantitation of pathway discrimination requires simulation.

Figure 8.12 shows the 2D delay space with all pathways present for $\omega_1, \omega_2 = \omega_{10}$. It illustrates the interplay of pulse width and system decay rates on the isolation of time-ordered pathways. The color bar shows the signal amplitude. Signal is symmetric about the $\tau_{21} = \tau_{22'}$ line because when $\omega_1 = \omega_2$, E_1 and $E_{2'}$ interactions are interchangeable: $S_{\text{tot}}(\tau_{21}, \tau_{22'}) = S_{\text{tot}}(\tau_{22'}, \tau_{21})$. The overlaid black contours represent signal "purity," P , defined as the relative amount of signal that comes from the dominant pathway at that delay value:

$$P(\tau_{21}, \tau_{22'}) = \frac{\max \{S_L(\tau_{21}, \tau_{22'})\}}{\sum_L S_L(\tau_{21}, \tau_{22'})}. \quad (8.52)$$

The dominant pathway ($\max \{S_L(\tau_{21}, \tau_{22'})\}$) at given delays can be inferred by the time-ordered region defined in Figure 8.2d. The contours of purity generally run parallel to the time-ordering boundaries with the exception of time-ordered regions II and IV, which involve the double quantum coherences that have been neglected.

A commonly-employed metric for temporal selectivity is how definitively the pulses are ordered. This metric agrees with our simulations. The purity contours have a weak dependence on $\Delta_t \Gamma_{10}$ for $|\tau_{22'}|/\Delta_t < 1$ or $|\tau_{21}|/\Delta_t < 1$ where there is significant pathway overlap and a stronger dependence at larger values where the pathways are well-isolated. Because responses decay exponentially, while pulses decay as Gaussians, there always exist delays where temporal discrimination is possible. As $\Delta_t \Gamma_{10} \rightarrow \infty$, how-

ever, such discrimination is only achieved at vanishing signal intensities; the contour of $P = 0.99$ across our systems highlights this trend.

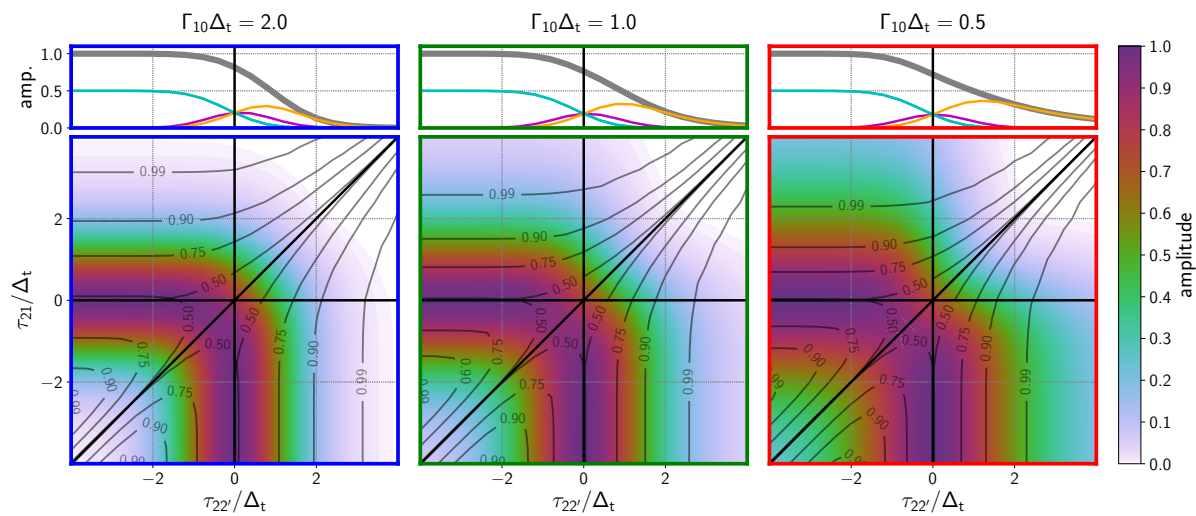


Figure 8.12: Comparison of the 2D delay response for different relative dephasing rates (labeled atop each column). All pulses are tuned to exact resonance. In each 2D delay plot, the signal amplitude is depicted by the colors. The black contour lines show signal purity, P (see Equation 8.52), with purity values denoted on each contour. The small plots above each 2D delay plot examine a τ_{22}' slice of the delay response ($\tau_{21} = 0$). The plot shows the total signal (black), as well as the component time-orderings VI (orange), V (purple), and III or I (teal).

8.4.4 Multidimensional line shape dependence on pulse delay time

In the previous sections we showed how pathway spectra and weights evolve with delay. This section ties the two concepts together by exploring the evolution of the spectral line shape over a span of τ_{21} delay times that include all pathways. It is a common practice to explore spectral evolution against τ_{21} because this delay axis shows population evolution in a manner analogous to pump-probe spectroscopies. The \vec{k}_2 and \vec{k}_2' interactions correspond to the pump, and the \vec{k}_1 interaction corresponds to the probe. Time-orderings V and VI are the normal pump-probe time-orderings, time-ordering III is a mixed pump-probe-pump ordering (so-called pump polarization coupling), and time-ordering I is the probe-pump ordering (so-called perturbed FID). Scanning τ_{21} through pulse overlap complicates interpretation of the line shape due to the changing nature and balance of the contributing time-orderings. At $\tau_{21} > 0$, time-ordering I dominates; at $\tau_{21} = 0$, all time-orderings contribute equally; at $\tau_{21} < 0$ time-orderings V and VI dominate (Figure 8.12). Conventional pump-probe techniques recognized these complications long ago, [171, 172] but the extension of these effects to MR-CMDS has not previously been done.

Figure 8.13 shows the MR-CMDS spectra, as well as histograms of the pathway weights, while scanning τ_{21} through pulse overlap. The colored histogram bars and line shape contours correspond to different values of the relative dephasing rate, $\Gamma_{10}\Delta_t$. The contour is the half-maximum of the line shape. The dependence of the line shape amplitude on τ_{21} can be inferred from Figure 8.12.

The qualitative trend, as τ_{21} goes from positive to negative delays, is a change from diagonal/compressed line shapes to much broader resonances with no correlation (ω_1 and ω_2 interact with independent resonances). Such spectral changes could be misinterpreted as spectral diffusion, where the line shape changes from correlated to uncorrelated as population time increases due to system dynamics. The system dynamics included here, however, contain no structure that would allow for such diffusion. Rather, the spectral changes reflect the changes in the majority pathway contribution, starting with time-ordering I pathways, proceeding to an equal admixture of I, III, V, and VI, and finishing at an equal balance of V and VI when E_1 arrives well after E_2 and E_2' . Time-orderings I and III both exhibit a spectral correlation in ω_1 and ω_2 when driven, but time-orderings V and VI do not. Moreover, such spectral correlation is forced near zero delay because the pulses time-gate the driven signals of the first two induced polarizations. The monochromator detection also plays a dynamic role, because time-

orderings V always VI always emit a signal at the monochromator frequency, while in time-orderings I and III the emitted frequency is not defined by ω_1 , as discussed above.

When we isolate time-orderings V and VI, we can maintain the proper scaling of FID bandwidth in the ω_1 direction because our monochromator can gate the final coherence. This gating is not possible in time-orderings I and III because the final coherence frequency is determined by ω_2 , which is identical to ω_2 .

There are differences in the line shapes for the different values of the relative dephasing rate, $\Gamma_{10}\Delta_t$. The spectral correlation at $\tau_{21}/\Delta_t = 2$ decreases in strength as $\Gamma_{10}\Delta_t$ decreases. As we illustrated in Figure 8.10, this spectral correlation is a signature of driven signal from temporal overlap of E_1 and E_2 ; the loss of spectral correlation reflects the increased prominence of FID in the first coherence as the field-matter interactions become more impulsive. This increased prominence of FID also reflects an increase in signal strength, as shown by τ_{21} traces in Figure 8.12. When all pulses are completely overlapped, ($\tau_{21} = 0$), each of the line shapes exhibit spectral correlation. At $\tau_{21}/\Delta_t = -2$, the line shape shrinks as $\Gamma_{10}\Delta_t$ decreases, with the elongation direction changing from horizontal to vertical. The general shrinking reflects the narrowing homogeneous linewidth of the ω_{10} resonance. In all cases, the horizontal line shape corresponds to the homogeneous linewidth because the narrow bandpass monochromator resolves the final ω_1 resonance. The change in elongation direction is due to the resolving power of ω_2 . At $\Gamma_{10}\Delta_t = 0.5$, the resonance is broader than our pulse bandwidth and is fully resolved vertically. It is narrower than the ω_1 resonance because time-orderings V and VI interfere to isolate only the absorptive line shape along ω_2 . This narrowing, however, is unresolvable when the pulse bandwidth becomes broader than that of the resonance, which gives rise to a vertically elongated signal when $\Gamma_{10}\Delta_t = 0.5$.

It is also common to represent data as “Wigner plots,” where one axis is delay and the other is frequency. [123, 129, 157, 122] In Figure 8.15 we show five τ_{21}, ω_1 plots for varying ω_2 with $\tau_{22'} = 0$. The plots are the analogue to the most common multidimensional experiment of Transient Absorption spectroscopy, where the non-linear probe spectrum is plotted as a function of the pump-probe delay. For each plot, the ω_2 frequency is denoted by a vertical gray line. Each Wigner plot is scaled to its own dynamic range to emphasize the dependence on ω_2 . The dramatic line shape changes between positive and negative delays can be seen. This representation also highlights the asymmetric broadening of the ω_1 line shape

near pulse overlap when ω_2 becomes non-resonant. Again, these features can resemble spectral diffusion even though our system is homogeneous.

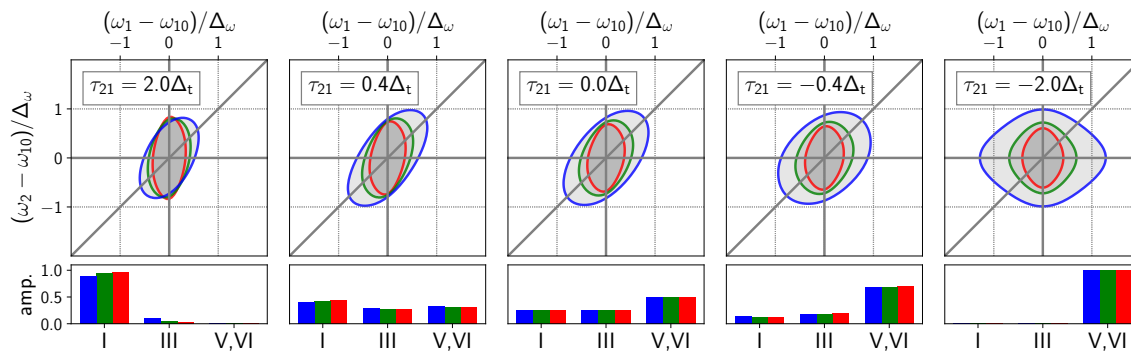


Figure 8.13: Evolution of the 2D frequency response as a function of τ_{21} (labeled inset) and the influence of the relative dephasing rate ($\Gamma_{10}\Delta_t = 0.5$ (red), 1.0 (green), and 2.0 (blue)). In all plots $\tau_{22'} = 0$. To ease comparison between different dephasing rates, the colored line contours (showing the half-maximum) for all three relative dephasing rates are overlaid. The colored histograms below each 2D frequency plot show the relative weights of each time-ordering for each relative dephasing rate. Contributions from V and VI are grouped together because they have equal weights at $\tau_{22'} = 0$.

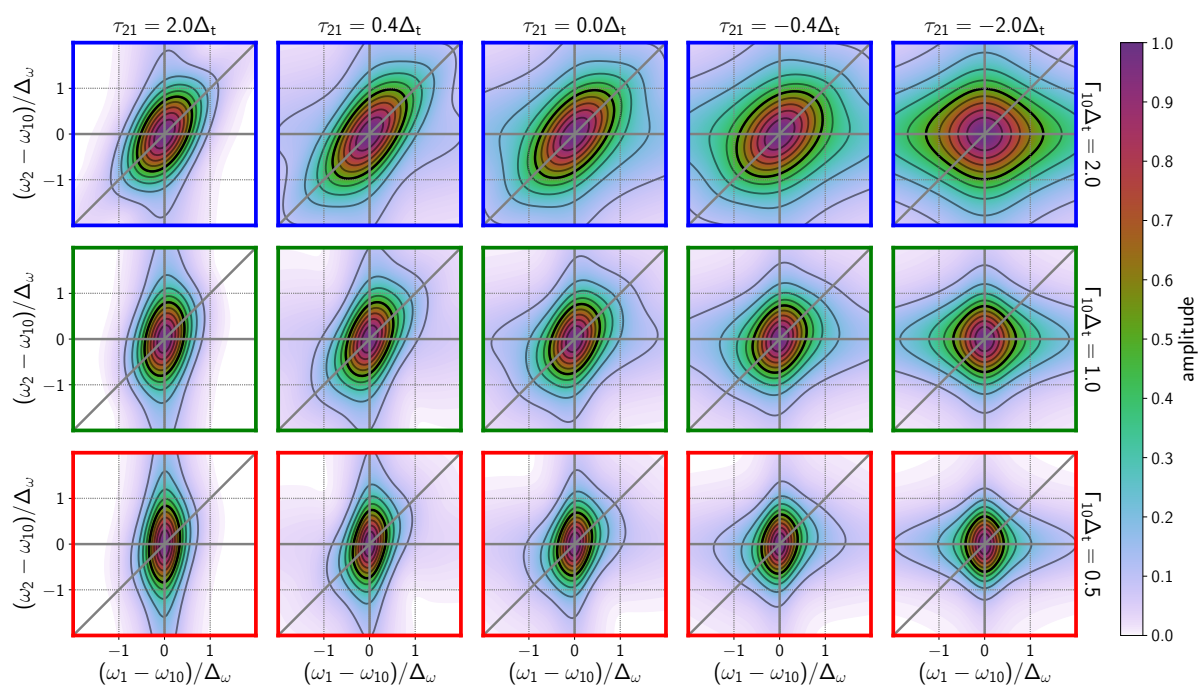


Figure 8.14: Spectral evolution of the homogeneous exciton resonance as a function of τ_{21} , with $\tau_{22'} = 0$. The 50% contour is darkened to ease comparison with Figure 7.

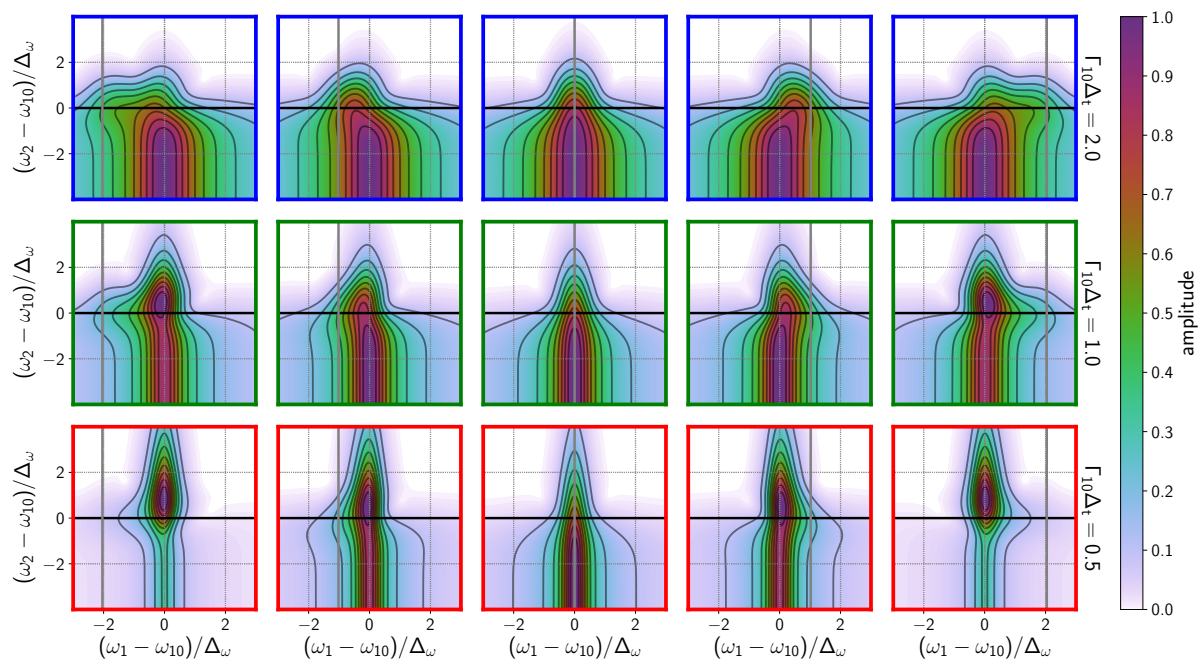


Figure 8.15: Mixed τ_{21} , ω_1 plots for each Γ_{10} value simulated in this work. For each plot, the corresponding ω_2 value is shown as a gray vertical line. Each plot is separately normalized.

8.4.5 Inhomogeneous broadening

With the homogeneous system characterized, we can now consider the effect of inhomogeneity. For inhomogeneous systems, time-orderings III and V are enhanced because their final coherence will rephase to form a photon echo, whereas time-orderings I and VI will not. In delay space, this rephasing appears as a shift of signal to time-ordered regions III and V that persists for all population times. Figure 8.17 shows the calculated spectra for relative dephasing rate $\Gamma_{10}\Delta_t = 1$ with a frequency broadening function of width $\Delta_{\text{inhom}} = 0.441\Gamma_{10}$. The inhomogeneity makes it easier to temporally isolate the rephasing pathways and harder to isolate the non-rephasing pathways, as shown by the purity contours.

A common metric of rephasing in delay space is the 3PEPS measurement. [165, 173, 174, 175] In 3PEPS, one measures the signal as the first coherence time, τ , is scanned across both rephasing and non-rephasing pathways while keeping population time, T , constant. The position of the peak is measured; a peak shifted away from $\tau = 0$ reflects the rephasing ability of the system. An inhomogeneous system will emit a photon echo in the rephasing pathway, enhancing signal in the rephasing time-ordering and creating the peak shift. In our 2D delay space, the τ trace can be defined if we assume E_2 and $E_{2'}$ create the population (time-orderings V and VI). The trace runs parallel to the III-V time-ordering boundary (diagonal) if $\tau_{22'} < 0$ and runs parallel to the IV-VI time-ordering boundary (horizontal) if $\tau_{22'} > 0$, and both intersect at $\tau_{22'} = 0$; the $-\tau_{21}$ value at this intersection is T . In our 2D delay plots (Figure 8.12, Figure 8.17), the peak shift is seen as the diagonal displacement of the signal peak from the $\tau_{21} = 0$ vertical line. Figure 8.17 highlights the peak shift profile as a function of population time with the yellow trace; it is easily verified that our static inhomogeneous system exhibits a non-zero peak shift value for all population times.

The unanticipated feature of the 3PEPS analysis is the dependence on T . Even though our inhomogeneity is static, the peak shift is maximal at $T = 0$ and dissipates as T increases, mimicking spectral diffusion. This dynamic arises from signal overlap with time-ordering III, which uses E_2 and E_1 as the first two interactions, and merely reflects E_1 and E_2 temporal overlap. At $T = 0$, the τ trace gives two ways to make a rephasing pathway (time-orderings III and V) and only one way to make a non-rephasing pathway (time-ordering VI). This pathway asymmetry shifts signal away from $\tau = 0$ into the rephasing direction. At large T (large τ_{21}), time-ordering III is not viable and pathway asymmetry disappears.

Peak shifts imply inhomogeneity only when time-orderings V and III are minimally contaminated by each other i.e. at population times that exceed pulse overlap. This fact is easily illustrated by the dynamics of homogeneous system (Fig. Figure 8.12); even though the homogeneous system cannot rephase, there is a non-zero peak shift near $T = 0$. The contamination of the 3PEPS measurement at pulse overlap is well-known and is described in some studies, [131, 166] but the dependence of pulse and system properties on the distortion has not been investigated previously. Peak-shifting due to pulse overlap is less important when $\omega_1 \neq \omega_2$ because time-ordering III is decoupled by detuning.

In frequency space, spectral elongation along the diagonal is the signature of inhomogeneous broadening. Figure 8.19 shows the line shape changes of a Gaussian inhomogeneous distribution. All systems are broadened by a distribution proportional to their dephasing bandwidth. As expected, the sequence again shows a gradual broadening along the ω_1 axis, with a strong spectral correlation at early delays ($\tau_{21} > 0$) for the more driven signals. The anti-diagonal width at early delays (e.g. Figure 8.19, $\tau_{21} = 2.0\Delta_t$) again depends on the pulse bandwidth and the monochromator slit width. At delay values that isolate time-orderings V and VI, however, the line shapes retain diagonal character, showing the characteristic balance of homogeneous and inhomogeneous width.

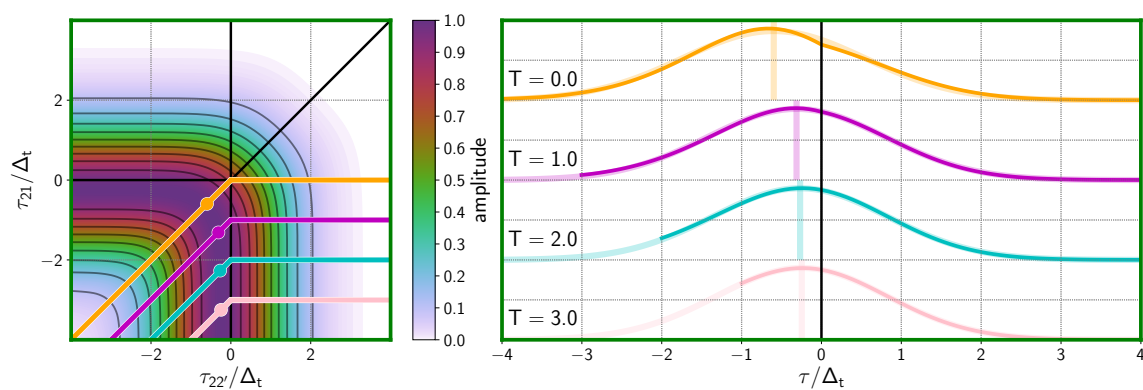


Figure 8.16: Extraction of 3PEPS peak shifts from MR-CMDS delay space. Left-hand plot: thick colored lines denote contours of constant τ for $T = 0, 1, 2, 3$. Dots indicate the fitted peak shift for each τ contour. Right-hand plot: numerically simulated amplitude traces (solid), Gaussian fits (transparent) and fit centers (vertical lines) for each T (colors matched).

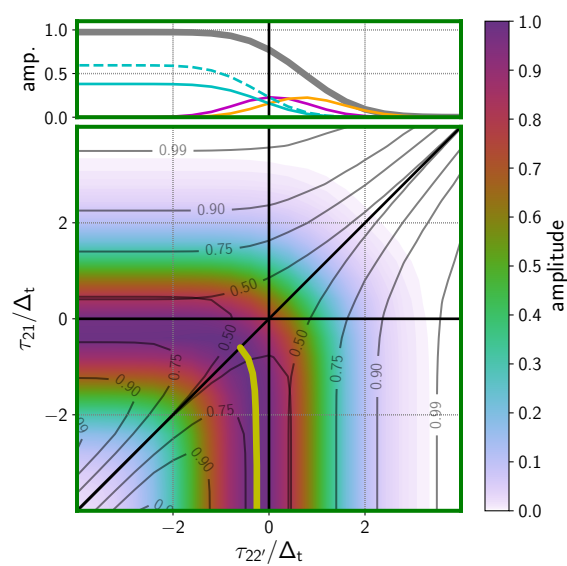


Figure 8.17: 2D delay response for $\Gamma_{10}\Delta_t = 1$ with sample inhomogeneity. All pulses are tuned to exact resonance. The colors depict the signal amplitude. The black contour lines show signal purity, P (see Equation 8.52), with purity values denoted on each contour. The thick yellow line denotes the peak amplitude position that is used for 3PEPS analysis. The small plot above each 2D delay plot examines a $\tau_{22'}$ slice of the delay response ($\tau_{21} = 0$). The plot shows the total signal (black), as well as the component time-orderings VI (orange), V (purple), III (teal, dashed), and I (teal, solid).

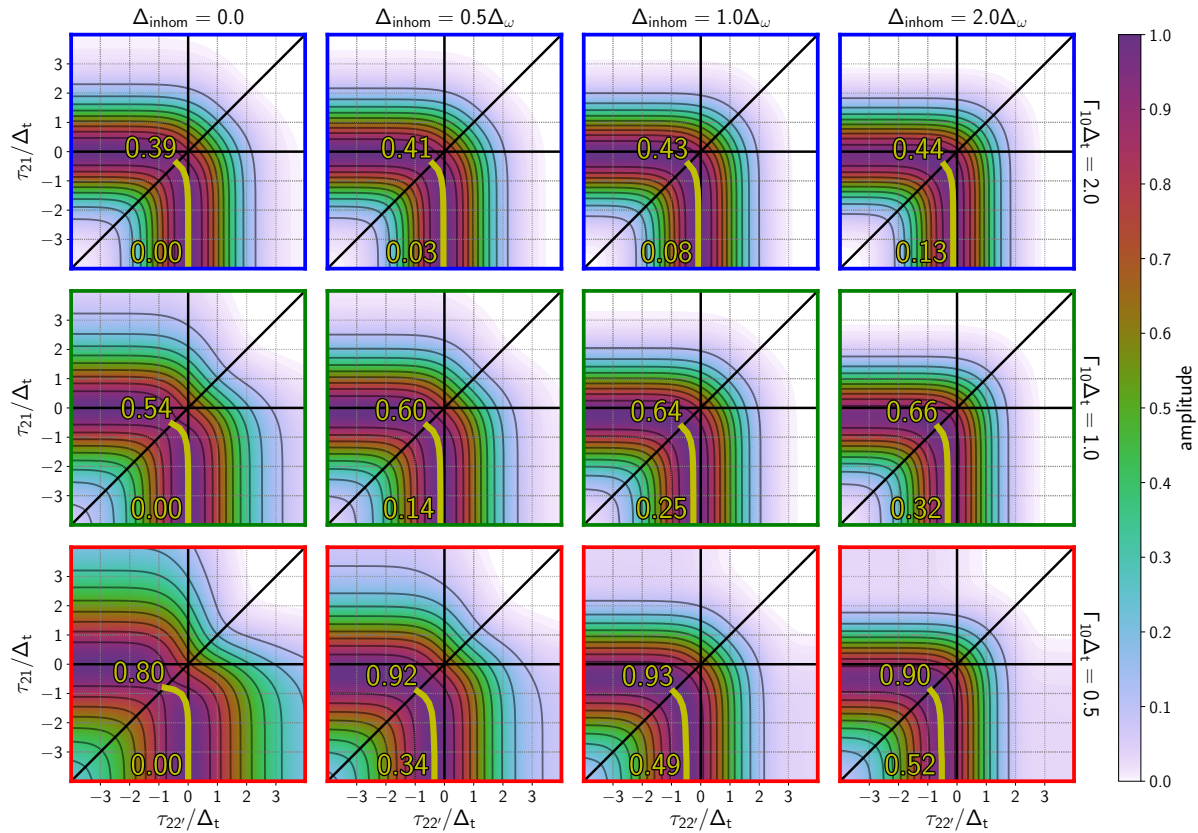


Figure 8.18: 2D delay scans at $\omega_1 = \omega_2 = \omega_{10}$ for all 12 combinations of Γ_{10} (rows) and Δ_{inhom} (columns) simulated in this work. The 3PEPS shift trace is plotted in yellow, annotated to indicate the magnitude of the τ shift at $T = 0$ and $T = 4\Delta_t$.

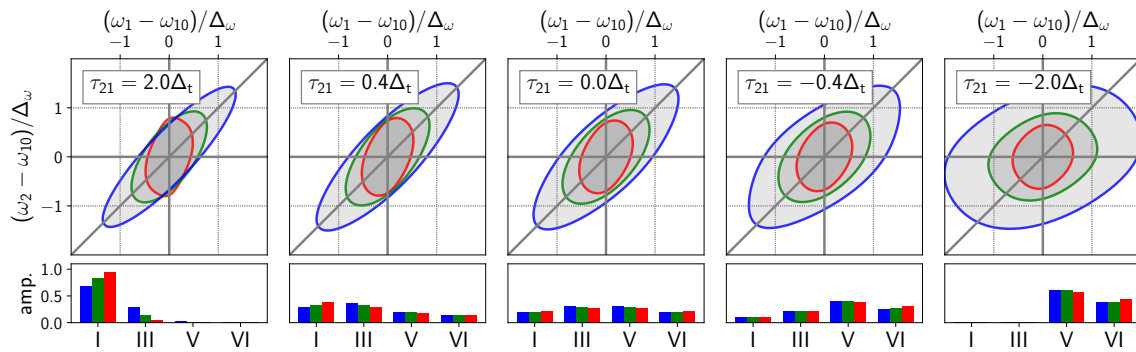


Figure 8.19: Same as Figure 8.13, but each system has inhomogeneity ($\Delta_{\text{inhom}} = 0.441\Gamma_{10}$). Relative dephasing rates are $\Gamma_{10}\Delta_t = 0.5$ (red), 1.0 (green), and 2.0 (blue). In all plots $\tau_{22'} = 0$. To ease comparison between different dephasing rates, the colored line shapes of all three systems are overlaid. Each 2D plot shows a single representative contour (half-maximum) for each $\Gamma_{10}\Delta_t$ value. The colored histograms below each 2D frequency plot show the relative weights of each time-ordering for each 2D frequency plot. In contrast to Figure 8.13, inhomogeneity makes the relative contributions of time-orderings V and VI unequal.

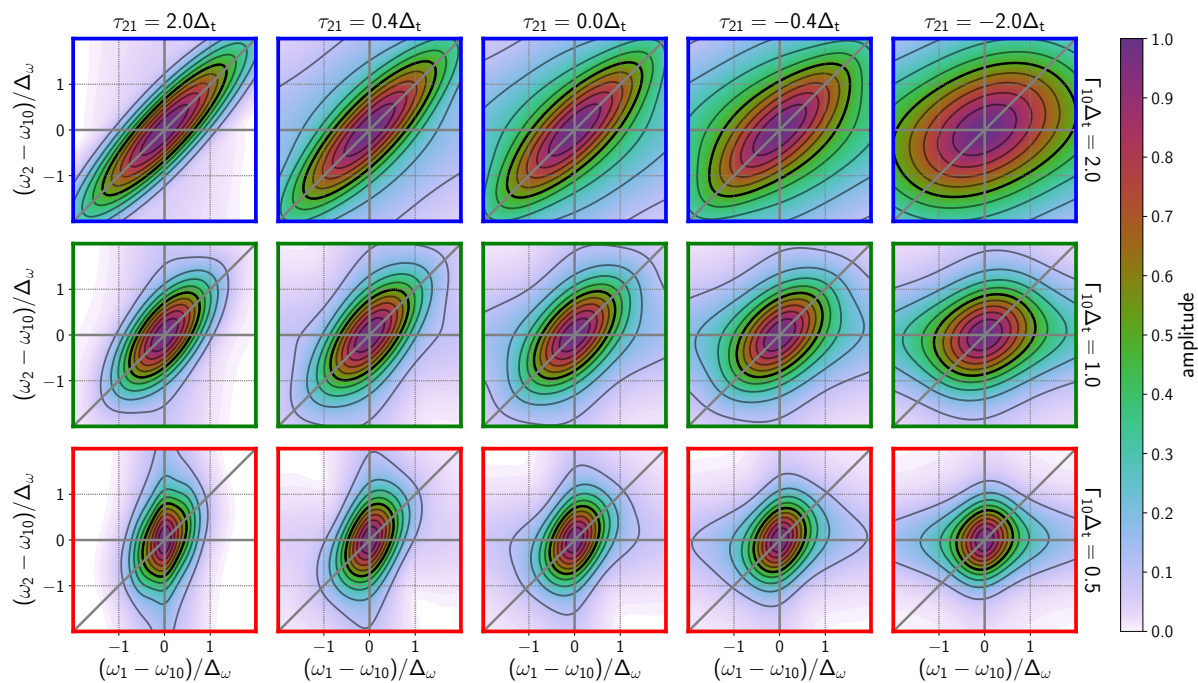


Figure 8.20: Spectral evolution of the exciton resonance as a function of τ_{21} , with $\tau_{22'} = 0$. For each system $\Delta_{inhom} = 0.441\Gamma_{10}$. The 50% contour is darkened to ease comparison with Figure 10.

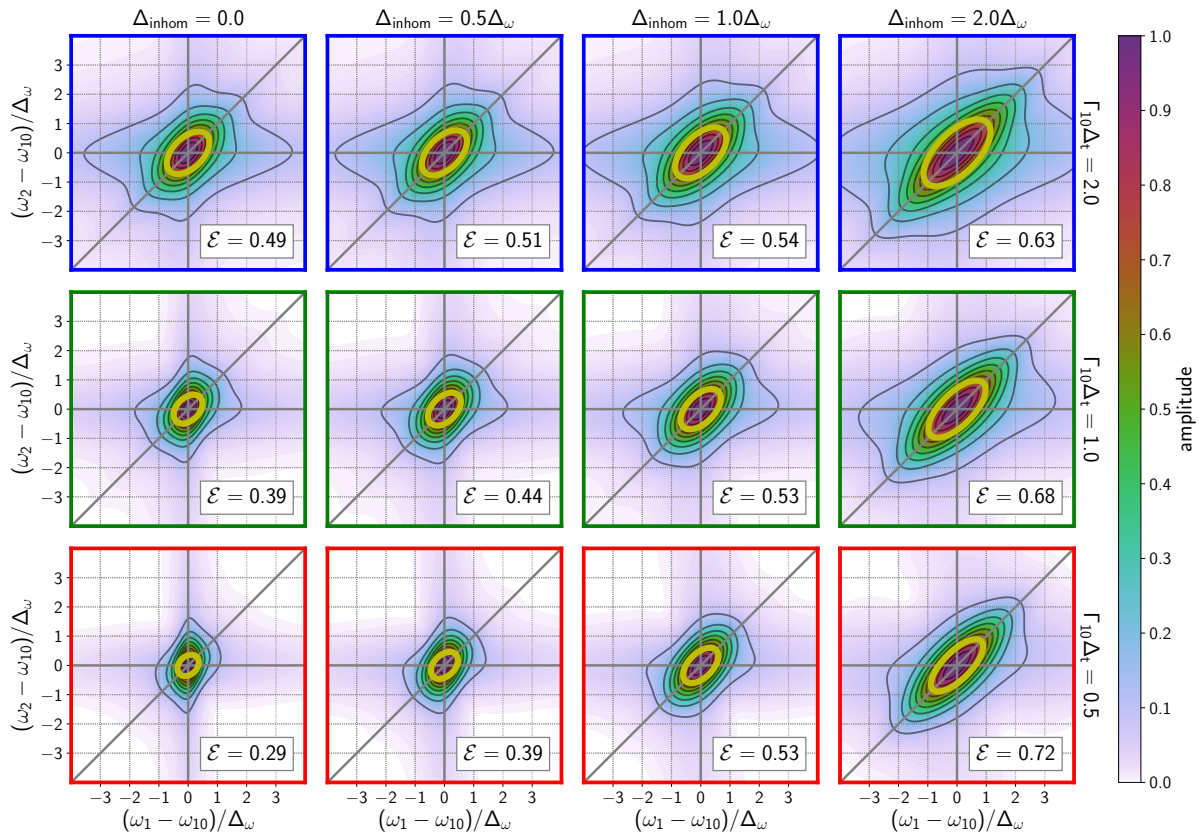


Figure 8.21: 2D frequency scans at $\tau_{21} = \tau_{22'} = 0$ for all 12 combinations of Γ_{10} (columns) and Δ_{inhom} (rows) simulated in this work. The eccentricity of each spectrum is inset and represented by the yellow ellipse (50% contour).

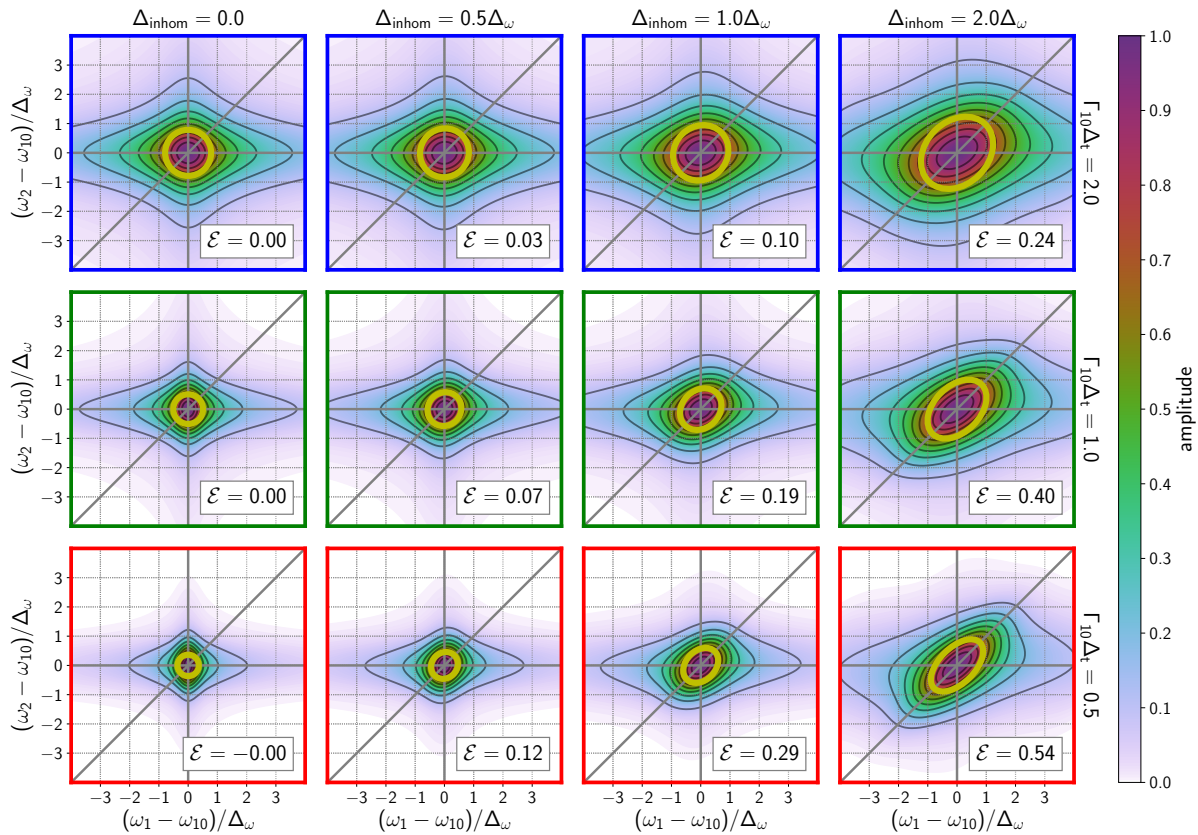


Figure 8.22: 2D frequency scans at large T ($\tau_{22'} = 0$, $\tau_{21} = -4\Delta_t$) for all 12 combinations of Γ_{10} (columns) and Δ_{inhom} (rows) simulated in this work. The eccentricity of each spectrum is inset and represented by the yellow ellipse (50% contour).

8.5 Discussion

8.5.1 An intuitive picture of pulse effects

Our chosen values of the relative dephasing time, $\Gamma_{10}\Delta_t$, describe experiments where neither the impulsive nor driven limit unilaterally applies. We have illustrated that in this intermediate regime, the multidimensional spectra contain attributes of both limits, and that it is possible to judge when these attributes apply. In our three-pulse experiment the second and third pulses time-gate coherences and populations produced by the previous pulse(s), and the monochromator frequency-gates the final coherence. Time-gating isolates different properties of the coherences and populations. Consequently, spectra evolve against delay. For any delay coordinate, one can develop qualitative line shape expectations by considering the following three principles:

1. When time-gating during the pulse, the system pins to the driving frequency with a buildup efficiency determined by resonance.
2. When time-gating after the pulse, the FID dominates the system response.
3. The emitted signal field contains both FID and driven components; the $\omega_{\text{out}} = \omega_1$ component is isolated by the tracking monochromator.

Figure 8.6 illustrates principles 1 and 2 and Figure 8.7 illustrates principle 2 and 3. Figure 8.10 provides a detailed example of the relationship between these principles and the multidimensional line shape changes for different delay times.

The principles presented above apply to a single pathway. For rapidly dephasing systems it is difficult to achieve complete pathway discrimination, as shown in Figure 8.12. In such situations the interference between pathways must be considered to predict the line shape. The relative weight of each pathway to the interference can be approximated by the extent of pulse overlap. The pathway weights exchange when scanning across pulse overlap, which creates the dramatic line shape changes observed in Figures 8.13 and 8.19.

8.5.2 Conditional validity of the driven limit

We have shown that the driven limit misses details of the line shape if $\Gamma_{10}\Delta_t \approx 1$, but we have also reasoned that in certain conditions the driven limit can approximate the response well (see principle 1). Here we examine the line shape at delay values that demonstrate this agreement. Fig. 8.23 compares the results of our numerical simulation (third column) with the driven limit expressions for populations where $\Gamma_{11}\Delta_t = 0$ (first column) or 1 (second column). The top and bottom rows compare the line shapes when $(\tau_{22'}, \tau_{21} = (0, 0))$ and $(0, -4\Delta_t)$, respectively. The third column demonstrates the agreement between the driven limit approximations with the simulation by comparing the diagonal and anti-diagonal cross-sections of the 2D spectra.

Note the very sharp diagonal feature that appears for $(\tau_{21}, \tau_{22'}) = (0, 0)$ and $\Gamma_{11} = 0$; this is due to population resonance in time-orderings I and III. This expression is inaccurate: the narrow resonance is only observed when pulse durations are much longer than the coherence time. A comparison of picosecond and femtosecond studies of quantum dot exciton line shapes (Yurs et al. [104] and Kohler et al. [123], respectively) demonstrates this difference well. The driven equation fails to reproduce our numerical simulations here because resonant excitation of the population is impulsive; the experiment time-gates only the rise time of the population, yet driven theory predicts the resonance to be vanishingly narrow ($\Gamma_{11} = 0$). In light of this, one can approximate this time-gating effect by substituting population lifetime with the pulse duration ($\Gamma_{11}\Delta_t = 1$), which gives good agreement with the numerical simulation (third column).

When $\tau_{22'} = 0$ and $\tau_{21} < \Delta_t$, signals can also be approximated by driven signal (Figure 8.23 bottom row). Only time-orderings V and VI are relevant. The intermediate population resonance is still impulsive but it depends on $\omega_{2'} - \omega_2$ which is not explored in our 2D frequency space.

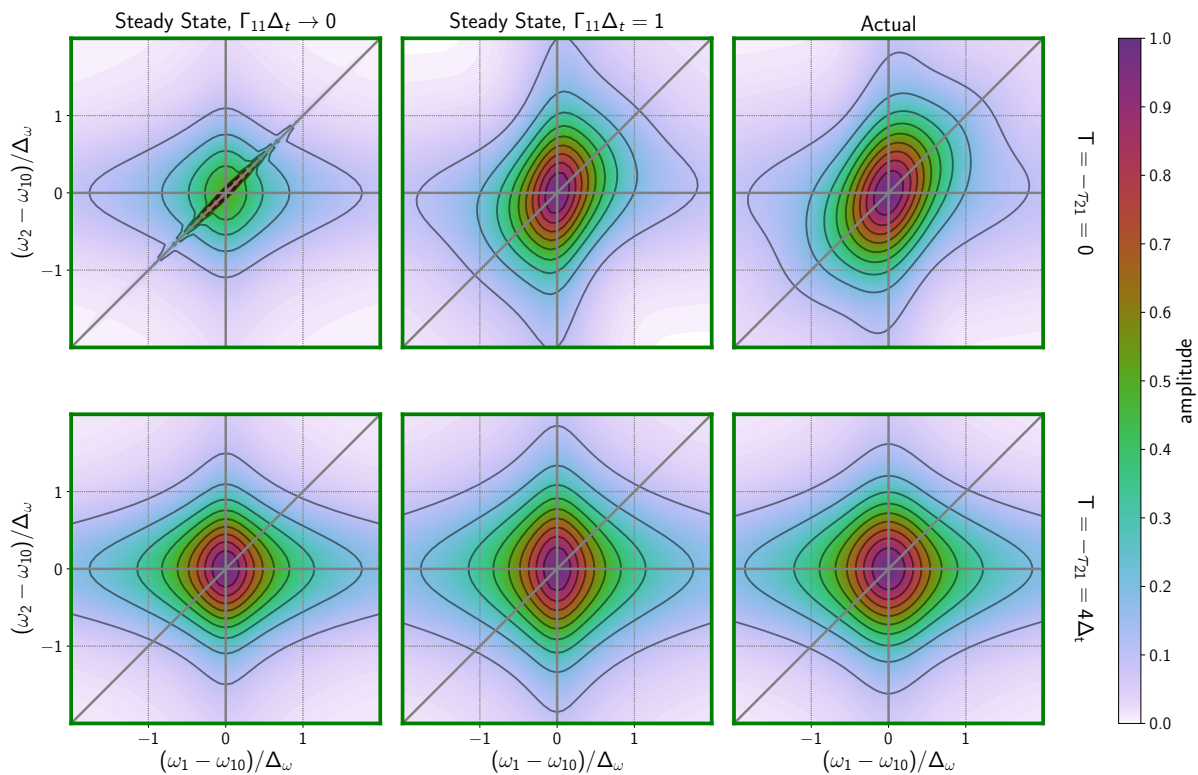


Figure 8.23: Comparing approximate expressions of the 2D frequency response with the directly integrated response. $\Gamma_{10}\Delta_t = 1$. The top row compares the 2D response of all time-orderings ($\tau_{21} = 0$) and the bottom row compares the response of time-orderings V and VI ($\tau_{21} = -4\Delta_t$). First column: The driven limit response. Note the narrow diagonal resonance for $\tau_{21} = 0$. Second column: Same as the first column, but with ad hoc substitution $\Gamma_{11} = \Delta_t$. Third column: The directly integrated response.

8.5.3 Extracting true material correlation

We have shown that pulse effects mimic the qualitative signatures of inhomogeneity. Here we address how one can extract true system inhomogeneity in light of these effects. We focus on two ubiquitous metrics of inhomogeneity: 3PEPS for the time domain and ellipticity for the frequency domain [155, 176]. There are many ways to characterize the ellipticity of a peak shape. We adopt the convention $\mathcal{E} = (a^2 - b^2) / (a^2 + b^2)$, where a is the diagonal width and b is the antidiagonal width. In the driven (impulsive) limit, ellipticity (3PEPS) corresponds to the frequency correlation function and uniquely extracts the inhomogeneity of the models presented here. In their respective limits, the metrics give values proportional to the inhomogeneity.

Figure 8.24 shows the results of this characterization for all Δ_{inhom} and $\Gamma_{10}\Delta_t$ values explored in this work. We study how the correlations between the two metrics depend on the relative dephasing rate, $\Gamma_{10}\Delta_t$, the absolute inhomogeneity, $\Delta_{\text{inhom}}/\Delta_\omega$, the relative inhomogeneity $\Delta_{\text{inhom}}/\Gamma_{10}$, and the population time delay. The top row shows the correlations of the $\Delta_{3\text{PEPS}}/\Delta_t$ 3PEPS metric that represents the normalized coherence delay time required to reach the peak intensity. The upper right graph shows the correlations for a population time delay of $T = 4\Delta_t$ that isolates the V and VI time-orderings. For this time delay, the $\Delta_{3\text{PEPS}}/\Delta_t$ metric works well for all dephasing times of $\Gamma_{10}\Delta_t$ when the relative inhomogeneity is $\Delta_{\text{inhom}}/\Delta_\omega \ll 1$. It becomes independent of $\Delta_{\text{inhom}}/\Delta_\omega$ when $\Delta_{\text{inhom}}/\Delta_\omega > 1$. This saturation results because the frequency bandwidth of the excitation pulses becomes smaller than the inhomogeneous width and only a portion of the inhomogeneous ensemble contributes to the 3PEPS experiment. [165] The corresponding graph for $T = 0$ shows a large peak shift occurs, even without inhomogeneity. In this case, the peak shift depends on pathway overlap, as discussed in Section 8.4.5.

The middle row in Figure 8.24 shows the ellipticity dependence on the relative dephasing rate and inhomogeneity assuming the measurement is performed when the first two pulses are temporally overlapped ($\tau_{22'} = 0$). For a $T = 4\Delta_t$ population time, the ellipticity is proportional to the inhomogeneity until $\Delta_{\text{inhom}}/\Delta_\omega \ll 1$ where the excitation bandwidth is wide compared with the inhomogeneity. Unlike 3PEPS, saturation is not observed because pulse bandwidth does not limit the frequency range scanned. The 3PEPS and ellipticity metrics are therefore complementary since 3PEPS works well for

$\Delta_{\text{inhom}}/\Delta\omega \ll 1$ and ellipticity works well for $\Delta_{\text{inhom}}/\Delta\omega \gg 1$. When all pulses are temporally overlapped at $T = 0$, the ellipticity is only weakly dependent on the inhomogeneity and dephasing rate. The ellipticity is instead dominated by the dependence on the excitation pulse frequency differences of time-orderings I and III that become important at pulse overlap.

It is clear from the previous discussion that both metrics depend on the dephasing and inhomogeneity. The dephasing can be measured independently in the frequency or time domain, depending upon whether the dephasing is very fast or slow, respectively. In the mixed frequency/time domain, measurement of the dephasing becomes more difficult. One strategy to address this challenge is to use both the 3PEPS and ellipticity metrics. The bottom row in Figure 8.24 plots 3PEPS against ellipticity to show how the relationship between the metrics changes for different amounts of dephasing and inhomogeneity. The anti-diagonal contours of constant relative inhomogeneity show that these metrics are complementary and can serve to extract the system correlation parameters.

Importantly, the metrics are uniquely mapped both in the presence and absence of pulse-induced effects (demonstrated by $T = 0$ and $T = 4\Delta_t$, respectively). The combined metrics can be used to determine correlation at $T = 0$, but the correlation-inducing pulse effects give a mapping significantly different than at $T = 4\Delta_t$. At $T = 0$, 3PEPS is almost nonresponsive to inhomogeneity; instead, it is an almost independent characterization of the pure dephasing. In fact, the $T = 0$ trace is equivalent to the original photon echo traces used to resolve pure dephasing rates. [177] Both metrics are offset due to the pulse overlap effects. Accordingly, the region to the left of homogeneous contour is non-physical, because it represents observed correlations that are less than that given by pulse overlap effects. If the metrics are measured as a function of T , the mapping gradually changes from the left figure to the right figure in accordance with the pulse overlap. Both metrics will show a decrease, even with static inhomogeneity. If a system has spectral diffusion, the mapping at late times will disagree with the mapping at early times; both ellipticity and 3PEPS will be smaller at later times than predicted by the change in mappings alone.

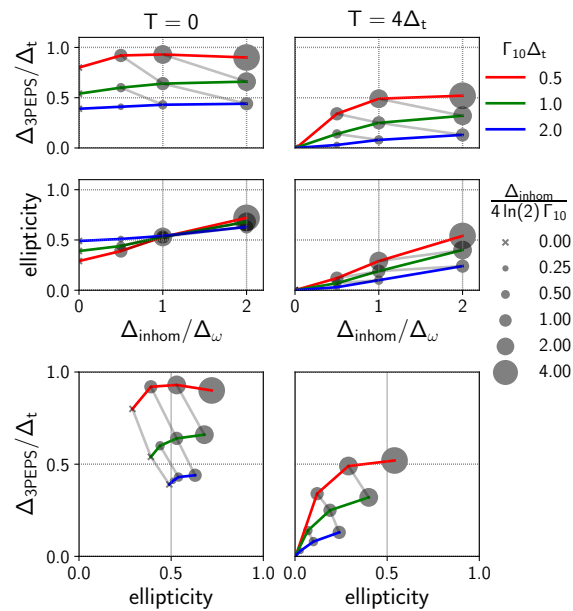


Figure 8.24: Temporal (3PEPS) and spectral (ellipticity) metrics of correlation and their relation to the true system inhomogeneity. The left column plots the relationship at pulse overlap ($T = 0$) and the right column plots the relationship at a delay where driven correlations are removed ($T = 4\Delta_t$). For the ellipticity measurements, $\tau_{22'} = 0$. In each case, the two metrics are plotted directly against system inhomogeneity (top and middle row) and against each other (bottom row). Colored lines guide the eyes for systems with equal relative dephasing rates ($\Gamma_{10}\Delta_t$, see upper legend), while the area of the data point marker indicates the relative inhomogeneity ($\Delta_{\text{inhom}}/\Gamma_{10}$, see lower legend). Gray lines indicate contours of constant relative inhomogeneity (scatter points with the same area are connected).

8.6 Conclusion

This study provides a framework to describe and disentangle the influence of the excitation pulses in mixed-domain ultrafast spectroscopy. We analyzed the features of mixed domain spectroscopy through detailed simulations of MR-CMDS signals. When pulse durations are similar to coherence times, resolution is compromised by time-bandwidth uncertainty and the complex mixture of driven and FID response. The dimensionless quantity $\Delta_t(\Gamma_f + i\kappa_f\Omega_{f_x})$ captures the balance of driven and FID character in a single field-matter interaction. In the nonlinear experiment, with multiple field-matter interactions, this balance is also controlled by pulse delays and frequency-resolved detection. Our analysis shows how these effects can be intuitive.

The dynamic nature of pulse effects can lead to misleading changes to spectra when delays are changed. When delays separate pulses, the spectral line shapes of individual pathways qualitatively change because the delays isolate FID contributions and de-emphasize driven response. When delays are scanned across pulse overlap, the weights of individual pathways change, further changing the line shapes. In a real system, these changes would all be present in addition to actual dynamics and spectral changes of the material.

Finally, we find that, in either frequency or time domain, pulse effects mimic signatures of ultrafast inhomogeneity. Even homogeneous systems take on these signatures. For mixed domain experiments, pulse effects induce spectral ellipticity and photon echo signatures, even in homogeneous systems. Driven character gives rise to pathway overlap peak shifting in the 2D delay response, which artificially produces rephasing near pulse overlap. Driven character also produces resonances that depend on $\omega_1 - \omega_2$ near pulse overlap. Determination of the homogeneous and inhomogeneous broadening at ultrashort times is only possible by performing correlation analysis in both the frequency and time domain.

Part III

Applications

Chapter 9

Resonant third-order susceptibility of PbSe quantum dots determined by standard dilution and transient grating spectroscopy

This Chapter borrows extensively from a work-in-progress publication. The authors are:

- 1. Daniel D. Kohler*
- 2. Blaise J. Thompson*
- 3. John C. Wright*

Here we detail the extraction of quantitative information from ultrafast multiresonant CMDS spectra. PbSe quantum dots are used as a model system. We investigate the severity of pulse propagation effects, as well as the increased prominence of solvent background contributions to signals. We characterize ultrafast Four-Wave Mixing signals of colloidal quantum dots using non-resonant solvent response as a standard. We show that dilution studies can be quantitatively described, and apply a robust, few-parameter fit to extract the peak susceptibility. These method should be generally applicable to solution phase systems and other CMDS techniques.

9.1 Introduction

Coherent multidimensional spectroscopy (CMDS) provides a wealth of information on the structure, energetics, and dynamics of solution phase systems. By using multiple excitations to probe resonances, CMDS elucidates correlations and couplings between electronic, vibrational, and vibronic states. It is now commonplace to interrogate such structures with femtosecond time resolution. The time resolved nature of the measurement in this regime allows characterization of transient states that are unresolvable in steady-state methods. Multiresonant CMDS (MR-CMDS) is a frequency-domain technique, whereby tunable lasers are scanned to obtain multidimensional spectra. It has been demonstrated on material systems with femtosecond pulses. [123, 157]

Traditionally CMDS employs line shape analysis (peak center, width, sign, homogeneous and inhomogeneous line width) and dynamics analysis (time constants, amplitudes) to give insight into the material of interest. The magnitudes of optical non-linearities, though commonly ignored, are intrinsic properties that also inform on microscopic properties. The microscopic mechanisms for optical nonlinearities are determined by a sequence of field-matter interactions (Liouville pathways) that depend on *linear* properties (cross-sections, etc.) with each interaction. Some methods (TSF) isolate single Liouville pathways so that the magnitude of the non-linearity is a simple product of cross-sections. Some cross sections are easily measured with conventional experiments (Raman, absorption), but others depend on exotic/inaccessible transitions. Techniques like CMDS must be employed to measure the cross-sections of these inaccessible transitions.

Other methods (2DES, TG, TA) have multiple similar Liouville pathways, such that the nonlinearity arises from imperfect cancellation between them. This lack of cancellation also is connected to the microscopic properties. For example, saturation effects, such as state-filling, have to do with the degeneracy of a transition. CMDS can explicitly measure the strengths of these saturation effects. This paper describes the measurement of state-filling.

Absolute susceptibility is not an observable with most experimental configurations. It is uncommon for CMDS spectra to obtain absolute units of susceptibility in the spectra they report. Measurements such as the z-scan [178, 10] and transient absorption, specialize in quantifying optical non-linearities,

but these methods are limited in the multidimensional space they can explore. TA cannot do 3-color non-linearities, and z-scan cannot interrogate dynamics.

Internal standards are a convenient means to quantify the non-linearity magnitude. [179] The method has also been demonstrated on MR-CMDS of quantum dots under picosecond excitation pulses. [180] These methods typically require a full characterization (phase and amplitude) of analyte spectral properties in order to relate the two signals. For CMDS methods that possess a multitude of Liouville pathways, this characterization requires a large number of parameters and can be quite complex to solve.

This work details a few-parameter extraction of the third-order susceptibility of the 1S band of PbSe quantum dots (QDs). We utilize standard additions for characterizing the absolute third-order susceptibility of resonant signals. Applying justified approximations, we extract the absolute susceptibility without explicitly modeling Liouville pathways of an excitonic manifold. We connect the common phenomenologies of optical non-linearities to the theory of state-filling. Finally, we compare our measurements with the diverse and numerous values of literature. Once pulse-length factors and propagation distortions are accounted for, we find that the measured non-linear susceptibility of these femtosecond experiments is in good agreement with previously published values.

9.2 Theory

These experiments consider the CMDS signal resulting from a chromophore resonance in a transparent solvent. We first formulate the CMDS intensity in terms of the separate contributions of solvent and solute. We then connect the well-known theory of optical bleaching of the 1S band to our measurements.

9.2.1 Extraction of susceptibility

In the Maker-Terhune convention, the relevant third-order polarization, $P^{(3)}$, is related to the non-linear susceptibility, $\chi^{(3)}$, by Maker and Terhune [181]

$$P^{(3)}(z, \omega) = D\chi^{(3)}(\omega; \omega_1, -\omega_2, \omega_{2'}) \times E_1(z, \omega_1)E_2(z, -\omega_2)E_{2'}(z, \omega_{2'}), \quad (9.1)$$

where z is the optical axis coordinate (the experiment is approximately collinear), E_i is the real-valued electric field of pulse i , and ω_i is the frequency of pulse i . The degeneracy factor $D = 3!/(3 - n)!$ accounts for the permutation symmetry that arises from the interference of n distinguishable excitation fields. $D = 6$ for transient absorption and transient grating, and $D = 3$ for z -scan. Permutation symmetry reflects the strength of the excitation fields and not the intrinsic non-linearity of the sample. Including D in our convention makes $\chi^{(3)}$ invariant to different beam geometries.

Equation 9.1 is the steady-state solution to the Liouville Equation, valid when excitation fields are greatly detuned from resonance and/or much longer than coherence times. This convention is invalid for impulsive excitation, where $\chi^{(3)}$ will be sensitive to pulse duration.

The non-linear polarization launches an output field. The intensity of this output depends on the accumulation of polarization throughout the sample. For a homogeneous material, the output intensity, I , is proportional to [24]

$$\begin{aligned} I &\propto \left| \int P^{(3)}(z, \omega) dz \right|^2 \\ &\propto \left| MP^{(3)}(0, \omega) \ell \right|^2 \\ &= |MD\chi^{(3)}E_1(0, \omega_1)E_2(0, -\omega_2)E_3(0, \omega_{2'})|^2. \end{aligned} \quad (9.2)$$

Here ℓ is the sample length and M is a frequency-dependent factor that accounts for phase mismatch and absorption effects. Phase mismatch is negligible in these experiments. For purely absorptive effects, M may be written as [24, 104]

$$M(\omega_1, \omega_2) = \frac{e^{-\alpha_1 \ell/2} (1 - e^{-\alpha_2 \ell})}{\alpha_2 \ell} \quad (9.3)$$

where $\alpha_i = \sigma_i N_{\text{QD}}$ is the absorptivity of the sample at frequency ω_i . Absorption effects disrupt the proportional relationship between I and $\chi^{(3)}$. Equation 9.2 shows that we can derive spectra free of pulse propagation effects by normalizing the output intensity by M^2 . The distortions incurred by optically thick samples are well-known and have been treated in similar CMDS experiments [20, 182, 183, 184].

For cuvettes, the sample solution is sandwiched between two transparent windows. Rather than Equation 9.2, the total polarization has three distinct homogeneous regions: the front window, the solution, and the back window. The windows each have the same thickness, ℓ_w , and susceptibility, $\chi_w^{(3)}$. The (absorption-corrected) output intensity is proportional to:

$$\frac{I}{l_1 l_2 l_2' M^2} \propto \left| \alpha_2 \ell_w \frac{1 + e^{-\alpha_2 \ell_s}}{1 - e^{-\alpha_2 \ell_s}} \chi_w^{(3)} + \chi_{\text{sol}}^{(3)} + \chi_{\text{QD}}^{(3)} \right|^2 \quad (9.4)$$

where $\chi_{\text{QD}}^{(3)}$ is the QD susceptibility and the $\chi_{\text{sol}}^{(3)}$ is the solvent susceptibility. Each susceptibility depends on the chromophore number density and local field enhancements for each wave:

$$\chi_i^{(3)} = f(\omega_1)^2 f(\omega_2)^2 N_i \gamma_i^{(3)}, \quad (9.5)$$

where $\gamma_i^{(3)}$ is the intrinsic (per-QD/per-molecule, *in vacuo*) hyperpolarizability, N_i is the number density of species i , and $f(\omega)$ is the local field enhancement factor. Since QDs constitute a negligible number/volume fraction of the solution, the field enhancement is derived entirely from the solvent: $f(\omega) = \left(\frac{n_{\text{sol}}(\omega)^2 + 2}{3} \right)$, where n_{sol} is the solvent refractive index. Both n and f are frequency dependent, but both vary small amounts ($\sim 0.1\%$) over the frequency ranges considered here. We approximate both as constants, and remove the frequency argument from further equations.

Equation 9.4 can be expressed as the classic signal-local oscillator interference,

$$I \propto |E_{\text{LO}}|^2 + N_{\text{QD}}^2 f^8 |\gamma_{\text{QD}}|^2 + 2N_{\text{QD}} f^4 \text{Re} [E_{\text{LO}} \gamma_{\text{QD}}^*] \quad (9.6)$$

where we have used the substitutions $E_{\text{LO}} = \alpha_2 \ell_w \frac{1+e^{-\alpha_2 \ell_s}}{1-e^{-\alpha_2 \ell_s}} \chi_w^{(3)} + \chi_{\text{sol}}^{(3)}$. The character of the interference depends both on the amplitude of the QD field and on the phase relationship between the two fields. The QD field amplitude can be controlled by N_{QD} . At low concentrations there is a linear dependence on N_{QD} , but this changes at high optical densities due to an α_2 dependence on the window contribution. The phase relationship cannot be externally controlled and is frequency dependent: it is defined by the resonant character of each material. The phase is defined by electronic resonances in QD and by Raman resonances in the solvent and the windows. The local oscillator and signal fields are non-additive unless the phase difference is $\pm\pi/2$.

9.2.2 Optical bleaching and dependencies on experimental conditions

Most non-linear experiments on QDs extract pulse propagation parameters, such as the non-linear absorptivity, β or non-linear index of refraction, n_2 . These parameters are connected to the third-order susceptibility (in the cgs unit system) by

$$\beta = \frac{32\pi^2 D \omega}{n_0^2 c^2} \text{Im} [\chi^{(3)}] \quad (9.7)$$

$$n_2 = \frac{16\pi^2 D}{n_0^2 c} \text{Re} [\chi^{(3)}]. \quad (9.8)$$

At the band edge, the non-linear absorptivity of semiconductor QDs is dominated by state-filling [185], where absorption is reduced in proportion to the average number of excitons pumped into the band, $\langle n \rangle$. Due to an 8-fold degeneracy, lead chalcogenide (PbX) QDs are bleached fractionally by the presence of an exciton. An 8-fold degeneracy predicts a bleach fraction of $\phi = 0.25$. [186, 187, 188, 189, 190] For a Gaussian pump pulse of peak intensity I , frequency ω , and full-width at half-maximum (FWHM) of Δ_t , $\langle n \rangle = \frac{\sqrt{2\pi}\sigma}{\hbar\omega} \Delta_t I$ where σ is the QD absorptive cross-section at frequency ω . We can then write the

non-linear change in absorptivity as

$$\begin{aligned}\beta I_2 &= -\phi \langle n \rangle \alpha \\ &= -\phi N_{\text{QD}} \frac{\sqrt{2\pi} \sigma_1 \sigma_2}{\hbar \omega} \Delta_t I_2\end{aligned}\tag{9.9}$$

where the indexes 1 and 2 denote properties of the probe and pump fields, respectively. In some techniques (e.g. z-scan), both probe and pump fields are the same, in which case the subscripts become unnecessary.

By combining Equations 9.5, 9.7, and 9.9 we can relate the bleach factor directly to the hyperpolarizability:

$$\text{Im} \left[\gamma^{(3)} \right] = -\phi \frac{\sqrt{2\pi} n^2 c^2}{32\pi^2 D f^4 \hbar \omega_1 \omega_2} \sigma_1 \sigma_2 \Delta_t.\tag{9.10}$$

Equation 9.10 will be useful for benchmarking our results because it connects our observable, γ_{QD} , with the nonlinearity of the microscopic model, ϕ .

9.3 Experimental

Sample preparation and characterization

QDs were created using a standard solution-phase technique. [191] QDs were washed in ethanol-toluene before being immersed in carbon tetrachloride (CCl_4) and stored in a nitrogen-pumped glovebox. Successive dilutions created the aliquots of different concentration used here. Aliquots were stored in 1 mm path length fused silica cuvettes with 1.25 mm thick windows.

Each aliquot was characterized by absorption spectroscopy (JASCO). The spectra are consistent between all dilutions (no agglomeration, see Supplementary Info). The 1S feature peaks at 0.937 eV and has a FWHM of 92 meV. Concentrations were extracted using Beer's law and published cross-sections. [192, 193] The peak ODs range from 0.06 to 0.86 (QD densities of $10^{16} - 10^{17} \text{ cm}^{-3}$).

Figure 9.1, top, shows the absorption spectra gathered for each of the aliquots used in this study. All spectra were baseline-subtracted to account for reflection of the cuvette. Differences in spectral properties between aliquots can be observed by normalizing each spectra to the 1S band (Figure 9.1, middle). No significant differences were observed; the small differences between aliquots near $\hbar\omega \approx 0.88 \text{ eV}$ is attributed to a small absorption feature of the fused silica cuvette, typically attributed to OH stretches). For computation of absorptive losses, the effective absorptivity of the pulses were computed. The effective absorptivity spectrum is achieved by convolving the absorption spectra with the pulse bandwidth. Figure 9.1, bottom, shows the differences between the effective absorptivity and the absorptivity of the darkest aliquot used.

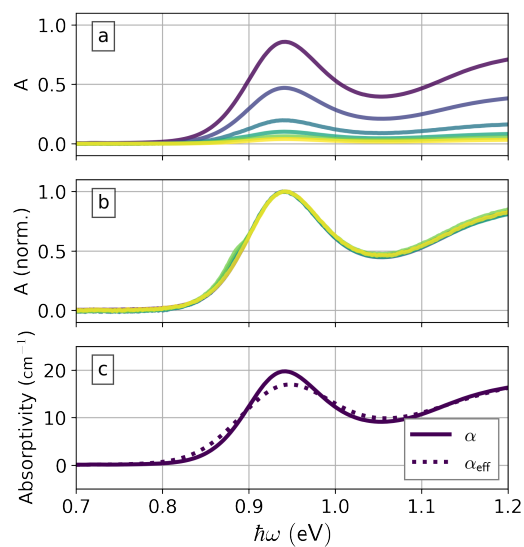


Figure 9.1: The absorption spectra of the QD aliquots used in this study. Top: the raw absorbance spectra of each aliquot used. Middle: the absorbance spectra normalized by the 1S peak. Bottom: Sample absorptivity (solid) and the effective absorption (assuming 50 meV FWHM pulse bandwidth).

Four-wave Mixing

An ultrafast oscillator (Tsunami, Spectra-Physics) produced a 80 MHz train of 35 fs pulses, which were amplified (Spitfire Pro XP, Spectra-Physics, 1kHz) and split to pump two independently tunable OPAs (TOPAS-C, Light Conversion): OPA1 and OPA2. The frequency-dependent OPA power output was measured (407-A Thermopile, Spectra-Physics) and used to normalize the non-linear spectra. Pulses from OPA2 were split again, for a total of three excitation pulses: E_1 , E_2 , and $E_{2'}$. These pulses were passed through motorized (Newport MFA-CC) retroreflectors to control their relative delays, defined as $\tau_{21} = \tau_2 - \tau_1$ and $\tau_{22'} = \tau_2 - \tau_{2'}$.

The three excitation pulses were focused (1m FL spherical mirror) into the sample using a BOXCAR geometry (~ 1 deg angle of incidence for all beams). All input fields were co-polarized. The coherent output at $k_1 - k_2 + k_{2'}$ was isolated using apertures and passed into a monochromator, with an InSb photodiode (Teledyne-Judson) at the exit slit. The monochromator was scanned with the FWM output frequency: $\omega_m = \omega_1$.

9.4 Results

In this section we describe the extraction of the QD susceptibility through standard dilution. First, we examine the window and solvent response, which will be our local oscillator, in frequency and time. Next, we isolate the pure QD response, using temporal discrimination, to validate the correction factors used to account for concentration dependence. Finally, we consider the interference between the solvent and QDs at pulse overlap, extracting the QD susceptibility by ratio.

9.4.1 Solvent and window response

Carbon tetrachloride is an ideal solvent because of the high QD solubility, transparency in the near infrared, and its well-studied non-linear properties. The FWM response of transparent solvents, like carbon tetrachloride, has components from nuclear and electronic non-linearities. [194, 195] The electronic perturbations renormalize nearly instantaneously and thus are only present with pulse overlap. The nuclear response depends on the vibrational dephasing times (ps and longer). [196, 197] Vibrational features appear in the 2D spectra when stimulated Raman pathways resonantly enhance the FWM at constant $(\omega_1 - \omega_2)$ frequencies.

Figure 9.2 summarizes the nonlinear measurements performed on neat CCl_4 . In general, our results corroborate with impulsive stimulated Raman experiments. [197, 198] When all pulses are overlapped (Figure 9.2a), the electronic response creates a featureless 2D spectrum. The horizontal and vertical structure observed in Figure 9.2a is believed to reflect the power levels of our OPAs, which were not accounted for in these scans. The weak diagonal enhancement observed may result from overdamped nuclear libration. The broad spectrum tracks with temporal pulse overlap, quickly disappearing at finite delays. If pulses E_1 and E_2 are kept overlapped and the E_2' is delayed (Figure 9.2b), the contributions from the Raman resonances can be resolved. These “TRIVE-Raman” [90] resonances have been observed in carbon tetrachloride previously. [123] The bright mode seen at approximately $\omega_1 - \omega_2 = \pm 50$ meV is the ν_1 symmetric stretch (459 cm^{-1}). [199]

Characterization of the solvent response at pulse overlap can be simplified if Raman resonances are negligible. If Raman resonances are important, their spectral phase needs to be characterized and included

in modeling. [180] To estimate the relative magnitude of Raman components at pulse overlap, we consider a delay trace. Figure 9.2c shows the signal dependence on $\tau_{22'}$ with pulse frequencies resonant with the large ν_1 resonance. The transient was fit to two components: a fast Gaussian (electronic) component and an exponential decay (Raman) component. The oscillations in the exponential decay are quantum beating between Raman modes of CCl_4 and are well-understood. [123] We determined the fast (non-resonant) component to be 4.0 ± 0.7 times larger than the long (Raman) contributions (amplitude level). At most colors, the ratio will be much less (confer Figure 9.2b). Since the Raman features are small in magnitude and spectrally sparse, we assume the CCl_4 spectrum near pulse overlap is well-approximated by non-resonant response (γ_{sol} is constant and real-valued). This simplifies Equation 9.4 because the dispersion of the interference term is determined completely by the real component of quantum dot response: $\text{Re} [E_{\text{LO}}\gamma_{\text{QD}}^*] = E_{\text{LO}}\text{Re} [\gamma_{\text{QD}}]$.

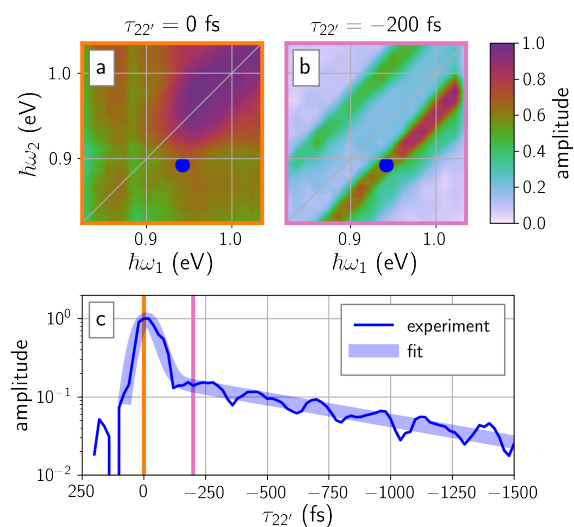


Figure 9.2: CMDS amplitude of neat CCl_4 . In all plots, E_1 and E_2 are coincident ($\tau_{21} = 0$ fs). Spectra are not normalized by the frequency-dependent OPA input powers. (a) The 2D frequency response at pulse overlap ($\tau_{21} = \tau_{22'} = 0$). (b) Same as (a), but $E_{2'}$ is latent by 200 fs. (c) The $\tau_{22'}$ dependence on CMDS amplitude (thin blue line) is tracked at $(\hbar\omega_1, \hbar\omega_2) = (.905, 0.955)$ eV, so that the ν_1 Raman mode is resonantly excited. The fit to the measured transient (thick blue line) is described further in the text. The ω_1, ω_2 frequency combination is represented in (a) and (b) as a blue dot.

9.4.2 Concentration-dependent corrections

It is important to address concentration effects on the CMDS output intensity because the resulting absorption dependence can dramatically change the signal features. Figure 9.3 shows ω_1 spectra gathered at the QD concentrations explored in this work. All spectra were gathered at delay values $\tau_{21} = -200$ fs, $\tau_{22'} = 0$ fs, and ω_2 is tuned to the exciton resonance. The pulse delays are chosen to remove all solvent and window contributions; the signal is due entirely to QDs ($\chi_w, \chi_{\text{solvent}} = 0$ in Equation 9.4). Power-normalized output amplitudes (Figure 9.3a) are positively correlated with QD concentration. Density-normalized (N_{QD}) output amplitudes (Figure 9.3b) are negatively correlated with concentration because of absorption effects. This normalization is adopted because the QD intensity term remains constant for any dilution level. This loss in efficiency is due to both absorption of the unresolved ω_2 axis (loss across all ω_1 values) and the plotted ω_1 axis (losses correlate to sample absorbance, thick grey line). After normalizing by M (Figure 9.3c), the density-normalized output amplitudes agree for all QD concentrations. The robustness of these corrections (derived from accurate absorption spectra) implies that data can be taken at large concentrations and corrected to reveal clean signal with large dynamic range. The nature of the corrected line shape, including the tail to lower energies, will be addressed in Chapter 10.

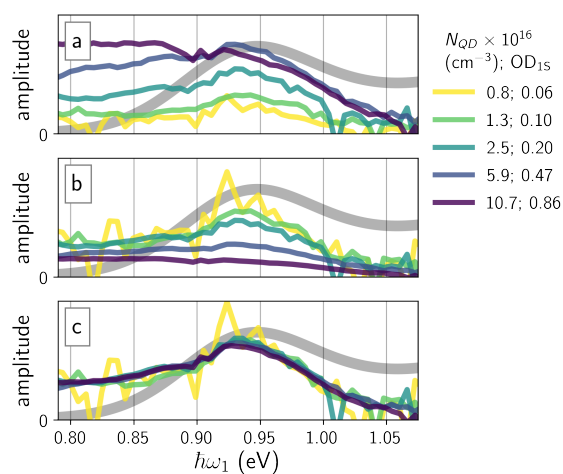


Figure 9.3: The three panels show the changes in the FWM spectra of the five QD concentrations when corrected for concentration and absorption effects. The legend at the top identifies each QD loading level by the number density (units of 10^{16} cm^{-3}). In all plots a representative QD absorption spectrum is overlaid (gray). Top: $I/l_1 l_2 l_2'$ spectra (intensity level). Middle: FWM amplitude spectra after normalizing by the carrier concentration ($\sqrt{I/(l_1 l_2 l_2' N_{\text{QD}}^2)}$). Bottom: same as middle, but with the additional normalization by the absorptive correction factor (M).

9.4.3 Quantum dot response

We now consider the behavior at pulse overlap, where solvent and window contributions are important. Figure 9.4a shows the (absorption-corrected) spectra for all samples at zero delay. The spectrum changes qualitatively with concentration, from peaked symmetric at high concentration (purple), to dispersed and antisymmetric at low concentration (yellow). This behavior contrasts with the signals at finite τ_{21} delays, where the sample spectra are independent of concentration (Figure 9.3c). Pulse overlap is complicated by the interference of multiple time-orderings and pulse effects. [112, 171, 116] These line shapes are not easily related to material properties, such as inhomogeneous broadening and pure dephasing.

The concentration dependence in Figure 9.4 can be understood with our knowledge of the solvent/window character and Equation 9.6. We approximate the solvent and window susceptibilities as real and constant, such that the frequency dependence of the interference is solely from the real projection of the QD nonlinearity. As a consequence, the interference term will be the Kramers-Kronig counterpart of the peaked transient absorption spectrum. This explains the observed antisymmetric, dispersed line shape at low concentrations. We analyze these spectra through two different methods: spectral integration and global line shape fitting.

Spectral integration

If we integrate Equation 9.4, the integral of the solvent-QD interference term disappears and the contributions are additive again. We can write

$$\int_a^{a+\Delta} \frac{I}{I_1 I_2 I_2' M^2} d\omega_1 = A\Delta \left(1 + \frac{\chi_w^{(3)}}{\chi_{\text{sol}}^{(3)}} f(N_{\text{QD}}) \right)^2 + \frac{AN_{\text{QD}}^2}{\gamma_{\text{sol}}^2 N_{\text{sol}}^2} \int_a^{a+\Delta} |\gamma_{\text{QD}}^{(3)}|^2 d\omega_1 \quad (9.11)$$

where A is a proportionality factor and $f(N_{\text{QD}}) = \sigma_2 N_{\text{QD}} \ell_w \frac{1+e^{-\sigma_2 N_{\text{QD}} \ell_s}}{1-e^{-\sigma_2 N_{\text{QD}} \ell_s}}$. Care must be taken when choosing integral bounds a and $a + \Delta$ so that the odd character of the interference is adequately destroyed.

Figure 9.4b shows the integral values for all five concentrations considered in this work (colored circles).

At high concentrations the QD intensity dominates and we see quadratic scaling with N_{QD} . The lower intensities converge to a fixed offset due to the solvent and window contributions. Our data fit well to Equation 9.11 (black dashed line).

Notably, our fit fails to distinguish between window and solvent contributions. The solvent integral is invariant to N_{QD} , while the window contribution changes only moderately over the concentrations studied ($f(N_{\text{QD}})$ varies by $\sim 0.3x$). In contrast, the QD integral will change by $\sim 100x$ over these concentration ranges, overwhelming the changes in window behavior. The approximation of $f(N_{\text{QD}})$ as constant produces equally good fits.

In order to distinguish between window and solvent contributions, we take literature values from Kerr lensing z-scan measurements of $\chi_w/\chi_{\text{sol}} \approx 0.13$. [200] This value agrees with our comparisons of FWM in cuvette windows and CCl_4 -filled cuvettes (data not shown).

The peak QD susceptibility can now be determined by assuming a line shape function; if γ_{QD} is a causal Lorentzian with half-width at half-maximum (HWHM) Γ ,

$$\gamma_{\text{QD}} = \gamma_{\text{QD,peak}} \frac{\Gamma}{\omega_{1S} - \omega_1 - i\Gamma}, \quad (9.12)$$

then $\gamma_{\text{QD, peak}} = 2\sqrt{\Gamma^{-1}\pi^{-1} \int |\gamma_{\text{QD}}|^2 d\omega_1}$. The peak width can be inferred, for instance, from spectra with high QD concentration (~ 25 meV HWHM). This gives a peak hyperpolarizability of $|\gamma_{\text{QD, peak}}| = 1.2 \times 10^6 \gamma_{\text{sol}}$.

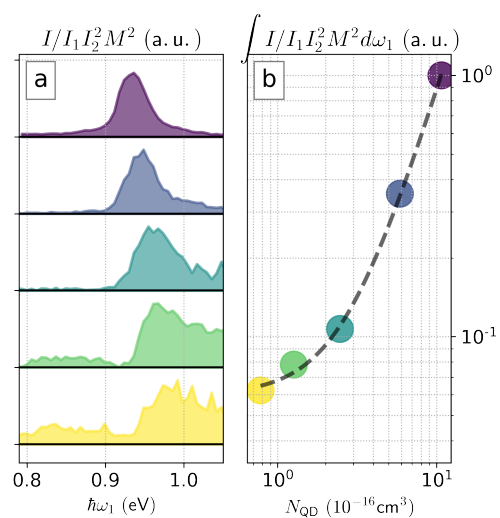


Figure 9.4: FWM at temporal pulse overlap ($\tau_{21} = \tau_{22'} = 0$ fs), with $\omega_2 = \omega_{1S}$. (a) Absorption-corrected ω_1 spectra for each of the concentrations, offset for clarity. Yellow is most dilute, purple is most concentrated. Each spectrum is individually normalized (amplification factors are shown by each spectrum). (b) The integrals of the FWM line shapes in part (a) are plotted against the QD concentration. The dashed black line is the result of a linear fit (the x-axis is logarithmic).

Global line shape fitting

The integration approach provides a simple means to separate the contributions to the non-linearity, but it relies on QDs having a purely resonant line shape. This approximation may not be appropriate for PbX QDs. Many studies have reported a broadband contribution, attributed to excited state absorption of excitons, in addition to the narrow 1S bleach feature. [180, 201, 202] To account for this feature, we perform a global fit of Equation 9.4 with the QD line shape definition

$$\gamma_{\text{QD}}^{(3)} = \gamma_{\text{QD,peak}}^{(3)} \frac{\Gamma}{\omega_{1\text{S}} - \omega_1 - i\Gamma} + B, \quad (9.13)$$

where Γ is a line width parameter and B is the broadband QD contribution. The results of the fit are overlaid with our data in Figure 9.5. The data is normalized by N_{QD}^2 (as in Figure 9.3c) so that least-squares fitting weighs all samples on similar scales. The fit parameters are listed in Table 9.1. Again, we use a literature value for χ_w/χ_{sol} . The extracted value of γ_{QD} is $\sim 35\%$ smaller than in the integral analysis because the integral method did not distinguish between the broadband contribution and the 1S state-filling feature.

Unlike the integral method, the line shape fit also gives signed information: we find that the sign of γ_{QD} is in fact negative, consistent with a photobleach. The broadband contribution has a positive imaginary component, consistent with excited state absorption.

Choice of γ_{sol}

To translate our result into an absolute susceptibility, $|\gamma_{\text{sol}}|$ must be known. There are numerous measurements in the literature, [194, 179, 203, 204, 196, 205, 206, 207, 208] but unfortunately, the variation between values is quite large ($\pm 50\%$) for quantitative analysis. This is the largest uncertainty in the determination of $\gamma_{\text{QD}}^{(3)}$. With this concern noted, we adopt the median susceptibility of $\gamma_{\text{sol}} = 4 \cdot 10^{-37} \frac{\text{cm}^6}{\text{erg}}$ as our value to give comparisons to literature. This yields a peak QD hyperpolarizability of $3 \cdot 10^{-31} \text{cm}^6/\text{erg}$.

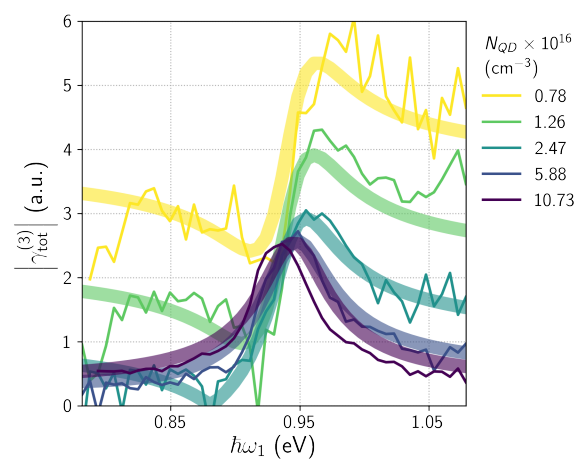


Figure 9.5: CMDS signal with different concentrations of PbSe. In all spectra $\omega_2 = \omega_{1S}$. Calculated $\gamma^{(3)}$ spectra (Equation 9.4) for the different QD concentrations. The thick, lighter lines are the result of a global fit.

variable	value
$N_{\text{sol}} \text{ (cm}^{-3}\text{)}$	6.23×10^{21}
$\hbar\omega_{1S} \text{ (eV)}$	0.945
$\sigma_2 \text{ (cm}^2\text{)}$	1.47×10^{-16} [192, 193]
χ_w/χ_{sol}	0.13 [200]
$\Gamma \text{ (meV)}$	25
$\gamma_{\text{QD,peak}}/\gamma_{\text{sol}}$	-7.7×10^5
$\mathbf{B}/\gamma_{\text{QD}}$	$0.10 - 0.13i$

Table 9.1: Parameters and extracted values from the global line shape fit using Equations 9.4 and 9.13. Bold items were extracted by least squares minimization. All other values were fixed parameters.

9.5 Discussion

We now consider the agreement of our non-linearity with those of literature. Comparison between different measured non-linearities is difficult because the effects of the excitation sources are often intertwined with the non-linear response. [112] Equation 9.10 shows that the metrics of non-linear response, ϕ and $\gamma^{(3)}$, are connected not just by intrinsic properties, but by experimental parameters D and Δ_t .

Since $\gamma^{(3)}$ assumes the driven limit for field-matter interactions (see Equation 9.1), signal scales with pulse intensity and not fluence. The third-order susceptibility will be proportional to the pulse duration of the experiment, Δ_t . Conversely, $\gamma^{(3)}$ factors out the degeneracy of the experiment, but non-linear absorptivity does not (Equation 9.7). Since ϕ is defined by the non-linear absorptivity, it is also proportional to D .

Equation 9.9 is valid for a temporally separated pump and probe, so that the probe sees the entire population created by the pump. Our experiments examine the non-linearity for temporally overlapped pump and probe pulses. The differences due to these effects can be calculated under reasonable assumptions (see the Supplementary Materials); we find the population seen at temporal overlap about 80% that of the excited state probed after the pump. This factor is needed for comparisons between our measurements and transient absorption with well separated pulses. This correction factor is small compared to our uncertainty, so we neglect it. It may be important in more precise measurements.

The most direct comparison of our measurements with literature is Yurs et al. [180], who performed the picosecond pulse analogue of the work herein. Both studies used CCl_4 as an internal standard, so we can compare the ratios directly to avoid uncertainty from the value of γ_{sol} . Note, however, that the picosecond study does not account for window contributions, which could mean their reported ratios are under-reported (the solvent field is actually the solvent and window fields). The values are shown in Table 9.2. Both Raman and 1S band susceptibilities differ by approximately the ratio of pulse durations, consistent with an intermediate state with lifetime longer than 1 ps (a Raman coherence and a 1S population, respectively).

The broadband QD hyperpolarizability (B) is similar with both pulse durations, indicating that this

contribution originates not from a 1S population, but something very fast (driven limit). Possible explanations are double/zero quantum coherences, ultrafast relaxation, or simply a non-resonant polarization. This broadband feature may be different from that observed in transient absorption because temporal pulse overlap isolates the fastest observable features (most TA features are analyzed at finite delays from pulse overlap).

Table 9.3 compares various non-linear quantities for this work, Yurs et al. [180], and a PbS experiment [9]. We will continue to refer to this table for the rest of this discussion. Note that $\gamma_{\text{QD}}/\Delta_t$ is similar between Yurs and our measurement, as expected from the pulse duration dependence.

The sample studied by Yurs et. al. was significantly degraded, and the authors described their QD spectra using mechanisms other than state-filling. The relative similarity of the absolute susceptibility, given such extraordinary spectral differences, is noteworthy. Yurs et. al. also uses a different γ_{sol} value to calculate their absolute susceptibility, which gives more disagreement in reported values than the literature suggests.

Omari et al. [9] performed z-scan measurements of PbS QDs to quantify the non-linear parameters (see right-hand column of Table 9.3). In contrast to our measurements, their degenerate susceptibility is primarily real in character and much larger than that reported here or in Yurs. While we cannot reconcile the real component, the imaginary component agrees with the standard bleach theory ($\phi = 0.15$). Omari et. al. report that their results do not agree with the $\phi = 0.25$ bleach theory of transient absorption, but we note that their observed bleach fractions is actually in great agreement once the experimental degeneracy is accounted for (a transient absorption measurement of their sample would give $\phi = 0.3$).

We now turn our focus to comparison between our measurement and ϕ . There is some variance in the value of ϕ reported for PbX quantum dots. The commonly accepted value for the bleach fraction of $\phi = 0.25$ is approximate, and runs counter to saturated absorption studies, where a fully inverted 1S band reduces by a factor of 1/8 after Auger recombination yields single-exciton species. [209, 210] Only a few transient absorption studies address the photobleach magnitude explicitly, rather than the more common state-filling analysis via the A:B ratio.

We can check our measured susceptibility with the accepted ϕ value using Equation 9.10. If the peak

susceptibility is mostly imaginary, we can attribute our TG peak $|\chi^{(3)}|$ susceptibility to the TA state-filling factor. Our peak TG hyperpolarizability measurements give values of $|\gamma^{(3)}| = 3 \pm 2 \cdot 10^{-31} \text{cm}^6 \text{erg}^{-1}$, while Equation 9.10 predicts $\text{Im} [\gamma^{(3)}] = -1.6 \cdot 10^{-31} \text{cm}^6 \text{erg}^{-1}$ for $\phi = 0.25$. Our method gives agreement with the $\phi = 0.25$ bleach factor.

	this work	Yurs et. al.	ratio
$ \gamma_{\text{QD,peak}} / \gamma_{\text{sol}} $	$7.3 \cdot 10^5$	$1.1 \cdot 10^7$	15
$ B / \gamma_{\text{sol}} $	$1.3 \cdot 10^5$	$1.6 \cdot 10^5$	1.3
$ \gamma_{\nu_1} / \gamma_{\text{sol}} $	0.25 ± 0.04	5.1	20.4

Table 9.2: Non-linear parameters relative to CCl_4 hyperpolarizability. γ_{ν_1} : hyperpolarizability of the ν_1 Raman transition.

QD measurement	this work	Yurs et. al.	Omari et. al. ¹
	PbSe $ \gamma $	PbSe $ \gamma $	PbS γ
Δ_t [fs]	~ 50	~ 1250	~ 2500
$ \gamma_{\text{QD}}^{(3)} \left[10^{-30} \frac{\text{cm}^6}{\text{erg}} \right]$	0.2	8.8	$-(1 + .03i) \cdot 10^4$
$ \gamma_{\text{QD}}^{(3)}/\Delta_t \left[10^{-18} \frac{\text{cm}^6}{\text{erg s}} \right]$	4	7	120
ϕ (Equation 9.10)	0.3	0.6	0.15

Table 9.3: Comparison of these measurements with PbX measurements in literature. $\gamma_{\text{raman}}^{(3)}$ refers to the 465 cm^{-1} (symmetric stretch) mode of CCl_4

9.6 Conclusion

We have shown that ultrafast CMDS can isolate the non-linearities of resonant signal and background solvent in manner similar to classic three-wave mixing analyses of the past. At pulse overlap, featureless solvent contributions can be especially large. The resonant Raman contributions decrease in prominence when pulses are shorter than the Raman lifetimes. Solvent contributions can also be suppressed by using large sample concentrations, in which case spectra have to be corrected by well-defined absorptive correction factors. The solute and solvent interference can be separated using simple, few parameter models, as we have demonstrated here with quantum dots. We have employed simple, few-parameter fits to easily disentangle the role of solvent and solute. These methods should be applicable to other CMDS spectroscopies, but the description of the solvent may change, especially when non-co-polarized excitations are used. [211] In fact, the polarization behavior provides a useful way to alter the balance of solvent and solute contributions in a predictable way, and is likely a viable method for separating solvent and solute contributions.

Solvent may be used as an internal standard to measure the solute non-linearity, but there are still large uncertainties in the non-linear susceptibility that propagate to the solute optical constants. More work should be done to reduce this uncertainty and characterize the dispersion of non-linear susceptibility of solvents.

Absolute nonlinearities are an important property to study in material systems because their relation to linear susceptibilities informs on the underlying physics. For MR-CMDS, it is important to identify how to extract these non-linearities. We have demonstrated ultrafast MR-CMDS as a viable method to extract the absolute non-linear susceptibility by using CCl_4 as an internal standard. The extraction requires accounting for the impulsive population creation, as well as absorptive propagation effects within the sample. These absorptive effects are also crucial factors for general analysis of MR-CMDS spectra.

Chapter 10

Global Analysis of Transient Grating and Transient Absorption of PbSe Quantum Dots

This Chapter borrows extensively from a work-in-progress publication. The authors are:

- 1. Daniel D. Kohler*
- 2. Blaise J. Thompson*
- 3. John C. Wright*

We examine the non-linear response of PbSe quantum dots about the 1S exciton using two-dimensional transient absorption and transient grating techniques. The combined analysis of both methods provides the complete amplitude and phase of the non-linear susceptibility. The phased spectra reconcile questions about the relationships between the PbSe quantum dot electronic states and the nature of nonlinearities measured by two-dimensional absorption and transient grating methods. The fits of the combined dataset reveal and quantify the presence of continuum transitions.

10.1 Introduction

Lead chalcogenide nanocrystals are among the simplest manifestations of quantum confinement [212] and provide a foundation for the rational design of nano-engineered photovoltaic materials. The time and frequency resolution capabilities of the different types of ultrafast pump-probe methods have provided the most detailed understanding of quantum dot (QD) photophysics. Transient absorption (TA) studies have dominated the literature. In a typical TA experiment, the pump pulse induces a change in the transmission of the medium that is measured by a subsequent probe pulse. The change in transmission is described by the change in the dissipative (imaginary) part of the complex refractive index, which is linked to the dynamics and structure of photoexcited species. TA does not provide information on the real-valued refractive index changes. Although the real component is less important for photovoltaic performance, it is an equal indicator of underlying structure and dynamics. In practice, having both real and imaginary components is often helpful. For example, the fully-phased response is crucial for correctly interpreting spectroscopy when interfaces are important, which is common in evaluation of materials. [213, 214, 215] The real and imaginary responses are directly related by the Kramers-Kronig relation, but it is experimentally difficult to measure the ultrafast response over the range of frequencies required for a Hilbert transform. Interferometric methods, such as two-dimensional electronic spectroscopy (2DES), can resolve both components. In this work we report a strategy for resolving fully phased spectra using frequency domain coherent multidimensional spectroscopy.

Transient grating (TG) is a pump-probe method closely related to TA. Figure 10.1 illustrates both methods. In TG, two pulsed and independently tunable excitation fields, E_1 and E_2 , are incident on a sample. The TG experiment modulates the optical properties of the sample by creating a population grating from the interference between the two crossed beams, E_2 and $E_{2'}$. The grating diffracts the E_1 probe field into a new direction defined by the phase matching condition $\vec{k}_{\text{sig}} = \vec{k}_1 - \vec{k}_2 + \vec{k}_{2'}$. In contrast, the TA experiment creates a spatially uniform excited population, but temporally modulates the ground and excited state populations with a chopper. TA can be seen as a special case of a TG experiment in which the grating fringes become infinitely spaced ($\vec{k}_2 - \vec{k}_{2'} \rightarrow \vec{0}$) and, instead of being diffracted, the nonlinear field overlaps and interferes with the probe beam (self heterodyne).

Like TA, TG does not fully characterize the non-linear response. Both imaginary and real parts of

the refractive index spatially modulate in the TG experiment. The diffracted probe is sensitive only to the total grating contrast (the response *amplitude*), and not the phase relationships of the grating. Since both techniques are sensitive to different components of the non-linear response, however, *the combination of both TA and TG can solve the fully-phased response.*

Here we report the results of dual 2DTA-2DTG experiments of PbSe quantum dots at the 1S exciton transition. We explore the three-dimensional experimental space of pump color, probe color, and population delay time. We define the important experimental factors that must be taken into account for accurate comparison of the two methods. We show that both methods exhibit reproducible spectra across different batches of different exciton sizes. Finally, we show that the methods can be used to construct a phased third-order response spectrum. Both experiments can be reproduced via simulations using the standard theory of PbSe excitons. Interestingly, the combined information reveals broadband contributions to the quantum dots non-linearity, barely distinguishable with transient absorption spectra alone. This work demonstrates TG and TA serve as complementary methods for the study of exciton structure and dynamics.

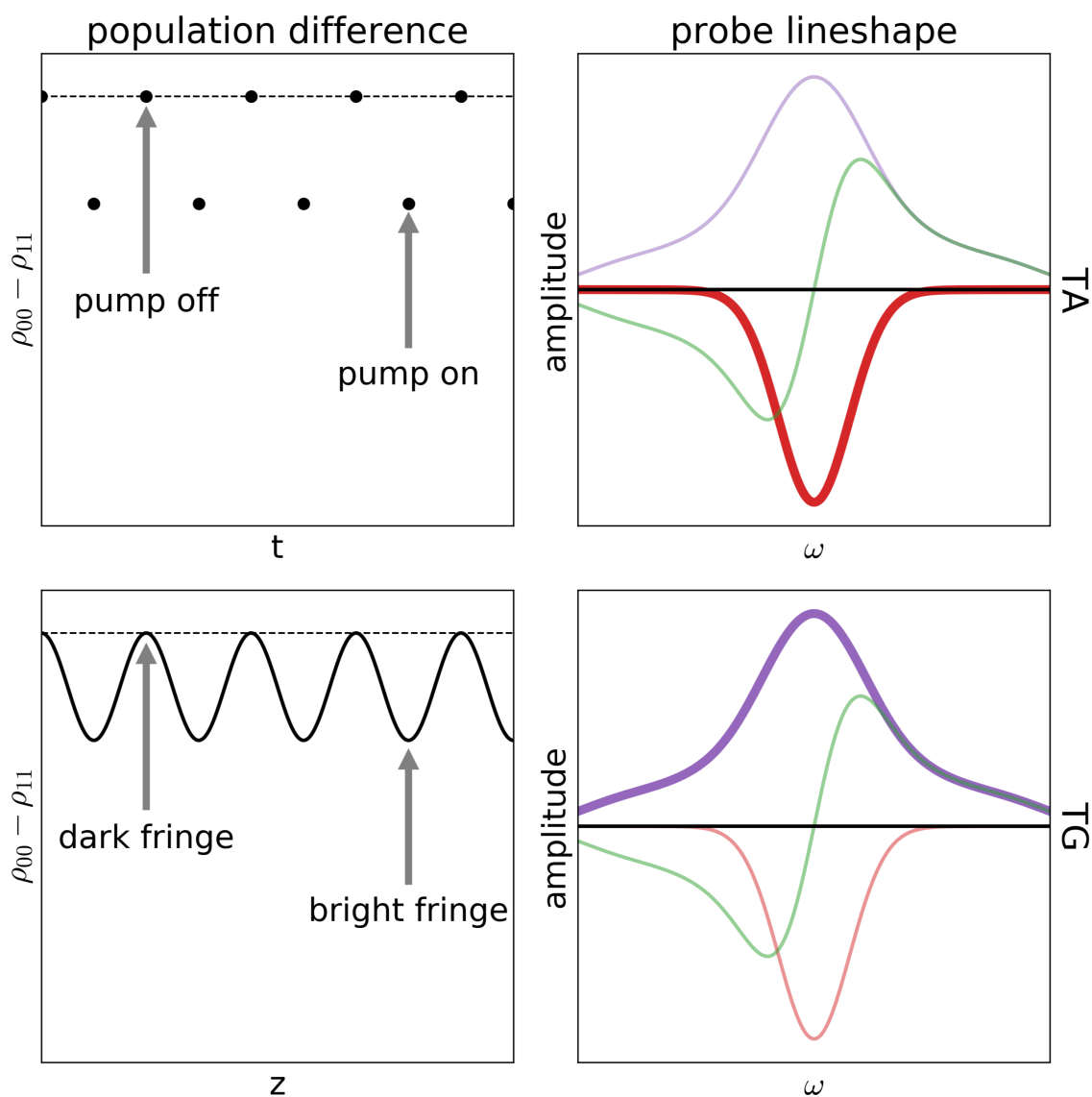


Figure 10.1: The similarities between transient grating and transient absorption measurements. Both signals are derived from creating a population difference in the sample. The intensity grating consequently spatially modulates the balance of ground state and excited state in the sample. The probe beam (E_1) is diffracted, and the diffracted intensity is measured. In transient absorption, the pump creates a monolithic population difference, which changes the attenuation the probe beam experiences through the sample. TA is sensitive to only the imaginary portion of the nonlinearity, while TG measures the total (magnitude) response.

10.2 Theory

The optical non-linearity of near-bandgap QD excitons has been extensively investigated. The response derives largely from state-filling and depends strongly on the exciton occupancy of the dots. In a PbSe quantum dot, the 1S peak is composed of transitions between 8 1S electrons and 8 1S holes. [216] Figure 10.2a shows the ground state configuration for a PbSe quantum dot. The 8-fold degeneracy means there are $8 \times 8 = 64$ states in the single-exciton ($|1\rangle$) manifold and $7 \times 7 = 49$ states in the biexciton ($|2\rangle$) manifold, so 1/4 of optical transitions are lost upon single exciton creation.

Figure 10.2 shows the model system used in this study and the parameters that control the third-order response.

We assign all electron-hole transitions the same dipole moment, μ_{eh} , so that the total cross-section between manifolds, $N_i \mu_{eh}^2$, is determined by the number optically active transitions available, N_i . Although this assumption has come under scrutiny [217, 218] it remains valid for the perturbative fluence used in this study. This model corresponds to the weakly interacting boson model, used to describe the four-wave mixing response of quantum wells, [219] in the limit of small quantum well area.

With this excitonic structure, we now describe the resulting non-linear polarization. We restrict ourselves to field-matter interactions in which the pump, E_2 , precedes the probe, E_1 (the “true” pump-probe time-ordering). Note that E_2 and $E_{2'}$ represent the same pulse in TA, but they are distinct fields ($\vec{k}_2 \neq \vec{k}_{2'}$) in TG. For brevity, we will write equations assuming these pulse parameters are interchangeable. We consider the limit of low pump fluence, so that only single absorption events need be considered: $\text{Tr}[\rho] = (1 - \bar{n})\rho_{00} + \bar{n}\rho_{11}$, where $\bar{n} \ll 1$ is the (average) fractional conversion of population. In this limit, $\bar{n} = \frac{\sigma_0(\omega_2)}{\hbar\omega_2} \int I_2(t)dt$, where $I_2(t)$ is the pump intensity and σ_0 is the ground state absorptive cross-section. For a Gaussian temporal profile of standard deviation Δ_t , $I_2(t) = I_{2,\text{peak}} \exp(-t^2/2\Delta_t^2)$, the exciton population is

$$\bar{n} = \frac{\sigma_0(\omega_2)\sqrt{2\pi}}{\hbar\omega_2} \Delta_t I_{2,\text{peak}}. \quad (10.1)$$

When the probe interrogates this ensemble; each population will interact linearly:

$$\begin{aligned} P_{\text{tot}} &= (1 - \bar{n}) \chi_0^{(1)}(\omega_1) E_1 + \bar{n} \chi_1^{(1)}(\omega_1) E_1 \\ &= \chi_0^{(1)} E_1 + \bar{n} \left(\chi_1^{(1)}(\omega_1) - \chi_0^{(1)}(\omega_1) \right) E_1. \end{aligned} \quad (10.2)$$

Here $\chi_0^{(1)}$ ($\chi_1^{(1)}$) denotes the linear susceptibility of the pure state $|0\rangle$ ($|1\rangle$). The third-order field scales as $I_2 E_1$, so from Equation 10.2

$$\chi^{(3)} \propto \sigma_0(\omega_2) \left(\chi_1^{(1)} - \chi_0^{(1)} \right). \quad (10.3)$$

This expression accounts for the familiar population-level pathways such as excited state absorption/emission and ground state depletion. Conforming the linear susceptibilities to our model, the non-linear portion of Equation 10.3 can be written as:

$$\chi^{(3)} \propto \mu_{eh}^4 \text{Im} [L_0(\omega_2)] [S L_1(\omega) - L_0(\omega_1)], \quad (10.4)$$

$$L_0(\omega) = \frac{1}{\omega - \omega_{10} + i\Gamma_{10}}, \quad (10.5)$$

$$L_1(\omega) = \frac{1}{\omega - (\omega_{10} - \epsilon) + i\xi\Gamma_{10}}, \quad (10.6)$$

where $S = \frac{N_2}{N_1+1}$. The denominator of S is larger than N_1 due to the contribution of stimulated emission; this contribution is often neglected. From Equation 10.4 we can see that a finite response can result from three conditions: $S \neq 1$, $\xi \neq 1$, and/or $\epsilon \neq 0$. The first inequality is the model's manifestation of state-filling, $S < 1$. If we assume that all 64 ground state transitions are optically active, then $S = 0.75$. The second condition is met by exciton-induced dephasing (EID), $\xi > 1$, and the third from the net attractive Coulombic coupling of excitons, $\epsilon > 0$. The finite bandwidth of the monochromator can be accounted for by convolving Equation 10.4 with the monochromator instrumental function.

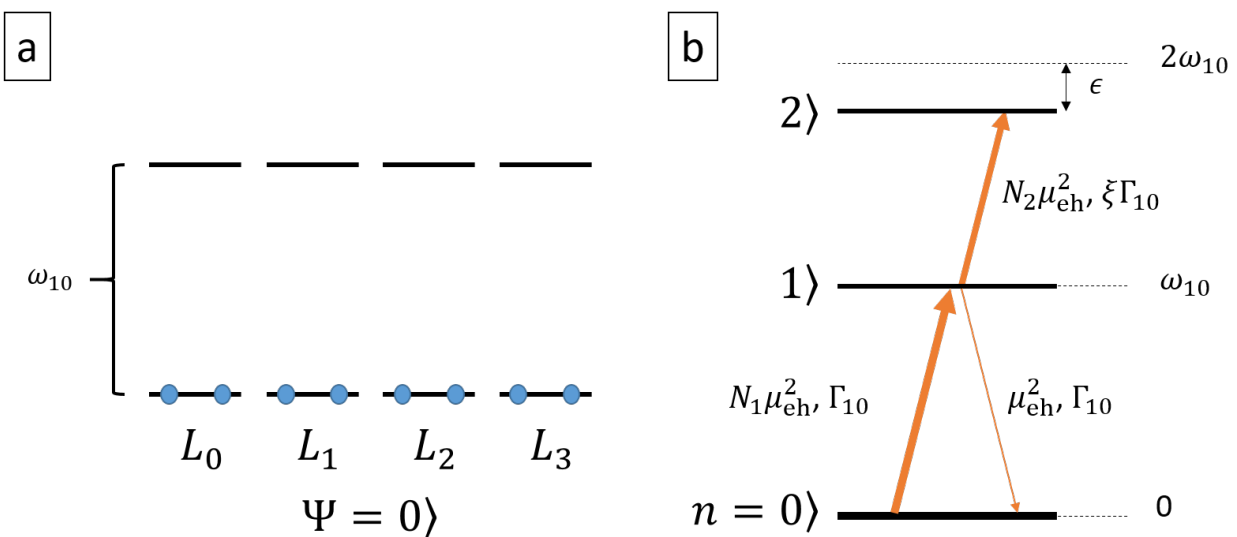


Figure 10.2: Model system for the 1S band of PbSe quantum dots. (a) The ground state shown in the electron-hole basis. All electrons (holes) are in the valence (conduction) band. There are two electrons and holes in each of the four degenerate L points. (b) The excitonic basis and the transitions accessible in this experiment. The arrows illustrate the available absorptive or emissive transitions that take place in the $\chi^{(3)}$ experiment, and are labeled by parameters that control the cross-sectional strength (arrow width qualitatively indicates transition strength).

10.2.1 The bleach nonlinearity

The bleach of the 1S band is the most studied nonlinear signature of the PbSe quantum dots. Experiments keep track of the bleach factor, ϕ , which is the proportionality factor that relates the relative change in the absorption coefficient at the exciton resonance, $\alpha_{\text{FWM}}(\omega_{10})$, with the average exciton occupation:

$$\frac{\alpha_{\text{FWM}}(\omega_{10})}{\alpha_0(\omega_{10})} = -\phi\bar{n} \quad (10.7)$$

where α_0 is the linear absorption coefficient. If QDs are completely bleached by the creation of a single exciton, then $\phi = 1$; if QDs are unperturbed by the exciton, then $\phi = 0$. For PbSe, 1S-resonant values of $\phi = 0.25$ and 0.125 have been reported in literature [188, 190, 209, 9, 201], each with supporting theories on how state-filling should behave in an 8-fold degenerate system. Inspection of Equation 10.4 shows that $\phi = \frac{\text{Im}[L_0 - SL_1]}{L_0}$; if the 1S nonlinearity is dominated by state-filling ($\xi = 1$ and $\epsilon = 0$), then the bleach fraction has perfect correspondence with the change in the number of optically active states: $\phi = 1 - S$. Because Coulombic shifts and EID act to decrease the resonant absorption of the excited state, we have the strict relation $\phi \geq 1 - S$.

More recently, a bleach factor metric has been adopted [186, 187] as the proportionality between the spectrally integrated probe and the carrier concentration:

$$\frac{\int \alpha_{\text{FWM}}(\omega) d\omega}{\int \alpha_0(\omega) d\omega} = -\phi_{\text{int}}\bar{n}. \quad (10.8)$$

This metric is a more robust description of state filling, because it is unaffected by Coulomb shifts or EID: Equation 10.4 gives $\phi_{\text{int}} = 1 - S$ regardless of ξ and ϵ . An experimental value of $\phi_{\text{int}} = 0.25$ has been reported [187] which consequently supports the measurement of $\phi = 0.25$.

10.2.2 TG/TA scaling

TG is often quantified in the framework of non-linear spectroscopy, which measures the non-linear susceptibility, $\chi^{(3)}$, while TA often uses metrics of absorption spectroscopy. The global study of both TA and TG requires relating the typical metrics of both experiments. Here we outline how the measured

signals from both methods compare. We assume perfect phase matching and collinear beams, and we neglect frequency dispersion of the linear refractive index.

When absorption is important ($\alpha_0 \ell \gg 0$), the spatial dependence of the electric field amplitudes must be considered. For TG, the polarization modulated in the phase-matched direction is given by

$$P_{\text{TG}}(z) = E_2^* E_{2'} e^{-\alpha_0(\omega_2)z} E_1 e^{-\alpha_0(\omega_1)z/2} \chi^{(3)} \quad (10.9)$$

The TG electric field propagation can be solved using the slowly varying envelope approximation, which yields an output intensity of [24]

$$I_{\text{TG}} \propto \omega_1^2 M_{\text{TG}}^2 I_1 I_2 I_2' \left| \chi^{(3)} \right|^2 \ell^2 \quad (10.10)$$

$$M_{\text{TG}}(\omega_1, \omega_2) = e^{-\alpha_0(\omega_1)\ell/2} \frac{1 - e^{-\alpha_0(\omega_2)\ell}}{\alpha_0(\omega_2)\ell}. \quad (10.11)$$

This motivates the following metric for TG:

$$S_{\text{TG}} \equiv \frac{1}{M_{\text{TG}} \omega_1} \sqrt{\frac{N_{\text{TG}}}{I_1 I_2^2}} \propto \left| \chi^{(3)} \right| \quad (10.12)$$

Here N_{TG} is the measured four-wave mixing signal from a squared-law detector (I_{TG}/ω_1). Again, the third-order response amplitude is extracted from this measurement.

We now derive a comparable metric for TA measurements. Because the third-order field is spatially overlapped with the probe field, E_1 , the relevant polarization includes the first- and third-order susceptibility:

$$P_{\text{TA}} = \left(\chi^{(3)}(z; \omega_1) + e^{-\alpha_0(\omega_2)z} |E_2|^2 \chi^{(3)}(z; \omega) \right) E_1. \quad (10.13)$$

Maxwell's equations show that the imaginary component of this polarization changes the intensity of the

E_1 beam with a total absorption coefficient that is a composite of the linear and non-linear propagation:

$$\begin{aligned}\alpha_{\text{tot}} &= \frac{2\omega_1}{c} \text{Im} \left[\sqrt{1 + 4\pi \left(\chi^{(1)} + \frac{1 - e^{-\alpha_0(\omega_2)\ell}}{\alpha_0(\omega_2)\ell} |E_2|^2 \chi^{(3)} \right)} \right] \\ &\approx \frac{4\pi\omega_1}{c} \left(\text{Im} [\chi^{(1)}] + \left[\frac{1 - e^{-\alpha_0(\omega_2)\ell}}{\alpha_0(\omega_2)\ell} \left(\frac{8\pi}{nc} l_2 \right) \right] \text{Im} [\chi^{(3)}] \right)\end{aligned}\quad (10.14)$$

The $\chi^{(3)}$ response term can be isolated by chopping the pump beam, which yields a measured transmittance indicative of the nominal absorption, $T_0 = I_1 e^{-\alpha_0(\omega_1)\ell}$, and the change in the absorption $T = I_1 e^{-\alpha_{\text{tot}}\ell}$. We quantify the non-linear absorption by $\alpha_{\text{FWM}} = 1/\ell \ln \frac{T-T_0}{T_0} = \alpha_0 - \alpha_{\text{tot}}$, which can now be written as

$$\alpha_{\text{FWM}}(\omega_1; \omega_2) = \frac{32\pi^2\omega_1}{nc^2} M_{\text{TA}}(\omega_2) l_2 \text{Im} [\chi^{(3)}], \quad (10.15)$$

$$M_{\text{TA}}(\omega_2) = \frac{1 - e^{-\alpha_0(\omega_2)\ell}}{\alpha_0(\omega_2)\ell}. \quad (10.16)$$

Similar to TG propagation, α_{FWM} suffers from absorption effects that distort the proportional relationship to $\text{Im} [\chi^{(3)}]$. It is notable that in this case distortions are only from the pump beam. The signal field heterodynes with the probe, which takes the absorption losses into account automatically. Note that M_{TA} simply accounts for the fraction of pump light absorbed in the sample, and consequently is closely related to the average exciton occupation across the entire path length of the sample, $\bar{n}_\ell = \ell^{-1} \int_0^\ell \bar{n}(z) dz$, which can be written using Equation 10.1 as:

$$\bar{n}_\ell = \frac{\sqrt{2}\sigma_0(\omega_2)\Delta_t}{\hbar\omega_2} I_{2,\text{peak}} M_{\text{TA}} \quad (10.17)$$

We define an experimental metric that isolates the $\chi^{(3)}$ tensor:

$$\begin{aligned}S_{\text{TA}} &= \frac{\alpha_{\text{FWM}}}{\omega_1 l_2 M_{\text{TA}}} \\ &\propto \text{Im} [\chi^{(3)}]\end{aligned}\quad (10.18)$$

For the general $\chi^{(3)}$ response, the imaginary and real components of the spectrum satisfy complicated relations owing to the causality of all three laser interactions. For the pump-probe time-ordered processes, the probe causality is separable from the pump excitation event, which makes the causality relation of

the pump and probe separable. [220] The causality relations of the probe spectrum are the famous Kramers-Kronig equations that relate ground state absorption to the index of refraction. This relation is foundational in analysis of TG [221] and TA measurements.

Theoretically, TA probe spectra alone could be transformed to generate the real spectrum. In practice, such a transform is difficult because the spectral breadth needed to accurately calculate the integral is experimentally difficult to achieve. When both the absolute value and the imaginary projection of $\chi^{(3)}$ are known, however, the real part can also be defined by the much simpler relation:

$$\text{Re} [\chi^{(3)}] = \pm \sqrt{|\chi^{(3)}|^2 - \text{Im} [\chi^{(3)}]^2} \quad (10.19)$$

10.2.3 The absorptive third-order susceptibility

Though the bleach factor is defined within the context of absorptive measurements, it can be converted into the form of a third-order susceptibility as well. Equations 10.2 and 10.7 motivate alternative expressions for differential absorptivity of the probe:

$$\begin{aligned} \alpha_{\text{FWM}} &= \bar{n}(\alpha_1(\omega_1) - \alpha_0(\omega_1)) \\ &= -\phi \bar{n} \alpha_0(\omega_1). \end{aligned} \quad (10.20)$$

Substituting Equation 10.17 into Equation 10.20, we can write the non-linear absorption as

$$\alpha_{\text{FWM}}(\omega_1) = -\phi \frac{\sqrt{2\pi}}{\hbar\omega_2} \alpha_0(\omega_1) \alpha_0(\omega_2) \Delta_t I_{2,\text{peak}} M_{\text{TA}} \quad (10.21)$$

By Equation 10.15, $\chi^{(3)}$ and the molecular hyperpolarizability, $\gamma_{\text{QD}}^{(3)} = \chi_{\text{QD}}^{(3)} / F^4 N_{\text{QD}}$, where F is the local field factor, can also be related to the bleach fraction:

$$\text{Im} [\chi_{\text{QD}}^{(3)}] = -\phi \frac{\sqrt{2\pi} n c^2 \Delta_t N_{\text{QD}}}{32\pi^2 \hbar \omega_1 \omega_2} \sigma_0(\omega_2) \sigma_0(\omega_1) \quad (10.22)$$

$$\text{Im} [\gamma_{\text{QD}}^{(3)}] = -\phi \frac{\sqrt{2\pi} n c^2 \Delta_t}{32\pi^2 \hbar \omega_1 \omega_2 F^4} \sigma_0(\omega_2) \sigma_0(\omega_1). \quad (10.23)$$

Because this formula only predicts the imaginary component of the signal, its magnitude gives an approximate lower limit for the peak susceptibility and hyperpolarizability. Absorptive cross-sections have been experimentally determined for PbSe QDs. [193, 192]

10.3 Methods

Quantum dot samples used in this study were synthesized using the hot injection method. [191] Samples were kept in a glovebox after synthesis and exposure to visible and UV light was minimized. These conditions preserved the dots for several months. Two samples, Batch A and Batch B, are presented in this study, in an effort to show the robustness of the results. Properties of their optical characterization are shown in Table 10.1. The 1S band of Batch A is broader than Batch B, an effect which is usually attributed to a wider size distribution and therefore greater inhomogeneous broadening.

The experimental system for the TG experiment has been previously explained. [123, 157] Briefly, two independently tunable OPAs are used to make pulses E_1 and E_2 with colors ω_1 and ω_2 . The third beam, $E_{2'}$, is split off from E_2 . The TG experiment utilized here uses temporally overlapped E_2 and $E_{2'}$. Previous ultrafast TG work has characterized the delay of E_1 as $\tau_{21} = \tau_2 - \tau_1$; to connect the experimental space with the TA measurements, we will report the population delay time between the probe and the pump as $T(= -\tau_{21})$. Pulse timing is controlled by a motorized stage that adjusts the arrival time of E_1 relative to E_2 and $E_{2'}$.

All three beams are focused onto the sample in a BOXCARS geometry and the direction $\vec{k}_1 - \vec{k}_2 + \vec{k}_{2'}$ is isolated and sent to a monochromator to isolate the ω_1 frequency with $\sim 120\text{cm}^{-1}$ detection bandwidth. The signal, N_{TG} , was detected with an InSb photodiode. Reflective neutral density filters (Inconel) limit the pulse fluence to avoid multi-photon absorption. To control for frequency-dependent changes in pulse arrival time due to the OPAs and the neutral density, a calibration table was established to assign a correct zero delay for each color combination.

The TA experiments were designed to minimally change the TG experimental conditions. The $E_{2'}$ beam was blocked and signal in the \vec{k}_1 direction was measured. E_2 was chopped and the differential signal and the average signal were measured to define T_0 and T needed to compute ΔA . Just as in TG experiments, the excitation frequencies were scanned while the monochromator was locked at $\omega_m = \omega_1$.

	A	B
ω_{10} (cm ⁻¹)	7570	6620
FWHM (cm ⁻¹)	780	540
$\langle d \rangle$ (nm)	4	4.8
σ_0 ($\times 10^{16}$ cm ²)	1.7	2.9

Table 10.1: Batch Parameters extracted from absorption spectra. $\langle d \rangle$: average QD diameter, as inferred by the 1S transition energy.

10.4 Results

10.4.1 Pump-probe 3D acquisitions for TA and TG

For both samples, 2D spectra were collected for increments along the population rise time. For these acquisitions, concentrated samples ($OD_{1S} \sim 0.6, 0.8$) were used to minimize contributions from non-resonant background. Both samples maintained constant signal amplitude for at least hundreds of picoseconds after initial excitation, indicating multiexcitons and trapping were negligible effects in these studies. The TA and TG results for both batches are shown in Figure 10.3. For $T < 0$ (probe arrives before pump), both collections show spectral line-narrowing in the anti-diagonal direction. This highly correlated line shape is indicative of an inhomogeneous distribution, but the correlation is enhanced by pulse overlap effects. When the probe arrives before or at the same time as the pump, the typical pump-probe pathways are suppressed and more unconventional pathways with probe-pump and pump-probe-pump pulse orderings are enhanced. Such pathways exhibit resonant enhancement when $\omega_1 = \omega_2$, even in the absence of inhomogeneity. The pulse overlap effect is well-understood in both TA [171] and TG [112] experiments.

After the initial excitation rise time ($T > 50$ fs), the signal reaches a maximum, followed by a slight loss of signal ($\sim 10\%$) over the course of 150 fs, after which the signal converges to a line shape that remains static over the dynamic range of our experiment (200 ps). This signal loss occurs in both samples in both TA and TG; in TA measurements, the loss of amplitude occurred on both the ESA feature and the bleach feature, so that the band integral [188] did not appreciably change. We do not know the cause of this loss, but speculate it could be a signature of bandgap renormalization.

The static line shape distinguishes the homogeneous and inhomogeneous contributions to the 1S band. The elongation of the peak along the diagonal, relative to the antidiagonal, demonstrates a persistent correlation between the pumped state and excited state; we attribute this correlation to the size distribution of the synthesized quantum dots. The diagonal elongation is much more noticeable in the TA spectrum; the TG spectra is much more elongated along the ω_1 axis, which makes discerning the antidiagonal and diagonal widths more difficult. The TG spectrum is elongated along ω_1 because it measures both the absorptive and refractive components of the probe spectrum, while it is sensitive

only to the absorptive components along the pump axis. At all delays, Batch A exhibits a much broader diagonal line shape than that of Batch B, indicative of its larger size distribution.

Our spectra show that the 2D line shape of the 1S exciton is significantly distorted by contributions from hot carrier excitation just above the 1S state. These hot carriers arise from transitions between the 1S and 1P resonances, which have been attributed to either the “rising edge” of the continuum or the pseudo-forbidden 1S-1P exciton transition [222, 223]. Contributions from these hot carriers distort the 1S 2D line shape for $\omega_2 > \omega_{1S}$, resulting in a bleach feature centered at $\omega_1 = \omega_{1S}$ and containing bleach contributions from the unresolved ensemble. The rise time of this feature is indistinguishable from the 1S rise time, indicating either extremely fast (≤ 50 fs) relaxation or direct excitation of a hot 1S exciton. Since the ensemble is inhomogeneous, these hot exciton contributions are presumably also present within the 1S band due to the larger (lower energy bandgap) members of the ensemble. Such contributions would not be recognized or resolved without scanning the pump frequency.

10.4.2 The skewed TG probe spectrum

The most surprising spectral feature presented here is the skew of the TG probe spectrum towards the red of $\omega_1 = \omega_{1S}$. If 1S state-filling completely describes the nonlinear response, the TG signal will mimic the absorptive bleach behavior of TA and show a line shape symmetric about ω_1 . Although the spectral range of our experimental system limits the measurement of the red skew of Batch A, this feature was reproducible across many batches and system alignments. We find no grounds to discount the red skew based on our experimental procedures or sample reproducibility issues.

As T is scanned, the skewed part rises in concert with the 1S-resonant signal that has the pump-probe pulse sequence. We therefore explain the skewness as either an instantaneous spectral signature of the photoexcited population or a feature with dynamics much faster than our pulses. For all pump colors, the skew maintains a magnitude of 30 – 40% of maximum TG signal for each probe slice. In contrast, TA signal red of the 1S excitation is no more than 10% of the maximum amplitude of the bleach. The difference in prominence shows that the redshifted feature is primarily refractive in character.

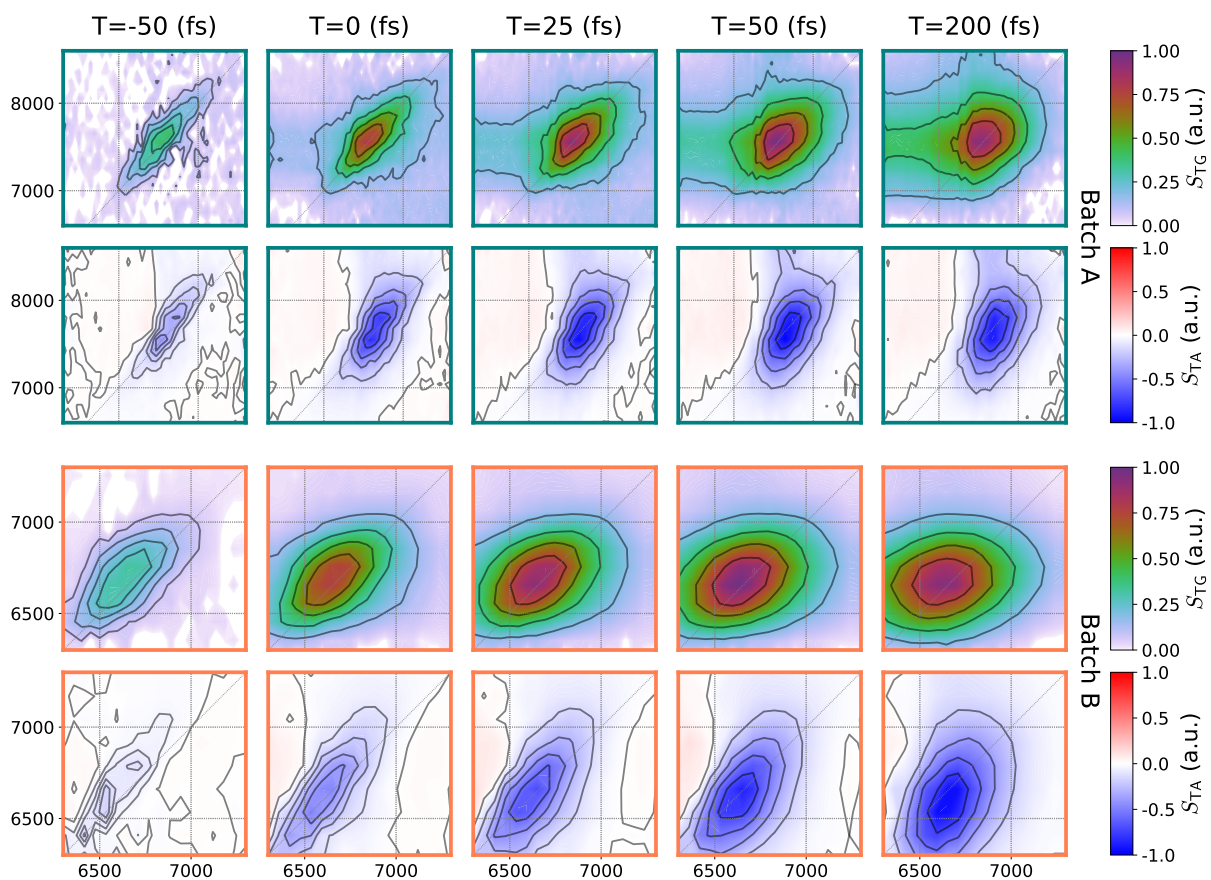


Figure 10.3: S_{TG} (left) and S_{TA} 2D spectra (see colorbar labels) of Batch A (top) and Batch B (bottom) as a function of T delay. The colors of each 2D spectrum are normalized to the global maximum of the 3D acquisition, while the contour lines are normalized to each particular 2D spectrum. The vertical axis is pump energy, horizontal axis probe energy.

10.5 Discussion

10.5.1 Comparison of TA and TG line shapes

We first attempted simple fits on a subset of the data to reduce the parameter complexity. We consider our experimental data at $T = 120$ fs to remove contributions from probe-pump and pump-probe-pump time-ordered processes. By further restricting our considerations to a single probe slice ($\omega_2 = \omega_{1S}$), we discriminate against inhomogeneous broadening and thus neglect ensemble effects for initial considerations. We fit our probe spectrum with Equation 10.4 along with the added treatment of convolving the response with our monochromator instrumental function. Proper accounting of all time-ordered pathways, finite pulse bandwidth, and inhomogeneity are treated later.

We find that the TA spectra are more sensitive to the model parameters than TG, and that the parameter interplay necessary to reproduce the spectra can be easily described. We note three features of the TA spectra that are crucial to reproduce in simulation: 1. the net bleach, 2. the photon energy of the bleach feature minimum is blue of the 1S absorption peak, 3. the probe spectrum decays quickly to zero away from the bleach resonance, leaving a minimal ESA feature to the red. These features are consistent with the vast majority of published TA spectra of the 1S exciton, [187, 222, 224, 188, 225, 202] and can only be reproduced when all three of our nonlinearities (state-filling, excitation-induced broadening, and Coulombic coupling) are invoked (exposition of this result is found in supporting information). The extracted fit parameters are listed in Table 10.2.

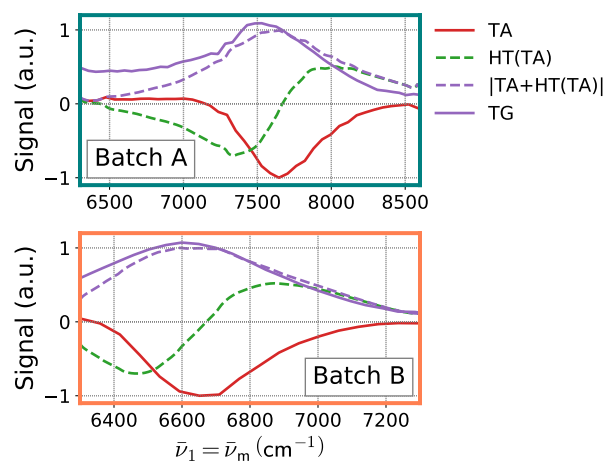


Figure 10.4: Kramers-Kronig analysis of TA spectra compared with TG spectra.

	Batch	
	A	B
$\varepsilon_{\text{Coul}} (\text{cm}^{-1})$	81	53
$\Gamma_{10} (\text{cm}^{-1})$	380	200
ξ	1.35	1.39

Table 10.2: Parameters used in fitting experimental probe slices using Equation 10.4; $S = 0.75$, $\omega_2 = \omega_{1S}$.

With the 1S band TA well-characterized, we now consider applying the fitted parameters to the TG signal. Transferring this simulation to the TG data poses technical challenges. A critical factor is appropriately scaling the TG signals relative to TA signals. The TA simulation can define the scaling through the Kramers-Kronig transform, which gives the transient refraction. The computed transient refraction is unique to within an arbitrary offset; for a single resonant TA/TG feature, the transient refraction offset is zero. We take the offset to be zero now and address this assumption later. The transient refraction shows highly dispersive character with a node near resonance. This means that there is a point in our spectrum at which $|\chi^{(3)}| = |\text{Im}[\chi^{(3)}]|$. Of course, we also have the constraint $|\chi^{(3)}| \geq |\text{Im}[\chi^{(3)}]|$ for every probe color. These two constraints uniquely determine the appropriate scaling factor as the minimum scalar c_0 that satisfies $c_0 S_{\text{TG}} \geq |S_{\text{TA}}|$ for all probe colors.

As we alluded, the arbitrary offset of the Kramers-Kronig transform deserves special consideration. The physical origin for this offset would be coupling between the 1S band and states outside our spectral range. If the coupling is sufficiently strong, the $\text{Re}[\chi^{(3)}]$ offset may be large enough to remove the node, invalidating the minimum scaling factor method. We believe such a large offset is not viable for several reasons. From a physical standpoint, it seems very unlikely a non-resonant state would have coupling stronger to the 1S band than the 1S band itself. Furthermore, TA studies with larger spectral ranges see no evidence of coupling features with strength comparable to the 1S bleach. [188, 187] Finally, a sufficiently large refractive offset would produce a dip in the $|\chi^{(3)}|$ line shape near the FWHM points; such features are definitively absent in the TG spectra.

While we have confidence in relating the TG and TA measurements using the minimum scaling factor, Equation 10.4 fails to accurately reproduce the TG spectrum. The errors are systematic: in both batches, our simulation misses the characteristic red skew of our experimental TG and instead skews signal to the blue. Based on the excellent agreement with S_{TA} , it follows that the chief source of error in our simulation lies in $\text{Re}[\chi^{(3)}]$. The experimentally-consistent $\text{Re}[\chi^{(3)}]$ can be calculated from Equation 10.19. The dark green curve highlights which of the two roots of Equation 10.19 is closest to our simulation. The choice of a negative non-linear refractive index value on resonance is in agreement with z-scan measurements. [226] The discrepancy between the experimental and simulated real components is well-approximated by a constant offset.

The presence of this offset forced a re-evaluation of the model. By revising our simulation to include a complex-valued offset term, $\Delta = |\Delta|e^{i\theta}$, so that

$$\chi^{(3)} = \text{Im} [L_0(\omega_2)] [SL_1(\omega_1) - L_0(\omega_1) + \Delta], \quad (10.24)$$

the discrepancy between S_{TA} and S_{TG} can be resolved. It was found, however, that minimizing error between Equation 10.24 and the two datasets alone does not confine all variables uniquely. Specifically, the effects of the EID non-linearity (ξ) on the line shapes were highly correlated with those of the broadband photoinduced absorption (PA) term, $\text{Im} [\Delta]$, so that getting a unique parameter combination was not possible. The fitting routine was robust, however, when the resonant bleach magnitude was pinned to the state-filling: $\phi \approx 1 - S$. Robustness here is defined as the ability to permute the fitting parameter order when minimizing the residual. For example, τ_{10} can be fit either before or after Δ is fit without significantly changing the resulting parameters. The resulting parameters are shown in Table 10.3, and the results of the fit are shown in Figure 10.5. As both $\phi = 0.25$ and $\phi_{\text{int}} = 0.25$ have been measured, this added constraint has a reasonable precedence. As mentioned earlier, EID and Coulombic coupling prevent this equality (as in Equation 10.4), but PA can counteract these terms and keep the resonant bleach near $1 - S$. In addition, since EID and PA are correlated, decreasing the resonant bleach also decreases the need for EID to influence the line shape, which results in a reduced EID non-linearity in this fit (compare ξ in Table 10.2 and Table 10.3).

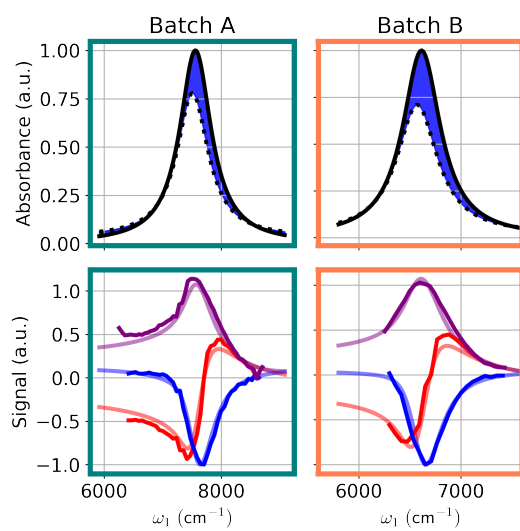


Figure 10.5: Top row: Final simulated absorption spectra for the excited state and the ground state. Bottom row: magnitude (purple), imaginary (blue) and real (red) projections and fits.

Batch	A	B
Γ_{10} (cm ⁻¹)	340 (320)	210 (210)
ξ	1.07 (1.04)	1.05 (1.02)
ϵ_{Coul} (cm ⁻¹)	54 (46)	28 (26)
$ \Delta /\text{Im}[L_0(\omega_{1S})]$	0.07 (0.06)	0.06 (0.06)
θ (deg)	151 (156)	146 (148)

Table 10.3: Parameters of the simulated $\chi^{(3)}$ response extracted by global fits of TA and TG at $T = 120$ fs using Equation 10.24 and with $S = 0.75$. Numbers in parentheses refer to fits at $T = 300$ fs.

The nature of Δ may be associated with resonant and non-resonant transitions from the 1S excitonic state to the continuum of intraband states involving the electron and/or hole. The magnitude and phase of this contribution would then depend on the ensemble average from all transitions. This contribution has been identified in previous TA studies. De Geyter et al. [202] isolated a net absorption at sub-bandgap probe frequencies. Geiregat et al. [201] found an absorptive contribution was needed to explain the fluence dependence of the 1S-resonant bleach. The absorptive contribution was 5% of the 1S absorbance, a value that agrees with this work (see Table 10.3). Our data unifies both observations by showing that additional contribution persists at both bandgap and sub-bandgap frequencies. In addition, our data provides the spectral phase of the contribution. It also shows that the red skew of the TG line shape is very sensitive to the relative importance of the 1S resonance and the additional contribution.

There may also be a relationship to studies on CdSe quantum dots where an additional broad ESA feature was observed for $\omega_1 < \omega_{1S}$. The feature had separate narrow and broad components. The narrow component closest to the band edge bleach corresponded to the Coulombically shifted biexciton transition. Since the broad component correlated with inadequate surface passivation, it was attributed to the surface inducing ESA transitions to the broad band of continuum states that would normally be forbidden. In addition to creating additional ESA transitions, it also created a short-lived transient that was similar to the transients attributed to multiexciton relaxation and multiexciton generation.

10.5.2 Determination of state filling factor

Our results show that the peak susceptibility is almost entirely imaginary, which means we can attribute our TG peak $|\chi^{(3)}|$ susceptibility to the TA state-filling factor. A standard addition method was used to extract the peak TG hyperpolarizability of $|\gamma^{(3)}| = 7 \pm 2 \cdot 10^{-31} \text{cm}^6 \text{erg}^{-1}$ for Batch A, while Equation 10.23 predicts a peak imaginary hyperpolarizability of $\text{Im}[\gamma^{(3)}] = 2.1\phi \cdot 10^{-30} \text{cm}^6 \text{erg}^{-1}$. The two values are similar for $\phi = 0.25$, while $\phi = 0.125$ gives an imaginary component that is much smaller than the total susceptibility. We conclude that only the $\phi = 0.25$ bleach factor is consistent with our observations.

10.5.3 Inhomogeneity and the pulse overlap response

Our parameter extraction above gives plausible parameters to explain the observed photophysics of a small slice of our multidimensional data. We now apply a more rigorous simulation of the model system to address the entire dataset and consider the broader experimental space. This rigorous simulation is meant to account for the complex signals that arise at temporal pulse overlap, the pulsed nature of our excitation beams, and sample inhomogeneity. We calculate signal through numerical integration techniques. [112] The homogeneous and inhomogeneous broadening were constrained to compensate each other so that the total ensemble line shape was kept constant and equal to that extracted from absorption measurements (Table 10.1). For a Lorentzian of FWHM $2\Gamma_{10}$ and a Gaussian line shape of standard deviation σ_{inhom} , the resulting Voigt line shape has a FWHM well-approximated by $\text{FWHM}_{\text{tot}} [\text{cm}^{-1}] \approx 5672\Gamma_{10} [\text{fs}^{-1}] + \sqrt{2298\Gamma_{10} [\text{fs}^{-1}] + 8 \ln 2 (\sigma_{\text{inhom}} [\text{cm}^{-1}])}$. [227]

Rather than a complete global fit of all parameters, we accept the non-linear parameters extracted earlier (Table 10.3) and optimize the inhomogeneous-homogeneous ratio only, using the ellipticity of the 2D peak shape [176] at late population times as the figure of merit. Table 10.4 shows the degree of inhomogeneity in the sample from matching the ellipticity of the peak shape. As expected, Batch B fits to a smaller Gaussian distribution of transition energies than Batch A, but it also exhibits a longer homogeneous dephasing time, suggesting different system-bath coupling strengths for both samples. Previous research on the coherent dynamics of PbSe 1S exciton feature noted homogeneous lifetimes are a significant source of broadening on the 1S exciton; [123] our results demonstrate that the relationship between exciton size distribution and 1S exciton linewidth is further complicated by sample-dependent system-bath coupling.

The results of this final simulation are compared with the experimental data in Figure Figure 10.6. It is important to note that the simulations get many details of the rise-time spectra correct. Particularly, the transition from narrow, diagonal line shape to a broad, less diagonal line shape is reproduced very well in both TA and TG simulations. Such behavior is expected for responses from excitonic peaks of material systems; the rise time behavior for such systems was studied in detail previously. [112] Because these simulations do not account for hot-exciton creation from the pump, simulations differ from experiment increasingly as the pump becomes bluer than the 1S center.

Batch	A	B
$\Gamma_{10} (\text{cm}^{-1})$	220	130
$\text{FWHM}_{\text{inhom}} (\text{cm}^{-1})$	520	360

Table 10.4: Homogeneous and inhomogeneous linewidths extracted by global analysis using numerical integration.

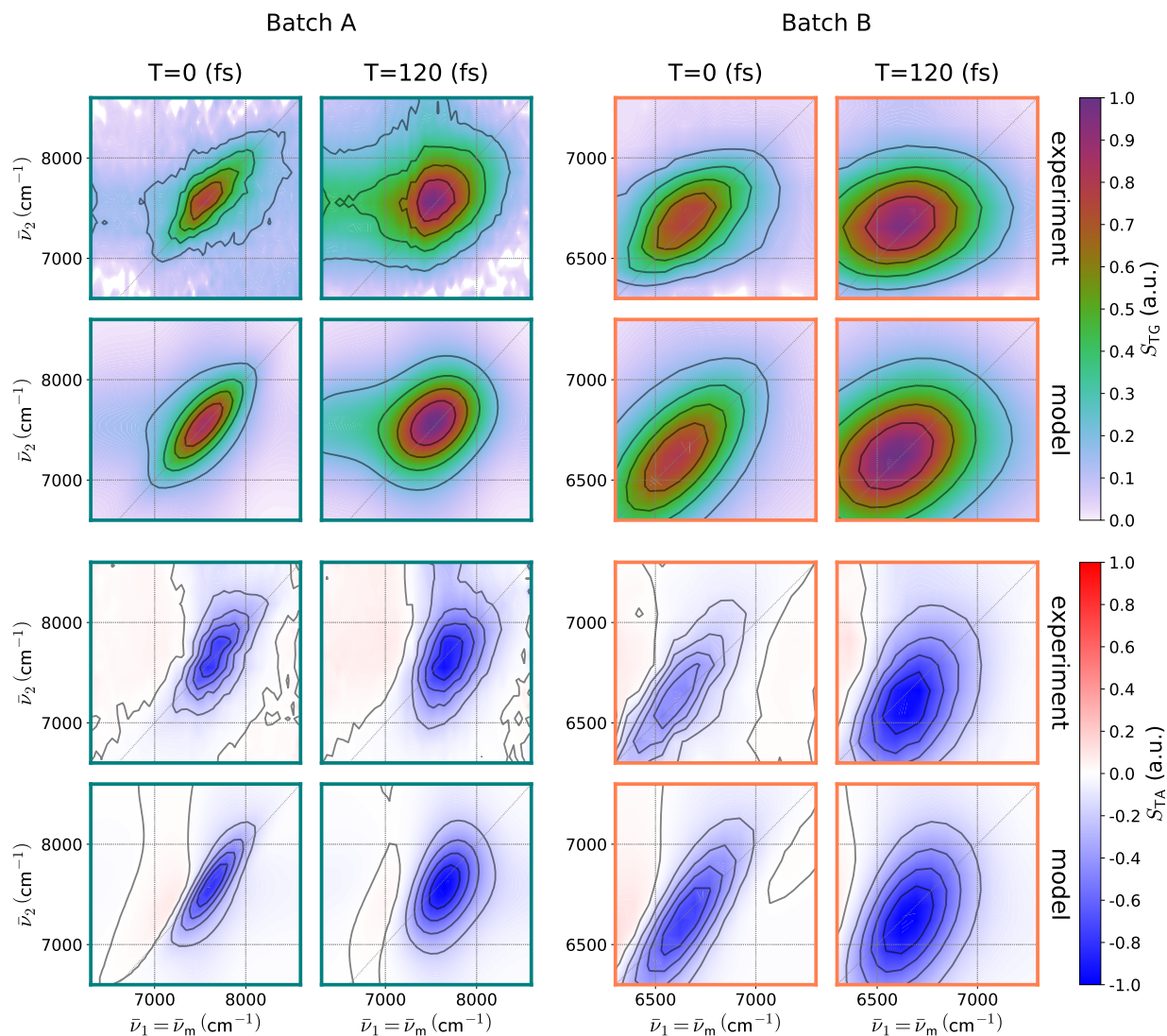


Figure 10.6: Global simulation using numerical integration and comparison with experiment. Batches A (left block) and B (right block) are shown, with the TG experimental (top), the simulated TG (2nd row), the experimental TA (3rd row), and the simulated TA (bottom row) data. Pump-probe delay times of $T = 0$, and 120 fs are shown in each case (see column labels). For each pair, the colors are globally normalized and the contours are locally normalized.

10.6 Conclusion

By combining TA and TG measurements, we have described the complex third-order, 2D susceptibility of the 1S resonance of PbSe quantum dots. We have demonstrated a parameter extraction procedure that is reproducible for different quantum dot samples, and that some of the parameters, such as the pure dephasing time, are batch dependent. Inhomogeneity, exciton-induced broadening, exciton-exciton coulombic coupling shifts, and intraband absorption are all required to reconcile both datasets. TA features about 1S exciton band are not exclusively assigned as 1S transitions, which can have important consequences for interpreting the evolution of the 1S bleach.

While the TA spectra show prominent 1S-resonant features, the intraband absorption and its associated refractive index signature are most visible in the TG dataset, so that disentangling the 1S resonant response and the broadband response is a more well-defined problem when both datasets are used together. This approach is thus useful for characterization of non-linear signals in spectrally congested systems.

Chapter 11

Measurement of Ultrafast Excitonic Dynamics of Few-Layer MoS₂ using State-Selective Coherent Multidimensional Spectroscopy

This Chapter borrows extensively from Czech et al. [157]. The authors are:

1. Kyle J. Czech
2. Blaise J. Thompson
3. Schuyler Kain
4. Qi Ding
5. Melinda J. Shearer
6. Robert J. Hamers
7. Song Jin
8. John C. Wright

We report the first coherent multidimensional spectroscopy study of a MoS₂ film. A four-layer sample of MoS₂ was synthesized on a silica substrate by a simplified sulfidation reaction and characterized by absorption and Raman spectroscopy, atomic force microscopy, and transmission electron microscopy. State-selective coherent multidimensional spectroscopy (CMDS) on the as-prepared MoS₂ film resolved the dynamics of a series of diagonal and cross-peak features involving the spin—orbit split A and B excitonic states and continuum states. The spectra are characterized by striped features that are similar to those observed in CMDS studies of quantum wells where the continuum states contribute strongly to the initial excitation of both the diagonal and cross-peak features, while the A and B excitonic states contributed strongly to the final output signal. The strong contribution from the continuum states to the initial excitation of both the diagonal and cross-peak features, while the A and B excitonic states contributed strongly to the final output signal. The strong contribution from the continuum states to the initial excitation shows that the continuum states are coupled to the A and B excitonic states and that fast intraband relaxation is occurring on a sub-70 fs time scale. A comparison of the CMDS excitation signal and the absorption spectrum shows that the relative importance of the continuum states is determined primarily by their absorption strength. Diagonal and cross-peak features decay with a 680 fs time constant characteristic of exciton recombination and/or trapping. The short time dynamics are complicated by coherent and partially coherent pathways that become important when the excitation pulses are temporally overlapped. In this region, the coherent dynamics create diagonal features involving both the excitonic states and continuum states, while the partially coherent pathways contribute to cross-peak features.

11.1 Introduction

Transition metal dichalcogenides (TMDCs), such as MoS₂, are layered semiconductors with strong spin-orbit coupling, high charge mobility, and an indirect band gap that becomes direct for monolayers. [228, 28] The optical properties are dominated by the A and B excitonic transitions between two HOMO spin-orbit split valence bands and the lowest state of the conduction band at the *K* and *K'* valleys of the two-dimensional hexagonal Brillouin zone. [229] The spin and valley degrees of freedom are coupled in individual TMDC layers as a result of the strong spin-orbit coupling and the loss of inversion symmetry. The coupling suppresses spin and valley relaxation since both spin and valley must change in a transition. These unusual properties have motivated the development of TMDC monolayers for next-generation nano/optoelectronic devices as well as model systems for spintronics and valleytronics applications. [28, 29, 30]

Ultrafast dynamics of the MoS₂ A and B electronic states have been measured by pump-probe, transient absorption, and transient reflection spectroscopy. [230, 231, 232, 233, 234] The spectra contain A and B excitonic features that result from ground-state bleaching (GSB), stimulated emission (SE), and excited-state absorption (ESA) pathways. The excitons exhibit biexponential relaxation times of ≈ 10 – 20 and ≈ 350 – 650 fs, depending on the fluence and temperature. The dependence on excitation frequency has not been explored in previous ultrafast experiments on MoS₂, but it has played a central role in understanding exciton cooling dynamics and exciton-phonon coupling in studies of quantum dots. [235]

Coherent multidimensional spectroscopy (CMDS) is a complementary four wave mixing (FWM) methodology that differs from pump-probe, transient absorption, and transient reflection methods. [30, 230, 231, 232, 234, 236, 233] Rather than measuring the intensity change of a probe beam caused by the state population changes induced by a pump beam, CMDS measures the intensity of a coherent output beam created by interactions with three excitation pulses. The interest in CMDS methods arise from their ability to remove inhomogeneous broadening, define interstate coupling, and resolve coherent and incoherent dynamics. [237, 238, 123, 104, 239, 240, 241, 242, 243] CMDS typically requires interferometric phase stability between excitation pulses, so CMDS has been limited to materials with electronic states within the excitation-pulse bandwidth. Multiresonant CMDS is a particularly attractive method for the broader range of complex materials because it does not require interferometric stability and is

able to use independently tunable excitation pulses over wide frequency ranges.

The multiresonant CMDS used in this work employs two independently tunable excitation beams with frequencies ω_1 and ω_2 . The ω_2 beam is split into two beams, denoted by ω_2 and ω'_2 . These three beams are focused onto the MoS₂ thin film at angles, creating an output beam in the phase-matched direction $\mathbf{k}_{\text{out}} = \mathbf{k}_1 - \mathbf{k}_2 + \mathbf{k}_{2'}$ where \mathbf{k} is the wave vector for each beam and the subscripts label the excitation frequencies. Multidimensional spectra result from measuring the output intensity dependence on frequency and delay times.

Figure 11.1 introduces our conventions for representing multidimensional spectra. Figure 11.1b,d are simulated data. Figure 11.1a shows one of the six time orderings of the three excitation pulses where $\tau_{22'} \equiv t_2 - t_{2'} > 0$ and $\tau_{21} \equiv t_2 - t_1 < 0$; that is, the $\omega_{2'}$ pulse interacts first and the ω_1 pulse interacts last. Figure 11.1b illustrates the 2D delay-delay spectrum for all six time orderings when ω_1 and ω_2 are both resonant with the same state. The color denotes the output amplitude. Along the negative ordinate where $\tau_{22'} = 0$, interactions with the ω_2 and $\omega_{2'}$ pulses create a population that is probed by ω_1 . Similarly, along the negative abscissa where $\tau_{21} = 0$, interactions with the ω_2 and ω_1 pulses create a population that is probed by $\omega_{2'}$. The decay along these axes measures the population relaxation dynamics. Note that these delay representations differ from previous publications by our group. [15] This paper specifically explores the dynamics along the ordinate where $\tau_{22'}$ is zero and the τ_{21} delay is changed.

Figure 11.1c depicts the A and B excitonic transitions between the spin-orbit split valence bands and the degenerate conduction band states of MoS₂. Figure 11.1d illustrates the 2D frequency-frequency spectrum when ω_1 and ω_2 are scanned over two narrow resonances. The spectrum contains diagonal and cross-peaks that we label according to the excitonic resonances AA, AB BA, and BB for illustrative purposes. The dynamics of the individual quantum states are best visualized by 2D frequency-delay plots, which combine the features seen in Figure 11.1b,d.

This work reports the first multiresonant CMDS spectra of MoS₂. It includes the excitation frequency dependence of the A and B excitonic-state dynamics. These experiments provide a fundamental understanding of the multidimensional MoS₂ spectra and a foundation for interpreting CMDS experiments on more complex TMDC heterostructures. The experimental spectra differ from the simple 2D spectrum

shown in Figure 11.1d and those of earlier CMDS experiments with model systems. The line shape of the CMDS excitation spectrum closely matches the absorption spectrum, but the line shape of the output coherence is dominated by the A and B excitonic features. The difference arises from fast, < 70 fs intraband relaxation from the hot A and B excitons of the continuum to the band edge. A longer, 680 fs relaxation occurs because of trapping and/or exciton dynamics. [234] The intensity of the cross-peaks depends on the importance of state filling and intraband relaxation of hot A excitons as well as the presence of interband population transfer of the A and B exciton states.

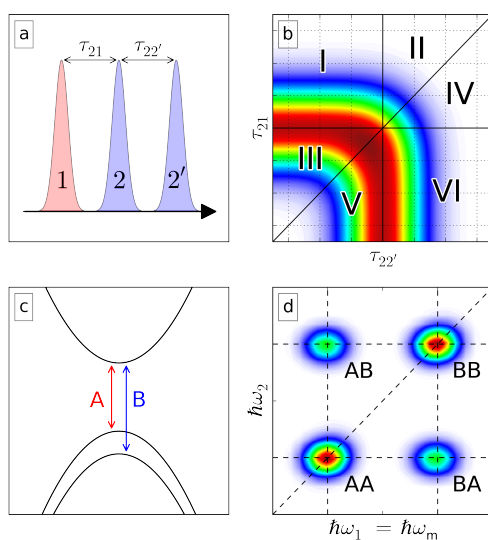


Figure 11.1: (a) Example delays of the ω_1 , ω_2 , and $\omega_{2'}$ excitation pulses. (b) Dependence of the output intensity on the $\tau_{22'}$ and τ_{21} time delays for $\omega_1 = \omega_2$. The solid lines define the regions for the six different time orderings of the ω_1 , ω_2 , and $\omega_{2'}$ excitation pulses. We have developed a convention for numbering these time orderings, as shown. (c) Diagram of the band structure of MoS₂ at the *K* point. The A and B exciton transitions are shown. (d) Two dimensional frequency-frequency plot labeling two diagonal and cross-peak features for the A and B excitons.

11.2 Methods

MoS₂ thin films were prepared *via* a Mo film sulfidation reaction, similar to methods reported by Laskar et al. [244]. A 1 nm amount of Mo (Kurt J. Lesker, 99.95%) metal was electron-beam evaporated onto a fused silica substrate at a rate of 0.05 Å/s. The prepared Mo thin films were quickly transferred to the center of a 1-inch fused silica tub furnace equipped with gas flow controllers (see Figure 11.2) and purged with Ar. The temperature of the Mo substrate was increased to 900 °C over the course of 15 min, after which 200 mg of sulfur was evaporated into the reaction chamber. Sulfidation was carried out for 30 min, and the furnace was subsequently cooled to room temperature; then the reactor tube was returned to atmospheric pressure, and the MoS₂ thin film samples were collected. The MoS₂ samples were characterized and used for CMDS experiments with no further preparation.

MoS₂ thin film absorption spectra were collected by a Shimadzu 2401PC ultraviolet-visible spectrophotometer. Raman and photoluminescence experiments were carried out in parallel using a Thermo DXR Raman microscope with a 100x 0.9 NA focusing objective and a 2.0 mW 532 nm excitation source. Raman/PL measurements were intentionally performed at an excitation power of <8.0 mW to prevent sample damage. [245] Contact-mode atomic force microscopy was performed with an Agilent 5500 AFM. MoS₂ film thickness was determined by scratching the sample to provide a clean step-edge between the MoS₂ film and the fused silica substrate. TEM samples were prepared following the method outlined by Shanmugam, Durcan, and Yu [246] using concentrated KOH in a 35 °C oil bath for 20 minutes. The delaminated MoS₂ sample was removed from the basic solution, rinsed five times with DI water, and transferred to a Cu-mesh TEM grid. TEM experiments were performed on a FEI Titan aberration corrected (S)TEM under 200 kV accelerating voltage.

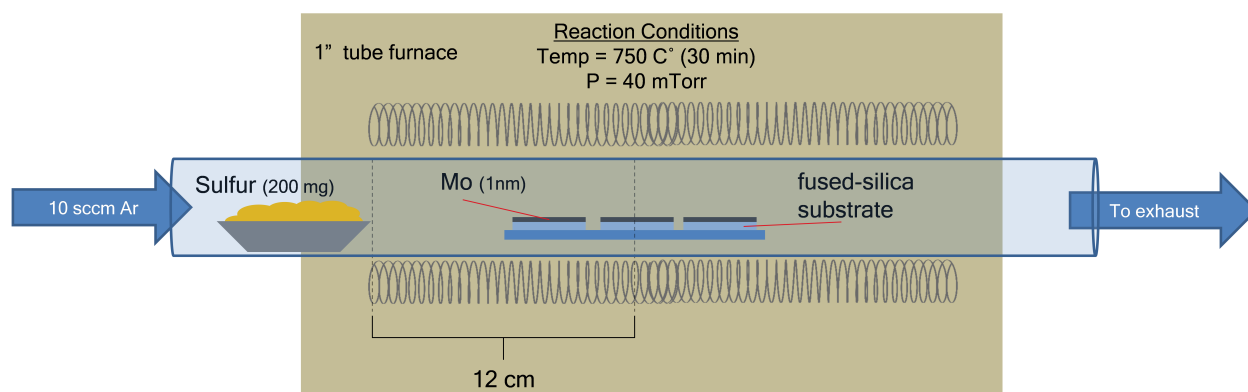


Figure 11.2: Schematic of the synthetic setup used for Mo thin film sulfidation reactions.

The coherent multidimensional spectroscopy system used a 35 fs seed pulse, centered at 800 nm and generated by a 1 kHz Tsunami Ti-sapphire oscillator. The seed was amplified by a Spitfire-Pro regenerative amplifier. The amplified output was split to pump two TOPAS-C collinear optical parametric amplifiers. OPA signal output was immediately frequency doubled with BBO crystals, providing two ≈ 50 fs independently tunable pulses denoted ω_1 and ω_2 with frequencies ranging from 1.62 to 2.12 eV. Signal and idler were not filtered out, but played no role due to their low photon energy. Pulse ω_2 was split into pulses labeled ω_2 and $\omega_{2'}$ to create a total of three excitation pulses.

In this experiment we use motorized OPAs which allow us to set the output color in software. OPA1 and OPA2 were used to create the ω_1 and ω_2 frequencies, respectively. In Figure 11.3 we compare the spectral envelope generated by the OPA at each set color. Negative detuning values correspond to regions of the envelope lower in energy than the corresponding set color. The colorbar allows for comparison between set color intensities. The fluence values reported correspond to the brightest set color for each OPA. A single trace of OPA2 output at set color = 1.95 eV can be found in Figure 11.7.

After passing through automated delay stages (Newport SMC100 actuators), all three beams were focused onto the sample surface by a 1 meter focal length spherical mirror in a distorted BOXCAR geometry to form a 630, 580, and 580 μm FWHM spot sizes for ω_1 , ω_2 , and $\omega_{2'}$, respectively.

Figure 11.4 represents delay corrections applied for each OPA. The corrections were experimentally determined using driven FWM output from fused silica. Corrections were approximately linear against photon energy, in agreement with the normal dispersion of transmissive optics inside our OPAs and between the OPAs and the sample. OPA2 required a relatively small correction along $\tau_{22'}$ (middle subplot) to account for any dispersion experienced differently between the two split beams. OPA1 was not split and therefore needed no such correction.

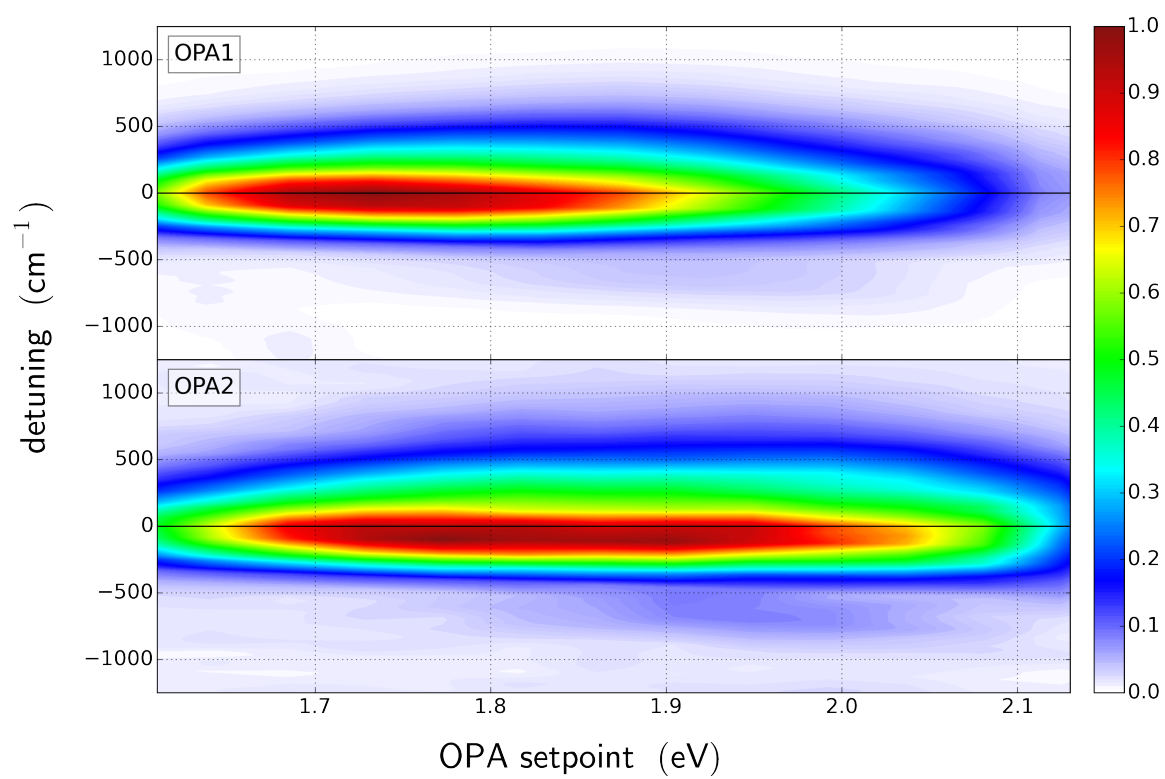


Figure 11.3: OPA outputs at each color explored.

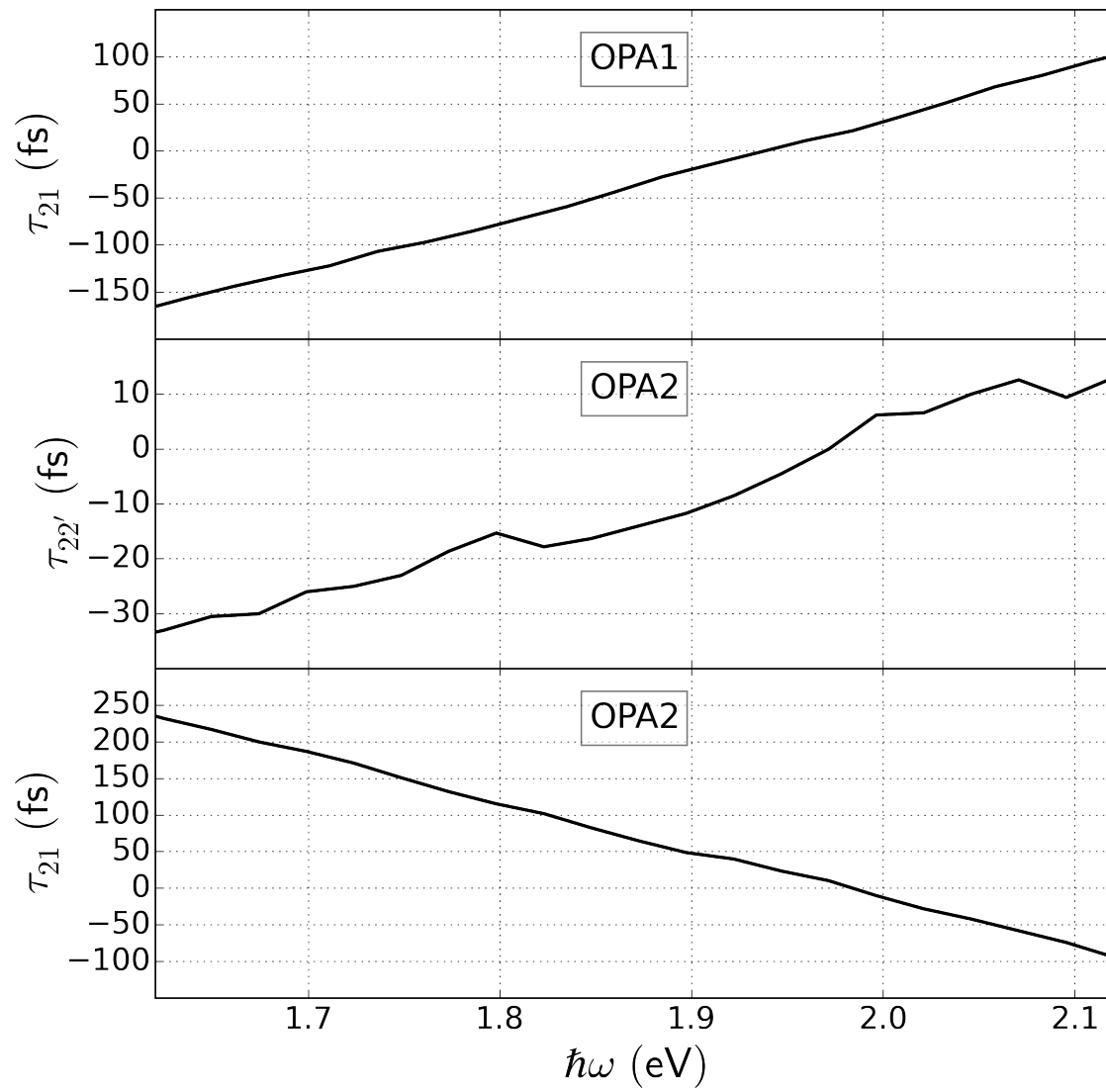


Figure 11.4: Spectral delay correction.

Figure 11.5a represents the to-scale mask that defines our distorted BOXCARS configuration. Relative to the center of the BOXCARS mask (small black dot), ω_1 , ω_2 , and $\omega_{2'}$ enter the sample at angles of 5.0, 1.5, and 1.0 degrees. Each is angled only along the vertical or horizontal dimension, as indicated in Figure 11.5a. This distortion allowed us to remove a large amount of unwanted ω_2 and $\omega_{2'}$ photons from our signal path (Figure 11.5a red star). ω_1 photons were less efficiently rejected, as we show below. The center of the BOXCARS mask was brought into the sample at ≈ 45 degrees. All three beams had S polarization. After reflection, the output beam was isolated using a series of apertures, spectrally resolved with a monochromator (spectral resolution 9 meV), and detected using a photomultiplier (RCA C31034A).

Our experimental setup allowed for the collection of both transmissive and reflective (epi-directional) FWM signal. The 2D delay spectra in Figure 11.5b show the presence of a large nonresonant contribution at the origin for the transmissive FWM signal and weaker signals from the MoS₂ thin film at negative values of τ_{21} and $\tau_{22'}$. The nonresonant contribution is much weaker than the signals from the film for the reflective signal and is the geometry chosen for this experiment. This discrimination between a film and the substrate was also seen in reflective and transmissive CARS microscopy experiments. [247]

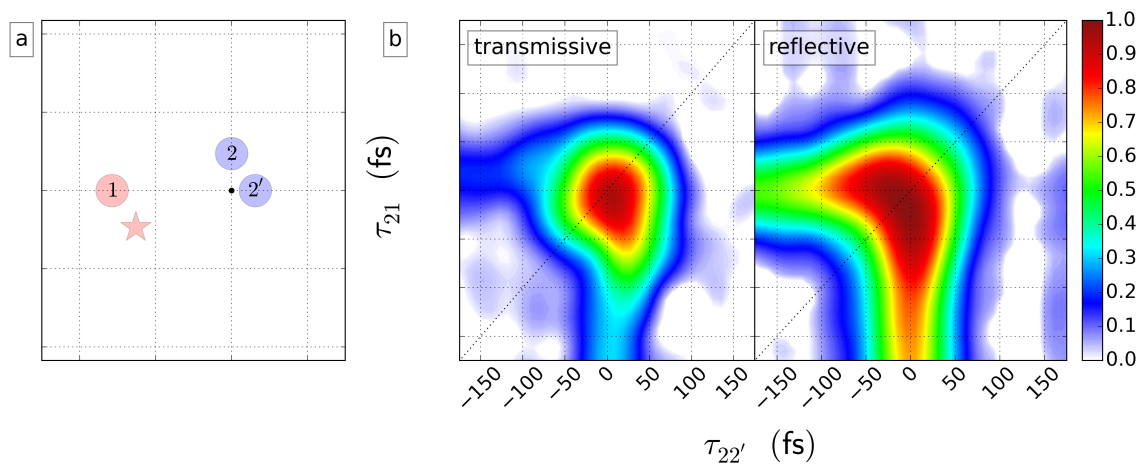


Figure 11.5: (a) Mask. (b) 2D delay spectra at the BB diagonal ($\omega_1 = \omega_2 \approx 1.95$ eV) for transmissive and reflective geometries. Transmissive signal is a mixture of MoS₂ signal and a large amount of driven signal from the substrate that only appears in the pulse overlap region. Reflective signal is representative of the pure MoS₂ response.

Once measured, the FWM signal was sent through a four-stage workup process to create the data set shown here. This workup procedure is visualized in Figure 11.6. We use a chopper and boxcar in active background subtraction mode (averaging 100 laser shots) to extract the FWM signal from ω_1 and ω_2 scatter. We collect this differential signal (Figure 11.6b) in software with an additional 50 shots of averaging. In post-process we subtract ω_2 scatter and smooth the data using a 2D Kaiser window. Finally, we represent the homodyne collected data as $(\text{sig})^{1/2}$ to make the dynamics and line widths comparable to heterodyne-collected techniques like absorbance and pump-probe spectra. Throughout this work, zero signal on the color bar is set to agree with the average rather than the minimum of noise. Values below zero due to measurement uncertainty underflow the color bar and are plotted in white. This is especially evident in lots such as +120 fs in Figure 11.13, where there is no real signal. IPython [248] and matplotlib [86] were important for data processing and plotting in this work.

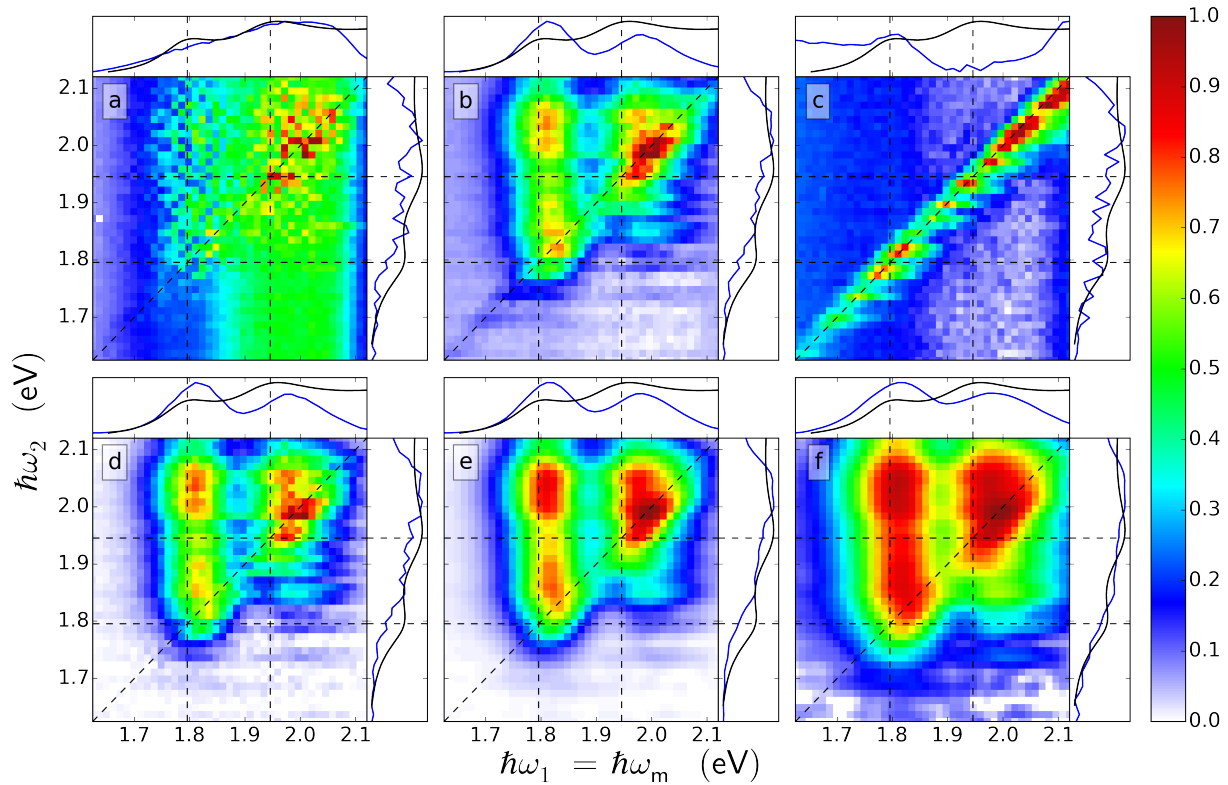


Figure 11.6: Visualization of data collection and processing. With the exception of (c), each subsequent pane represents an additional processing step on top of previous processing. The color bar of each image is separate. (a) Voltages read by the detector at each color combination. The large vertical feature is ω_1 scatter; the shape is indicative of the power curve of the OPA. MoS₂ response can be barely seen above this scatter. (b) Data after chopping and active background subtraction at the boxcar (100 shots). (c) The portion of chopped signal that is not material response. This portion is extracted by averaging several collections at very positive τ_{21} values, where no material response is present due to the short coherence times of MoS₂ electronic states. The largest feature is ω_2 scatter. Cross-talk between digital-to-analog channels can also be seen as the negative portion that goes as ω_1 intensity. (d) Signal after (c) is subtracted. (e) Smoothed data. (f) Amplitude level (square root) data. This spectrum corresponds to that at 0 delay in Figure 11.9. Note that the color bar's range is different than in Figure 11.9.

11.3 Results and discussion

The few-layer MoS₂ thin film sample studied in this work was prepared on a transparent fused silica substrate by a simple sulfidation reaction of a Mo thin film using a procedure modified from a recent report. [244] Figure 11.7a and b show the homogeneous deposition and surface smoothness of the sample over the centimeter-sized fused silica substrate, respectively. The Raman spectrum shows the E_{2g}^1 and A_{1g} vibrational modes (Figure 11.7c) that are characteristic of MoS₂. jteLiSongLin2012a The transmission electron micrograph (TEM) in Figure 11.7d shows the lattice fringes of the film with an inset fast Fourier transform (FFT) of the TEM image indicative of the hexagonal crystal structure of the film corresponding to the 0001 plane of MoS₂. [249] The MoS₂ film thickness was determined to be 2.66 nm by atomic force microscopy and corresponds to approximately four monolayers. Figure 11.73 shows the absorption and fluorescence spectrum of the film along with the A and B excitonic line shapes that were extracted from the absorption spectrum. A representative excitation pulse profile is also shown in red for comparison.

Extracting the exciton absorbance spectrum is complicated by the large “rising background” signal from other MoS₂ bands. With this in mind, we fit the second derivative absorption spectrum to a sum of two second derivative Gaussians, as seen in Figure 11.8. Conceptually, this method can be thought of as maximizing the smoothness (as opposed to minimizing the amplitude) of the remainder between the fit and the absorption spectrum. The fit parameters can be found in the inset table in Figure 11.8. The Gaussians themselves and the remainder can be found in Figure 11.8.

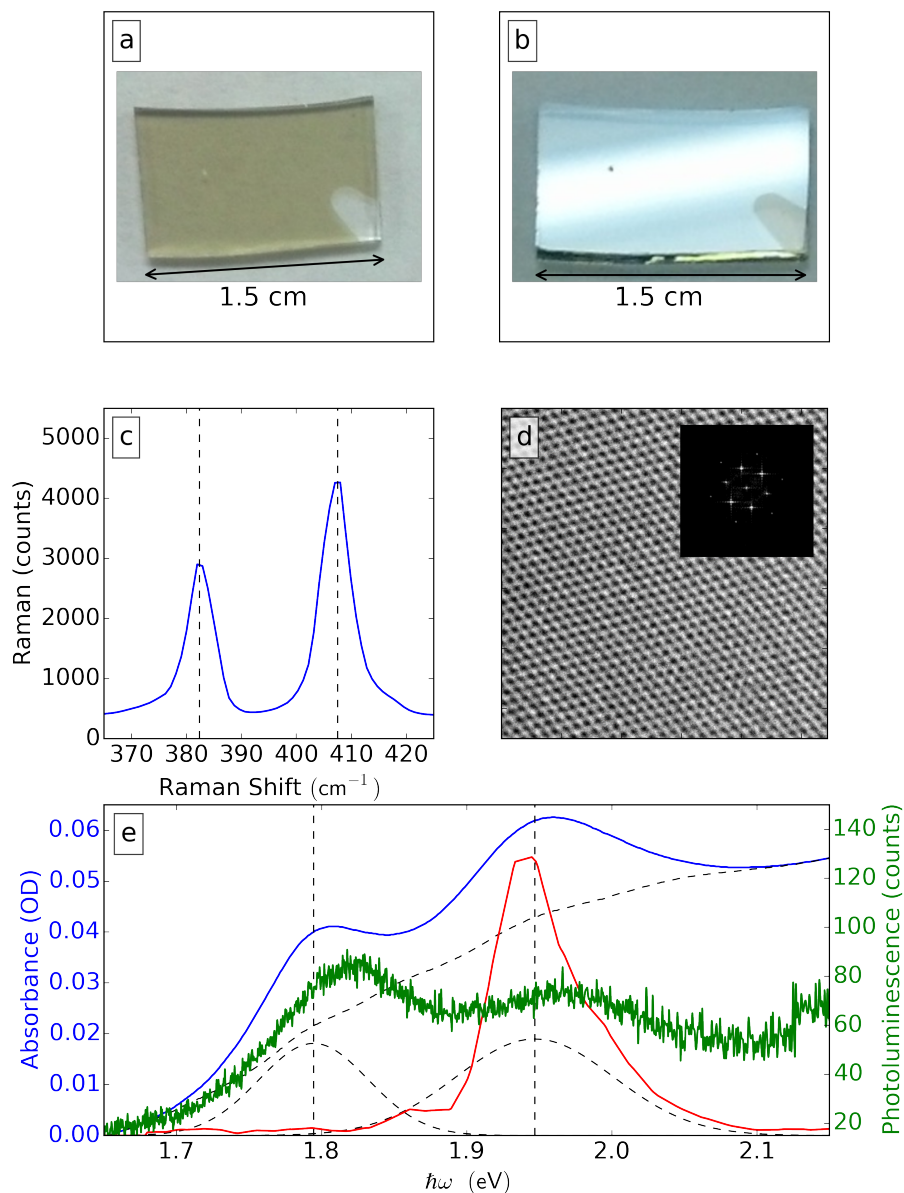


Figure 11.7: Characterization of the few-layer MoS₂ film studied in this work. Optical images of the MoS₂ thin film on fused silica substrate in (a) transmission and (b) reflection. (c) Raman spectrum of the E_{2g}^1 and A_{1g} vibrational modes. (d) High-resolution TEM image and its corresponding FFT shown in the inset. (e) Absorption (blue), photoluminescence (green), Gaussian fits to the A and B excitons, along with the residues between the fits and absorbance (dotted), A and B exciton centers (dotted) and representative excitation pulse shape (red).

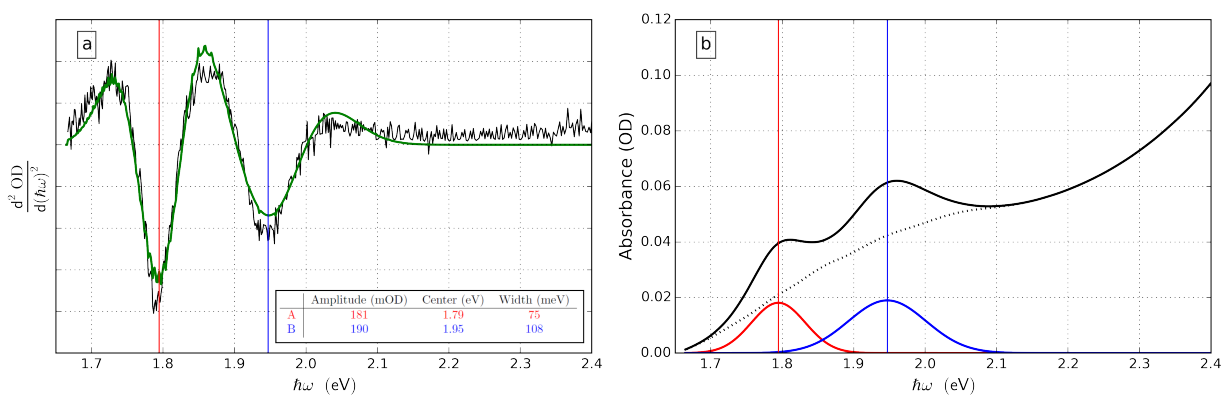


Figure 11.8: Extraction of excitonic features from absorbance spectrum. (a) Second derivative spectra of absorbance (black) and fit second derivative spectrum (green). Gaussian fit parameters are shown in the inset table. (b) Absorption curve (black), Gaussian fits (blue and red), and remainder (black dotted).

The multiresonant CMDS experiment uses ≈ 70 fs excitation pulses created by two independently tunable optical parametric amplifiers (OPAs). Automated delay stages and neutral density filters set the excitation time delays over all values of τ_{21} with $\tau_{22'} = 0$ and the pulse fluence to $90 \mu\text{J}/\text{cm}^2$ ($114 \mu\text{J}/\text{cm}^2$) for the ω_1 (ω_2 and $\omega_{2'}$) beam(s). Each pulse was focused onto the sample using a distorted BOXCARS configuration. [250] The FWM signal was spatially isolated and detected with a monochromator that tracks the output frequency so $\omega_m = \omega_1$. In order to compare the FWM spectra with the absorption spectrum, the signal has been defined as the square root of the measured FWM signal since FWM depends quadratically on the sample concentration and path length.

The main set of data presented in this work is an $\omega_1\omega_2\tau_{21}$ “movie” with $\tau_{22'} = 0$. Figure 11.9 shows representative 2D frequency-frequency slices from this movie at increasingly negative τ_{21} times. Each 2D frequency spectrum contains side plots along both axes that compare the absorbance spectrum (black) to the projection of the integrated signal onto the axis (blue). Along ω_1 (which for negative τ_{21} times acts as the “probe”) we observe two peaks corresponding to the A and B excitons. In contrast, we see no well-defined excitonic peaks along the ω_2 “pump” axis. Instead, the signal amplitude increases toward bluer ω_2 values. The decrease in FWM above 2.05 eV is caused by a drop in the ω_2 OPA power.

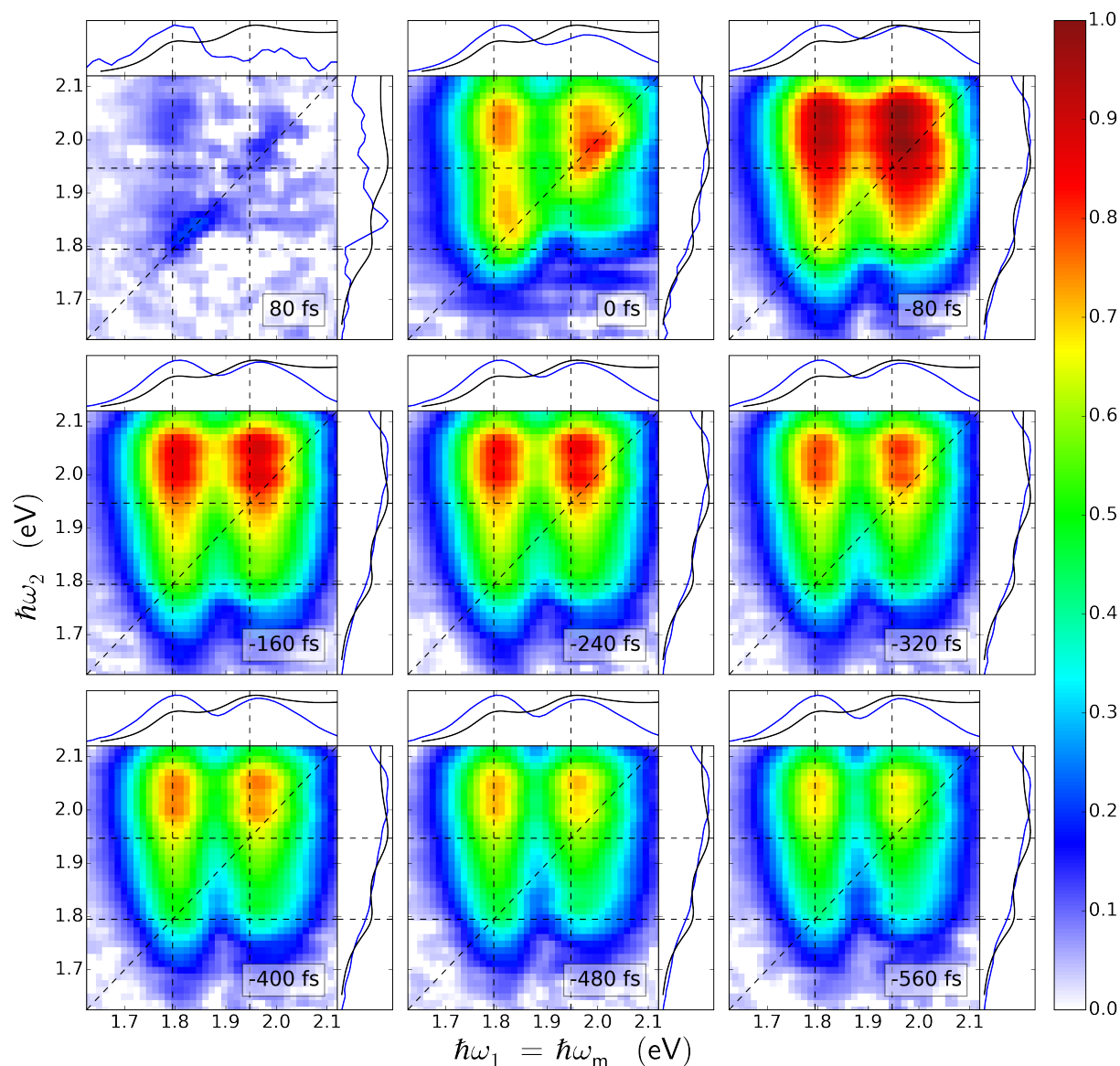


Figure 11.9: 2D frequency-frequency spectra of the MoS₂ sample in the epi configuration. In all spectra $\tau_{22'} = 0$ fs, while τ_{21} is designated in the bottom-right corner of each spectral panel. The color bar defines the square root of the intensity normalized to the most intense feature in the series of spectra. The integration of the signal onto the $\hbar\omega_1 = \hbar\omega_m$ and $\hbar\omega_2$ axes are represented as the blue curves in the top and right side plots, respectively. The side plots also contain the absorbance spectrum (black line) to aid interpretation of the dynamics of the integrated 2D signals. The dashed lines mark the centers of the A and B excitons, as designated from the absorbance spectrum.

Figures 11.10 and 11.11 show representative 2D frequency-delay slices from this movie, where the abscissa is the ω_1 or ω_2 frequency, respectively, the ordinate is the τ_{21} delay time, and the solid bold lines represent five different ω_2 or ω_1 frequencies. The color bar is normalized to the brightest feature in each subplot. This normalization allows comparison of the time dependence of the line shapes, positions, and relative signal amplitudes along the ω_1 or ω_2 axis directly.

Each subplot in Figure 11.10 is similar to published pump-probe, transient absorption, and transient reflection experiments that have measured the electronic dynamics of the A and B excitons. [30, 230, 231, 232, 233, 234, 236, 251] These previous experiments measure relaxation dynamics on the same ≈ 400 -600 fs time scale that is characteristic of Figure 11.10.

Our experiments also show how the spectral features change as a function of the ω_2 excitation frequency. The top two subplots of Figure 11.10 reflect the changes in the AA and BA features, while the bottom two subplots reflect the changes in the AB and BB features. The figure highlights the changes in the relative amplitude of the A and B features as a function of excitation frequency. Both the line shapes and the dynamics of the spectral features are very similar. Figure 11.11 is an excitation spectrum that shows that the dynamics of the spectral features do not depend strongly on the ω_1 frequency.

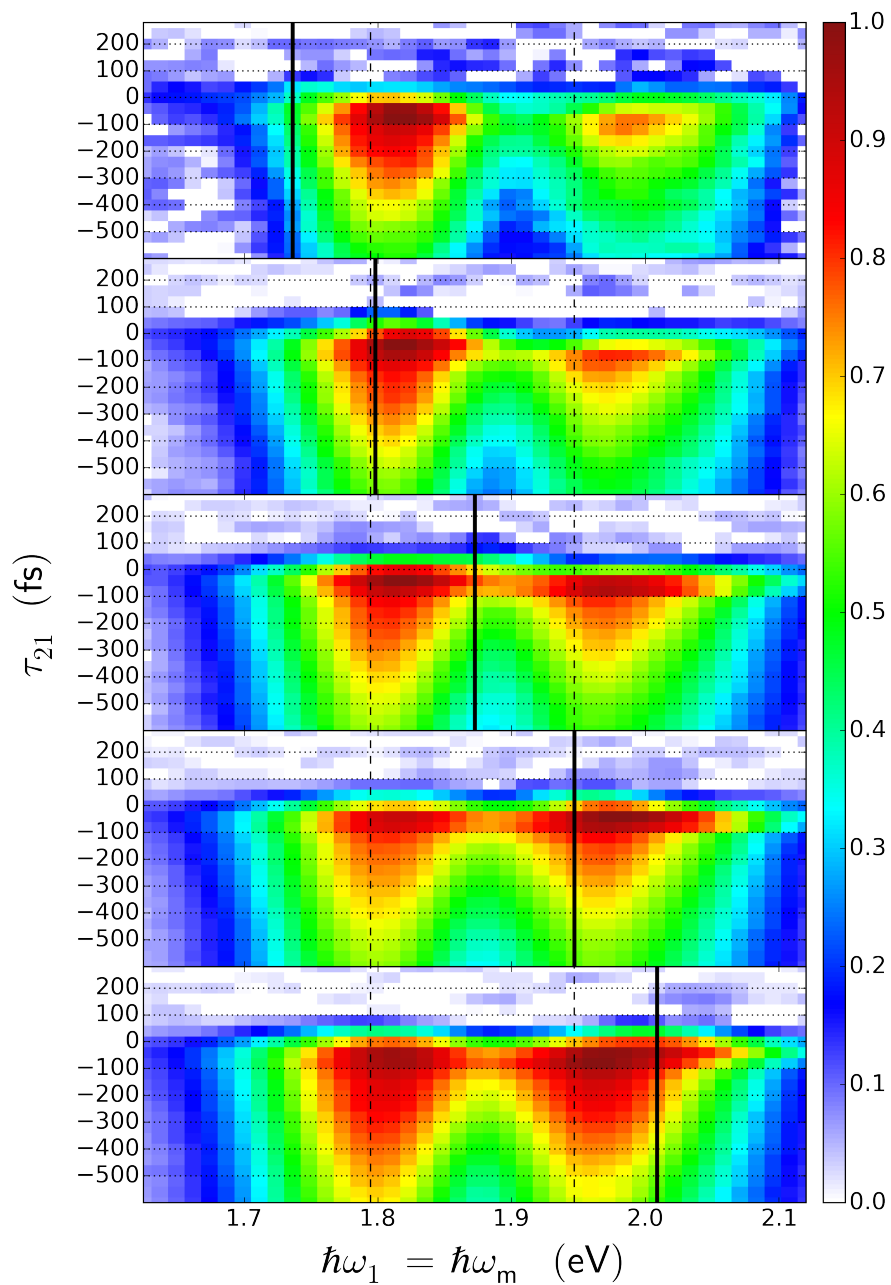


Figure 11.10: Mixed ω_1 — τ_{21} time—frequency representations of the 3D data set at five ascending ω_2 excitation frequencies (solid black lines) showing the impact of the ω_2 excitation frequency on the ω_1 spectral line shape as a function of time. The A and B exciton energies are marked as dashed lines within each spectrum.

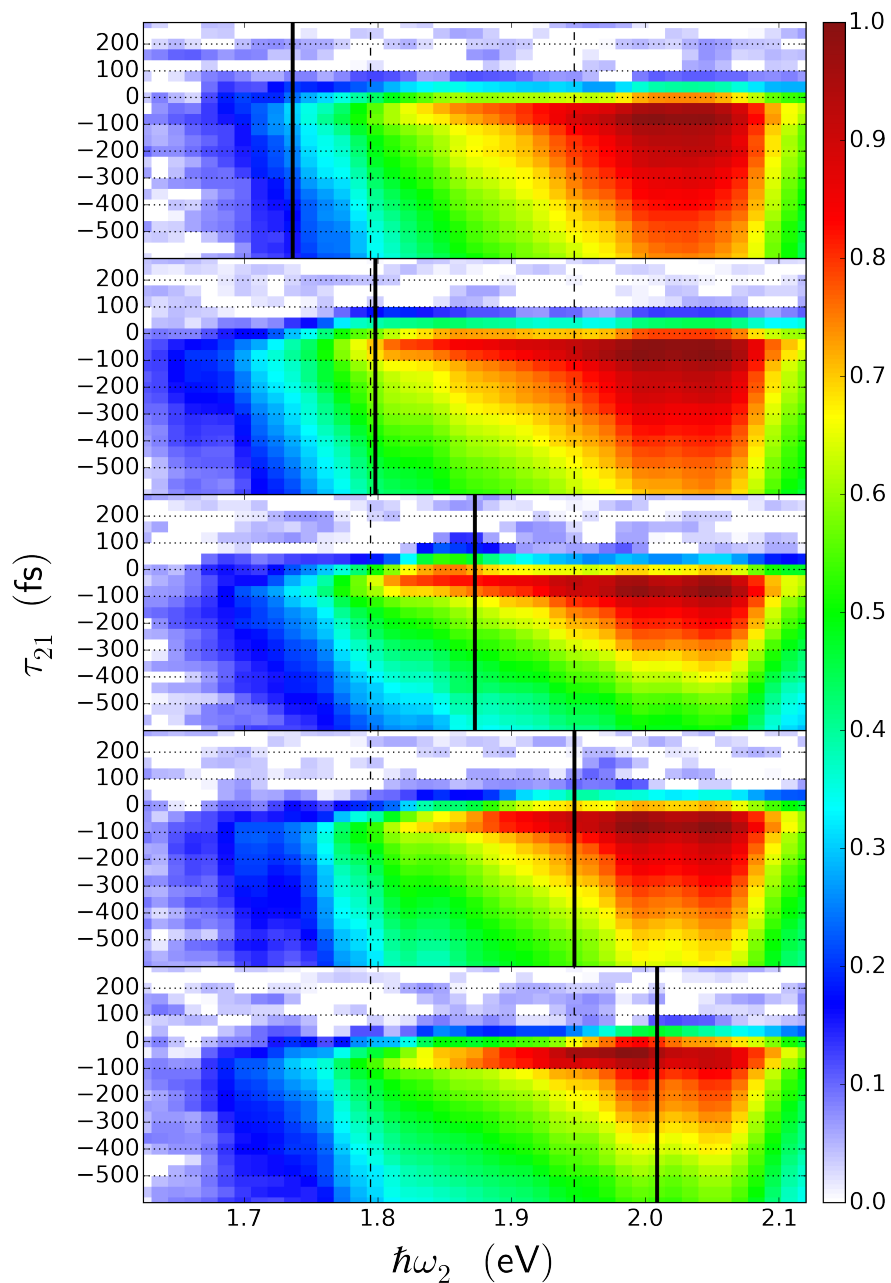


Figure 11.11: Mixed ω_2 — τ_{21} time—frequency representations of the 3D data set at five ascending ω_1 probe frequencies (solid black lines) showing the impact of the ω_1 excitation frequency on the ω_2 spectral line shape as a function of time. The A and B exciton energies are marked as dashed lines within each spectrum.

The spectral features in Figures 11.9, 11.10 and 11.11 depend on the quantum mechanical interference effects caused by the different pathways. [252, 6] These pathways correspond to the time orderings labeled V and VI in Figure 11.7b. The letters denote the density matrix elements, ρ_{ij} , where g represent the ground state and e, e' represent any excitonic state. Interaction with the temporally overlapped ω_2 and ω_2' pulses excites the ee excited-state population and bleaches the ground-state population. Subsequent interaction with ω_1 creates the output coherences for the diagonal spectral features when $\omega_1 = \omega_2$ or the cross-peak features when $\omega_1 \neq \omega_2$. The stimulated emission (SE) and ground-state bleaching (GSB) pathways create the eg or $e'g$ output coherences from the ee and gg populations, respectively, while the excited-state absorption pathway creates the $2e,e$ or $e'+e,e$ biexcitonic output coherences. There is also a population transfer pathway from the ee excited-state population to an $e'e'$ population from which similar SE and ESA pathways occur. Since the ESA pathways destructively interfere with the SE and GSB pathways, the output signal depends on the differences between the pathways. Factors that change the biexcitonic output coherences such as the transition moments, state filling (Pauli blocking), frequency shifts, or dephasing rate changes will control the output signal. State filling and ground-state depletion are important factors for MoS_2 since the transitions excite specific electron and hole spin and valley states in individual layers.

The state-filling and ground-state bleaching effects on the diagonal and cross-peak features in Figure 11.9 depend on the spin and valley states in the output coherence. [29] The effects of these spins will disappear as the spin and valley states return to equilibrium. [253, 254, 255] If we assume spin relaxation is negligible, the FWM transitions that create the diagonal features involve either the A or B ESA transitions, so the resulting $2e,e$ state includes two spin-aligned conduction band electrons and valence band holes. Similarly, the cross-peak regions denoted by AB or BA in Figure 11.1d will have two transitions involving an A exciton, so the initial $e'+e,e$ state will include antialigned spins. A quantitative treatment of the cancellation effects between the GSB, SE, and ESA pathways requires knowledge of the transition moments and state degeneracies and is beyond the scope of this paper. [256]

The most important characteristic of the experimental spectra is the contrast between the absence of well-resolved excitonic features that depend on ω_2 in Figures 11.9 and 11.11 and the well-defined excitonic features that depend on ω_1 . It is also important to note that the projections of the signal amplitude onto the ω_2 axis in Figure 11.9 match closely with the continuum features in the absorption

spectrum and that the line shapes of the features along the ω_2 axis in Figure 11.11 do not change appreciably for different delay times or ω_1 values. It should be noted that the excitation pulse bandwidth (see Figure 11.7e) contributes to the absence of well-resolved A and B excitonic features along ω_2 . The similarity to the continuum states in the absorption spectrum and the absence of a strong dependence on ω_1 show that the continuum states observed at higher ω_2 frequencies participate directly in creating the final output coherences and that their increasing importance reflects the increasing absorption strength of higher energy continuum states. In contrast, the features dependent on ω_1 in Figures 11.9 and 11.10 match the line shapes of the A and B excitonic resonances. Although the relative amplitudes of the spectral features depend on the ω_2 frequency, they do not depend strongly on the delay times. These characteristics show that hot A and B excitonic states undergo rapid intraband population relaxation over a <70 fs time scale set by excitation pulses to the A and B excitonic states excited by ω_1 .

A central feature of Figures 11.9 and 11.10 is that the AB region is much brighter than the BA region. This difference is surprising because simple models predict cross-peaks of equal amplitude, as depicted in Figure 11.1d. The symmetry in simple models arises because the AB and BA cross-peaks involve the same four transitions. The symmetry between AB and BA may be broken by material processes such as population relaxation and transfer, the output coherence dephasing rates, and the bleaching and state-filling effects of the valence and conduction band states involved in the transitions.

We believe that ultrafast intraband population transfer breaks the symmetry of AB and BA cross-peaks. For the BA peak, the interactions with ω_2 and ω_2' generate only A excitons that do not relax on <70 fs time scales. For the AB peak, ω_2 and ω_2' generate two kinds of excitons: (1) B excitons and (2) hot excitons in the A band. Relaxation to A may occur by interband transitions of B excitons or intraband transitions of hot A excitons. Either will lead to the GSB, SE, and ESA of the $ee \rightarrow e'e'$ population transfer pathways. This relaxation must occur on the time scale of our pulse-width since the cross-peak asymmetry is observed even during temporal overlap. We believe that intraband relaxation of hot A excitons is the main factor in breaking the symmetry between AB and BA cross-peaks. Figure 11.10 shows that $B \rightarrow A$ interband relaxation occurs on a longer time scale. The B/A ratio is higher when ω_2 is resonant with the B excitonic transition than when ω_2 is lower than the A exciton frequency (the top subplot). If population transfer of holes from the B to A valence bands occurred during temporal overlap, the B/A ratio would be independent of pump frequency at $\tau_{21} < 0$. Figure 11.12 shows the

delay transients at the different frequencies shown in the 2D spectrum. The colors of the dots on the 2D frequency-frequency spectrum match the colors of the transients. The transients were taken with a smaller step size and a longer time scale than the delay space explored in the 3D data set. The transients are quite similar. Our data are consistent with both monomolecular biexponential and bimolecular kinetic models and cannot discriminate between them. We have fit the decay kinetics to a single exponential model with a time constant of 680 fs and an offset that represents the long time decay. The 680 fs decay is similar to previously published pump-probe and transient absorption experiments. [232, 233, 257]

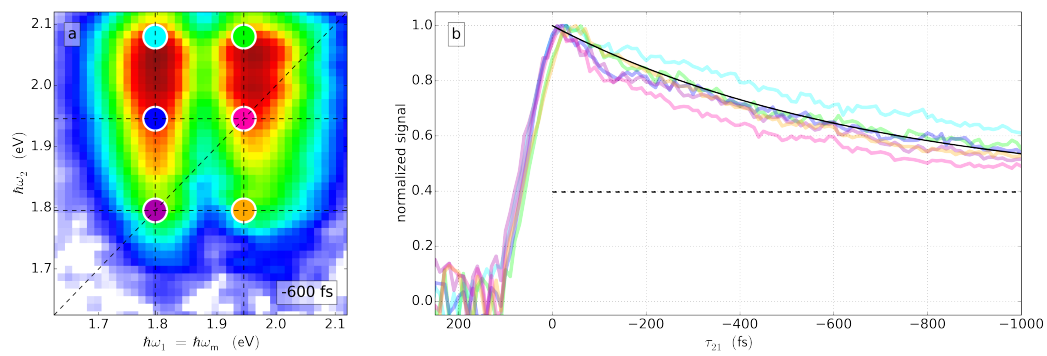


Figure 11.12: Transients taken at the different ω_1 and ω_2 frequencies indicated by the colored markers on the 2D spectrum. The dynamics are assigned to a 680 fs fast time constant (black solid line) and a slow time constant represented as an unchanging offset over this timescale (black dashed line).

The spectral features change quantitatively for delay times near temporal overlap. Figure 11.13 shows a series of 2D spectra for both positive and negative τ_{21} delay times with $\tau_{22'} = 0$. Each spectrum is normalized to its brightest feature. The spectra at positive τ_{21} delay times become rapidly weaker as the delay times become more positive until the features vanish into the noise at +120 fs. The spectra also develop more diagonal character as the delay time moves from negative to positive values. The AB cross-peak is also a strong feature in the spectrum at early times.

The pulse overlap region is complicated by the multiple Liouville pathways that must be considered. Additionally, interference between scattered light from the ω_1 excitation beam and the output signal becomes a larger factor as the FWM signal decreases. Since $\tau_{22'} = 0$, the initial ω_1 pulse creates an excited-state coherence, while the subsequent ω_2 and $\omega_{2'}$ pulses create the output coherence. The output signal is only important at short $\tau_{21} > 0$ values because the initially excited coherence dephases very rapidly. When $\omega_1 \neq \omega_2$, the first two interactions create an $e'e$ zero quantum coherence that also dephases rapidly. However, when $\omega_1 = \omega_2$, the first two interactions create an ee , gg population difference that relaxes on longer time scales. The resulting signal will therefore appear as the diagonal feature in Figure 11.13 (e.g., see the +40 fs 2D spectrum). In addition to the diagonal feature in Figure 11.13, there is also a vertical feature when ω_1 is resonant with the A excitonic transition as well as the AB cross-peak. Although these pathways are depressed when $\tau_{21} > 0$, there is sufficient temporal overlap between the ω_2 , $\omega_{2'}$, and ω_1 pulses to make their contribution comparable to V and VI. More positive values of τ_{21} emphasize the I, III pathways over the V, VI pathways, accounting for the increasing percentage of diagonal character at increasingly positive delays.

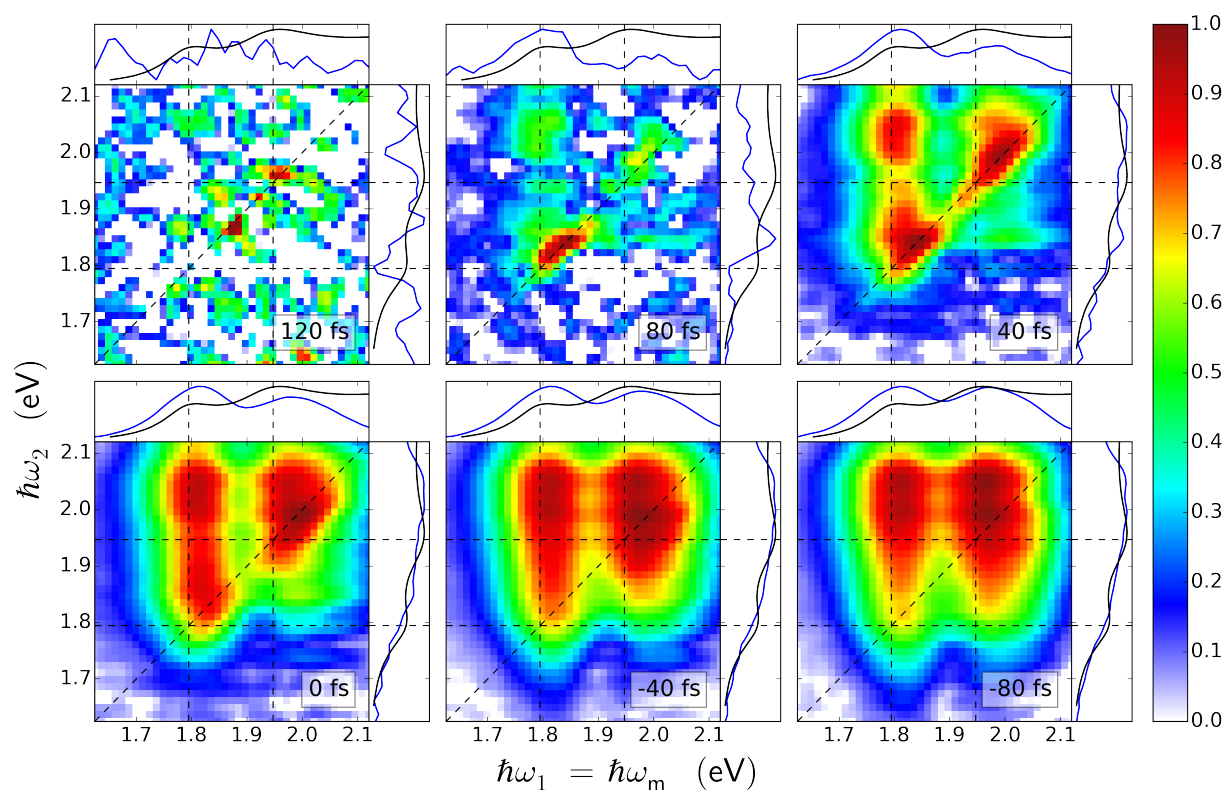


Figure 11.13: 2D frequency-frequency spectra near zero τ_{21} delay times. The signal amplitude is normalized to the brightest features in each spectrum.

11.4 Conclusions

This paper presents the first coherent multidimensional spectroscopy of MoS₂ thin films. CMDS methods are related to the earlier ultrafast pump-probe and transient absorption methods since they all share bleaching, stimulated emission, Pauli blocking, and excited-state absorption pathways, but they differ in how these pathways define the spectra. In addition, CMDS methods have many additional pathways that become important when the coherence dephasing times are longer than the excitation pulse widths. In this work, the dephasing times are short so the pathways are identical to transient absorption. This work reports the first frequency-frequency-delay spectra of MX₂ samples. These spectra are complementary to previous work because they allow a direct comparison between the initially excited excitonic states and the states creating the final output coherence. The spectra show that the same hot A and B exciton continuum states that are observed in the absorption spectrum also dominate the CMDS excitation spectra. They also show that rapid, < 70 fs intraband relaxation occurs to create the band-edge A and B excitonic features observed in the CMDS spectrum. The relative intensity of the diagonal peak features depends on the relative absorption strength of the A and B excitons. The relative intensity of cross-peak features in the 2D spectra depends on the excitation frequency. Excitation at or above the B exciton feature creates strong cross-peaks associated with hot A and B excitons that undergo ultrafast intraband population transfer. Excitation below the B excitonic feature creates a weak cross-peak indicating A-induced B-state bleaching but at a lower signal level corresponding to the lower optical density at this energy. Population relaxation occurs over ≈ 680 fs, either by transfer to traps or by bimolecular charge recombination.

These experiments provide the understanding of MoS₂ coherent multidimensional spectra that will form the foundation required to measure the dynamical processes occurring in more complex MoS₂ and other TMDC heterostructures with quantum-state resolution. The frequency domain based multiresonant CMDS methods described in this paper will play a central role in these measurements. They use longer, independently tunable pulses that provide state-selective excitation over a wide spectral range without the requirement for interferometric stability.

11.5 Errata

The following is an errata for the of this chapter, published in November 2015. [157]

- Reference 13 is identical to reference 9.
- In the last paragraph of the introduction the sentence “The experimental spectra differ from the simple 2D spectrum shown in Figure 1d and those of earlier CMDS experiments with model systems” appears. This sentence cites references 6 through 10. Instead, it should cite references 15 through 20.
- In the last paragraph beginning on page 12148, the text “Automated delay stages and neutral density filters set the excitation time delays over all values of τ_{21} with $\tau_{22} = 0$ ” appears. For the second τ , the subscript should read $22'$, not 22.
- Caption of Figure 5 reads, in part: “showing the impact of the ω_1 excitation frequency on the ω_1 spectral line shape”. This should instead read “showing the impact of the ω_1 excitation frequency on the ω_2 spectral line shape”. The subscript on the last ω should be a 2 and not a 1.
- Figure 6 $e'+e,e'$ should read $e'+e,e$ and vice versa.

Chapter 12

Measurement of ultrafast dynamics within PEDOT:PSS using three-pulse photon echo spectroscopy

This Chapter presents content first published in Horak et al. [258]. The authors are:

1. Erik H. Horak
2. Morgan T. Rea
3. Kevin D. Heylman
4. David Gelbwaser-Klimovsky
5. Semion K. Saikin
6. Blaise J. Thompson
7. Daniel D. Kohler
8. Kassandra A. Knapper
9. Wei Wei
10. Feng Pan
11. Padma Gopalan
12. John C. Wright
13. Alán Aspuru-Guzik
14. Randall H. Goldsmith

This Chapter focuses on my contribution.

12.1 Introduction

Poly(3,4-ethylenedioxythiophene)-poly(styrenesulfonate) (PEDOT:PSS) is a transparent, electrically conductive (up to 4380 S cm^{-1} [259]) polymer. It has found widespread use as a flexible, cheap alternative to inorganic transparent electrodes such as indium tin oxide.

As a polymer, PEDOT:PSS implicitly contains a large amount of structural inhomogeneity. On top of this, PEDOT:PSS is a two component material, composed of PEDOT (low molecular weight, p-doped, highly conductive) and PSS (high molecular-weight, insulating, stabilizing). These two components segment into domains of conductive and non-conductive material, leading to even more structural inhomogeneity. In summary, PEDOT:PSS has a complex, nested microstructure. From smallest to largest:

- PEDOT oligomers (6—18-mers) [260]
- these oligomers π -stack to form small nanocrystalites, 3 to 14 oligomers for pristine films to as many as 13—14 oligomers for more conductive solvent treated films [261]
- nanocrystalites then arrange into globular conductive particles in a pancake-like shape [262, 263]
- these particles themselves are then linked via PSS-rich domains and assembled into nanofibril geometry akin to a string of pearls [259, 264]
- nanofibrils interweave to form thin films, with PSS capping layer at surface [265]

In order to be conductive, PEDOT:PSS needs to have good spatial and energetic overlap between electronic states throughout the thin film. The exact energetics and dynamics of these electronic states, then, is a crucial piece of information needed to understand the mechanism of conductivity in PEDOT:PSS. The electronic states responsible for conductivity have very broad and featureless transitions in the mid infrared. Bulk, linear spectroscopy cannot tease out the relative contribution of homogeneous and inhomogeneous broadening in the breadth of these transitions. Multidimensional spectroscopy is able to tease these two broadening mechanisms apart.

In this chapter, I report on my usage of three pulse echo (3PE) spectroscopy to constrain homogeneous and inhomogeneous linewidths in PEDOT:PSS.

12.2 Methods

PEDOT:PSS (Orgacon Dry, Sigma Aldrich) was dropcast onto a glass microscope slide at 1 mg/mL at a tilt to ensure homogeneous film formation. The sample was heated at 100 °C for ~15 min to evaporate water.

An ultrafast oscillator (Spectra-Physics Tsunami) was used to prepare ~35 fs seed pulses. These were amplified (Spectra-Physics Spitfire Pro XP, 1 kHz), split, and converted into 1300 nm 40 fs pulses using two separate optical parametric amplifiers (Light Conversion TOPAS-C): “OPA1” and “OPA2”. Pulses from OPA2 were split again, for a total of three excitation pulses: ω_1 , ω_2 and $\omega_{2'}$. These were passed through motorized (Newport MFA-CC) retroreflectors to control their relative arrival time (“delay”) at the sample: $\tau_{21} = \tau_2 - \tau_1$ and $\tau_{22'} = \tau_2 - \tau_{2'}$. The three excitation pulses were focused into the sample in a 1° right-angle isocetes triange, as in the BOXCARS configuration. [250] Each excitation beam was 67 nJ focused into a 375 μm symmetric Gaussian mode for an intensity of 67 $\mu\text{J}/\text{cm}^2$. A new beam, emitted coherently from the sample, was isolated with apertures and passed into a monochromator (HORIBA Jobin Yvon MicroHR, 140 mm focal length) with a visible grating (500 nm blaze 300 groves per mm). The monochromator was set to pass all colors (0 nm, 250 μm slits) to keep the measurement impulsive. Signal was detected using an InSb photodiode (Teledyne Judson J10D-M204-R01M-3C-SP28). Four wave mixing was isolated from excitation scatter using dual chopping and digital signal processing.

Two dimensional τ_{21} , $\tau_{22'}$ scans were taken for two phase matching configurations: (1) $k_{\text{out}} = k_1 - k_2 + k_{2'}$ (3PE) and (2) $k_{\text{out}} = k_1 + k_2 - k_{2'}$ (3PE*). Figure 12.1 diagrams the phase matching mask used in this set of experiments. The rephasing and nonrephasing pathways exchange their time dependance between these two configurations. Comparing both pathways, rephasing-induced peak shifts can be extracted as in 3PE. [165] All data was modeled using numerical integration of the Liouville-von Numann equation.

Continuously variable ND filters (THORLABS NDC-100C-4M, THORLABS NDL-10C-4) were used to ensure that all three excitation pulse powers were equal within measurement error.

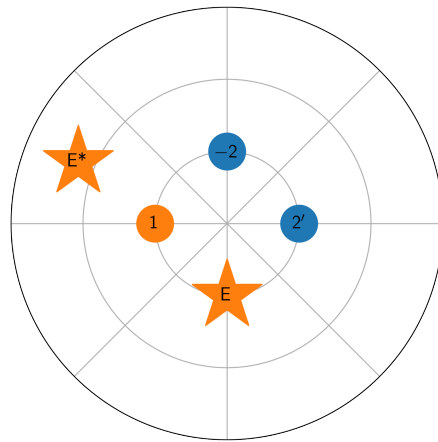


Figure 12.1: Phase matching mask used in this experiment. Each successive ring subtends 1 degree, such that the excitation pulses are each angled one degree relative to the mask center. The two stars mark the two output poyntings detected in this work.

12.3 Results

Figure 12.2 shows the transmission, reflectance, and extinction spectrum of the thin film used in this work. The region under investigation is shaded green. Reflectance is remarkably low across the visible and near infrared, and transmission does not change much at all over the region under investigation.

Figure 12.3 shows the ten raw 2D delay-delay scans that comprise the primary dataset described in this section. The rows correspond to the two phase matching conditions, as labeled. The data is presented on the intensity level, as raw as possible. The five repetitions of each experiment are truly remarkably similar, showing that no damage was being done during the experiment. Each row is normalized to the same factor, showing the remarkable *quantitative* agreement between scans. In total, these 10 scans comprise roughly eight hours of continuous illumination for this sample.

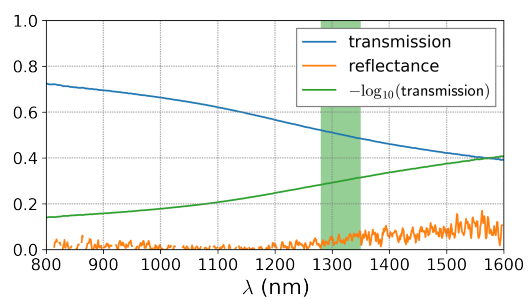


Figure 12.2: Thin film spectra. Transmission, reflectance, and extinction spectrum of the thin film used in this work. Extinction is $\log_{10}(\text{transmission})$.

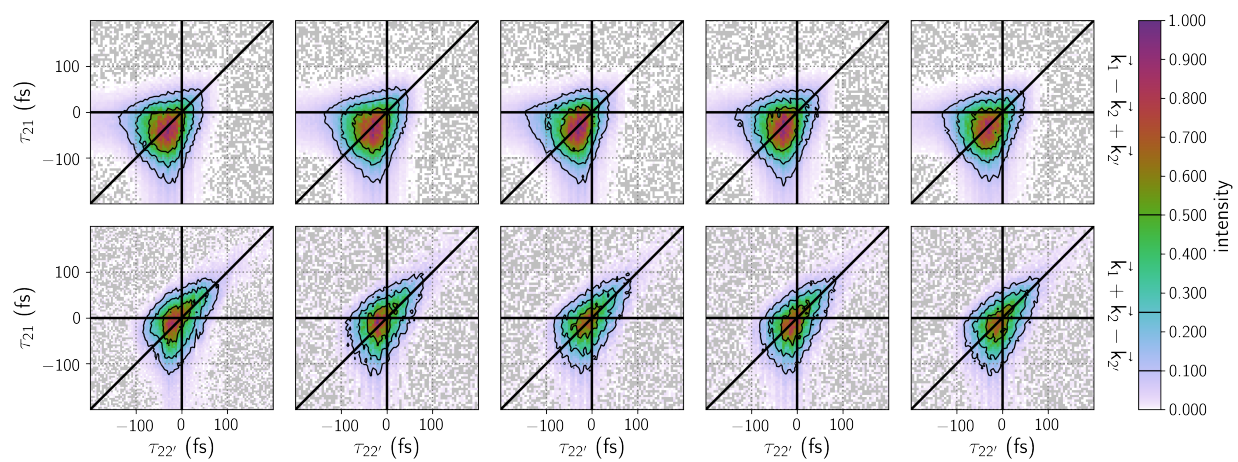


Figure 12.3: Raw ultrafast data. Unprocessed two-dimensional delay-delay plots. Each discrete acquisition is plotted as a single colored pixel. Grey pixels correspond to negative results, which appear in the no-signal regions due to random noise.

12.4 Discussion

12.4.1 Assignment of zero delay

The absolute position of complete temporal overlap of the excitation pulses (zero delay) is a crucial step in determining the magnitude of the peak shift and therefore the total rephasing ability of the material. The strategy for assigning zero delay relies upon the intrinsic symmetry of the two-dimensional delay space. Figure 12.4 labels the six time-orderings (TOs) of the three pulses that are possible with two delays. The TO labeling scheme follows from a convention first defined by Meyer, Wright and Thompson. [90] Kohler et al. [123] first discussed how these TOs relate to traditional 3PE experiments. Briefly, spectral peak shifts into the rephasing TOs III and V when inhomogeneous broadening creates a photon echo in the III and V rephasing pathways colored orange in Figure 12.4. For both phase-matching conditions, there are two separate 3PE peak shift traces (represented as black arrows in Figure 12.4), yielding four different measurements of the photon echo. Since both 3PE and 3PE* were measured using the same alignment on the same day, the zero delay position is identical for the four photon echo measurements. We focus on this signature when assigning zero delay—zero is correct only when all four peak shifts agree. Conceptually, this is the two-dimensional analogue to the traditional strategy of placing zero such that the two conjugate peak shifts (3PE and 3PE*) agree. [165]

We found that the 3PEPS traces agree best when the data in Figure 12.3 is offset by 19 fs in $\tau_{22'}$ and 4 fs in τ_{21} . The entire 3PEPS trace (τ vs T) is shown for regions I, III (purple and light green traces) and V, VI (yellow and light blue traces) for the $\vec{k}_1 - \vec{k}_2 + \vec{k}_{2'}$ and $\vec{k}_1 + \vec{k}_2 - \vec{k}_{2'}$ phase matching conditions, respectively. Peak-shift magnitudes were found with Gaussian fits on the intensity level, in accordance with 3PEPS convention. [165] The bottom subplot of Figure 12.6 shows the agreement between the four traces for $T > 50$ fs where pulse-overlap effects become negligible. These pulse-overlap effects cause the 3PEPS at small T even without inhomogeneous broadening. [165] At long T , the average static 3PEPS is 2.5 fs.

There is a deviation of the TO I-III 3PEPS* trace (green line) from the other traces. It is attributed to a combination of excitation pulse distortions and line shape differences between OPA1 and OPA2 (see Figure 12.2) and small errors in the zero delay correction. Figure 12.5 shows what the four 3PEPS

traces would look like for different choices of zero-delay. The inset numbers in each subplot denote the offset (from chosen zero) in each delay axis.

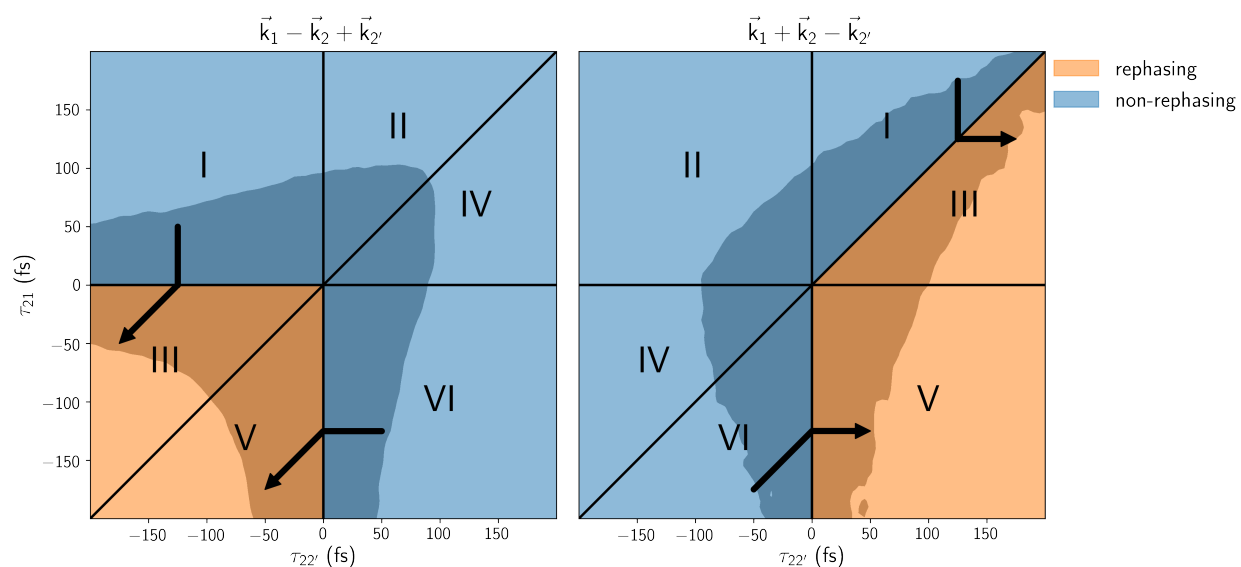


Figure 12.4: Representation of 2D delay space. Representation of symmetry between the two phase-matched experiments performed in this work. In each two-dimensional delay space, the six TOs are labeled. Pathways III and V are rephasing (orange), all other pathways are non-rephasing (blue). Thick black arrows are drawn along the τ trace for constant $T = 125$ fs, with arrowheads pointing in the direction of shift for positively correlated systems. The region with signal above 10% (processed dataset, amplitude level) is shaded to guide the eye.

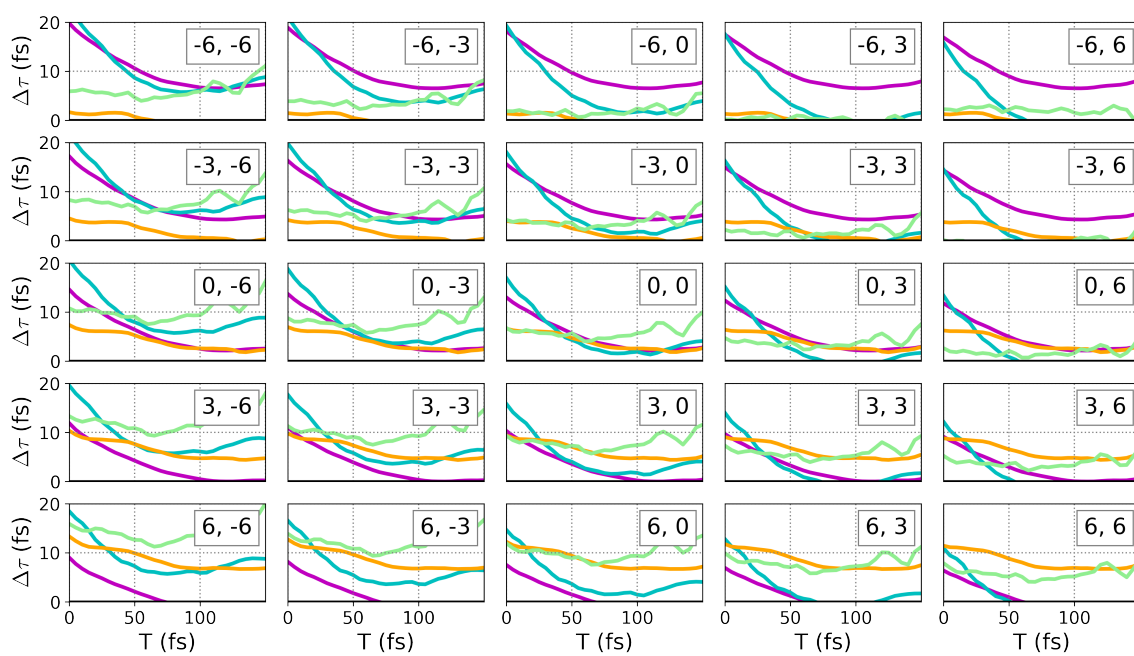


Figure 12.5: Delay offsets. Comparison between 3PEPS traces at different delay offsets. Inset is D_1 , D_2 offset in fs.

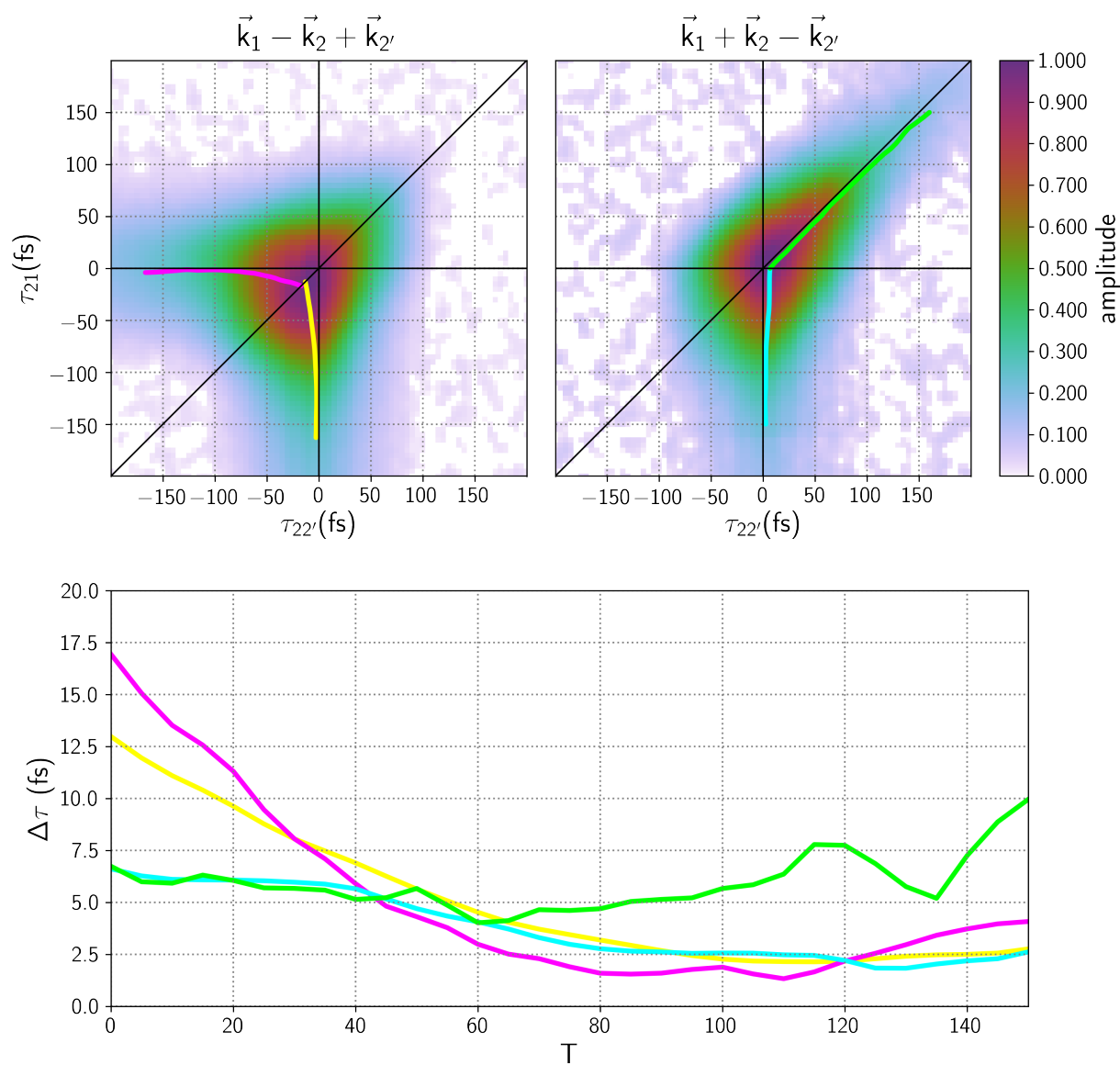


Figure 12.6: 3PEPS traces. Fully processed 2D delay scans (upper) and 3PEPS traces for both rephasing pathways and both phase matching conditions. The 3PEPS traces are shown mapped onto the 2D space (upper) and overlaid for comparison (lower).

12.4.2 Numerical model

We simulated the 3PEPS response of PEDOT:PSS through numerical integration of the Liouville-von Neumann Equation. Integration was performed on a homogeneous, three-level system with coherent dynamics described by

$$\frac{1}{T_2} = \frac{1}{2T_1} + \frac{1}{T_2^*}, \quad (12.1)$$

where T_2 , T_1 and T_2^* are the net dephasing, population relaxation, and pure dephasing rates, respectively. A three-level system was used because a two-level system cannot explain the population relaxation observed at long populations times, T . This slow decay may be the same as the slowly decaying optical nonlinearities in PEDOT:PSS. [266] Inhomogeneity was incorporated by convolving the homogeneous response with a Gaussian distribution function of width Δ_{inhom} and allowing the resultant polarization to interfere on the amplitude level. This strategy captures rephasing peak shifts and ensemble dephasing.

It is difficult to determine the coherence dephasing and the inhomogeneous broadening using 3PE if both factors are large. To extract T_2^* and Δ_{inhom} , we focused on two key components of the dataset, coherence duration and peak shift at large T . Since dephasing is very fast in PEDOT:PSS, we cannot directly resolve an exponential free induction decay (FID). Instead, our model focuses on the FWHM of the τ trace to determine the coherence duration. At $T > 50$ fs, the transient has a FWHM of ~ 80 fs (intensity level). For comparison, our instrumental response is estimated to be 70-90 fs, depending on the exact value of our pulse duration Δ_t (35-45 fs FWHM, intensity level). An experimental peak shift of 2.5 fs was extracted using the strategy described above. Taken together, it is clear that both pure dephasing and ensemble dephasing influence FWHM and peak shift so it is important to find values of T_2^* and Δ_{inhom} that uniquely constrain the measured response.

We simulated the τ trace for a variety of Δ_{inhom} and T_2 values. The results for $\Delta_t = 40$ fs are summarized in Figure 12.7. The lines of constant T_2 span from $\Delta_{\text{inhom}} = 0$ (green left ends of curves) to the limit $\Delta_{\text{inhom}} \rightarrow \infty$ (blue right ends of curves). The lines of constant T_2 demonstrate that ensemble dephasing reduces the transient duration and introduces a peak shift. The influence of inhomogeneity

on the observables vanishes as $T_2 \rightarrow \infty$.

We performed simulations analogous to those in Figure 12.7 for pulse durations longer and smaller than $\Delta_t = 40$ fs. Longer pulse durations create solutions that do not intersect our experimental point (see right-most subplot of Figure 12.7), but shorter pulse durations do. [TABLE] summarizes the coherence dephasing time and inhomogeneous broadening values that best matches the experimental FWHM and inhomogeneous broadening value for $\Delta_t = 35, 40$ and 45 fs. Clearly, there is no upper limit that can provide an upper limit for the inhomogeneous broadening.

Our model system does an excellent job of reproducing the entire 2D transient within measurement error (Figure 12.8). The most dramatic disagreement is in the upper right, where the experiment decays much slower than the simulation. Our system description does not account for signal contributions in TOs II and IV, where double quantum coherence resonances are important. In addition, excitation pulse shapes may cause such distortions. Regardless, these contributions do not affect our analysis.

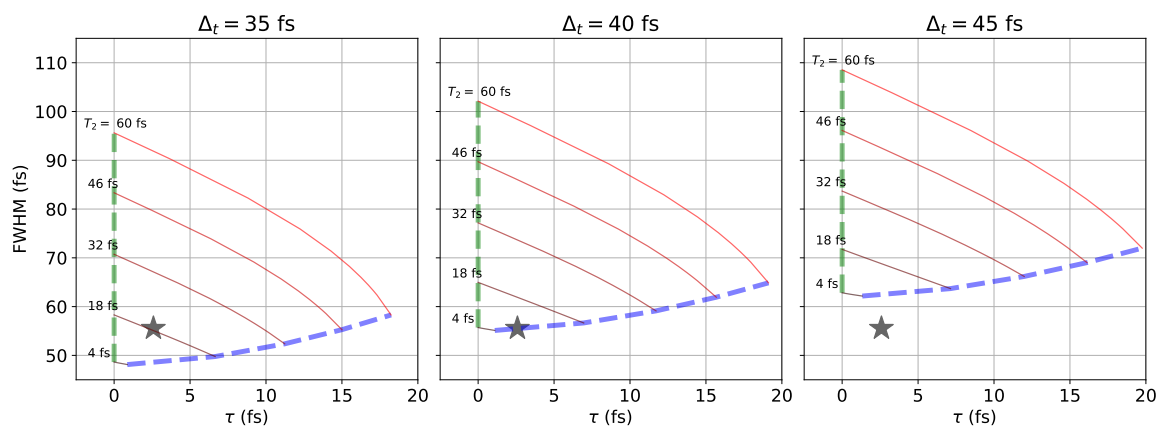


Figure 12.7: 3PEPS parameter space. Interplay of pure and ensemble dephasing on the coherent transient duration and the peak shift value for the three pulse-widths considered in Table 12.1. Red lines represent the parameters for constant values of T_2 and varying amounts of Δ_{inhom} . The domain of possible observables is bounded (blue hash for $\Delta_{inhom} \rightarrow \infty$, green hash for $\Delta_{inhom} = 0$). Also shown is the measured FWHM for the PEDOT:PSS thin film (star).

Δ_t (fs)	T_2 (fs)	$\hbar T_2^{-1}$ (meV)	Δ_{inhom} (meV)
45	—	—	—
40	10	66	∞

Table 12.1: Fitted parameters for the coherent transient. The FWHM of the homogeneous line shape is $\hbar T_2^{-1}$.

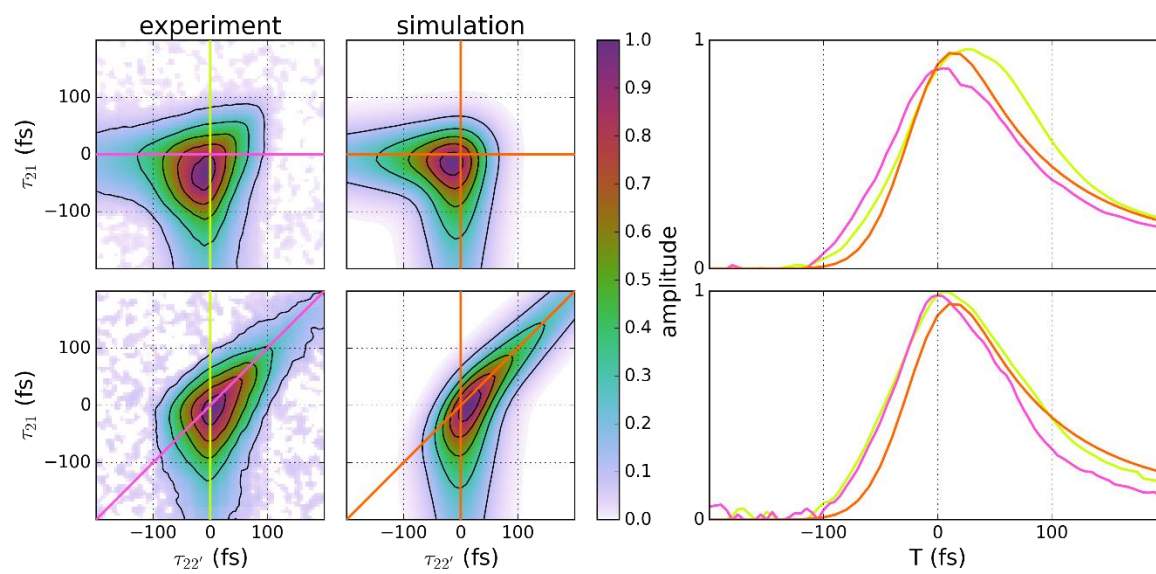


Figure 12.8: Agreement between simulation and experiment. Experiment and simulation in the full 2D representation (left) and transient grating slices (right), for both phase matching conditions (top and bottom). The identity of each slice can be inferred from its color. In this case the displayed simulation is for $\Delta_t = 35$ fs, with the appropriate T_2 and Δ_{inhom} as seen in Table 12.1. Simulations for other pulse-widths look very similar.

12.5 Conclusions

To assess homogeneous and inhomogeneous linewidth, we performed ultrafast four wave mixing spectroscopy on a drop-cast PEDOT:PSS thin film. Under Redfield theory, the homogeneous linewidth of any transition is determined by pure dephasing and population relaxation [267], although ensemble dephasing can become relevant for very inhomogeneously broadened systems. Three-pulse photon echo (3PE) analysis can distinguish between homogeneous and inhomogeneous broadening. [165] We collected the transient grating population relaxation trace and 3PE traces and find that the net dephasing and population relaxation are both fast, comparable to our pulse width. Through numerical modeling, we extract a population time of 80 fs, a homogeneous dephasing time of < 18 fs (> 73 meV), and an inhomogeneous broadening factor of > 43 meV. Extremely fast (single fs) carrier scattering time constants have also been observed for PEDOT-base conductive films. [268, 269, 270]

Part IV

Appendix

Appendix A

Procedures

These are the vegetables we must eat before we can have our pixels of ice cream.

– Wright Group saying.

In this chapter I document the various procedures that I have performed to maintain the MR-CMDS instruments.

A.1 “Six-month” maintenance

The laser system that the Wright Group’s MR-CMDS instruments use requires regular maintenance. Each component is sensitive to lab conditions such as temperature, humidity, and vibrations. Small deviations in upstream components can cause large problems in downstream optics. Regular maintenance involves “tune-up” of upstream components so that upstream lasers function well and ideally couple into downstream lasers.

Historically, the Wright Group has engaged in reactive maintenance: a “fix it if it’s broken, don’t touch otherwise” kind of approach. This approach makes a lot of sense for instruments that are quick to fix, and have few active users. I instituted a proactive, regular maintenance procedure (described below) that has improved the predictability of instrumental performance. Predictability is key for instruments with multiple users.

Supplies needed:

1. 5 gallons distilled water.
2. Filters.
 - (a) 2x 5 micron Liquatec SDF-25-0505
 - (b) 2x 20 micron general purpose water filter (sold in stockroom)
3. Nalco fluids.
 - (a) 4 gallons Nalco 460-PCCL104 (the pink stuff).
 - (b) 1 gallon Nalco 460-CCL2567 (the clear stuff).
4. 150x 5 mg activated charcoal packets (Newport Part # 90035762)

Procedure:

1. Turn off all lasers in lab.
2. In no particular order:
 - Computers in lab.
 - (a) Backup contents.
 - (b) Uninstall unnecessary software.
 - (c) Update all necessary software.
 - (d) Restart.
 - Chiller maintenance.
 - See Section A.2
 - See Section A.3
 - See Section A.4
 - Lab cleaning.
 - (a) Mop the floor, trying to get as much dust as possible. Sweeping in the laser lab is discouraged, because it knocks more dust into the air and onto the lasers.
 - (b) Take out trash.
 - (c) Empty homeless cables box.
 - (d) Empty homeless screws box.
 - (e) Organize any cabling mess.
 - (f) Remove unused components (optics, electronics, cabling, etc) from laser tables and

surroundings. Place into appropriate locations.

(g) Organize optics cabinet.

(h) Organize electronics bench.

3. Let lab sit overnight to allow dust to settle and the air to dehumidify.
4. Start up system again.

A.2 Lytron Kodiak RC006

We have one Lytron Kodiak RC006: Model Number RC006G03BB1C002, Serial Number 739383-02.

Regular chiller maintenance:

1. Gather supplies.
 - (a) 1 gallon distilled water (do not use deionized)
 - (b) 1 gallon Nalco 460-PCCL104 (the pink stuff)
2. Drain system completely.
 - (a) Turn off chiller.
 - (b) Break tubing at push-to-connect, plugging return.
 - (c) Allow chiller to run until liquid stops flowing. You will get a pressure error, ignore (silence) it.
 - (d) Use tube to mouth-syphon remaining liquid from within chiller
 - (e) Reconnect at push-to-connect
3. Fill chiller with distilled water (should require \approx 1 gallon), let run for 30 minutes.
4. Drain system again.
5. Replace filter.
6. Fill system with one gallon Nalco 460-PCCL104 (the pink stuff).
7. Turn chiller on, top-off with distilled water.
8. Record maintenance, order new supplies for next time if necessary.

In the past we have had trouble with low flow errors upon system startup. These seem to have been fixed by adding a “high” flow loop connecting the outlet and inlet of the chiller. Ideally the pressure drop across this loop is sufficient to still drive fluid through the laser.

A.3 PolyScience 6000 Series

We own two PolyScience chillers—different models but functionally equivalent.

Grey: Serial Number 3E1161245

White and Blue: Serial Number 4K1050550

Regular chiller maintenance:

1. Gather supplies.
 - (a) Filter (sold in stockroom).
 - (b) 1 gallon Nalco 460-PCCL104 (the pink stuff).
2. Drain system completely.
 - (a) Disconnect red water line and allow chiller to pump water out.
 - (b) Push nipple of male end with flat object to release check valve.
3. Check filters.
 - (a) If air filter dirty, wash with water (let dry after washing).
 - (b) If water filter is dirty, replace.
4. Reassemble.
 - (a) Fill with Nalco 460-PCCL104 (the pink stuff).
 - (b) Turn on, top-off with distilled water.

A.4 NesLab Merlin M33

We have one NesLab Merlin M33 Chiller, Serial Number 106227049.

This chiller serves four different lasers on the ultrafast system. The cycle goes chiller out → Millennia → Tsunami → Spitfire Ace → Spitfire Pro → chiller return.

Regular chiller maintenance

1. Gather supplies.
2. Drain system completely.
3. Clean chiller internals.
4. Disconnect tubing, plug return.
5. Remove old filter from chiller, discard, replace filter holder.
6. Drain system completely and flush with several gallons of distilled water (should require 2), until no suds are formed as new water goes through system.
7. Fill system with one gallon distilled water, circulate for at least 15 minutes, then drain completely.
8. Place new filter into chiller.
9. Reconstruct tubing to include lasers in cycle. Ensure that cycle direction is correct.
10. Fill system with one gallon Nalco 460-PCCL104 (the pink stuff).
11. Turn chiller on, and allow system to flow for a while to get air bubbles out.
12. Top off with distilled water until chiller and tubing is full.

A.5 Calibrating the 407A

Calibrating the 407A

You may sometimes notice that the zero position changes dramatically from sensitivity to sensitivity with the 407A. If this happens, iterate through the following until zero stays consistent:

- Use the fine adjust (knob on side) to zero the 407A on the highest sensitivity.
- Use the front adjust (flathead screwdriver needed) to zero on the lowest sensitivity.

A.6 Millennia

A.6.1 Startup

The Millennia has problems with ‘thermal runaway’ upon startup if a user naively sends the laser to 4.0 W in power mode (thermal runaway causes the infamous FAULT 146 system shut off: power adjust timeout error). The following procedure wakes the Millennia up gently enough to prevent thermal runaway. It assumes that the Millennia starts completely shutdown and cold.

1. Check desiccant inside Millennia, replace if pink.
2. Ensure that Millennia shutter is closed such that no light is going downstream.
3. Flip orange power switch on diode box.
4. Wait for system to warm up (~ 30 minutes).
 - (a) “System Warming Up” message should appear on control box.
 - (b) Warm-up is finished when percentage complete indicator reaches 100.
 - (c) The Millennia will start in SP Current mode.
5. Record diode hours (choose “Info”, scroll down).
6. Turn Millennia to 1.0 W in “power mode”, wait for equilibration.

→ Equilibration occurs when diode temperature and current are stable (within 0.1) over 5 minutes.

7. Switch Millennia to “current mode”.
8. Slowly ramp current until you have more than 4 W output (probably $\sim 70\%$).
9. Again, wait for diode temperature and current equilibration.
10. Switch Millennia to “power mode” at 4.0 W.
11. Ensure one last time that diode temperature and current are not changing over 5 minute timescale.
12. Measure and record actual Millennia output power, currents, temperatures.

A.6.2 Toggling service mode

On the control board inside the laser, DIP switch #4 toggles service mode. Service mode unlocks special SP modes. These allow us to record things like Diode hours. Service mode can be buggy, so it's best to leave the Millennia in normal Power mode during regular operation.

A.7 Spitfire Pro

Only tune up the Spitfire if you need to, and do not treat it casually—set aside an entire day. Merely opening the spitfire box exposes the optics to dust. Treating the Spitfire with the respect it deserves will only save time in the long run.

The Spitfire contains 70 charcoal packets. These should be replaced every ~ 6 months.

A.7.1 Startup

Spitfire Pro startup procedure, from cold.

1. Ensure that the software is closed on control laptop.
2. Turn on Empower power supply (orange switch).
3. Turn on Timing and Delay Generator “TDG” (orange switch).
4. Turn on temperature control box (orange switch).
5. Turn key on Empower power supply.
6. Turn key on TDG.
7. Start Spitfire software.
 - Often have many faults, most will clear immediately.
 - Empower LBO temp fault may take a while to clear—normally clears after 5 minutes.
8. If Empower has been off (totally, or just at zero amps), warm up at 20 Amps for 1 hour.
 - Use a *good* beam block to block entry into the Cavity during this time.

A.7.2 Common alignment

Preparation

1. Check desiccant in temperature control box (blue is good, pink is bad).
2. Connect the fast oscilloscope so that you can see the pulse train.
3. If Spitfire is off, go through startup procedure.
4. If it has not been done in awhile, measure and record an Empower power curve.
5. Place the 407A after the telescope between the large square mirrors on the empty mount.
 - This mount is dedicated for this purpose, and should be left in the Spitfire at all times.
6. Set Empower current such that the Empower is delivering ~ 20 W to the crystal (refer to Empower power curve).

Cavity and pump

The first goal in alignment will be to ensure that the cavity and pump are healthy without seeding (in ns-lasing mode). A well-aligned Spitfire will deliver 5 to 5.2 W to the 407A when pumped with 20 W. The idea of this procedure is to be minimally invasive, while demanding good performance from the laser. Do not move on to seeded operation until ns-lasing is healthy, but at the same time do not do more than you need to.

1. Switch to ns lasing mode.
 - (a) Block seed at output of stretcher.
 - Note that seed must still enter the stretcher with good enough alignment to defeat the bandwidth detector.
 - (b) Activate Pockels Cells 2 and 3—do not use Pockels Cell 1.
 - (c) Expect 20 to 30 ns (2 to 3 round trips) additional build-up time in ns operation.
 - Means you must change timing of Pockels Cell 3 when working in ns lasing mode.
 - Operate the laser at its ideal switch out time when aligning ns lasing.
 - Refer to the records to see if your switch-out time is unusual for recent performance.
2. Clean all green optics in order of light hitting them.
 - You must get new spectrophotometric grade methanol for this cleaning.
3. Loop through the following until ns lasing delivers more than 5 W to the 407A.
 - (a) Check to ensure no optics are damaged, have dust on them etc.
 - (b) Clean optics. Check power after each cleaning—at the very least you don't want to lose power. When cleaning, wait at least a second before letting the light hit the optic after cleaning.
 - (c) Align pump.
 - i. Decrease the pump power to ~ 17 A. This should decrease the ns lasing output to around 2 Watts, so you should be able to increase the sensitivity on the 407A if you wish.
 - ii. Find the ideal switch out (channel 3) time for this lower pump power. It will be still later than the aforementioned ns lasing switch out time. Back off the ideal ns switch out time by ~ 2 round trips for optimization.
 - iii. Typically the two mirrors on either side of the cavity (immediate to the lenses) are the only pump mirrors touched during alignment. The first pump mirror may be touched in special cases but not for regular touch-up.
 - iv. Block the back reflection with the mirror mount when aligning the second pump mirror.
 - v. Align the back reflection mirror (without beam block).
 - vi. Iterate through the mirrors until you are satisfied that pump pointing is ideal.
 - (d) Align cavity.
 - Never touch anything except the two outermost end mirrors.
 - Note that the alignment tool is poor to the cavity mode intentionally.

Seed

Now that the cavity is good, you must couple the seed into it. If you have just aligned the oscillator you should wait at least 10 hours before attempting to align the seed.

Expect 100 to 150 mW less in seeded mode.

1. Remove stretcher cavity flange, being careful not to damage wires.
2. Align the two apertures either side of the Faraday isolator using mirrors external to Spitfire.
3. Align to two “A” mask positions in stretcher.
4. Align to three “F” mask positions on the way to cavity (this mirror system is under-constrained; consider using the final mirror of the “A” mask alignment as a tweaking mirror).
5. Let seed into cavity by turning on channel 1.
6. Optimize (mirror at D18 and periscope) to putout power before compressor—often useful to go one or two round trips less (~ 210 ns on channel 3) when aligning seed to power (output depends on seed more sensitively and you want to optimize for largest buildup reduction time).
7. Can adjust quarter waveplate to minimize post-pulsing if necessary but be very careful to not send the output back into the stretcher.
8. Ensure pulse is let out of the cavity at the optimal time (check oscilloscope and power).
9. Remove 407A.
10. Reconstruct flanges on stretcher cavity.

Output

Now you must ensure that the cavity output is properly routed through the telescope and compressor before leaving the Spitfire.

Note that only $\sim 80\%$ of the cavity output power transmits through the compressor due to absorbance in the grating and other losses.

1. Align to compressor positions (two “H”) using backwards alignment tool. You will need to unplug the compressor stage cable—remember to plug back in.
2. If the cavity or pump was touched, consider aligning the compressor.

Cleanup

1. Make sure you have taken all of your tools out of the laser.
2. Make sure you remembered to plug the compressor stage back in.
3. Wait at least 2 hours before moving on to downstream alignment—best to wait overnight if you can. It’s OK to make a first pass at rough alignment, but compression and pointing will probably change slightly as the laser equilibrates.

It’s normal for the Spitfire output power change by 50 to 100 mW in the first days after alignment. After that initial change the Spitfire tends to be stable for weeks. During experiments it’s a good idea to measure Spitfire output on a daily basis.

A.7.3 Stretcher alignment

This is our current best strategy for stretcher alignment. Use caution and keep your brain in gear when working on this. Since we have not messed with the stretcher frequently this guide cannot be trusted blindly.

Preparation

1. Ensure that the Spitfire is off (no pump present in cavity, Pockels cells powered down).
2. Force upstream oscillator to go CW, adjust output color to be at center of mode-locked bandwidth.
3. Use external mirrors to align through Faraday isolator.
4. Use first two mirrors after isolator to alignment tool prior to stretcher grating.

Adjustment

1. Adjust the stretcher grating until the four dots are overlapped.
2. Use pickoff mirror after stretcher to get the beam to alignment tool.

A.7.4 Compressor alignment

1. Use the first telescope mirror to align to the alignment tool before the first square mirror preceding the compressor.
2. Use the second telescope mirror to align to the alignment tool when place between the square mirrors preceding the compressor.

A.8 TOPAS-C

A.8.1 Common alignment

This section discusses common alignment operations that will probably need adjustment on a monthly basis to ensure ideal OPA behavior. All fs table users should feel comfortable performing these tuneups.

Tips and tricks:

- As a rule of thumb, if you don't need light through the poweramp the light should be blocked to avoid hot spots and damage. The light should be blocked before M8.
- Use a fluorescent card to visually get a better idea of the centering of the 800 nm beam through the apertures.
- Align OPA1 before OPA2. For the sake of consistency, we have agreed that compression should be adjusted for OPA1 best performance. OPA2 will have to "live with" the compression that is best suited for OPA1.

Preparation

1. Ensure that the Spitfire is working (between 3.8 and 4.0 W).
2. Ensure that the pump is not clipping on any mirrors between Spitfire and OPAs.
3. Inspect mode structure of pump for hot spots or diffraction. Clean any dust off of mirrors between Spitfire and OPAs.
4. Open OPA lid.
5. Set OPA to 1300 nm, ensure motors are homed.

Preamp

1. Block pump into poweramp upstream of M8 using block of metal.
2. Open OPA shutter.
3. Ensure that the WL plate is not drilled—look for "sparking". This takes an experienced eye—ask if you have a hard time deciding. If drilled, rotate the WL plate.
4. Ensure that input poynting and compression are good for the preamp by iterating through the following adjustments. Stop iteration once all metrics are good without further adjustment.
 - (a) Iteratively align through A0 and A2. The orange-colored white light should go through A2.
 - (b) Ensure that your poynting changes have not introduced clipping on external mirrors.
 - (c) Ensure that white light is good.
 - If you are aligning OPA1, adjust compression such that WLG is maximized. Do not adjust compression to OPA2 WL.
 - Adjust A_{WL} until WL is symmetric and stable.
 - (d) Ensure that no OPG is present in C1.
 - (e) Manually adjust D1 to maximize seed intensity.

5. Align seed down entire row of holes in-which L6, L7, DM2, NC2, and DM3 lie. Alignment is accomplished using M5 and M6.

→ Consider blocking pump into C1 (passing only WL) to “toggle” the seed—this helps distinguish between orange WL and slightly redder seed.

Poweramp

1. Unblock pump into poweramp.
2. Setup 407A power meter outside of OPA.
3. Adjust manual D2 until over 600 mW is achieved.
4. Put “Caution fs OPA free to coldwave” sign on laser-lab doors.
5. Remove all optics downstream of Mixer 3: filter periscopes (wavelength selectors), periscope, beam splitter, and beam dump.
6. Iterate the following until optimal power and collinearity are simultaneously reached:
 - (a) Adjust collinearity of three beams using DM2
 - All beams should be overlapped far away.
 - Use surveyor’s telescope to observe beams.
 - (b) Maximize 407A-measured power using manual D2.
 - (c) Maximize 407A-measured power using M10.
7. Reassemble optics downstream of Mixer 3.
8. Close OPA lid.
9. Allow a minute for equilibration.
10. Measure and record power—should be over 600 mW.

A.8.2 Full alignment

The following discussion endeavors to be as complete as possible. The goal is to have a procedure that produces a well-aligned OPA regardless of initial conditions. Experienced OPA users may find only pieces of this guide necessary to solve their particular problem.

Preparation

1. Ensure that the spitfire is working well (between 3.8 and 4.0 W).
2. Ensure that the pump is not clipping on any mirrors between Spitfire and OPA.
3. Inspect mode structure of pump for hot spots or diffraction. Clean any dust off of mirrors between Spitfire and OPAs.
4. Open OPA lid.
5. Set OPA to 1300 nm, ensure motors are homed.
6. Block pump into poweramp upstream of M8 using block of metal.
7. Block seed between compensating crystal and M5.
8. Remove all side walls from OPA.

Input poynting

Input poynting is adjusted to ensure good alignment through L1 and L2 into D1.

1. Remove A1/L3, VF, A_{WL} .
2. Place D2 at nominal position (45 degrees).
3. Using external mirrors, ensure that beam propagates through the alignment tool at the holes just after L2 and just before M1 (in D1).
4. Ensure that you have not introduced clipping external to the OPA.
5. Ensure that the L1/L2 telescope is outputting a collimated, undistorted beam.

D1 alignment

1. If you haven't already, remove A1/L3, VF.
2. Ensure that you are blocking light between compensating crystal and M5.
3. Remove WLG plate, L4, TD, DM1, knife edge.
4. Set C1 to surface normal (should be 0 degrees if affix is set correctly in software.)
5. Using M1 and M2, ensure that beam propagates through the alignment tool at the holes just after M2 and just before M5.

White light

1. Block preamp pump after M3 during this procedure.
2. Remove WL plate if it is present.
3. Replace/adjust AL/L3, adjusting focus to be at white light plate desired position.
4. Replace VF and A_{WL} if they are absent.
5. Replace the WL plate—ensure that the plate is normal to input beam by visual inspection.
6. Optimize WLG using VF, A_{WL} , and plate position.
 - (a) Begin with A_{WL} closed.
 - (b) Open A_{WL} 10%.
 - (c) Adjust VF to just allow for onset of WLG.
 - (d) Adjust position of sapphire plate to maximize visible component of continuum.
 - (e) Adjust VF to attenuate WLG pump to lowest WLG threshold where central mode and first outer rings are visible.
 - (f) Adjust compression for WLG symmetry.
 - (g) Continue to open A_{WL} and adjust VS until A_{WL} is as open as possible and VF is as dark as possible (while maintaining stable WLG).
7. Replace / adjust L4.
 - (a) Remove DM1 if it is present.
 - (b) Adjust L4 so that the visible component of the WL continuum is focused on A2. This will ensure that the NIR component focuses at NC1.
 - (c) Ensure that the WL remains centered on the alignment tool / A2.

→ L4 may be rotated to adjust height.

- (d) Replace DM1.

Preamp pump

1. Remove L5, M4.
2. Adjust M3 to alignment tool holes near edge of OPA to M5 mount. Note that M3 may be rotated to change height.
3. Ensure that beam is passing through alignment tool near M3. If not, consider translating M3 or moving BS2. Before making these adjustments ensure that the beam is true into BS2 (see input poynting section above).
4. Replace L5. Ensure that beam through L5 is on-axis with alignment tool in far field.
5. Replace M4. Point pump so that it intersects with the first red ring in the WL at DM1.
6. Adjust DM1 to spatially overlap pump with WL in NC1.
7. Adjust L5 so pump focus in in C1. If OPG in C1 is seen, back-off L5 by moving towards M3 until OPG disappears.

Seed

1. Ensure that both WL and pump are entering C1 properly.
 2. Remove M5 if present.
 3. Manually adjust D1 to optimize seed generation.
 4. Make fine adjustments to M4 and DM1 to ensure that the seed travels along alignment tools all the way to the OPA wall. If large adjustments need to be made something upstream must be wrong.
 5. Replace M5.
 6. Replace knife edge, if absent.
- Ensure you are not clipping the red seed profile.
7. Remove L6 and L7 if present.
 8. Use M5 and M6 to align the seed to the alignment guide through C2 and out of the OPA.
 9. Replace L6 and L7.

Poweramp

1. Ensure that pump is not clipping on BS1, M7.
 2. Center pump on M8 using M7.
 3. Remove L8 if present.
 4. Using M8, M9, alignment tool ensure that pump travels along holes from M9 to M10.
 5. Replace L8.
- Back of L8 Mount to front of M10 should be ~ 15 cm.
6. Without clipping, place M11 and point M10 to minimize off-axis angle at M10 (this requires us to put the pump 1/4 inch right of center as viewed while facing M11).

7. Center pump on M12 using M11.
8. Center pump on DM2 using M12.
9. Overlap with seed in NC2 using DM2.
10. Make small adjustments to M10, DM2 to perfect collinearity and overlap.
11. Adjust manual D2 until over 600 mW is achieved.
12. Put "Caution fs OPA free to coldwave" sign on laser-lab doors.
13. Remove all optics downstream of Mixer 3: filter periscopes (wavelength selectors), periscope, beam splitter, and beam dump.
14. Iterate the following until optimal power and collinearity are simultaneously reached:
 - (a) Adjust collinearity of three beams using DM2
 - All beams should be overlapped far away.
 - Use surveyor's telescope to observe beams.
 - (b) Maximize 407A-measured power using manual D2.
 - (c) Maximize 407A-measured power using M10.
15. Reassemble optics downstream of Mixer 3.
16. Close OPA lid.
17. Allow a minute for equilibration.
18. Measure and record power—should be over 600 mW.

A.9 MicroHR Monochromator

Visible Grating.

Align the HeNe as perpendicular as possible to the monochromator entrance slit.

Move the grating angle until the HeNe falls on the exit slit.

Shine a flashlight through the entrance slit and observe the colour on the exit slit: if white, then you are at 0-order (0 nm), if red, then you are at 1st order (632.8 nm).

Go to 0-order, narrow the slits, and slowly adjust the angle until the HeNe is going through the exit slit.

Go to Jovin Yvon/utilities and find the motor configuration program.

In the Gratings tab, select the 1st grating (1200 line density) and hit Calibrate.

In theoretical wavelength, enter 0 nm.

In experimental wavelength, enter the wavelength you observe from the control program.

Hit set.

Appendix B

Instrumental response function

The instrumental response function (IRF) is a classic concept in analytical science. Defining IRF becomes complex with instruments as complex as these, but it is still useful to attempt.

It is particularly useful to define bandwidth.

Time Domain

I will use four wave mixing to extract the time-domain pulse-width. I use a driven signal *e.g.* near infrared carbon tetrachloride response. I'll homodyne-detect the output. In my experiment I'm moving pulse 1 against pulses 2 and 3 (which are coincident).

The driven polarization, P , goes as the product of my input pulse *intensities*:

$$P(T) = I_1(t - T) \times I_2(t) \times I_3(t) \quad (\text{B.1})$$

In our experiment we are convolving I_1 with $I_2 \times I_3$. Each pulse has an *intensity-level* width, σ_1 , σ_2 , and σ_3 . $I_2 \times I_3$ is itself a Gaussian, and

$$\sigma_{I_2 I_3} = \dots \quad (\text{B.2})$$

$$= \sqrt{\frac{\sigma_2^2 \sigma_3^2}{\sigma_2^2 + \sigma_3^2}}. \quad (\text{B.3})$$

The width of the polarization (across T) is therefore

$$\sigma_P = \sqrt{\sigma_1^2 + \sigma_2^2 l_3} \quad (\text{B.4})$$

$$= \dots \quad (\text{B.5})$$

$$= \sqrt{\frac{\sigma_1^2 + \sigma_2^2 \sigma_3^2}{\sigma_1^2 + \sigma_2^2}}. \quad (\text{B.6})$$

I assume that all of the pulses have the same width. l_1 , l_2 , and l_3 are identical Gaussian functions with FWHM σ . In this case, Equation B.6 simplifies to

$$\sigma_P = \sqrt{\frac{\sigma^2 + \sigma^2 \sigma^2}{\sigma^2 + \sigma^2}} \quad (\text{B.7})$$

$$= \dots \quad (\text{B.8})$$

$$= \sigma \sqrt{\frac{3}{2}} \quad (\text{B.9})$$

Finally, since we measure σ_P and wish to extract σ :

$$\sigma = \sigma_P \sqrt{\frac{2}{3}} \quad (\text{B.10})$$

Again, all of these widths are on the *intensity* level.

Frequency Domain

We can directly measure σ (the width on the intensity-level) in the frequency domain using a spectrometer. A tune test contains this information.

Time-Bandwidth Product

For a Gaussian, approximately 0.441

Appendix C

Quantitative differential absorbance

Transient absorbance (TA) spectroscopy is a self-heterodyned technique. Through chopping you can measure nonlinearities quantitatively much easier than with homodyne detected (or explicitly heterodyned) experiments.

Typically one attempts to calculate the change in absorbance ΔA ...

$$\Delta A = A_{\text{on}} - A_{\text{off}} \quad (\text{C.1})$$

$$= -\log_{10} \left(\frac{I_{\text{T}} + I_{\text{R}} + I_{\Delta\text{T}} + I_{\Delta\text{R}}}{I_0} \right) + \log \left(\frac{I_{\text{T}} + I_{\text{R}}}{I_0} \right) \quad (\text{C.2})$$

$$= -(\log_{10}(I_{\text{T}} + I_{\text{R}} + I_{\Delta\text{T}} + I_{\Delta\text{R}}) - \log_{10}(I_0)) + (\log_{10}(I_{\text{T}} + I_{\text{R}}) - \log_{10}(I_0)) \quad (\text{C.3})$$

$$= -(\log_{10}(I_{\text{T}} + I_{\text{R}} + I_{\Delta\text{T}} + I_{\Delta\text{R}}) - \log_{10}(I_{\text{T}} + I_{\text{R}})) \quad (\text{C.4})$$

$$= -\log_{10} \left(\frac{I_{\text{T}} + I_{\text{R}} + I_{\Delta\text{T}} + I_{\Delta\text{R}}}{I_{\text{T}} + I_{\text{R}}} \right) \quad (\text{C.5})$$

Equation C.5 simplifies beautifully if reflectivity is negligible ...

Now I define a variable for each experimental measurable:

V_{T}		voltage recorded from transmitted beam, without pump
V_{R}		voltage recorded from reflected beam, without pump
$V_{\Delta\text{T}}$		change in voltage recorded from transmitted beam due to pump
$V_{\Delta\text{R}}$		change in voltage recorded from reflected beam due to pump

We will need to calibrate using a sample with a known transmissivity and reflectivity constant:

$V_{\text{T,ref}}$		voltage recorded from transmitted beam, without pump
$V_{\text{R,ref}}$		voltage recorded from reflected beam, without pump
\mathcal{T}_{ref}		transmissivity
\mathcal{R}_{ref}		reflectivity

Define two new proportionality constants...

$$C_{\text{T}} \equiv \frac{\mathcal{T}}{V_{\text{T}}} \quad (\text{C.6})$$

$$C_{\text{R}} \equiv \frac{\mathcal{R}}{V_{\text{R}}} \quad (\text{C.7})$$

These are explicitly calibrated (as a function of probe color) prior to the experiment using the calibration sample.

Given the eight experimental measurables (V_{T} , V_{R} , $V_{\Delta\text{T}}$, $V_{\Delta\text{R}}$, $V_{\text{T,ref}}$, $V_{\text{R,ref}}$, \mathcal{T}_{ref} , \mathcal{R}_{ref}) I can express all of the intensities in Equation C.5 in terms of I_0 .

$$C_T = \frac{\mathcal{T}_{\text{ref}}}{V_{T, \text{ref}}} \quad (\text{C.8})$$

$$C_R = \frac{\mathcal{R}_{\text{ref}}}{V_{R, \text{ref}}} \quad (\text{C.9})$$

$$I_T = I_0 C_T V_T \quad (\text{C.10})$$

$$I_R = I_0 C_R V_R \quad (\text{C.11})$$

$$I_{\Delta T} = I_0 C_T V_{\Delta T} \quad (\text{C.12})$$

$$I_{\Delta R} = I_0 C_R V_{\Delta R} \quad (\text{C.13})$$

Wonderfully, the I_0 cancels when plugged back into Equation C.5, leaving a final expression for ΔA that only depends on my eight measurables.

$$\Delta A = -\log_{10} \left(\frac{C_T(V_T + V_{\Delta T}) + C_R(V_R + V_{\Delta R})}{C_T V_T + C_R V_R} \right) \quad (\text{C.14})$$

Appendix D

Abandon the random phase approximation

Historically, we've assumed that OPAs have random phase on every shot. This makes interference processes quickly average to zero over many shots—we rarely take fewer than 100 shots per pixel. Here I demonstrate that this assumption is very poor, at least for the femtosecond OPAs.

In these experiments, I simply send OPA1 and OPA2 simultaneously into the array detector. The crucial detail is that the beams are exactly collinear—overlapped in a beamsplitter. I then scan delay between them while collecting single shot spectra using the array detector.

Figure D.1 shows the results of these experiments for OPA2 vs itself (“auto-interference”) and vs OPA1 (“cross interference”). At zero delay all colors arise simultaneously, so there are no modulations along the array axis (vertical). As I scan further from zero modulations set in as each wavelength within the pulse has a different period in delay space. It is crucial to remember that the monochromator acts like a stretcher, so we see interference between the two pulses even when separated by 400 fs.

The fringe pattern is expected in the case of auto-interference, but it is also quite stable in cross-interference. In the next experiment, I explore just how stable the cross-phase is.

Figure D.2 shows the same single-shot spectrum taken 1000 times at a fixed delay of 500 fs. The phase does drift, but it is certainly not quickly randomized. In fact, the period shifts by 180 degrees in roughly one minute—much much longer than any single pixel that we have taken.

This result forces us to reconsider our assumptions when identifying potential sources of artifact in our measurements.

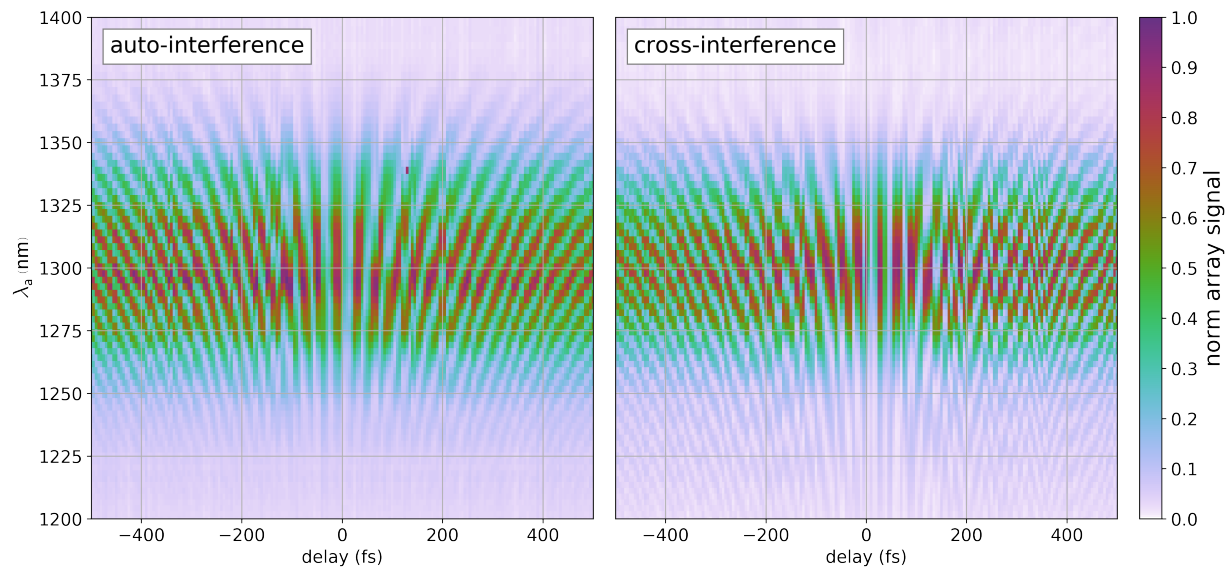


Figure D.1: Interference between OPA outputs as function of relative arrival time. In the left hand plot, OPA2 interferes with itself. In the right hand plot, OPA1 interferes with OPA2. Signal is intensity level.

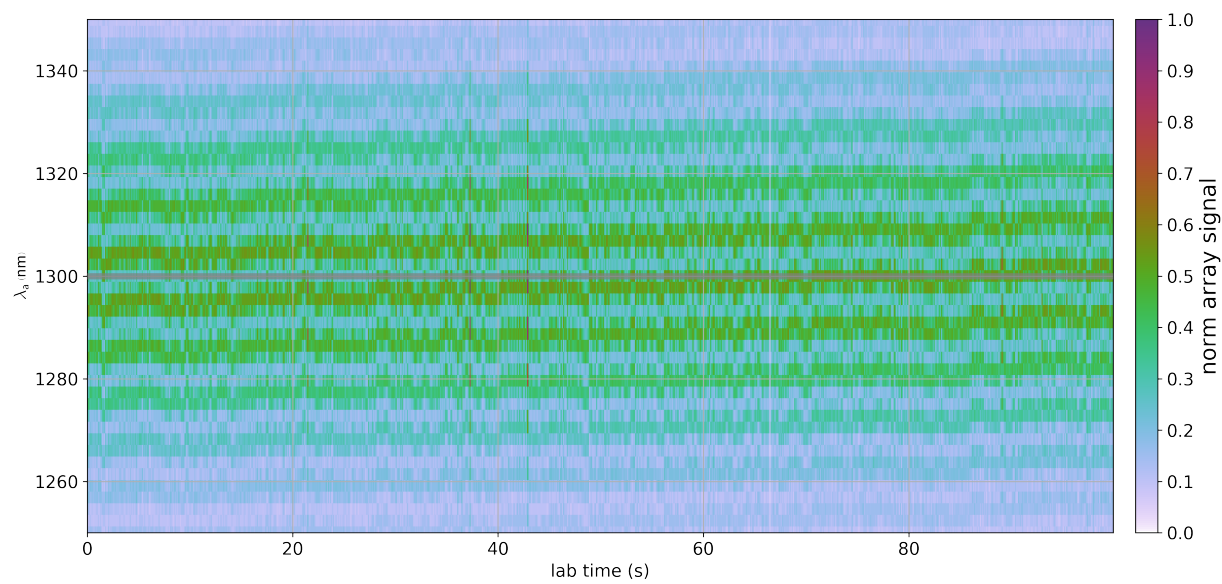


Figure D.2: Cross interference at fixed delay of 500 fs. 1000 single-shot acquisitions over a period of 100 seconds in lab time.

Appendix E

Colophon

This document was prepared using \LaTeX . The vast majority of figures were not remade for this document, and are simply reproduced from other locations. The source, including raw data, python scripts, image files and all other associated files can be found at <https://git.chem.wisc.edu/bthompson/dissertation>. If using git, simply clone:

```
git clone https://git.chem.wisc.edu/bthompson/dissertation.git (5.1)
```

The easiest way to reproduce this document is to use the bash script `build.sh`, located in the top directory of the repository.

The python scripts within the repository require WrightTools 3 or newer.

Please contact me directly for any reason at my permanent email address: blaise@untzag.com.

Bibliography

- [1] Daniel Kahneman. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, 2013.
- [2] Wei Zhao and John C. Wright. "Spectral Simplification in Vibrational Spectroscopy Using Doubly Vibrationally Enhanced Infrared Four Wave Mixing". In: *Journal of the American Chemical Society* 121.47 (Dec. 1999), pp. 10994–10998. DOI: [10.1021/ja9926414](https://doi.org/10.1021/ja9926414).
- [3] Paul M Donaldson, Rui Guo, Frederic Fournier, Elizabeth M Gardner, Ian R Gould, and David R Klug. "Decongestion of methylene spectra in biological and non-biological systems using picosecond 2DIR spectroscopy measuring electron-vibration–vibration coupling". In: *Chemical Physics* 350.1-3 (June 2008), pp. 201–211. DOI: [10.1016/j.chemphys.2008.02.050](https://doi.org/10.1016/j.chemphys.2008.02.050).
- [4] Erin S Boyle, Nathan A Neff-Mallon, and John C Wright. "Triply Resonant Sum Frequency Spectroscopy: Combining Advantages of Resonance Raman and 2D-IR". In: *The Journal of Physical Chemistry A* 117.47 (Nov. 2013), pp. 12401–12408. DOI: [10.1021/jp409377a](https://doi.org/10.1021/jp409377a).
- [5] Erin S Boyle, Nathan A. Neff-Mallon, Jonathan D Handali, and John C Wright. "Resonance IR: A Coherent Multidimensional Analogue of Resonance Raman". In: *The Journal of Physical Chemistry A* 118.17 (May 2014), pp. 3112–3119. DOI: [10.1021/jp5018554](https://doi.org/10.1021/jp5018554).
- [6] Andrei V. Pakoulev, Mark A. Rickard, Kathryn M. Kornau, Nathan A. Mathew, Lena A. Yurs, Stephen B. Block, and John C. Wright. "Mixed Frequency-/Time-Domain Coherent Multidimensional Spectroscopy: Research Tool or Potential Analytical Method?" In: *Accounts of Chemical Research* 42.9 (Sept. 2009), pp. 1310–1321. DOI: [10.1021/ar900032g](https://doi.org/10.1021/ar900032g).
- [7] John C. Wright. "Multiresonant Coherent Multidimensional Spectroscopy". In: *Annual Review of Physical Chemistry* 62.1 (May 2011), pp. 209–230. DOI: [10.1146/annurev-physchem-032210-103551](https://doi.org/10.1146/annurev-physchem-032210-103551).
- [8] Kisam Park and Minhaeng Cho. "Time- and frequency-resolved coherent two-dimensional IR spectroscopy: Its complementary relationship with the coherent two-dimensional Raman scattering spectroscopy". In: *The Journal of Chemical Physics* 109.24 (Dec. 1998), pp. 10559–10569. DOI: [10.1063/1.477756](https://doi.org/10.1063/1.477756).
- [9] Abdoulghafar Omari, Iwan Moreels, Francesco Masia, Wolfgang Langbein, Paola Borri, Dries Van Thourhout, Pascal Kockaert, and Zeger Hens. "Role of interband and photoinduced absorption in the nonlinear refraction and absorption of resonantly excited PbS quantum dots around 1550 nm". In: *Physical Review B* 85.11 (Mar. 2012). DOI: [10.1103/physrevb.85.115318](https://doi.org/10.1103/physrevb.85.115318).
- [10] M Sheik-Bahae, AA Said, T-H Wei, DJ Hagan, and EW Van Stryland. "Sensitive measurement of optical nonlinearities using a single beam". In: *IEEE journal of quantum electronics*. 26.4 (1990), pp. 760–769.

- [11] Frederic Fournier, Rui Guo, Elizabeth M. Gardner, Paul M. Donaldson, Christian Loeffeld, Ian R. Gould, Keith R. Willison, and David R. Klug. "Biological and Biomedical Applications of Two-Dimensional Vibrational Spectroscopy: Proteomics, Imaging, and Structural Analysis". In: *Accounts of Chemical Research* 42.9 (Sept. 2009), pp. 1322–1331. DOI: [10.1021/ar900074p](https://doi.org/10.1021/ar900074p).
- [12] Roger John Carlson. "Quantitative Aspects of High Resolution, Fully Resonant, Four-Wave Mixing Spectroscopy For the Analysis of Vibronic Mode Coupling in Molecules". PhD thesis. University of Wisconsin-Madison, 1988.
- [13] Dorine Keusters, Howe-Siang Tan, and Warren. "Role of Pulse Phase and Direction in Two-Dimensional Optical Spectroscopy". In: *The Journal of Physical Chemistry A* 103.49 (Dec. 1999), pp. 10369–10380. DOI: [10.1021/jp992325b](https://doi.org/10.1021/jp992325b).
- [14] W. Zhao and J. C. Wright. "Doubly Vibrationally Enhanced Four Wave Mixing: The Optical Analog to 2D NMR". In: *Physical Review Letters* 84 (Feb. 2000), pp. 1411–1414. DOI: [10.1103/PhysRevLett.84.1411](https://doi.org/10.1103/PhysRevLett.84.1411).
- [15] Andrei V. Pakoulev, Mark a Rickard, Kent a. Meyer, Kathryn Kornau, Nathan a. Mathew, David E. Thompson, and John C. Wright. "Mixed Frequency/Time Domain Optical Analogues of Heteronuclear Multidimensional NMR". In: *The Journal of Physical Chemistry A* 110.10 (Mar. 2006), pp. 3352–3355. DOI: [10.1021/jp057339y](https://doi.org/10.1021/jp057339y).
- [16] P. M. Rentzepis. "Ultrafast Processes". In: *Science* 169.3942 (July 1970), pp. 239–247. DOI: [10.1126/science.169.3942.239](https://doi.org/10.1126/science.169.3942.239).
- [17] Shaul Mukamel, Yoshitaka Tanimura, and Peter Hamm. "Coherent Multidimensional Optical Spectroscopy". In: *Accounts of Chemical Research* 42.9 (Sept. 2009), pp. 1207–1209. DOI: [10.1021/ar900227m](https://doi.org/10.1021/ar900227m).
- [18] Sarah M Gallagher, Allison W Albrecht, John D Hybl, Brett L Landin, Bhavani Rajaram, and David M Jonas. "Heterodyne detection of the complete electric field of femtosecond four-wave mixing signals". In: *Journal of the Optical Society of America B* 15.8 (Aug. 1998), p. 2338. DOI: [10.1364/JOSAB.15.002338](https://doi.org/10.1364/JOSAB.15.002338).
- [19] E. Harel, A. F. Fidler, and G. S. Engel. "Real-time mapping of electronic structure with single-shot two-dimensional electronic spectroscopy". In: *Proceedings of the National Academy of Sciences* 107.38 (Sept. 2010), pp. 16444–16447. DOI: [10.1073/pnas.1007579107](https://doi.org/10.1073/pnas.1007579107).
- [20] Austin P. Spencer, Hebin Li, Steven T. Cundiff, and David M. Jonas. "Pulse Propagation Effects in Optical 2D Fourier-Transform Spectroscopy: Theory". In: *The Journal of Physical Chemistry A* 119.17 (2015), pp. 3936–3960. DOI: [10.1021/acs.jpca.5b00001](https://doi.org/10.1021/acs.jpca.5b00001).
- [21] Nicholas M. Kearns, Randy D. Mehlenbacher, Andrew C. Jones, and Martin T. Zanni. "Broad-band 2D electronic spectrometer using white light and pulse shaping: noise and signal evaluation at 1 and 100 kHz". In: *Optics Express* 25.7 (Mar. 2017), p. 7869. DOI: [10.1364/oe.25.007869](https://doi.org/10.1364/oe.25.007869).
- [22] W. Xiong, J. E. Laaser, R. D. Mehlenbacher, and M. T. Zanni. "Adding a dimension to the infrared spectra of interfaces using heterodyne detected 2D sum-frequency generation (HD 2D SFG) spectroscopy". In: *Proceedings of the National Academy of Sciences* 108.52 (Dec. 2011), pp. 20902–20907. DOI: [10.1073/pnas.1115055108](https://doi.org/10.1073/pnas.1115055108).
- [23] Giulio Cerullo and Sandro De Silvestri. "Ultrafast optical parametric amplifiers". In: *Review of Scientific Instruments* 74.1 (Jan. 2003), pp. 1–18. DOI: [10.1063/1.1523642](https://doi.org/10.1063/1.1523642).

- [24] Roger J. Carlson and John C. Wright. "Absorption and Coherent Interference Effects in Multiply Resonant Four-Wave Mixing Spectroscopy". In: *Applied Spectroscopy* 43.7 (Sept. 1989), pp. 1195–1208. DOI: [10.1366/0003702894203408](https://doi.org/10.1366/0003702894203408). URL: <https://doi.org/10.1366/0003702894203408>.
- [25] Keith M. Murdoch, David E. Thompson, Kent A. Meyer, and John C. Wright. "Modeling Window Contributions to Four-Wave Mixing Spectra and Measurements of Third-Order Optical Susceptibilities". In: *Appl. Spectrosc.* 54.10 (Oct. 2000), pp. 1495–1505.
- [26] Christopher L. Smallwood, Travis M. Autry, and Steven T. Cundiff. "Analytical solutions to the finite-pulse Bloch model for multidimensional coherent spectroscopy". In: *Journal of the Optical Society of America B* 34.2 (Feb. 2017), p. 419. DOI: [10.1364/JOSAB.34.000419](https://doi.org/10.1364/JOSAB.34.000419).
- [27] Václav Perlík, Jürgen Hauer, and František Šanda. "Finite pulse effects in single and double quantum spectroscopies". In: *Journal of the Optical Society of America B* 34.2 (Feb. 2017), p. 430. DOI: [10.1364/JOSAB.34.000430](https://doi.org/10.1364/JOSAB.34.000430).
- [28] Kin Fai Mak, Changgu Lee, James Hone, Jie Shan, and Tony F. Heinz. "Atomically Thin MoS₂: A New Direct-Gap Semiconductor". In: *Physical Review Letters* 105.13 (Sept. 2010). DOI: [10.1103/physrevlett.105.136805](https://doi.org/10.1103/physrevlett.105.136805).
- [29] Xiaodong Xu, Wang Yao, Di Xiao, and Tony F. Heinz. "Spin and pseudospins in layered transition metal dichalcogenides". In: *Nature Physics* 10.5 (Apr. 2014), pp. 343–350. DOI: [10.1038/nphys2942](https://doi.org/10.1038/nphys2942).
- [30] Di Xiao, Gui-Bin Liu, Wanxiang Feng, Xiaodong Xu, and Wang Yao. "Coupled Spin and Valley Physics in Monolayers of MoS₂ and Other Group-VI Dichalcogenides". In: *Physical Review Letters* 108.19 (May 2012). DOI: [10.1103/physrevlett.108.196802](https://doi.org/10.1103/physrevlett.108.196802).
- [31] Christiaan Huygens. "Treatise on Light". In: *Science* 37.955 (Apr. 1913), pp. 610–611. DOI: [10.1126/science.37.955.610](https://doi.org/10.1126/science.37.955.610).
- [32] Giles Henderson, Robert C. Rittenhouse, John C. Wright, and Jon L. Holmes. "How a Photon is Created or Absorbed". In: *Journal of Chemical Education* 71.4 (Apr. 1994), p. 300. DOI: [10.1021/ed071p300.2](https://doi.org/10.1021/ed071p300.2).
- [33] P. A. M. Dirac. "A new notation for quantum mechanics". In: *Mathematical Proceedings of the Cambridge Philosophical Society* 35.03 (July 1939), p. 416. DOI: [10.1017/s0305004100021162](https://doi.org/10.1017/s0305004100021162).
- [34] J. F. WARD. "Calculation of Nonlinear Optical Susceptibilities Using Diagrammatic Perturbation Theory". In: *Reviews of Modern Physics* 37.1 (Jan. 1965), pp. 1–18. DOI: [10.1103/revmodphys.37.1](https://doi.org/10.1103/revmodphys.37.1).
- [35] T. K. Yee and T. K. Gustafson. "Diagrammatic analysis of the density operator for nonlinear optical calculations: Pulsed and cw responses". In: *Physical Review A* 18.4 (Oct. 1978), pp. 1597–1617. DOI: [10.1103/PhysRevA.18.1597](https://doi.org/10.1103/PhysRevA.18.1597).
- [36] Christoph A. Marx, Upendra Harbola, and Shaul Mukamel. "Nonlinear optical spectroscopy of single, few, and many molecules: Nonequilibrium Green's function QED approach". In: *Physical Review A* 77.2 (Feb. 2008). DOI: [10.1103/physreva.77.022110](https://doi.org/10.1103/physreva.77.022110).
- [37] Oleksiy Roslyak, Christoph A. Marx, and Shaul Mukamel. "Generalized Kramers-Heisenberg expressions for stimulated Raman scattering and two-photon absorption". In: *Physical Review A* 79.6 (June 2009). DOI: [10.1103/physreva.79.063827](https://doi.org/10.1103/physreva.79.063827).
- [38] Shaul Mukamel. *Principles of nonlinear optical spectroscopy*. Oxford series in optical and imaging sciences. Oxford University Press, 1995.

- [39] J. D. Biggs, J. A. Voll, and S. Mukamel. "Coherent nonlinear optical studies of elementary processes in biological complexes: diagrammatic techniques based on the wave function versus the density matrix". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 370.1972 (July 2012), pp. 3709–3727. DOI: [10.1098/rsta.2011.0219](https://doi.org/10.1098/rsta.2011.0219).
- [40] Duckhwan Lee and Andreas C. Albrecht. "A Unified View of Raman, Resonance Raman, and Fluorescence Spectroscopy (and their Analogues in Two-Photon Absorption)". In: *Advances in infrared and Raman Spectroscopy*. Ed. by R. J. H. Clark and R. E. Hester. 1st ed. London; New York, 1985. Chap. 4, pp. 179–213.
- [41] *NanoRam Handheld Raman Spectrometer for Material ID*. Accessed: 2018-04-15. URL: <http://bwtek.com/products/nanoram/>.
- [42] Ji-xin Cheng, Andreas Volkmer, Lewis D Book, and X Sunney Xie. "An Epi-Detected Coherent Anti-Stokes Raman Scattering (E-CARS) Microscope with High Spectral Resolution and High Sensitivity". In: *The Journal of Physical Chemistry B* 105.7 (Feb. 2001), pp. 1277–1280. DOI: [10.1021/jp003774a](https://doi.org/10.1021/jp003774a).
- [43] Juha Koivistoinen, Pasi Myllyperkiö, and Mika Pettersson. "Time-Resolved Coherent Anti-Stokes Raman Scattering of Graphene: Dephasing Dynamics of Optical Phonon". In: *The Journal of Physical Chemistry Letters* 8.17 (Aug. 2017), pp. 4108–4112. DOI: [10.1021/acs.jpcllett.7b01711](https://doi.org/10.1021/acs.jpcllett.7b01711).
- [44] David N. Card, Victor E. Church, and William W. Agresti. "An empirical study of software design practices". In: *IEEE Transactions on Software Engineering* SE-12.2 (Feb. 1986), pp. 264–271. DOI: [10.1109/tse.1986.6312942](https://doi.org/10.1109/tse.1986.6312942).
- [45] Carolyn B. Seaman and Victor R. Basili. "An empirical study of communication in code inspections". In: *Proceedings of the 19th international conference on Software engineering - ICSE '97*. ACM Press, 1997. DOI: [10.1145/253228.253248](https://doi.org/10.1145/253228.253248).
- [46] M.M. Muller and W.F. Tichy. "Case study: extreme programming in a university environment". In: *Proceedings of the 23rd International Conference on Software Engineering. ICSE 2001*. IEEE Comput. Soc. DOI: [10.1109/icse.2001.919128](https://doi.org/10.1109/icse.2001.919128).
- [47] Judith Segal. "Models of scientific software development". In: *SECSE 08, First International Workshop on Software Engineering in Computational Science and Engineering*. 2008.
- [48] Judith Segal. "When Software Engineers Met Research Scientists: A Case Study". In: *Empirical Software Engineering* 10.4 (Oct. 2005), pp. 517–536. DOI: [10.1007/s10664-005-3865-y](https://doi.org/10.1007/s10664-005-3865-y).
- [49] Jeffrey C. Carver, Richard P. Kendall, Susan E. Squires, and Douglass E. Post. "Software Development Environments for Scientific and Engineering Software: A Series of Case Studies". In: *29th International Conference on Software Engineering (ICSE'07)*. IEEE, May 2007. DOI: [10.1109/icse.2007.77](https://doi.org/10.1109/icse.2007.77).
- [50] Jo Erskine Hannay, Carolyn MacLeod, Janice Singer, Hans Petter Langtangen, Dietmar Pfahl, and Greg Wilson. "How do scientists develop and use scientific software?" In: *2009 ICSE Workshop on Software Engineering for Computational Science and Engineering*. Institute of Electrical and Electronics Engineers (IEEE), May 2009. DOI: [10.1109/secse.2009.5069155](https://doi.org/10.1109/secse.2009.5069155).
- [51] Prakash Prabhu, Yun Zhang, Soumyadeep Ghosh, David I. August, Jialu Huang, Stephen Beard, Hanjun Kim, Taewook Oh, Thomas B. Jablin, Nick P. Johnson, Matthew Zoufaly, Arun Raman, Feng Liu, and David Walker. "A survey of the practice of computational science". In: *State of the Practice Reports on - SC '11*. ACM Press, 2011. DOI: [10.1145/2063348.2063374](https://doi.org/10.1145/2063348.2063374).

- [52] Prakash Prabhu, Yun Zhang, Soumyadeep Ghosh, David I. August, Jialu Huang, Stephen Beard, Hanjun Kim, Taewook Oh, Thomas B. Jablin, Nick P. Johnson, Matthew Zoufaly, Arun Raman, Feng Liu, and David Walker. "A survey of the practice of computational science". In: *State of the Practice Reports on - SC '11*. ACM Press, 2011. DOI: [10.1145/2063348.2063374](https://doi.org/10.1145/2063348.2063374).
- [53] *A Vision and Strategy for Software for Science, Engineering and Education: Cyberinfrastructure Framework for the 21st Century*. Accessed: 2018-04-07. URL: <https://www.nsf.gov/pubs/2012/nsf12113/nsf12113.pdf>.
- [54] L. N. Joppa, G. McInerney, R. Harper, L. Salido, K. Takeda, K. O'Hara, D. Gavaghan, and S. Emmott. "Troubling Trends in Scientific Software Use". In: *Science* 340.6134 (May 2013), pp. 814–815. DOI: [10.1126/science.1231535](https://doi.org/10.1126/science.1231535).
- [55] R. Sanders and D. Kelly. "Dealing with Risk in Scientific Software Development". In: *IEEE Software* 25 (July 2008), pp. 21–28. DOI: [10.1109/MS.2008.84](https://doi.org/10.1109/MS.2008.84).
- [56] Greg Wilson, Jennifer Bryan, Karen Cranston, Justin Kitzes, Lex Nederbragt, and Tracy K. Teal. "Good enough practices in scientific computing". In: *PLOS Computational Biology* 13.6 (June 2017). Ed. by Francis Ouellette, e1005510. DOI: [10.1371/journal.pcbi.1005510](https://doi.org/10.1371/journal.pcbi.1005510).
- [57] G. Wilson. "Software Carpentry: Getting Scientists to Write Better Code by Making Them More Productive". In: *Computing in Science & Engineering* 8 (Nov. 2006), pp. 66–69. DOI: [10.1109/MCSE.2006.122](https://doi.org/10.1109/MCSE.2006.122).
- [58] *PyPI - the Python Package Index*. Accessed: 2018-04-04. URL: <https://pypi.org/>.
- [59] *File Exchange - MATLAB Central*. Accessed: 2018-04-04. URL: <https://www.mathworks.com/matlabcentral/fileexchange>.
- [60] *The Comprehensive R Archive Network*. Accessed: 2018-04-04. URL: <https://cran.r-project.org/>.
- [61] Susan M. Baxter, Steven W. Day, Jacquelyn S. Fetrow, and Stephanie J. Reisinger. "Scientific Software Development Is Not an Oxymoron". In: *PLoS Computational Biology* 2.9 (2006), e87. DOI: [10.1371/journal.pcbi.0020087](https://doi.org/10.1371/journal.pcbi.0020087).
- [62] Hadley Wickham. "Tidy data". In: *Journal of Statistical Software* 59.10 (2014), pp. 1–23.
- [63] Mike Folk, Gerd Heber, Quincey Koziol, Elena Pourmal, and Dana Robinson. "An overview of the HDF5 technology suite and its applications". In: *Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases - AD '11*. ACM Press, 2011. DOI: [10.1145/1966895.1966900](https://doi.org/10.1145/1966895.1966900).
- [64] *The world's leading software development platform · GitHub*. Accessed: 2018-04-04. URL: <https://github.com/>.
- [65] *The only single product for the complete DevOps lifecycle - GitLab*. Accessed: 2018-04-07. URL: <https://about.gitlab.com/>.
- [66] *GitLab (University of Wisconsin-Madison Chemistry)*. Accessed: 2018-04-04. URL: <https://git.chem.wisc.edu/>.
- [67] *Semantic Versioning*. Accessed: 2018-04-04. URL: <https://semver.org/>.
- [68] Nick Barnes. "Publish your computer code: it is good enough". In: *Nature* 467.7317 (Oct. 2010), pp. 753–753. DOI: [10.1038/467753a](https://doi.org/10.1038/467753a).

- [69] Greg Wilson, D. A. Aruliah, C. Titus Brown, Neil P. Chue Hong, Matt Davis, Richard T. Guy, Steven H. D. Haddock, Kathryn D. Huff, Ian M. Mitchell, Mark D. Plumbley, Ben Waugh, Ethan P. White, and Paul Wilson. “Best Practices for Scientific Computing”. In: *PLoS Biology* 12.1 (Jan. 2014). Ed. by Jonathan A. Eisen, e1001745. DOI: [10.1371/journal.pbio.1001745](https://doi.org/10.1371/journal.pbio.1001745). URL: <https://doi.org/10.1371/journal.pbio.1001745>.
- [70] *The Python Profilers (Python 3.5)*. Accessed: 2018-04-04. URL: <https://docs.python.org/3.5/library/profile.html>.
- [71] *SnakeViz*. Accessed: 2018-04-04. URL: <https://jiffyclub.github.io/snakeviz/>.
- [72] Deborah J. Armstrong. “The quarks of object-oriented development”. In: *Communications of the ACM* 49.2 (Feb. 2006), pp. 123–128. DOI: [10.1145/1113034.1113040](https://doi.org/10.1145/1113034.1113040).
- [73] Lloyd A. Treinish and Michael L. Gough. “A software package for the data-independent management of multidimensional data”. In: *Eos, Transactions American Geophysical Union* 68.28 (1987), p. 633. DOI: [10.1029/eo068i028p00633](https://doi.org/10.1029/eo068i028p00633).
- [74] R. Rew and G. Davis. “NetCDF: an interface for scientific data access”. In: *IEEE Computer Graphics and Applications* 10.4 (July 1990), pp. 76–82. DOI: [10.1109/38.56302](https://doi.org/10.1109/38.56302).
- [75] *FITS: Flexible image transport*. Accessed: 2018-03-27. URL: <https://fits.gsfc.nasa.gov/>.
- [76] D. C. Wells, E. W. Greisen, and R. H. Harten. “FITS - a Flexible Image Transport System”. In: *Astronomy and Astrophysics Supplement* 44 (June 1981), p. 363.
- [77] *h5py: HDF5 for Python*. Accessed: 2018-04-04. URL: <https://www.h5py.org/>.
- [78] Matthew T. Dougherty, Michael J. Folk, Erez Zadok, Herbert J. Bernstein, Frances C. Bernstein, Kevin W. Eliceiri, Werner Bengler, and Christoph Best. “Unifying biological image formats with HDF5”. In: *Communications of the ACM* 52.10 (Oct. 2009), p. 42. DOI: [10.1145/1562764.1562781](https://doi.org/10.1145/1562764.1562781).
- [79] Benjamin Watts and Jörg Raabe. “A NeXus/HDF5 based file format for STXM”. In: AIP, 2016. DOI: [10.1063/1.4937536](https://doi.org/10.1063/1.4937536). URL: <https://doi.org/10.1063/1.4937536>.
- [80] Paul Theodor Pyl, Julian Gehring, Bernd Fischer, and Wolfgang Huber. “h5vc: scalable nucleotide tallies with HDF5”. In: *Bioinformatics* 30.10 (Jan. 2014), pp. 1464–1466. DOI: [10.1093/bioinformatics/btu026](https://doi.org/10.1093/bioinformatics/btu026).
- [81] Luke Prince, Dwight Clayton, and Hector Santos-Villalobos. “An HDF5-based framework for the distribution and analysis of ultrasonic concrete data”. In: 2017. DOI: [10.1063/1.4974639](https://doi.org/10.1063/1.4974639).
- [82] Travis E. Oliphant. “Python for Scientific Computing”. In: *Computing in Science & Engineering* 9.3 (2007), pp. 10–20. DOI: [10.1109/MCSE.2007.58](https://doi.org/10.1109/MCSE.2007.58).
- [83] K. Jarrod Millman and Michael Aivazis. “Python for Scientists and Engineers”. In: *Computing in Science & Engineering* 13.2 (Mar. 2011), pp. 9–12. DOI: [10.1109/mcse.2011.36](https://doi.org/10.1109/mcse.2011.36).
- [84] Travis E Oliphant. *A guide to NumPy*. Trelgol Publishing, 2006.
- [85] Eric Jones, Travis Oliphant, Pearu Peterson, et al. *SciPy: Open source scientific tools for Python*. Accessed: 2018-04-04. URL: <http://www.scipy.org/>.
- [86] John D. Hunter. “Matplotlib: A 2D Graphics Environment”. In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: [10.1109/mcse.2007.55](https://doi.org/10.1109/mcse.2007.55).
- [87] *A Singularity Sensation*. Accessed: 2018-03-24. URL: <http://sm.stanford.edu/archive/stanmed/2012summer/article4.html>.

- [88] Nathan A. Neff-Mallon and John C. Wright. "Multidimensional Spectral Fingerprints of a New Family of Coherent Analytical Spectroscopies". In: *Analytical Chemistry* 89.24 (Nov. 2017), pp. 13182–13189. DOI: [10.1021/acs.analchem.7b02917](https://doi.org/10.1021/acs.analchem.7b02917).
- [89] Schuyler Kain. "Transition of Frequency-Domain Coherent Multidimensional Spectroscopic Methods to the Femtosecond Time Regime with Applications to Nanoscale Semiconductors". PhD thesis. University of Wisconsin-Madison, 2017.
- [90] Kent A Meyer, John C. Wright, and David E Thompson. "Frequency and Time-Resolved Triply Vibrationally Enhanced Four-Wave Mixing Spectroscopy Frequency and Time-Resolved Triply Vibrationally Enhanced Four-Wave Mixing Spectroscopy". In: *The Journal of Physical Chemistry A* 108.52 (Dec. 2004), pp. 11485–11493. DOI: [10.1021/jp046137j](https://doi.org/10.1021/jp046137j).
- [91] *h5py Groups documentation*. Accessed: 2018-03-25. URL: <http://docs.h5py.org/en/latest/high/group.html#groups>.
- [92] *Mayavi: 3D scientific data visualization and plotting in Python*. Accessed: 2018-03-25. URL: <http://docs.enthought.com/mayavi/mayavi/>.
- [93] Yang Liu and Jeffrey Heer. "Somewhere Over the Rainbow". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press, 2018. DOI: [10.1145/3173574.3174172](https://doi.org/10.1145/3173574.3174172).
- [94] D. A. Green. "A colour scheme for the display of astronomical intensity images". In: *Bulletin of the Astronomical Society of India* 39 (Dec. 2011), pp. 289–295. DOI: [10.1111/j.1365-2966.2004.08414.x](https://doi.org/10.1111/j.1365-2966.2004.08414.x).
- [95] *Read the Docs: Create, host, and browse documentation*. Accessed: 2018-03-27. URL: <https://readthedocs.org/>.
- [96] Kent A Meyer. "Frequency-Scanned Ultrafast Spectroscopic Techniques Applied to Infrared Four-Wave Mixing Spectroscopy". PhD thesis. University of Wisconsin-Madison, 2004.
- [97] W. Jänig. "Autonomic Nervous System". In: *Human Physiology*. Springer Berlin Heidelberg, 1989, pp. 333–370. DOI: [10.1007/978-3-642-73831-9_16](https://doi.org/10.1007/978-3-642-73831-9_16).
- [98] *QMutex Class*. Accessed: 2018-03-27. URL: <https://doc.qt.io/archives/qt-4.8/qmutex.html>.
- [99] *PyQtGraph: Scientific Graphics and GUI Library for Python*. Accessed: 2018-03-27. URL: <http://pyqtgraph.org/>.
- [100] *Miniature Steel Linear Stages*. Accessed: 2018-04-15. URL: <https://www.newport.com/f/miniature-mfa-motorized-linear-stages>.
- [101] *NI PCI-6251*. Accessed: 2018-04-15. URL: <http://sine.ni.com/nips/cds/view/p/lang/en/nid/14124>.
- [102] *numpy.ndindex*. Accessed: 2018-04-15. URL: <https://docs.scipy.org/doc/numpy/reference/generated/numpy.ndindex.html>.
- [103] Tobias Brixner, Tomáš Mančal, Igor V. Stiopkin, and Graham R. Fleming. "Phase-stabilized two-dimensional electronic spectroscopy". In: *The Journal of Chemical Physics* 121.9 (Sept. 2004), pp. 4221–4236. DOI: [10.1063/1.1776112](https://doi.org/10.1063/1.1776112).
- [104] Lena A Yurs, Stephen B. Block, Andrei V Pakoulev, Rachel S. Selinsky, Song Jin, and John Wright. "Multiresonant Coherent Multidimensional Electronic Spectroscopy of Colloidal PbSe Quantum Dots". In: *The Journal of Physical Chemistry C* 115.46 (Nov. 2011), pp. 22833–22844. DOI: [10.1021/jp207273x](https://doi.org/10.1021/jp207273x).

- [105] Ismael A. Heisler, Roberta Moca, Franco V. A. Camargo, and Stephen R. Meech. “Two-dimensional electronic spectroscopy based on conventional optics and fast dual chopper data acquisition”. In: *Review of Scientific Instruments* 85.6 (June 2014), p. 063103. DOI: [10.1063/1.4879822](https://doi.org/10.1063/1.4879822).
- [106] Koichi Furuta, Masanori Fuyuki, and Akihide Wada. “Cross-Term Selective, Two-Pulse Correlation Measurements by Phase-Shifted Parallel Modulation for Analysis of a Multi-Photon Process”. In: *Applied Spectroscopy* 66.12 (Dec. 2012), pp. 1475–1479. DOI: [10.1366/12-06657](https://doi.org/10.1366/12-06657).
- [107] Ramūnas Augulis and Donatas Zigmantas. “Two-dimensional electronic spectroscopy with double modulation lock-in detection: enhancement of sensitivity and noise resistance.” In: *Optics express* 19.14 (June 2011), pp. 13126–13133. DOI: [10.1364/OE.19.013126](https://doi.org/10.1364/OE.19.013126).
- [108] Ismael A. Heisler, Roberta Moca, Franco V A Camargo, and Stephen R. Meech. “Two-dimensional electronic spectroscopy based on conventional optics and fast dual chopper data acquisition”. In: *Review of Scientific Instruments* 85.6 (June 2014), p. 063103. DOI: [10.1063/1.4879822](https://doi.org/10.1063/1.4879822).
- [109] Ivan C. Spector, Courtney M. Olson, Christopher J. Huber, and Aaron M. Massari. “Simple fully reflective method of scatter reduction in 2D-IR spectroscopy”. In: *Optics Letters* 40.8 (Apr. 2015), pp. 1850–1852. DOI: [10.1364/OL.40.001850](https://doi.org/10.1364/OL.40.001850).
- [110] Brian L. McClain, Ilya J. Finkelstein, and M. D. Fayer. “Vibrational echo experiments on red blood cells: Comparison of the dynamics of cytoplasmic and aqueous hemoglobin”. In: *Chemical Physics Letters* 392.4-6 (July 2004), pp. 324–329. DOI: [10.1016/j.cplett.2004.05.080](https://doi.org/10.1016/j.cplett.2004.05.080).
- [111] Paul Murray Donaldson. “Two Dimensional Infrared Four Wave Mixing Spectroscopy of Simple Molecules, Peptides, and Proteins”. PhD thesis. Imperial College London, 2012.
- [112] Daniel D. Kohler, Blaise J. Thompson, and John C. Wright. “Frequency-domain coherent multidimensional spectroscopy when dephasing rivals pulsewidth: Disentangling material and instrument response”. In: *The Journal of Chemical Physics* 147.8 (Aug. 2017), p. 084202. DOI: [10.1063/1.4986069](https://doi.org/10.1063/1.4986069).
- [113] Shaul Mukamel. “Multidimensional Femtosecond Correlation Spectroscopies of Electronic and Vibrational Excitations”. In: *Annual Review of Physical Chemistry* 51.1 (Oct. 2000), pp. 691–729. DOI: [10.1146/annurev.physchem.51.1.691](https://doi.org/10.1146/annurev.physchem.51.1.691).
- [114] M V Lebedev, O V Misochko, T Dekorsy, and N Georgiev. “On the nature of “coherent artifact””. In: *Journal of Experimental and Theoretical Physics* 100.2 (Feb. 2005), pp. 272–282. DOI: [10.1134/1.1884668](https://doi.org/10.1134/1.1884668).
- [115] Z. Vardeny and J. Tauc. “Picosecond coherence coupling in the pump and probe technique”. In: *Optics Communications* 39.6 (Nov. 1981), pp. 396–400. DOI: [10.1016/0030-4018\(81\)90231-5](https://doi.org/10.1016/0030-4018(81)90231-5).
- [116] M Joffre, C. Benoit à la Guillaume, N Peyghambarian, M Lindberg, D Hulin, A Migus, S W Koch, and A Antonetti. “Coherent effects in pump-probe spectroscopy of excitons”. In: *Optics Letters* 13.4 (Apr. 1988), p. 276. DOI: [10.1364/OL.13.000276](https://doi.org/10.1364/OL.13.000276).
- [117] W. Pollard. “Analysis of Femtosecond Dynamic Absorption Spectra of Nonstationary States”. In: *Annual Review of Physical Chemistry* 43.1 (Jan. 1992), pp. 497–523. DOI: [10.1146/annurev.physchem.43.1.497](https://doi.org/10.1146/annurev.physchem.43.1.497).
- [118] S.A.J. Druet, J.-P.E. Taran, and Ch J Bordé. “Line shape and Doppler broadening in resonant CARS and related nonlinear processes through a diagrammatic approach”. In: *Journal de Physique* 40.9 (1979), pp. 819–840. DOI: [10.1051/jphys:01979004009081900](https://doi.org/10.1051/jphys:01979004009081900).
- [119] J-L. Oudar and Y. R. Shen. “Nonlinear spectroscopy by multiresonant four-wave mixing”. In: *Physical Review A* 22.3 (Sept. 1980), pp. 1141–1158. DOI: [10.1103/PhysRevA.22.1141](https://doi.org/10.1103/PhysRevA.22.1141).

- [120] John C. Wright, Peter C. Chen, James P. Hamilton, Arne Zilian, and Mitchell J. Labuda. "Theoretical Foundations for a New Family of Infrared Four-Wave Mixing Spectroscopies". In: *Applied Spectroscopy* 51.7 (July 1997), pp. 949–958. DOI: [10.1366/0003702971941601](https://doi.org/10.1366/0003702971941601).
- [121] John C. Wright, Roger J. Carlson, Gregory B. Hurst, John K. Steehler, Michael T. Riebe, Bradford B. Price, Dinh C. Nguyen, and Steven H. Lee. "Molecular, multiresonant coherent four-wave mixing spectroscopy". In: *International Reviews in Physical Chemistry* 10.4 (Oct. 1991), pp. 349–390. DOI: [10.1080/01442359109353262](https://doi.org/10.1080/01442359109353262).
- [122] Andrei V. Pakoulev, Mark A Rickard, Nathan A Mathew, Kathryn M. Kornau, and John C. Wright. "Spectral Quantum Beating in Mixed Frequency/Time-Domain Coherent Multidimensional Spectroscopy". In: *The Journal of Physical Chemistry A* 111.30 (Aug. 2007), pp. 6999–7005. DOI: [10.1021/jp071929+](https://doi.org/10.1021/jp071929+).
- [123] Daniel D. Kohler, Stephen B. Block, Schuyler Kain, Andrei V. Pakoulev, and John C. Wright. "Ultrafast Dynamics within the 1S Exciton Band of Colloidal PbSe Quantum Dots Using Multiresonant Coherent Multidimensional Spectroscopy". In: *The Journal of Physical Chemistry C* 118.9 (Mar. 2014), pp. 5020–5031. DOI: [10.1021/jp412058u](https://doi.org/10.1021/jp412058u).
- [124] Maxim F. Gelin, Dassia Egorova, and Wolfgang Domcke. "Efficient Calculation of Time- and Frequency-Resolved Four-Wave-Mixing Signals". In: *Accounts of Chemical Research* 42.9 (Sept. 2009), pp. 1290–1298. DOI: [10.1021/ar900045d](https://doi.org/10.1021/ar900045d).
- [125] Peter Hamm, Manho Lim, William F. DeGrado, and Robin M Hochstrasser. "Pump/probe self heterodyned 2D spectroscopy of vibrational transitions of a small globular peptide". In: *The Journal of Chemical Physics* 112.4 (Jan. 2000), pp. 1907–1916. DOI: [10.1063/1.480772](https://doi.org/10.1063/1.480772).
- [126] J R Salcedo, A E Siegman, D D Dlott, and M D Fayer. "Dynamics of Energy Transport in Molecular Crystals: The Picosecond Transient-Grating Method". In: *Physical Review Letters* 41.2 (July 1978), pp. 131–134. DOI: [10.1103/PhysRevLett.41.131](https://doi.org/10.1103/PhysRevLett.41.131).
- [127] John T. Fourkas, Rick Trebino, and M. D. Fayer. "The grating decomposition method: A new approach for understanding polarization-selective transient grating experiments. I. Theory". In: *The Journal of Chemical Physics* 97.1 (July 1992), pp. 69–77. DOI: [10.1063/1.463565](https://doi.org/10.1063/1.463565).
- [128] John T Fourkas, Rick Trebino, and M D Fayer. "The grating decomposition method: A new approach for understanding polarization-selective transient grating experiments. II. Applications". In: *The Journal of Chemical Physics* 97.1 (July 1992), pp. 78–85. DOI: [10.1063/1.463525](https://doi.org/10.1063/1.463525).
- [129] Gerald Auböck, Cristina Consani, Frank van Mourik, and Majed Chergui. "Ultrabroadband femtosecond two-dimensional ultraviolet transient absorption". In: *Optics Letters* 37.12 (June 2012), p. 2337. DOI: [10.1364/OL.37.002337](https://doi.org/10.1364/OL.37.002337).
- [130] H. J. Bakker, H.-K. Nienhuys, G. Gallot, N. Lascoux, G. M. Gale, J.-C. Leicknam, and S. Bratos. "Transient absorption of vibrationally excited water". In: *The Journal of Chemical Physics* 116.6 (Feb. 2002), pp. 2592–2598. DOI: [10.1063/1.1432687](https://doi.org/10.1063/1.1432687).
- [131] Wim P de Boeij, Maxim S Pshenichnikov, and Douwe A Wiersma. "On the relation between the echo-peak shift and Brownian-oscillator correlation function". In: *Chemical Physics Letters* 253.1-2 (Apr. 1996), pp. 53–60. DOI: [10.1016/0009-2614\(96\)00207-2](https://doi.org/10.1016/0009-2614(96)00207-2).
- [132] F.G. Patterson, H.W.H. Lee, William L. Wilson, and M.D. Fayer. "Intersystem crossing from singlet states of molecular dimers and monomers in mixed molecular crystals: picosecond stimulated photon echo experiments". In: *Chemical Physics* 84.1 (Feb. 1984), pp. 51–60. DOI: [10.1016/0301-0104\(84\)80005-1](https://doi.org/10.1016/0301-0104(84)80005-1).

- [133] A. Tokmakoff and M. D. Fayer. "Infrared Photon Echo Experiments: Exploring Vibrational Dynamics in Liquids and Glasses". In: *Accounts of Chemical Research* 28.11 (Nov. 1995), pp. 437–445. DOI: [10.1021/ar00059a001](https://doi.org/10.1021/ar00059a001).
- [134] Peter Hamm, Manho Lim, W. F. DeGrado, and Robin M Hochstrasser. "The two-dimensional IR nonlinear spectroscopy of a cyclic penta-peptide in relation to its three-dimensional structure". In: *Proceedings of the National Academy of Sciences* 96.5 (Mar. 1999), pp. 2036–2041. DOI: [10.1073/pnas.96.5.2036](https://doi.org/10.1073/pnas.96.5.2036).
- [135] M. C. Asplund, M. T. Zanni, and R. M. Hochstrasser. "Two-dimensional infrared spectroscopy of peptides by phase-controlled femtosecond vibrational photon echoes". In: *Proceedings of the National Academy of Sciences* 97.15 (July 2000), pp. 8219–8224. DOI: [10.1073/pnas.140227997](https://doi.org/10.1073/pnas.140227997).
- [136] Martin T Zanni, Matthew C Asplund, and Robin M Hochstrasser. "Two-dimensional heterodyned and stimulated infrared photon echoes of N-methylacetamide-D". In: *The Journal of Chemical Physics* 114.10 (2001), p. 4579. DOI: [10.1063/1.1346647](https://doi.org/10.1063/1.1346647).
- [137] John D. Hybl, Yannick Christophe, and David M. Jonas. "Peak shapes in femtosecond 2D correlation spectroscopy". In: *Chemical Physics* 266.2-3 (May 2001), pp. 295–309. DOI: [10.1016/S0301-0104\(01\)00233-6](https://doi.org/10.1016/S0301-0104(01)00233-6).
- [138] Michel F Emde, Andrius Baltuska, Andreas Kummrow, Maxim S Pshenichnikov, and Douwe A Wiersma. "Ultrafast Librational Dynamics of the Hydrated Electron". In: *Physical Review Letters* 80.21 (May 1998), pp. 4645–4648. DOI: [10.1103/PhysRevLett.80.4645](https://doi.org/10.1103/PhysRevLett.80.4645).
- [139] Wim P. de Boeij, Maxim S. Pshenichnikov, and Douwe A. Wiersma. "Phase-locked heterodyne-detected stimulated photon echo. A unique tool to study solute—solvent interactions". In: *Chemical Physics Letters* 238.1-3 (May 1995), pp. 1–8. DOI: [10.1016/0009-2614\(95\)00452-1](https://doi.org/10.1016/0009-2614(95)00452-1).
- [140] Minhaeng Cho, Norbert F Scherer, Graham R Fleming, and Shaul Mukamel. "Photon echoes and related four-wave-mixing spectroscopies using phase-locked pulses". In: *The Journal of Chemical Physics* 96.8 (Apr. 1992), pp. 5618–5629. DOI: [10.1063/1.462686](https://doi.org/10.1063/1.462686).
- [141] Sean A. Passino, Yutaka Nagasawa, Taiha Joo, and Graham R. Fleming. "Three-Pulse Echo Peak Shift Studies of Polar Solvation Dynamics". In: *The Journal of Physical Chemistry A* 101.4 (Jan. 1997), pp. 725–731. DOI: [10.1021/jp9621383](https://doi.org/10.1021/jp9621383).
- [142] Sangwoon Yoon, David W. McCamant, Philipp Kukura, Richard A Mathies, Donghui Zhang, and Soo-Y Lee. "Dependence of line shapes in femtosecond broadband stimulated Raman spectroscopy on pump-probe time delay". In: *The Journal of Chemical Physics* 122.2 (Jan. 2005), p. 024505. DOI: [10.1063/1.1828044](https://doi.org/10.1063/1.1828044).
- [143] David W. McCamant, Philipp Kukura, and Richard A Mathies. "Femtosecond Stimulated Raman Study of Excited-State Evolution in Bacteriorhodopsin". In: *The Journal of Physical Chemistry B* 109.20 (May 2005), pp. 10449–10457. DOI: [10.1021/jp050095x](https://doi.org/10.1021/jp050095x).
- [144] Wei Zhao and John C. Wright. "Measurement of(3)for Doubly Vibrationally Enhanced Four Wave Mixing Spectroscopy". In: *Physical Review Letters* 83.10 (Sept. 1999), pp. 1950–1953. DOI: [10.1103/physrevlett.83.1950](https://doi.org/10.1103/physrevlett.83.1950).
- [145] Wei Zhao, Keith M M Urdoch, Daniel M Besemann, Nicholas J. Condon, Kent a. Meyer, and John C W Right. "Nonlinear Two - Dimensional Vibrational Spectroscopy". In: *Applied Spectroscopy* 54.7 (2000).

- [146] Kent A Meyer and John C Wright. "Interference, Dephasing, and Coherent Control in Time-Resolved Frequency Domain Two-Dimensional Vibrational Spectra". In: *The Journal of Physical Chemistry A* 107.41 (Oct. 2003), pp. 8388–8395. DOI: [10.1021/jp035146+](https://doi.org/10.1021/jp035146+).
- [147] Paul M Donaldson, Rui Guo, Frederic Fournier, Elizabeth M Gardner, Laura M C Barter, Chris J Barnett, Ian R Gould, David R Klug, D Jason Palmer, and Keith R Willison. "Direct identification and decongestion of Fermi resonances by control of pulse time ordering in two-dimensional IR spectroscopy". In: *The Journal of Chemical Physics* 127.11 (Sept. 2007), p. 114513. DOI: [10.1063/1.2771176](https://doi.org/10.1063/1.2771176).
- [148] Frédéric Fournier, Elizabeth M Gardner, Darek a Kedra, Paul M Donaldson, Rui Guo, Sarah a Butcher, Ian R Gould, Keith R Willison, and David R Klug. "Protein identification and quantification by two-dimensional infrared spectroscopy: implications for an all-optical proteomic platform." In: *Proceedings of the National Academy of Sciences of the United States of America* 105.40 (Oct. 2008), pp. 15352–7. DOI: [10.1073/pnas.0805127105](https://doi.org/10.1073/pnas.0805127105).
- [149] Erin Selene Boyle, Andrei V. Pakoulev, and John C. Wright. "Fully Coherent Triple Sum Frequency Spectroscopy of a Benzene Fermi Resonance". In: *The Journal of Physical Chemistry A* 117.27 (July 2013), pp. 5578–5588. DOI: [10.1021/jp404713x](https://doi.org/10.1021/jp404713x).
- [150] Alexei Lagutchev, Selezion A. Hambir, and Dana D. Dlott. "Nonresonant Background Suppression in Broadband Vibrational Sum-Frequency Generation Spectroscopy". In: *The Journal of Physical Chemistry C* 111.37 (Sept. 2007), pp. 13645–13647. DOI: [10.1021/jp075391j](https://doi.org/10.1021/jp075391j).
- [151] Roger J Carlson and John C Wright. "Analysis of vibrational correlations and couplings in the lowest two singlet states of pentacene by high resolution, fully resonant, coherent four-wave mixing spectroscopy". In: *The Journal of Chemical Physics* 92.9 (May 1990), pp. 5186–5195. DOI: [10.1063/1.458553](https://doi.org/10.1063/1.458553).
- [152] Roger J Carlson, Dinh C Nguyen, and John C Wright. "Analysis of vibronic mode coupling in pentacene by fully resonant coherent four-wave mixing". In: *The Journal of Chemical Physics* 92.3 (Feb. 1990), pp. 1538–1546. DOI: [10.1063/1.458084](https://doi.org/10.1063/1.458084).
- [153] Roger J. Carlson and John C Wright. "Enhanced selectivity for spectrochemical measurement by mode selection in fully resonant nonlinear mixing". In: *Analytical Chemistry* 63.14 (July 1991), pp. 1449–1451. DOI: [10.1021/ac00014a019](https://doi.org/10.1021/ac00014a019).
- [154] J. K. Steehler and J. C. Wright. "Parametric and nonparametric four-wave mixing in pentacene: p-terphenyl". In: *The Journal of Chemical Physics* 83.7 (Oct. 1985), pp. 3200–3208. DOI: [10.1063/1.449177](https://doi.org/10.1063/1.449177).
- [155] Kijeong Kwac and Minhaeng Cho. "Two-Color Pump-Probe Spectroscopies of Two- and Three-Level Systems: 2-Dimensional Line Shapes and Solvation Dynamics". In: *The Journal of Physical Chemistry A* 107.31 (Aug. 2003), pp. 5903–5912. DOI: [10.1021/jp034727w](https://doi.org/10.1021/jp034727w).
- [156] John C. Wright. "Analytical chemistry, multidimensional spectral signatures, and the future of coherent multidimensional spectroscopy". In: *Chemical Physics Letters* 662 (Oct. 2016), pp. 1–13. DOI: [10.1016/j.cplett.2016.07.045](https://doi.org/10.1016/j.cplett.2016.07.045).
- [157] Kyle J. Czech, Blaise J. Thompson, Schuyler Kain, Qi Ding, Melinda J. Shearer, Robert J. Hamers, Song Jin, and John C. Wright. "Measurement of Ultrafast Excitonic Dynamics of Few-Layer MoS₂ Using State-Selective Coherent Multidimensional Spectroscopy". In: *ACS Nano* 9.12 (Dec. 2015), pp. 12146–12157. DOI: [10.1021/acsnano.5b05198](https://doi.org/10.1021/acsnano.5b05198).

- [158] John C. Wright, Nicholas J. Condon, Keith M. Murdoch, Daniel M. Besemann, and Kent A. Meyer. "Quantitative Modeling of Nonlinear Processes in Coherent Two-Dimensional Vibrational Spectroscopy †". In: *The Journal of Physical Chemistry A* 107.40 (Oct. 2003), pp. 8166–8176. DOI: [10.1021/jp034018i](https://doi.org/10.1021/jp034018i).
- [159] Paul M Donaldson, Keith R Willison, and David R Klug. "Generation of Simplified Protein Raman Spectra Using Three-Color Picosecond Coherent Anti-Stokes Raman Spectroscopy". In: *The Journal of Physical Chemistry B* 114.37 (Sept. 2010), pp. 12175–12181. DOI: [10.1021/jp1061607](https://doi.org/10.1021/jp1061607).
- [160] J. A. Armstrong, N. Bloembergen, J. Ducuing, and P. S. Pershan. "Interactions between Light Waves in a Nonlinear Dielectric". In: *Physical Review* 127.6 (Sept. 1962), pp. 1918–1939. DOI: [10.1103/PhysRev.127.1918](https://doi.org/10.1103/PhysRev.127.1918).
- [161] Igor V. Schweigert and Shaul Mukamel. "Simulating multidimensional optical wave-mixing signals with finite-pulse envelopes". In: *Physical Review A* 77.3 (Mar. 2008), p. 033802. DOI: [10.1103/PhysRevA.77.033802](https://doi.org/10.1103/PhysRevA.77.033802).
- [162] Daniel M. Besemann, Kent A. Meyer, and John C. Wright. "Spectroscopic Characteristics of Triply Vibrationally Enhanced Four-Wave Mixing Spectroscopy †". In: *The Journal of Physical Chemistry B* 108.29 (July 2004), pp. 10493–10504. DOI: [10.1021/jp0495971](https://doi.org/10.1021/jp0495971).
- [163] Roger J. Carlson and John C. Wright. "Line narrowing in multiresonant third order molecular spectroscopies". In: *Journal of Molecular Spectroscopy* 143.1 (Sept. 1990), pp. 1–17. DOI: [10.1016/0022-2852\(90\)90256-P](https://doi.org/10.1016/0022-2852(90)90256-P).
- [164] Michael T. Riebe and John C. Wright. "Spectral line-narrowing and saturation effects in fully resonant nondegenerate four wave mixing". In: *The Journal of Chemical Physics* 88.5 (Mar. 1988), pp. 2981–2994. DOI: [10.1063/1.453939](https://doi.org/10.1063/1.453939).
- [165] A M Weiner, S. De Silvestri, and E P Ippen. "Three-pulse scattering for femtosecond dephasing studies: theory and experiment". In: *Journal of the Optical Society of America B* 2.4 (Apr. 1985), p. 654. DOI: [10.1364/JOSAB.2.000654](https://doi.org/10.1364/JOSAB.2.000654).
- [166] Ritesh Agarwal, Bradley S. Prall, Abbas H. Rizvi, Mino Yang, and Graham R. Fleming. "Two-color three pulse photon echo peak shift spectroscopy". In: *The Journal of Chemical Physics* 116.14 (Apr. 2002), pp. 6243–6252. DOI: [10.1063/1.1459414](https://doi.org/10.1063/1.1459414).
- [167] Bernhard Dick and R.M. Hochstrasser. "Resonant non-linear spectroscopy in strong fields". In: *Chemical Physics* 75.2 (Mar. 1983), pp. 133–155. DOI: [10.1016/0301-0104\(83\)85015-0](https://doi.org/10.1016/0301-0104(83)85015-0).
- [168] Maxim F. Gelin, Dussia Egorova, and Wolfgang Domcke. "Efficient method for the calculation of time- and frequency-resolved four-wave mixing signals and its application to photon-echo spectroscopy". In: *The Journal of Chemical Physics* 123.16 (Oct. 2005), p. 164112. DOI: [10.1063/1.2062188](https://doi.org/10.1063/1.2062188).
- [169] Maxim F. Gelin, Dussia Egorova, and Wolfgang Domcke. "Efficient calculation of the polarization induced by N coherent laser pulses." In: *The Journal of chemical physics* 131.19 (Nov. 2009), p. 194103. DOI: [10.1063/1.3265213](https://doi.org/10.1063/1.3265213).
- [170] Alexei Lagutchev, Aaron Lozano, Prabuddha Mukherjee, Selezione a Hambir, and Dana D Dlott. "Compact broadband vibrational sum-frequency generation spectrometer with nonresonant suppression". In: *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy* 75.4 (Apr. 2010), pp. 1289–1296. DOI: [10.1016/j.saa.2009.12.066](https://doi.org/10.1016/j.saa.2009.12.066).

- [171] C.H. Brito Cruz, J.P. Gordon, P.C. Becker, R.L. Fork, and C.V. Shank. "Dynamics of spectral hole burning". In: *IEEE Journal of Quantum Electronics* 24.2 (Feb. 1988), pp. 261–269. DOI: [10.1109/3.122](https://doi.org/10.1109/3.122).
- [172] S L Palfrey and T F Heinz. "Coherent interactions in pump–probe absorption measurements: the effect of phase gratings". In: *Journal of the Optical Society of America B* 2.4 (Apr. 1985), p. 674. DOI: [10.1364/JOSAB.2.000674](https://doi.org/10.1364/JOSAB.2.000674).
- [173] G. R. Fleming, S A Passino, and Y. Nagasawa. "The interaction of solutes with their environments". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 356.1736 (Feb. 1998), pp. 389–404. DOI: [10.1098/rsta.1998.0172](https://doi.org/10.1098/rsta.1998.0172).
- [174] Wim P de Boeij, Maxim S Pshenichnikov, and Douwe A Wiersma. "ULTRAFast SOLVATION DYNAMICS EXPLORED BY FEMTOSECOND PHOTON ECHO SPECTROSCOPIES". In: *Annual Review of Physical Chemistry* 49.1 (Oct. 1998), pp. 99–123. DOI: [10.1146/annurev.physchem.49.1.99](https://doi.org/10.1146/annurev.physchem.49.1.99).
- [175] Mayrose R. Salvador, P. Sreekumari Nair, Minhaeng Cho, and Gregory D Scholes. "Interaction between excitons determines the non-linear response of nanocrystals". In: *Chemical Physics* 350.1-3 (June 2008), pp. 56–68. DOI: [10.1016/j.chemphys.2007.12.020](https://doi.org/10.1016/j.chemphys.2007.12.020).
- [176] Ko Okumura, Andrei Tokmakoff, and Yoshitaka Tanimura. "Two-dimensional line-shape analysis of photon-echo signal". In: *Chemical Physics Letters* 314.5-6 (Dec. 1999), pp. 488–495. DOI: [10.1016/S0009-2614\(99\)01173-2](https://doi.org/10.1016/S0009-2614(99)01173-2).
- [177] Thijs J Aartsma and Douwe A Wiersma. "Photon-echo relaxation in molecular mixed crystals". In: *Chemical Physics Letters* 42.3 (Sept. 1976), pp. 520–524. DOI: [10.1016/0009-2614\(76\)80667-7](https://doi.org/10.1016/0009-2614(76)80667-7).
- [178] M. Sheik-bahae, A A Said, and E. W. Van Stryland. "High-sensitivity, single-beam n_2 measurements". In: *Optics Letters* 14.17 (Sept. 1989), p. 955. DOI: [10.1364/OL.14.000955](https://doi.org/10.1364/OL.14.000955).
- [179] M. D. Levenson and N. Bloembergen. "Dispersion of the nonlinear optical susceptibilities of organic liquids and solutions". In: *The Journal of Chemical Physics* 60.4 (Feb. 1974), pp. 1323–1327. DOI: [10.1063/1.1681198](https://doi.org/10.1063/1.1681198).
- [180] Lena A Yurs, Stephen B Block, Andrei V. Pakoulev, Rachel S. Selinsky, Song Jin, and John Wright. "Spectral Isolation and Measurement of Surface-Trapped State Multidimensional Non-linear Susceptibility in Colloidal Quantum Dots". In: *The Journal of Physical Chemistry C* 116.9 (Mar. 2012), pp. 5546–5553. DOI: [10.1021/jp3014139](https://doi.org/10.1021/jp3014139).
- [181] P D Maker and R. W. Terhune. "Study of Optical Effects Due to an Induced Polarization Third Order in the Electric Field Strength". In: *Physical Review* 137.3A (Feb. 1965), A801–A818. DOI: [10.1103/PhysRev.137.A801](https://doi.org/10.1103/PhysRev.137.A801).
- [182] Michael K Yetzbacher, Nadia Belabas, Katherine a Kitney, and David M. Jonas. "Propagation, beam geometry, and detection distortions of peak shapes in two-dimensional Fourier transform spectra". In: *The Journal of Chemical Physics* 126.4 (Jan. 2007), p. 044511. DOI: [10.1063/1.2426337](https://doi.org/10.1063/1.2426337).
- [183] Byungmoon Cho, Michael K Yetzbacher, Katherine A. Kitney, Eric R Smith, and David M. Jonas. "Propagation and Beam Geometry Effects on Two-Dimensional Fourier Transform Spectra of Multilevel Systems †". In: *The Journal of Physical Chemistry A* 113.47 (Nov. 2009), pp. 13287–13299. DOI: [10.1021/jp904504z](https://doi.org/10.1021/jp904504z).

- [184] Dorine Keusters and Warren S. Warren. "Propagation effects on the peak profile in two-dimensional optical photon echo spectroscopy". In: *Chemical Physics Letters* 383.1-2 (Jan. 2004), pp. 21–24. DOI: [10.1016/j.cplett.2003.10.130](https://doi.org/10.1016/j.cplett.2003.10.130).
- [185] Victor I. Klimov. "Optical Nonlinearities and Ultrafast Carrier Dynamics in Semiconductor Nanocrystals". In: *The Journal of Physical Chemistry B* 104.26 (July 2000), pp. 6112–6123. DOI: [10.1021/jp9944132](https://doi.org/10.1021/jp9944132).
- [186] M Tuan Trinh, Arjan J Houtepen, Juleon M Schins, Jorge Piris, and Laurens D a Siebbeles. "Nature of the second optical transition in PbSe nanocrystals." In: *Nano letters* 8.7 (July 2008), pp. 2112–7. DOI: [10.1021/nl8010963](https://doi.org/10.1021/nl8010963).
- [187] M. Tuan Trinh, Matthew Y. Sfeir, Joshua J. Choi, Jonathan S. Owen, and Xiaoyang Zhu. "A Hot Electron–Hole Pair Breaks the Symmetry of a Semiconductor Quantum Dot". In: *Nano Letters* 13.12 (Dec. 2013), pp. 6091–6097. DOI: [10.1021/nl403368y](https://doi.org/10.1021/nl403368y).
- [188] Itay Gdor, Chunfan Yang, Diana Yanover, Hanan Sachs, Efrat Lifshitz, and Sanford Ruhman. "Novel Spectral Decay Dynamics of Hot Excitons in PbSe Nanocrystals: A Tunable Femtosecond Pump–Hyperspectral Probe Study". In: *The Journal of Physical Chemistry C* 117.49 (Dec. 2013), pp. 26342–26350. DOI: [10.1021/jp409530z](https://doi.org/10.1021/jp409530z).
- [189] Frank C M Spoor, Stanko Tomić, Arjan J Houtepen, and Laurens D A Siebbeles. "Broadband Cooling Spectra of Hot Electrons and Holes in PbSe Quantum Dots". In: *ACS Nano* 11.6 (June 2017), pp. 6286–6294. DOI: [10.1021/acsnano.7b02506](https://doi.org/10.1021/acsnano.7b02506).
- [190] Richard D Schaller, M. a. Petruska, and Victor I. Klimov. "Tunable Near-Infrared Optical Gain and Amplified Spontaneous Emission Using PbSe Nanocrystals". In: *The Journal of Physical Chemistry B* 107.50 (Dec. 2003), pp. 13765–13768. DOI: [10.1021/jp0311660](https://doi.org/10.1021/jp0311660). URL: <http://pubs.acs.org/doi/abs/10.1021/jp0311660>.
- [191] Brian L. Wehrenberg, Congjun Wang, and Philippe Guyot-Sionnest. "Interband and Intraband Optical Studies of PbSe Colloidal Quantum Dots". In: *The Journal of Physical Chemistry B* 106.41 (Oct. 2002), pp. 10634–10640. DOI: [10.1021/jp021187e](https://doi.org/10.1021/jp021187e).
- [192] Iwan Moreels, Karel Lambert, David De Muynck, Frank Vanhaecke, Dirk Poelman, José C. Martins, Guy Allan, and Zeger Hens. "Composition and Size-Dependent Extinction Coefficient of Colloidal PbSe Quantum Dots". In: *Chemistry of Materials* 19.25 (Dec. 2007), pp. 6101–6106. DOI: [10.1021/cm071410q](https://doi.org/10.1021/cm071410q).
- [193] Quanqin Dai, Yingnan Yiding Wang, Xinbi Li, Yu Zhang, Donald J. Pellegrino, Muxun Zhao, Bo Zou, JaeTae Seo, Yingnan Yiding Wang, and William W. Yu. "Size-dependent composition and molar extinction coefficient of PbSe semiconductor nanocrystals." In: *ACS nano* 3.6 (June 2009), pp. 1518–24. DOI: [10.1021/nn9001616](https://doi.org/10.1021/nn9001616).
- [194] R W Hellwarth, Adelbert Owyong, and Nicholas George. "Origin of the Nonlinear Refractive Index of Liquid CCl₄". In: *Physical Review A* 4.6 (Dec. 1971), pp. 2342–2347. DOI: [10.1103/PhysRevA.4.2342](https://doi.org/10.1103/PhysRevA.4.2342).
- [195] R.W. Hellwarth. "Third-order optical susceptibilities of liquids and solids". In: *Progress in Quantum Electronics* 5 (1977), pp. 1–68. DOI: [10.1016/0079-6727\(79\)90002-8](https://doi.org/10.1016/0079-6727(79)90002-8). URL: <http://linkinghub.elsevier.com/retrieve/pii/0079672779900028>.
- [196] P P Ho and R R Alfano. "Optical Kerr effect in liquids". In: *Physical Review A* 20.5 (Nov. 1979), pp. 2170–2187. DOI: [10.1103/PhysRevA.20.2170](https://doi.org/10.1103/PhysRevA.20.2170).

- [197] Shigeki Matsuo and Tahei Tahara. "Phase-stabilized optical heterodyne detection of impulsive stimulated Raman scattering". In: *Chemical Physics Letters* 264.6 (Jan. 1997), pp. 636–642. DOI: [10.1016/S0009-2614\(96\)01399-1](https://doi.org/10.1016/S0009-2614(96)01399-1).
- [198] Peter Voehringer and Norbert F Scherer. "Transient Grating Optical Heterodyne Detected Impulsive Stimulated Raman Scattering in Simple Liquids". In: *The Journal of Physical Chemistry* 99.9 (Mar. 1995), pp. 2684–2695. DOI: [10.1021/j100009a027](https://doi.org/10.1021/j100009a027).
- [199] T. Shimanouchi. "Molecular Vibrational Frequencies". In: *NIST Standard Reference Database Number 69*. Ed. by P.J. Linstrom and W.G. Mallard. DOI: [10.18434/T4D303](https://doi.org/10.18434/T4D303).
- [200] Ileana Rau, François Kajzar, Jérôme Luc, Bouchta Sahraoui, and Georges Boudebs. "Comparison of Z-scan and THG derived nonlinear index of refraction in selected organic solvents". In: *Journal of the Optical Society of America B* 25.10 (Oct. 2008), p. 1738. DOI: [10.1364/JOSAB.25.001738](https://doi.org/10.1364/JOSAB.25.001738).
- [201] Pieter Geiregat, Arjan J. Houtepen, Yolanda Justo, Ferdinand C. Grozema, Dries Van Thourhout, and Zeger Hens. "Coulomb Shifts upon Exciton Addition to Photoexcited PbS Colloidal Quantum Dots". In: *The Journal of Physical Chemistry C* 118.38 (Sept. 2014), pp. 22284–22290. DOI: [10.1021/jp505530k](https://doi.org/10.1021/jp505530k).
- [202] Bram De Geyter, Pieter Geiregat, Yunan Gao, Sybren Ten Cate, Arjan J. Houtepen, Juleon M. Schins, Dries Van Thourhout, Laurens D A Siebbeles, and Zeger Hens. "Broadband and picosecond intraband absorption in lead based colloidal quantum dots". In: *ACS Nano* 7 (2012), pp. 6067–6074. DOI: [10.1109/ICTON.2012.6254469](https://doi.org/10.1109/ICTON.2012.6254469).
- [203] B. F. Levine and C. G. Bethea. "Second and third order hyperpolarizabilities of organic molecules". In: *The Journal of Chemical Physics* 63.6 (Sept. 1975), pp. 2666–2682. DOI: [10.1063/1.431660](https://doi.org/10.1063/1.431660).
- [204] J. Cherlow, T. Yang, and R. Hellwarth. "Nonlinear optical susceptibilities of solvents". In: *IEEE Journal of Quantum Electronics* 12.10 (Oct. 1976), pp. 644–646. DOI: [10.1109/JQE.1976.1069058](https://doi.org/10.1109/JQE.1976.1069058).
- [205] M Thalhammer and A Penzkofer. "Measurement of third-order nonlinear susceptibilities by non-phase matched third-harmonic generation". In: *Applied Physics B Photophysics and Laser Chemistry* 32.3 (Nov. 1983), pp. 137–143. DOI: [10.1007/BF00688819](https://doi.org/10.1007/BF00688819).
- [206] J Etchepare, G Grillon, I Thomazeau, A Migus, and A Antonetti. "Third-order electronic susceptibilities of liquids measured by femtosecond kinetics of optical Kerr effect". In: *Journal of the Optical Society of America B* 2.4 (Apr. 1985), p. 649. DOI: [10.1364/JOSAB.2.000649](https://doi.org/10.1364/JOSAB.2.000649).
- [207] Etj Nibbering, M.A. Franco, BS Prade, G. Grillon, C Le Blanc, and A Mysyrowicz. "Measurement of the nonlinear refractive index of transparent materials by spectral analysis after nonlinear propagation". In: *Optics Communications* 119.5-6 (Sept. 1995), pp. 479–484. DOI: [10.1016/0030-4018\(95\)00394-N](https://doi.org/10.1016/0030-4018(95)00394-N).
- [208] A R P Rau. "Perspectives on the Fano Resonance Formula". In: *Physica Scripta* 69.1 (Jan. 2004), pp. C10–C13. DOI: [10.1238/Physica.Regular.069a00C10](https://doi.org/10.1238/Physica.Regular.069a00C10).
- [209] Gero Nootz, Lazaro A. Padilha, Larissa Levina, Vlad Sukhovatkin, Scott Webster, Lukasz Brzozowski, Edward H. Sargent, David J. Hagan, and Eric W. Van Stryland. "Size dependence of carrier dynamics and carrier multiplication in PbS quantum dots". In: *Physical Review B* 83.15 (Apr. 2011), p. 155302. DOI: [10.1103/PhysRevB.83.155302](https://doi.org/10.1103/PhysRevB.83.155302).

- [210] Emanuel Istrate, Sjoerd Hoogland, Vlad Sukhovatkin, Larissa Levina, Stefan Myrskog, Peter W E Smith, and Edward H Sargent. "Carrier Relaxation Dynamics in Lead Sulfide Colloidal Quantum Dots". In: *The Journal of Physical Chemistry B* 112.10 (Mar. 2008), pp. 2757–2760. DOI: [10.1021/jp076809g](https://doi.org/10.1021/jp076809g).
- [211] F. W. Deeg and M. D. Fayer. "Analysis of complex molecular dynamics in an organic liquid by polarization selective subpicosecond transient grating experiments". In: *The Journal of Chemical Physics* 91.4 (Aug. 1989), pp. 2269–2279. DOI: [10.1063/1.457034](https://doi.org/10.1063/1.457034).
- [212] Frank W. Wise. "Lead salt quantum dots: the limit of strong quantum confinement." In: *Accounts of chemical research* 33.11 (Nov. 2000), pp. 773–80.
- [213] Michael B. Price, Justinas Butkus, Tom C. Jellicoe, Aditya Sadhanala, Anouk Briane, Jonathan E. Halpert, Katharina Broch, Justin M. Hodgkiss, Richard H. Friend, and Felix Deschler. "Hot-carrier cooling and photoinduced refractive index changes in organic–inorganic lead halide perovskites". In: *Nature Communications* 6.May (2015), p. 8420. DOI: [10.1038/ncomms9420](https://doi.org/10.1038/ncomms9420).
- [214] Ye Yang, Yong Yan, Mengjin Yang, Sukgeun Choi, Kai Zhu, Joseph M Luther, and Matthew C Beard. "Low surface recombination velocity in". In: *Nature Communications* 6 (2015), pp. 1–6. DOI: [10.1038/ncomms8961](https://doi.org/10.1038/ncomms8961).
- [215] Ye Yang, Mengjin Yang, David T. Moore, Yong Yan, Elisa M. Miller, Kai Zhu, and Matthew C. Beard. "Top and bottom surfaces limit carrier lifetime in lead iodide perovskite films". In: *Nature Energy* 2.2 (Jan. 2017), p. 16207. DOI: [10.1038/nenergy.2016.207](https://doi.org/10.1038/nenergy.2016.207).
- [216] Inuk Kang and Frank W. Wise. "Electronic structure and optical properties of PbS and PbSe quantum dots". In: *Journal of the Optical Society of America B* 14.7 (July 1997), p. 1632. DOI: [10.1364/JOSAB.14.001632](https://doi.org/10.1364/JOSAB.14.001632).
- [217] Khadga J Karki, Fei Ma, Kaibo Zheng, Karel Zidek, Abdelrazek Mousa, Mohamed Abdellah, Maria E Messing, L Reine Wallenberg, Arkadi Yartsev, and Tõnu Pullerits. "Multiple exciton generation in nano-crystals revisited: consistent calculation of the yield based on pump-probe spectroscopy." In: *Scientific reports* 3 (Jan. 2013), p. 2287. DOI: [10.1038/srep02287](https://doi.org/10.1038/srep02287).
- [218] Itay Gdor, Dianna Yanover, Chunfan Yang, Arthur Shapiro, Efrat Lifshitz, and Sanford Ruhman. "Three Pulse Femtosecond Spectroscopy of PbSe Nano-Crystals ; 1S Bleach Nonlinearity and Sub Band-Edge Excited State Absorption Assignment Three Pulse Femtosecond Spectroscopy of PbSe nano-Crystals ; 1S Bleach Nonlinearity and Sub Band-edge Excited State A". In: *ACS Nano* (2015), pp. 2–5. DOI: [10.1021/nm5074868](https://doi.org/10.1021/nm5074868).
- [219] Yuri P. Svirko, Masayuki Shirane, Hidekatsu Suzuura, and Makoto Kuwata-Gonokami. "Four-Wave Mixing Theory at the Excitonic Resonance: Weakly Interacting Boson Model". In: *Journal of the Physical Society of Japan* 68.2 (Feb. 1999), pp. 674–682. DOI: [10.1143/JPSJ.68.674](https://doi.org/10.1143/JPSJ.68.674).
- [220] D C Hutchings, M Sheik-Bahae, D J Hagan, and E W Van Stryland. "Kramers-Kronig relations in nonlinear optics". In: *Optical and Quantum Electronics* 24.1 (1992), pp. 1–30. DOI: [10.1007/BF01234275](https://doi.org/10.1007/BF01234275).
- [221] Claudia Högemann, Marc Pauchard, and Eric Vauthey. "Picosecond transient grating spectroscopy: The nature of the diffracted spectrum". In: *Review of Scientific Instruments* 67.10 (Oct. 1996), pp. 3449–3453. DOI: [10.1063/1.1147157](https://doi.org/10.1063/1.1147157).
- [222] Juleon Schins, M. Trinh, Arjan Houtepen, and Laurens Siebbeles. "Probing formally forbidden optical transitions in PbSe nanocrystals by time- and energy-resolved transient absorption spectroscopy". In: *Physical Review B* 80.3 (July 2009), p. 035323. DOI: [10.1103/PhysRevB.80.035323](https://doi.org/10.1103/PhysRevB.80.035323).

- [223] JJ Peterson, Libai Huang, C. Delerue, and Guy Allan. "Uncovering forbidden optical transitions in PbSe nanocrystals". In: *nano Letters* (2007).
- [224] F Gesuele, M Y Sfeir, C B Murray, T F Heinz, and C W Wong. "Biexcitonic Effects in Excited Ultrafast Supercontinuum Spectroscopy of Carrier Multiplication and Biexcitonic Effects in Excited States of PbS Quantum Dots". In: *Nano letters* 12 (2012), p. 2658.
- [225] Ingvar T Kraatz, Matthew Booth, Benjamin J Whitaker, Michael G D Nix, and Kevin Critchley. "Sub-Bandgap Emission and Intraband Defect-Related Excited-State Dynamics in Colloidal CuInS₂/ZnS Quantum Dots Revealed by Femtosecond Pump - Dump - Probe Spectroscopy". In: (2014).
- [226] I. Moreels, Z. Hens, P. Kockaert, J. Loicq, and D. Van Thourhout. "Spectroscopy of the nonlinear refractive index of colloidal PbSe nanocrystals". In: *Applied Physics Letters* 89.19 (Nov. 2006), p. 193106. DOI: [10.1063/1.2385658](https://doi.org/10.1063/1.2385658).
- [227] J.J. Olivero and R.L. Longbothum. "Empirical fits to the voigt line width: A brief review". In: *J. Quant. Spectrosc. Radiat. Transfer* 17 (1977), pp. 233–236.
- [228] Qing Hua Wang, Kourosh Kalantar-Zadeh, Andras Kis, Jonathan N. Coleman, and Michael S. Strano. "Electronics and optoelectronics of two-dimensional transition metal dichalcogenides". In: *Nature Nanotechnology* 7.11 (Nov. 2012), pp. 699–712. DOI: [10.1038/nnano.2012.193](https://doi.org/10.1038/nnano.2012.193).
- [229] Alejandro Molina-Sánchez, Davide Sangalli, Kerstin Hummer, Andrea Marini, and Ludger Wirtz. "Effect of spin-orbit interaction on the optical spectra of single-layer, double-layer, and bulk MoS₂". In: *Physical Review B* 88.4 (July 2013). DOI: [10.1103/physrevb.88.045412](https://doi.org/10.1103/physrevb.88.045412).
- [230] H. Fang, C. Battaglia, C. Carraro, S. Nemsak, B. Ozdol, J. S. Kang, H. A. Bechtel, S. B. Desai, F. Kronast, A. A. Unal, G. Conti, C. Conlon, G. K. Palsson, M. C. Martin, A. M. Minor, C. S. Fadley, E. Yablonovitch, R. Maboudian, and A. Javey. "Strong interlayer coupling in van der Waals heterostructures built from single-layer chalcogenides". In: *Proceedings of the National Academy of Sciences* 111.17 (Apr. 2014), pp. 6198–6202. DOI: [10.1073/pnas.1405435111](https://doi.org/10.1073/pnas.1405435111).
- [231] Nardeep Kumar, Jiaqi He, Dawei He, Yongsheng Wang, and Hui Zhao. "Charge carrier dynamics in bulk MoS₂ crystal studied by transient absorption microscopy". In: *Journal of Applied Physics* 113.13 (2013), p. 133702. DOI: [10.1063/1.4799110](https://doi.org/10.1063/1.4799110).
- [232] Zhaogang Nie, Run Long, Linfeng Sun, Chung-Che Huang, Jun Zhang, Qihua Xiong, Daniel W. Hewak, Zexiang Shen, Oleg V. Prezhdo, and Zhi-Heng Loh. "Ultrafast Carrier Thermalization and Cooling Dynamics in Few-Layer MoS₂". In: *ACS Nano* 8.10 (Oct. 2014), pp. 10931–10940. DOI: [10.1021/nn504760x](https://doi.org/10.1021/nn504760x).
- [233] Dezheng Sun, Yi Rao, Georg A. Reider, Gugang Chen, Yumeng You, Louis Brézin, Avetik R. Harutyunyan, and Tony F. Heinz. "Observation of Rapid Exciton–Exciton Annihilation in Monolayer Molybdenum Disulfide". In: *Nano Letters* 14.10 (Oct. 2014), pp. 5625–5629. DOI: [10.1021/nl5021975](https://doi.org/10.1021/nl5021975).
- [234] Sangwan Sim, Jusang Park, Jeong-Gyu Song, Chihun In, Yun-Shik Lee, Hyungjun Kim, and Hyunyoung Choi. "Exciton dynamics in atomically thin MoS₂: Interexcitonic interaction and broadening kinetics". In: *Physical Review B* 88.7 (Aug. 2013). DOI: [10.1103/physrevb.88.075434](https://doi.org/10.1103/physrevb.88.075434).
- [235] Patanjali Kambhampati. "Unraveling the Structure and Dynamics of Excitons in Semiconductor Quantum Dots". In: *Accounts of Chemical Research* 44.1 (Jan. 2011), pp. 1–13. DOI: [10.1021/ar1000428](https://doi.org/10.1021/ar1000428).

- [236] Kin Fai Mak, Keliang He, Jie Shan, and Tony F. Heinz. "Control of valley polarization in monolayer MoS₂ by optical helicity". In: *Nature Nanotechnology* 7.8 (June 2012), pp. 494–498. DOI: [10.1038/nnano.2012.96](https://doi.org/10.1038/nnano.2012.96).
- [237] Steven T. Cundiff. "Coherent spectroscopy of semiconductors". In: *Optics Express* 16.7 (Mar. 2008), p. 4639. DOI: [10.1364/oe.16.004639](https://doi.org/10.1364/oe.16.004639).
- [238] Daniel B. Turner, Katherine W. Stone, Kenan Gundogdu, and Keith A. Nelson. "Three-dimensional electronic spectroscopy of excitons in GaAs quantum wells". In: *The Journal of Chemical Physics* 131.14 (Oct. 2009), p. 144510. DOI: [10.1063/1.3245964](https://doi.org/10.1063/1.3245964).
- [239] Graham B. Griffin, Sandrine Ithurria, Dmitriy S. Dolzhenkov, Alexander Linkin, Dmitri V. Talapin, and Gregory S. Engel. "Two-dimensional electronic spectroscopy of CdSe nanoparticles at very low pulse power". In: *The Journal of Chemical Physics* 138.1 (Jan. 2013), p. 014705. DOI: [10.1063/1.4772465](https://doi.org/10.1063/1.4772465).
- [240] Elad Harel, Sara M. Rupich, Richard D. Schaller, Dmitri V. Talapin, and Gregory S. Engel. "Measurement of electronic splitting in PbS quantum dots by two-dimensional nonlinear spectroscopy". In: *Physical Review B* 86.7 (Aug. 2012). DOI: [10.1103/physrevb.86.075412](https://doi.org/10.1103/physrevb.86.075412).
- [241] S. T. Cundiff, M. Koch, W. H. Knox, J. Shah, and W. Stolz. "Optical Coherence in Semiconductors: Strong Emission Mediated by Nondegenerate Interactions". In: *Physical Review Letters* 77.6 (Aug. 1996), pp. 1107–1110. DOI: [10.1103/physrevlett.77.1107](https://doi.org/10.1103/physrevlett.77.1107).
- [242] D. Birkedal, V. G. Lyssenko, J. M. Hvam, and K. El Sayed. "Continuum contribution to excitonic four-wave mixing due to interaction-induced nonlinearities". In: *Physical Review B* 54.20 (Nov. 1996), R14250–R14253. DOI: [10.1103/physrevb.54.r14250](https://doi.org/10.1103/physrevb.54.r14250).
- [243] M. U. Wehner, D. Steinbach, and M. Wegener. "Ultrafast coherent transients due to exciton-continuum scattering in bulk GaAs". In: *Physical Review B* 54.8 (Aug. 1996), R5211–R5214. DOI: [10.1103/physrevb.54.r5211](https://doi.org/10.1103/physrevb.54.r5211).
- [244] Masihur R. Laskar, Lu Ma, Santhakumar Kannappan, Pil Sung Park, Sriram Krishnamoorthy, Digbijoy N. Nath, Wu Lu, Yiyang Wu, and Siddharth Rajan. "Large area single crystal (0001) oriented MoS₂". In: *Applied Physics Letters* 102.25 (June 2013), p. 252108. DOI: [10.1063/1.4811410](https://doi.org/10.1063/1.4811410).
- [245] A. Castellanos-Gomez, M. Barkelid, A. M. Goossens, V. E. Calado, H. S. J. van der Zant, and G. A. Steele. "Laser-Thinning of MoS₂: On Demand Generation of a Single-Layer Semiconductor". In: *Nano Letters* 12.6 (June 2012), pp. 3187–3192. DOI: [10.1021/nl301164v](https://doi.org/10.1021/nl301164v).
- [246] Mariyappan Shanmugam, Chris A. Durcan, and Bin Yu. "Layered semiconductor molybdenum disulfide nanomembrane based Schottky-barrier solar cells". In: *Nanoscale* 4.23 (2012), p. 7399. DOI: [10.1039/c2nr32394j](https://doi.org/10.1039/c2nr32394j).
- [247] Andreas Volkmer, Ji-Xin Cheng, and X. Sunney Xie. "Vibrational Imaging with High Sensitivity via Epidetected Coherent Anti-Stokes Raman Scattering Microscopy". In: *Physical Review Letters* 87.2 (June 2001). DOI: [10.1103/physrevlett.87.023901](https://doi.org/10.1103/physrevlett.87.023901).
- [248] Fernando Perez and Brian E. Granger. "IPython: A System for Interactive Scientific Computing". In: *Computing in Science & Engineering* 9.3 (2007), pp. 21–29. DOI: [10.1109/mcse.2007.53](https://doi.org/10.1109/mcse.2007.53).
- [249] Mark A. Lukowski, Andrew S. Daniel, Fei Meng, Audrey Forticaux, Linsen Li, and Song Jin. "Enhanced Hydrogen Evolution Catalysis from Chemically Exfoliated Metallic MoS₂ Nanosheets". In: *Journal of the American Chemical Society* 135.28 (July 2013), pp. 10274–10277. DOI: [10.1021/ja404523s](https://doi.org/10.1021/ja404523s).

- [250] Alan C. Eckbreth. "BOXCARS: Crossed-beam phase-matched CARS generation in gases". In: *Applied Physics Letters* 32.7 (Apr. 1978), pp. 421–423. DOI: [10.1063/1.90070](https://doi.org/10.1063/1.90070).
- [251] Markus Thomalla and Helmut Tributsch. "Photosensitization of Nanostructured TiO₂ with WS₂ Quantum Sheets". In: *The Journal of Physical Chemistry B* 110.24 (June 2006), pp. 12167–12171. DOI: [10.1021/jp061371q](https://doi.org/10.1021/jp061371q).
- [252] Andrei V. Pakoulev, Stephen B. Block, Lena A. Yurs, Nathan A. Mathew, Kathryn M. Kornau, and John C. Wright. "Multiply Resonant Coherent Multidimensional Spectroscopy: Implications for Materials Science". In: *The Journal of Physical Chemistry Letters* 1.5 (Mar. 2010), pp. 822–828. DOI: [10.1021/jz9003476](https://doi.org/10.1021/jz9003476).
- [253] Hualing Zeng, Junfeng Dai, Wang Yao, Di Xiao, and Xiaodong Cui. "Valley polarization in MoS₂ monolayers by optical pumping". In: *Physical Review Letters* 83.10 (Sept. 1999), pp. 1950–1953. DOI: [10.1103/PhysRevLett.83.1950](https://doi.org/10.1103/PhysRevLett.83.1950).
- [254] Bairen Zhu, Hualing Zeng, Junfeng Dai, and Xiaodong Cui. "The Study of Spin-Valley Coupling in Atomically Thin Group VI Transition Metal Dichalcogenides". In: *Advanced Materials* 26.31 (Apr. 2014), pp. 5504–5507. DOI: [10.1002/adma.201305367](https://doi.org/10.1002/adma.201305367).
- [255] Cong Mai, Andrew Barrette, Yifei Yu, Yuriy G. Semenov, Ki Wook Kim, Linyou Cao, and Kenan Gundogdu. "Many-Body Effects in Valleytronics: Direct Measurement of Valley Lifetimes in Single-Layer MoS₂". In: *Nano Letters* 14.1 (Dec. 2013), pp. 202–206. DOI: [10.1021/nl403742j](https://doi.org/10.1021/nl403742j).
- [256] Cathy Y. Wong and Gregory D. Scholes. "Using two-dimensional photon echo spectroscopy to probe the fine structure of the ground state biexciton of CdSe nanocrystals". In: *Journal of Luminescence* 131.3 (Mar. 2011), pp. 366–374. DOI: [10.1016/j.jlumin.2010.09.015](https://doi.org/10.1016/j.jlumin.2010.09.015).
- [257] Callum J. Docherty, Patrick Parkinson, Hannah J. Joyce, Ming-Hui Chiu, Chang-Hsiao Chen, Ming-Yang Lee, Lain-Jong Li, Laura M. Herz, and Michael B. Johnston. "Ultrafast Transient Terahertz Conductivity of Monolayer MoS₂ and WSe₂ Grown by Chemical Vapor Deposition". In: *ACS Nano* 8.11 (Nov. 2014), pp. 11147–11153. DOI: [10.1021/nm5034746](https://doi.org/10.1021/nm5034746).
- [258] Erik H. Horak, Morgan T. Rea, Kevin D. Heylman, David Gelbwaser-Klimovsky, Semion K. Saikin, Blaise J. Thompson, Daniel D. Kohler, Cassandra A. Knapper, Wei Wei, Feng Pan, Padma Gopalan, John C. Wright, Alán Aspuru-Guzik, and Randall H. Goldsmith. "Exploring Electronic Structure and Order in Polymers via Single-Particle Microresonator Spectroscopy". In: *Nano Letters* 18.3 (2018), pp. 1600–1607. DOI: [10.1021/acs.nanolett.7b04211](https://doi.org/10.1021/acs.nanolett.7b04211).
- [259] Nara Kim, Seyoung Kee, Seoung Ho Lee, Byoung Hoon Lee, Yung Ho Kahng, Yong-Ryun Jo, Bong-Joong Kim, and Kwanghee Lee. "Highly Conductive PEDOT:PSS Nanofibrils Induced by Solution-Processed Crystallization". In: *Advanced Materials* 26.14 (Dec. 2013), pp. 2268–2272. DOI: [10.1002/adma.201304611](https://doi.org/10.1002/adma.201304611).
- [260] Stephan Kirchmeyer and Knud Reuter. "Scientific importance, properties and growing applications of poly(3, 4-ethylenedioxythiophene)". In: *Journal of Materials Chemistry* 15.21 (2005), p. 2077. DOI: [10.1039/b417803n](https://doi.org/10.1039/b417803n).
- [261] Takumi Takano, Hiroyasu Masunaga, Akihiko Fujiwara, Hidenori Okuzaki, and Takahiko Sasaki. "PEDOT Nanocrystal in Highly Conductive PEDOT:PSS Polymer Films". In: *Macromolecules* 45.9 (Apr. 2012), pp. 3859–3865. DOI: [10.1021/ma300120g](https://doi.org/10.1021/ma300120g).
- [262] A. M. Nardes, M. Kemerink, R. A. J. Janssen, J. A. M. Bastiaansen, N. M. M. Kiggen, B. M. W. Langeveld, A. J. J. M. van Breemen, and M. M. de Kok. "Microscopic Understanding of the Anisotropic Conductivity of PEDOT:PSS Thin Films". In: *Advanced Materials* 19.9 (May 2007), pp. 1196–1200. DOI: [10.1002/adma.200602575](https://doi.org/10.1002/adma.200602575).

- [263] Alexandre Mantovani Nardes, René A. J. Janssen, and Martijn Kemerink. "A Morphological Model for the Solvent-Enhanced Conductivity of PEDOT:PSS Thin Films". In: *Advanced Functional Materials* 18.6 (Mar. 2008), pp. 865–871. DOI: [10.1002/adfm.200700796](https://doi.org/10.1002/adfm.200700796). URL: <https://doi.org/10.1002/adfm.200700796>.
- [264] Kevin van de Ruit, Racheli Itzhak Cohen, Dirk Bollen, Ton van Mol, Rachel Yerushalmi-Rozen, René A. J. Janssen, and Martijn Kemerink. "Quasi-One Dimensional in-Plane Conductivity in Filamentary Films of PEDOT:PSS". In: *Advanced Functional Materials* 23.46 (June 2013), pp. 5778–5786. DOI: [10.1002/adfm.201301174](https://doi.org/10.1002/adfm.201301174).
- [265] S. Timpanaro, M. Kemerink, F.J. Touwslager, M.M. De Kok, and S. Schrader. "Morphology and conductivity of PEDOT/PSS films studied by scanning-tunneling microscopy". In: *Chemical Physics Letters* 394.4-6 (Aug. 2004), pp. 339–343. DOI: [10.1016/j.cpllett.2004.07.035](https://doi.org/10.1016/j.cpllett.2004.07.035).
- [266] S.C.J. Meskers, J.K.J. van Duren, and R.A.J. Janssen. "Thermally Induced Transient Absorption of Light by Poly(3, 4-ethylenedioxythiophene):Poly(styrene sulfonic acid) (PEDOT:PSS) Films: A Way to Probe Charge-Carrier Thermalization Processes". In: *Advanced Functional Materials* 13.10 (Oct. 2003), pp. 805–810. DOI: [10.1002/adfm.200304398](https://doi.org/10.1002/adfm.200304398).
- [267] J L Skinner. "Theory of Pure Dephasing in Crystals". In: *Annual Review of Physical Chemistry* 39.1 (Oct. 1988), pp. 463–478. DOI: [10.1146/annurev.pc.39.100188.002335](https://doi.org/10.1146/annurev.pc.39.100188.002335).
- [268] Yunhee Chang, Kwanghee Lee, R Kiebooms, A Aleshin, and A.J Heeger. "Reflectance of conducting poly(3, 4-ethylenedioxythiophene)". In: *Synthetic Metals* 105.3 (Sept. 1999), pp. 203–206. DOI: [10.1016/s0379-6779\(99\)00095-8](https://doi.org/10.1016/s0379-6779(99)00095-8). URL: [https://doi.org/10.1016/s0379-6779\(99\)00095-8](https://doi.org/10.1016/s0379-6779(99)00095-8).
- [269] Shinuk Cho, Sungheum Park, and Kwanghee Lee. "Reflectance study on the metal-insulator transition driven by crystallinity change in poly(3,4-ethylenedioxy thiophene)/poly(styrenesulfonate) films". In: *Journal of the Korean Physical Society* 47.3 (2005), pp. 474–478.
- [270] Jing-Mei Zhuo, Li-Hong Zhao, Perq-Jon Chia, Wee-Sun Sim, Richard H. Friend, and Peter K. H. Ho. "Direct Evidence for Delocalization of Charge Carriers at the Fermi Level in a Doped Conducting Polymer". In: *Physical Review Letters* 100.18 (May 2008). DOI: [10.1103/physrevlett.100.186601](https://doi.org/10.1103/physrevlett.100.186601).