Comparison of Spectral Estimation Techniques Applied to Molecular Dynamics Spectroscopy

by

Marc Thomson

B.S. in Applied Mathematics, University of Colorado - Boulder, 2018

B.S. in Chemical Engineering, University of Colorado - Boulder, 2018

A thesis submitted to

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Master of Science

Department of Applied Mathematics

2019

This thesis entitled:
Comparison of Spectral Estimation Techniques Applied to Molecular Dynamics Spectroscopy
written by Marc Thomson
has been approved for the Department of Applied Mathematics

Dr. Stephen Becker

Dr. William Kleiber

Dr. Charles Musgrave

Date: _____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Thomson, Marc (M.S., Applied Mathematics)

Comparison of Spectral Estimation Techniques Applied to Molecular Dynamics Spectroscopy

Thesis directed by Dr. Stephen Becker

Compared with experiments, molecular dynamics (MD) simulations provide a quick and inexpensive way to study the properties of chemical systems. In many cases, it is necessary to extract spectral data from these simulations, such as infrared or Raman spectra. For instance, to validate that the computational system matches a physical system, the spectral fingerprints can be examined. For complicated systems, Raman spectroscopy calculations are computationally expensive, providing an incentive to reduce the amount of data required. Currently, spectral estimation from MD simulations relies on the discrete Fourier transform (DFT); however, alternative methods can more precisely model the spectra using fewer data points. These methods are particularly effective when prior knowledge of the spectral shape is considered. Several methods, including the direct regression, Welch power estimation, the regularized resolvent transform (RRT), and a modified version of the filter diagonalization method (FDM) are compared to the DFT when applied to MD simulations of methanol and sodium chloride. We propose a novel modification of the FDM, including use of the LASSO (least absolute shrinkage and selection operator) to improve the method's accuracy. Moreover, 'windowing' present in FDM is modified to produce a significantly more accurate spectrum. The performance of these methods is then compared with each other to determine which methods are prone to include incorrect spectral features or lack correct spectral features. In brief, the modified FDM and RRT far outperformed other methods: the modified FDM produces the lowest rate incorrect spectral peaks while the RRT produces the lowest rate of missing peaks.

Dedicated to my mom, for her unwavering support

Acknowledgements

First of all, I would like to thank Dr. Stephen Becker[1] for his continuous support throughout this whole project. I do not have a significant background in signal processing, and he was extremely patient as I got up to speed. Dr. Becker provided immense support for this project, giving me a multitude of ideas on how to improve my work.

I would also like to thank Dr. Charles Musgrave[2], for the support he has offered over the last several years. Prior to this research, I worked exclusively with Dr. Musgrave, who inspired my interest in quantum chemistry. A significant portion of my understanding of solid state physics and quantum mechanics comes directly from Dr. Musgrave.

I am also grateful to Dr. Aaron Holder[3] for technical support and advice during the course of this project. Dr. Holder provided the original motivation to solve this problem, as it was relevant to his work. He provided a significant amount of useful information to make sure that my thesis would be relevant to those actually running molecular dynamics simulations.

Lastly, I am immensely appreciative of all of my friends who supported me over the last five years of my education. My success stems largely from their continual encouragement, understanding, and help.

[1] Assistant Professor, Department of Applied Math
[2] Professor and Department Chair, Department of Chemical and Biological Engineering
[3] Assistant Professor Adjunct, Department of Chemical and Biological Engineering

Contents

List of Figures

Chapter 1

Introduction

1.1 Molecular Dynamics Simulations

Molecular dynamics (MD) simulations offer a relatively inexpensive way to study the properties of chemical systems in great depth. Rather than having to synthesize a material to test its properties, these properties can be extracted simply by using computational resources. Alternatively, as MD studies the system on the atomic scale, it can target aspects of a system not easily observed at a macro-scale.

In brief, molecular dynamics is performed through repeatedly solving Newton's equations of motion, taking into account the potentials surrounding the atoms in a system. A brief exploration of the theoretical background of MD can be found in [6]. The result is a 'trajectory', or a series of atomic positions in time. A trajectory can be used to find the average properties of the system over time.

More advanced systems might require a modified version of MD, in which the quantum mechanical properties of the atomic interactions are considered. In these cases, called QM/MD, the system becomes much more computationally expensive to evaluate, as Schrödinger's Equation must be approximated to describe the nature of the interactions. In such cases, there is ample need to reduce the number of points in the trajectory necessary to make conclusions.

1.2 Spectroscopy

Molecular systems can be characterized by a variety of spectra that act as fingerprints. The basic spectrum is the vibrational, or phonon, spectra, which characterizes the relative motion of the atoms over time. Some of these phonon modes will be visible through experimentally-convenient forms of spectroscopy, including Raman spectroscopy and infrared spectroscopy.

In certain cases, it is useful to extract the spectral data, be it vibrational, infrared, or Raman, from a theoretical system. For instance, this can be done to make sure that the system being simulated matches a system being observed. To do so, a number of approaches have been developed [7]. Many rely on the construction of autocorrelation functions, as defined in Equation (1.1):

$$\rho(n) = \langle f_i \cdot f_{i+n} \rangle \tag{1.1}$$

where f is a sequence of observations in time [18]. The notation $\langle\rangle$ indicates an average over all particles in the system and all data with the appropriate difference in time.

This autocorrelation function, when transformed from the time domain to the frequency domain, provides a spectrum. This result stems from the Wiener-Khinchin Theorem, which demonstrates that the Fourier transform of an autocorrelation function is the Fourier transform of the square of the property under consideration [19]. The precise identity of the function f determines the type of spectroscopy. If f is the vector of atomic velocities, the result is the phonon spectrum. If f is the polarizability of the system, the result is the Raman spectrum [7].

The computation of the vibrational autocorrelation function is quite cheap, as velocities are inherently computed in an MD simulation. Polarizability, however, is a much more expensive computation, especially for QM/MD. These calculations are not done during the main simulation but are done after the fact on certain molecular configurations along the trajectory. To reduce computational cost, it is best to reduce the required number of points at which the polarizability is calculated. Primarily, the spectrum of an MD autocorrelation function is recovered using the Fourier transform, approximated with the Discrete Fourier Transform (DFT) [2, 7, 12], but more efficient methods might be possible. This thesis will pursue more efficient methods, exploiting assumptions about the structure of the spectrum.

Many prior studies have effectively been able to perform 'super-resolution', in which spectra are extracted in much more detail than possible with the traditional DFT. However, these methods often rely on the true spectrum being fundamentally a sum of Dirac-δ functions [1].

Typical chemical spectra do not appear just as Dirac-δ functions in the frequency domain. Instead, these peaks are broadened, often into a Lorentzian shape [3]. A Lorentzian function

appears with much the same form as a Gaussian, and is defined by Equation (1.2):

$$f(x) = \frac{A}{2\pi\left((x - x_0)^2 + \left(\frac{1}{2}w\right)^2\right)} \tag{1.2}$$

Note that for this work, the Lorentzian functions will be normalized with respect to height as opposed to area. The Lorentzian function is characterized by three parameters. The amplitude A corresponds to its maximum height, w characterizes the width, and x_0 describes the center of the function. If this function is transformed into the frequency domain, it is a decaying sinusoid, as shown in Equation (1.3) [14]:

$$F(t) = \frac{A}{w} e^{2\pi i t x_0 - w\pi|t|} \tag{1.3}$$

This knowledge of the fundamental lineshape should allow for more precise spectral recovery. The goal of this work is to reduce the number of points necessary to get an accurate spectrum from autocorrelation data. In doing so, it will reduce the amount of computational resources necessary to perform spectroscopy with molecular dynamics.

Chapter 2

Methods

2.1 Assumptions

In order to make the problem of spectral estimation from molecular dynamics tractable, a number of assumptions must be made and the scope of the work must be defined.

The purpose of this work is to reduce the number of points necessary to produce a spectrum from the output of a simulation. A successful method should output a spectrum equivalent to the converged spectrum of the simulation. Any deviation of the physical chemical spectrum from the converged spectrum produced by molecular dynamics is outside the scope of the problem.

For the parametric methods tested, it is assumed that the converged spectrum of each chemical species is a sum of Lorentzian functions, as discussed in the Introduction. For the nonparametric methods, this assumption is generally not necessary, except in the case of the regularized resolvent transform. This assumption will not significantly alter the results. If this

assumption is incorrect, then methods that rely on its truth will produce less accurate results. These methods will then be truthfully classified as less effective when applied to MD simulations.

To test the effectiveness of the proposed techniques of spectral recovery, each method is used to calculate the spectra of several data sets, as discussed later in this chapter. To estimate the true spectra of the chemical species being studied, the simulations are run for a long time. The spectra are extracted by applying the Discrete Fourier Transform (DFT) to the converged region of these long simulations. These approximate spectra are assumed to be the converged spectra of the simulations. This assumption is justified by comparing spectra produced by all of the long-time simulation and only a portion of the long-time data, which shows strong agreement in the spectral shape. This is presented later in this chapter.

It is also necessary to assume that the species being simulated, methanol (organic) and sodium chloride (inorganic), are sufficiently representative of systems of interest to researchers using molecular dynamics. This assumption is limiting, as more complicated systems likely demonstrate much more complicated behavior. Future work should be done on more advanced systems to test the applicability of this work to the broader range of MD simulations.

To reduce the computational overhead necessary for this work, the vibrational, or phonon, spectrum is used in place of more complicated spectrum. Calculations of Raman or infrared spectra tend to be significantly more computationally expensive. The modes active in a Raman or Infrared spectrum tend to be a subset of the modes active in the phonon spectrum [11]. Thus, vibrational spectra serve as a good approximation for the forms of more complicated spectra.

2.2 Data Collection

To examine the performance of the various methods, several primary data sets are used. In some cases, 'toy' data is used to validate the effectiveness of some techniques. The toy data is generated using Lorentzian modes in the spectral domain which are analytically transferred to the time domain with the Fourier transform. Additionally, Gaussian noise is added to the time signal to roughly simulate the error from calculating the autocorrelation function. The toy data is not used to compare the effectiveness of the various methods, as this would unfairly favor the methods that assume Lorentzian lineshape without sufficient justification.

The comparative performance of the spectral estimation methods is assessed using MD simulations run using the free program LAMMPS from Sandia National Labs [4]. In particular, two example simulations of sodium chloride and methanol are run. Images from the simulations are shown in Figure 2.1. The setup scripts for these simulations are included in the download package for LAMMPS, with the names "eim" for the sodium chloride simulation and "dreiding" for the methanol simulation. As the simulations run, all atomic velocities are recorded every five timesteps. The timesteps for these simulations are 1 fs (methanol) and 0.3 fs (sodium chloride.)

These velocities can be subsampled and processed to produce autocorrelation functions, which are then analyzed using spectral estimation methods to recover the vibrational spectra. When calculating the autocorrelation function using Equation 1.1, the dot product of the velocities is used.

To ensure the accuracy of the spectra, it is necessary to ensure that the velocities have equilibrated. Temperature directly measures the average velocity of a system, so convergence

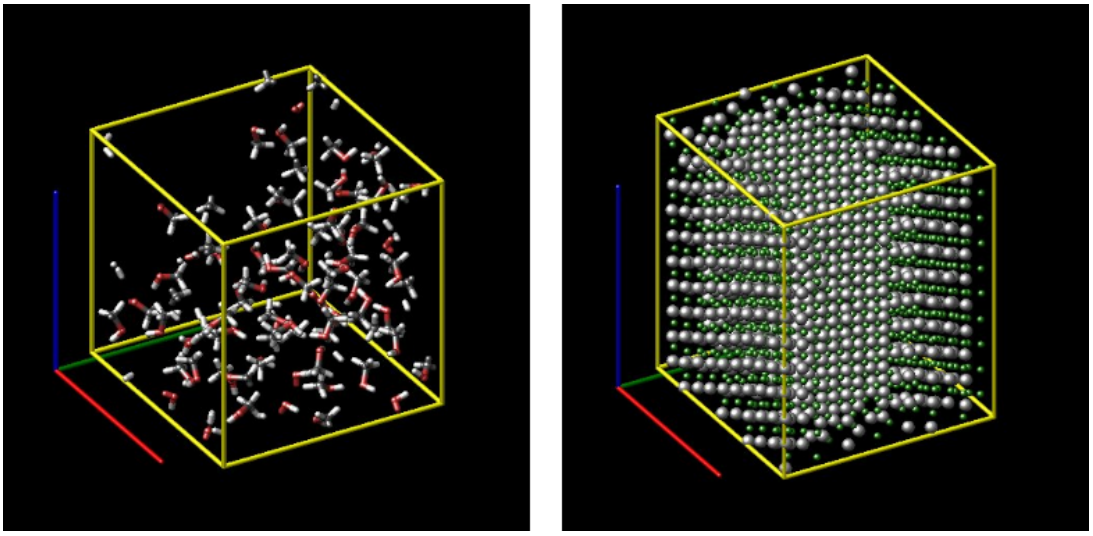

FIGURE 2.1: Images of the unit cells used for MD simulations of methanol (left) and sodium chloride (right)

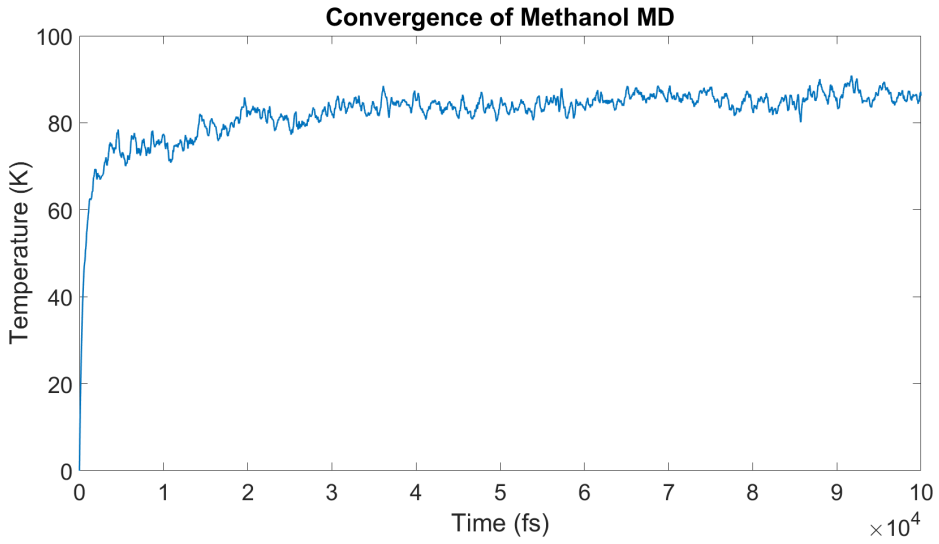

FIGURE 2.2: Temperature profile of methanol MD simulation

in temperature indicates convergence in velocity. Temperature profiles for methanol and NaCl are shown in Figure 2.2 and Figure 2.3 respectively.

The simulation of NaCl appears to go through a significant change around 5,000 ps, with equilibrium beginning around 10,000 ps. Images output for this simulation seem to imply a phase change of the system around 5,000 ps. As stated in the assumptions, the ability of the

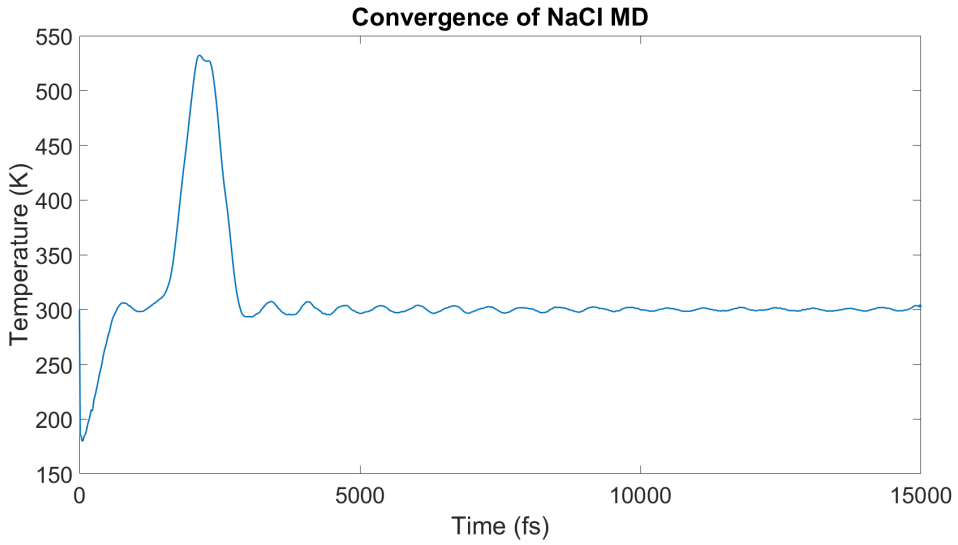

FIGURE 2.3: Temperature profile of NaCl MD simulation

MD simulations to accurately model the real system lies outside the scope of this problem. Thus, it is sufficient to observe that the NaCl system does eventually equilibrate. Both system temperatures eventually level out, with equilibrium certainly established by halfway through the simulation time. Thus, the second half of the simulation time is taken as the data sets to be tested.

Using the full sets of equilibrated data, the converged spectra can be estimated. Figure 2.4 and Figure 2.5 show the converged spectra for methanol and NaCl, respectively. On each plot, the spectrum is shown with the full set of converged data, and with 80% of the converged data. In both cases, the two spectra demonstrate strong agreement, indicating that the spectra have indeed converged.

The sodium chloride spectrum is quite basic, with only one primary defined peak. In particular, the spectrum has no behavior in the higher frequency region of the spectrum, indicating that the timestep is finer than necessary to capture its spectral behavior. It is then possible that

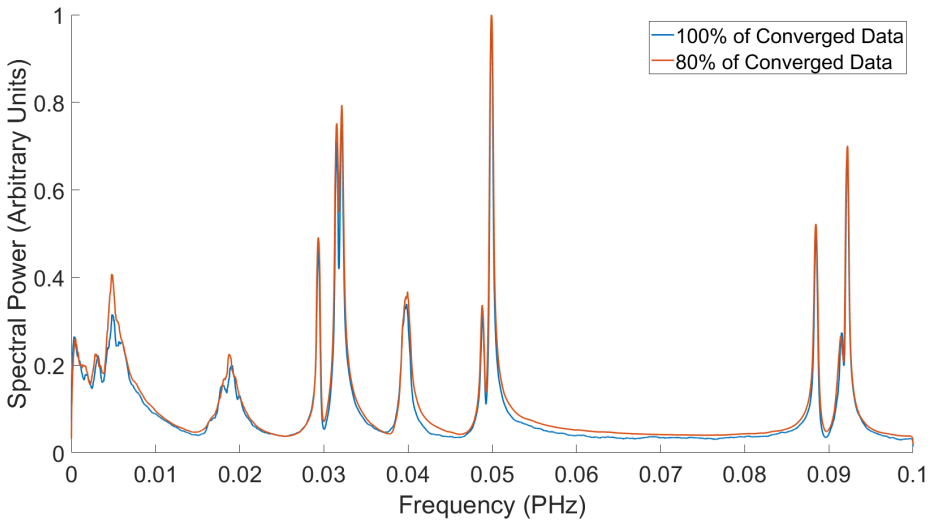

FIGURE 2.4: Converged vibrational spectrum of methanol

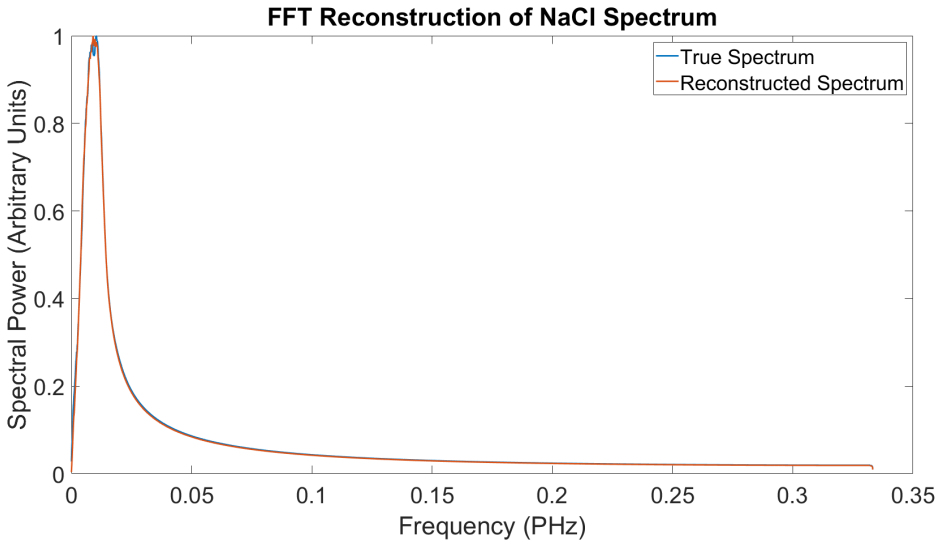

FIGURE 2.5: Converged vibrational spectrum of NaCl

a longer time simulation run with a longer timestep might better resolve the region of spectral density. However, given the simplicity of the system, a simple spectrum makes sense.

Because the goal of this work is the reduction in the number of points necessary to produce an accurate spectrum, the spectral estimation methods will be tested on a small subset of the converged data. For each test, an interval size and number of points is specified. Then, a random interval of appropriate size is selected from the converged velocity data, and randomly

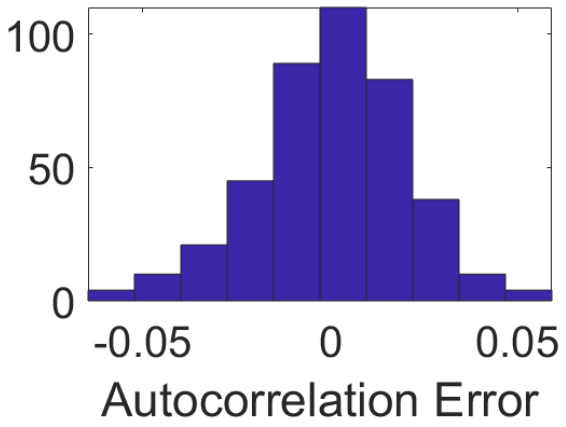

FIGURE 2.6: Histogram of example error in the autocorrelation function. The coarse grid may include some time points interpolated with a spline, as gaps in the autocorrelation function are possible. The time grid used in the plotting is the coarser time grid.

sampled without replacement, so that the resulting time series is not uniformly spaced. These points are used to estimate the velocity autocorrelation function (VACF). As the interval may not be fully sampled, some gaps in the VACF are possible, which are filled by interpolating the signal with a spline. An autocorrelation function is fundamentally a series of averages, so its accuracy at any time is dependent on the number of samples used to average. By the central limit theorem, typical variation in the value at any point will be $\mathcal{O}\left(\frac{1}{\sqrt{m}}\right)$, with m representing the number of points used in the average. For the purposes of this work, m represents the number of pairs of data points with a given lag time.

Before proceeding with the spectral estimation, it is useful to examine the nature of the error in the VACF. To do this, an interval from the methanol simulation is sampled twice with different numbers of points. The difference between the two calculated autocorrelation functions is then calculated. In this case, the two autocorrelation functions are calculated with a number of points differing by a factor of two. A histogram of these errors is shown in Figure 2.6, and a plot of the error against time is shown in Figure 2.7.

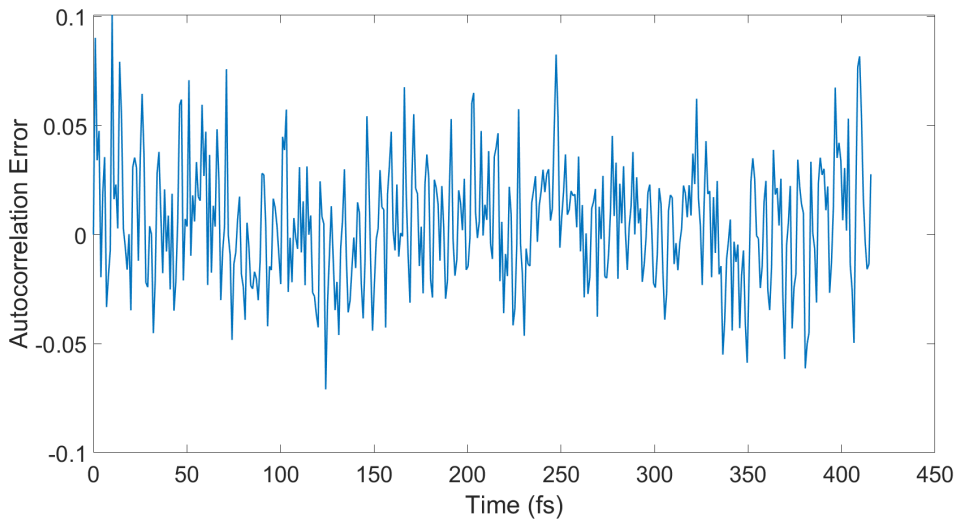

FIGURE 2.7: Plot of error in the autocorrelation function against time

The errors appear roughly normally distributed based on the histogram. In addition, the errors appear relatively random when plotted against time. Thus, it is reasonable to conclude that the error in the VACF is sufficiently similar to Gaussian noise. Future work could examine this assumption more rigorously, including performing an Anderson-Darling test to test the normality assumption.

2.3 Methods of Spectral Estimation

For this work, a number of spectral estimation methods are tested.[1] These methods range from extremely common, like the discrete Fourier transform, to specialized, like the filter diagonalization method. In addition, filter diagonalization is modified to make it more effective when dealing with the problem at hand. Three of the methods tested are parametric, meaning that they directly fit Lorentzian functions to the spectrum. The remaining three techniques

[1]The code used for these analyses will be uploaded to the GitHub repository https://github.com/MarcThomson/MDSpectralAnalysis.git

are nonparametric, or spectral. The parametric methods' accuracy strongly depends on the assumption on the form of the data, but have the additional advantage that they return solutions with infinite resolution.

2.3.1 Discrete Fourier Transform (DFT)

The discrete Fourier transform requires little introduction, being ubiquitous in signal processing [10, 16]. The idea of the method is to discretely approximate the Fourier transform, which transforms data from the time domain to the frequency domain. The basic formula for the DFT is presented in Equation (2.1):

$$F_n = \sum_{k=0}^{N-1} f_k e^{-2\pi i n k/N} \tag{2.1}$$

for a discrete signal $\{f_0, ..., f_{N-1}\}$. The speed of this algorithm is greatly enhanced by the Fast Fourier Transform (FFT), allowing the DFT to dominate signal processing. For the purposes of this work, the terms DFT and FFT will be used interchangeably. Prior studies involving spectroscopy from MD have used the DFT [2, 7, 12]. Thus, this is the baseline method to which other methods should be compared. The goal of this work is to identify a method more effective than the DFT.

2.3.2 Regularized Regression

Given the assumption that the spectral signal is a summation of Lorentzian functions, it seems reasonable to create a set of Lorentzian functions and use a regularized regression technique to fit the data in the time domain. This set of functions is referred to as a 'dictionary.'

With no knowledge of the centers of widths of the true Lorentzian functions, the dictionary must contain functions with a wide range in both of these parameters. To reduce the sample space, a method that generates sparsity, like the least absolute shrinkage and selection operator (LASSO) [5] is preferred. The lasso returns the minimizer to the penalized linear regression problem defined in Equation (2.2):

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2 + \lambda \|x\|_1 \tag{2.2}$$

where $\|x\|_1 = \sum_{i=1}^{n} |x_i|$. In addition, a constraint is added for this problem that all x_i must be positive, as a power spectrum consists of positive peaks. This problem is convex and therefore a local minimum must be a global minimum. The lasso has the property that the coefficients of unnecessary features tend to be set to zero, leading to the desired sparsity property.

The parameter $\lambda > 0$ must be supplied by the user. A popular method involves k-fold cross-validation [5], in which the data is partitioned into k sets. For each set, the model is fit on the remaining data, and its accuracy is tested on the held-out data. By doing so, the error and its variance can be assessed for a given value of λ. By varying the value of λ, a minimum in the error is discovered. At this value of λ, the model is accurate without being too overfit. A common practice when selecting λ involves taking not the value which minimizes the error, but the largest value of λ for which the error is within one standard error of the minimum [13]. This adds an extra layer of protection against overfitting, as larger values of λ generally lead to a sparser solution. For the purposes of this work, the lasso is used with 10-fold cross-validation. In addition, as the cross-validation step is random and therefore variable, this step is repeated three times.

The application of the lasso in this case is hindered by the nature of the Lorentzian, as it is linear in one of its parameters (amplitude) and nonlinear in the other two (width and center). The lasso can only be used for linear estimation, so a wide dictionary of Lorentzian functions, with variable center and width, is necessary. These are generated by creating a 2-dimensional grid in width and center. The maximum frequency possibly observed by the DFT is called the Nyquist critical frequency, defined by

$$\omega^* = \frac{1}{2\tau} \tag{2.3}$$

with τ representing the timestep. Thus, it is reasonable to use this as the maximum frequency under consideration for the lasso basis functions. The n_c centers of the Lorentzian functions are evenly spaced from $\frac{\omega^*}{n_c}$ to ω^*. The widths are slightly more arbitrary. In this work, they are chosen on a logarithmic scale from 0.05% of ω^* to 10% of ω^*. The choice of grid size is also arbitrary and must be tested on a specific problem. The lack of guidance in choosing widths highlights a key problem with this approach: it is necessary to create a large swatch of unnecessary functions, significantly increasing the cost of the calculation. Later, the filter diagonalization method will be presented as an alternative means of generating basis functions.

The lasso naturally shrinks the magnitude of the coefficients in addition to performing variable selection. This poses a slight problem, as it introduces bias into the resulting curve [13]. To mitigate this bias, the dictionary functions selected by the lasso are refit without the magnitude penalty, instead only keeping the positivity constraint on the coefficients. Unless otherwise specified, this step will be performed in all cases where the lasso is used. The lasso calculation is performed using MATLAB.

2.3.3 Regularized Regression With Nonlinear Optimization

Another problem plaguing the use of the lasso on this problem is spectral leakage [15]. This problem commonly occurs with the DFT, in which the grid of possible output does not perfectly align with the true frequency peaks. This problem will also occur when the centers of the dictionary functions in the lasso method do not properly align. This challenge can be mitigated by adding a second step to the lasso fitting procedure, in which the surviving dictionary functions are individually optimized in amplitude, width, and center. This optimization is performed by iterating through the selected dictionary functions, and reducing the error while holding the remaining functions constant. The nonlinear optimization step is done with the *fmincon* function in MATLAB [9], which uses sequential quadratic programming. The exact nature of this algorithm is outside the scope of this work. During this optimization, the amplitude, center, and width of each function are constrained to be within a range defined by their initial values, but crucially, they are not constrained to be on a grid.

The addition of nonlinear optimization offers a promising way to reduce the computational cost of using the lasso. Because it is no longer necessary to use an extremely fine grid, the number of dictionary functions can theoretically be reduced. The comparative accuracy of using nonlinear optimization as opposed to using a finer grid for the lasso is explored in the results section.

2.3.4 Welch's Method

Welch's Method is a nonparametric method in which the time data is partitioned into n segments, each of length m [20]. There is no need for the product of n and m to equal the size of

the full data set, as the intervals are allowed to overlap. Commonly, an overlap of 50% is used. The discrete Fourier transform is applied to each data interval, and a periodogram is produced with an additional weighting function, called the windowing function, designed to reduce the effects of spectral leakage. The periodograms are averaged to produce the full spectrum. This method trades resolution for noise reduction, which might be useful compared with the regular DFT. In this work, four partitions are used with an overlap of 50%.

2.3.5 Filter Diagonalization Method (FDM)

The filter diagonalization method was designed to extract spectra from nuclear magnetic resonance (NMR) readings [8]. Fortunately, NMR signals have the form of decaying sinusoids in time, which correspond to Lorentzian functions in the frequency domain. Thus, this technique offers a promising method of spectral estimation in our scenario. In particular, this method is able to generate a set of dictionary functions that accurately describe the signal under examination. The FDM is ultimately derived from a method called Krylov basis diagonalization, which will be described first. As the FDM is a major focus of this work, the details of its derivation from Mandelshtam's work [8] are summarized here, although the method will be slightly altered to increase applicability to the MD problem.

2.3.5.1 Krylov Basis Diagonalization

Both the FDM and Krylov basis diagonalization assume that the signal is of the form

$$f(t) = \sum_{k=1}^{K} A_k e^{-it\omega_k} \tag{2.4}$$

with ω_k representing a set of complex frequencies. Equating this to our previous definition of a Lorentzian function, ω_k is decomposed into parts describing the center and the width of the Lorentzian functions:

$$w_k = \frac{-Im(\omega_k)}{\pi}$$
$$x_{0,k} = \frac{-Re(\omega_k)}{2\pi}$$

To solve this problem, Mandelshtam takes an abstract approach based on quantum mechanical theory. He defines a linear vector space $\mathbb{V}$ with an inner product such that $(\Phi \mid \Psi) = (\Psi \mid \Phi)$, as opposed to the more classical Hermitian inner product $(\Phi \mid \Psi) = (\Psi \mid \Phi)^*$. This inner product actually violates the definition of an inner product, as it is possible to find a nonzero Ψ such that $(\Psi \mid \Psi) = 0$. Nevertheless, this 'inner product' is necessary for the properties of $\mathbb{V}$.

He then defines a diagonalizable, complex-symmetric operator $\hat{\Omega}$, acting on vectors from $\mathbb{V}$, with eigenvalues ω_k and eigenvectors v_k. This operator represents the Hamiltonian operator in quantum mechanics. Based on the properties of $\hat{\Omega}$, the operator has the spectral representation

$$\hat{\Omega} = \sum_k \omega_k v_k^T v_k \tag{2.5}$$

Functions of the operator can thus be defined in a spectral sense:

$$f\left(\hat{\Omega}\right) = \sum_k f(\omega_k) v_k^T v_k. \tag{2.6}$$

The new operator $f\left(\hat{\Omega}\right)$ has the same eigenvectors as $\hat{\Omega}$ with eigenvalues equal to $f(\omega_k)$. Of particular interest is the time-evolution operator, $\hat{U}(t) = e^{-it\hat{\Omega}}$.

Given an initial vector $u_0 \in \mathbb{V}$, u_t is defined by the action of the time-evolution operator:

$$u_t = \hat{U}(t)u_0. \tag{2.7}$$

The autocorrelation function for u_t is

$$c(t) = u_0^T u_t = u_0^T \hat{U}(t)u_0 = \sum_k (u_0^T \mid v_k)^2 e^{-it\omega_k}. \tag{2.8}$$

Equation (2.8) matches the form of Equation (2.4), so finding the eigenvalues of $\hat{U}(t)$ will produce the parameters of the constituent Lorentzian functions of $c(t)$. We are only interested in finding the terms with nonzero amplitude, so we will only see eigenvalues such that $(u_0 \mid v_k) \neq 0$. Instead of continuous time, consider discrete time $t = n\tau$, in which case $\hat{U}$ becomes $e^{-in\tau\hat{\Omega}}$. The problem can thus be represented as the eigenvalue problem

$$\hat{U}v_k = e^{-i\tau\omega_k}v_k. \tag{2.9}$$

The initial vector u_0 must exist within the space spanned by the eigenvectors v_k. Each time the time-evolution operator is applied to u_0, a new vector u_n is created, creating a Krylov subspace. This set of $\{u_0, ..., u_{M-1}\}$ will eventually span the space defined by v_k, meaning the eigenvectors can be written in terms of this basis as follows:

$$v_k = \sum_{n=0}^{M-1} [B_k]_n u_n \tag{2.10}$$

where B_k is an unknown vector of weights. This equation can be plugged into the eigenvalue problem and left multiplied by u_m^T, yielding the generalized eigenvalue problem

$$\sum_{n=0}^{M-1} u_m^T \hat{U}^1 u_n[B_k] = e^{-i\tau\omega_k} \sum_{n=0}^{M-1} u_m^T \hat{U}^0 u_n[B_k]. \tag{2.11}$$

Define U_n as a matrix with elements $(U_n)_{i,j} = c(\tau(n+i+j))$. Because $\hat{U}^n u_0 = u_n$, Equation (2.11) can then be written in matrix form as

$$U_1 B_k = e^{-i\tau\omega_k} U_0 B_k \tag{2.12}$$

Solving this generalized eigenvalue problem yields the parameters of the Lorentzian functions present in the vector $c(n)$. Mandelshtam continues to provide a means of extracting the amplitudes of these Lorentzian functions. However, in practice, this method breaks down in the face of even minor noise. To demonstrate this, a set of toy data is generated using a set of known Lorentzian functions. One data set is noiseless, while the other has small-amplitude Gaussian noise added. Then, Krylov basis diagonalization is applied to these data sets and the coefficients recovered. Plots of the real and imaginary parts of the recovered data are shown in Figures 2.8 and 2.9.

The data shown in these figures demonstrate that the coefficient recovery of Krylov basis diagonalization is unstable to deviation from Lorentzian curves. The real parts of the coefficients show a significant deviation from the true values when noise is introduced. Moreover, the imaginary parts, initially not present in the coefficients, appear with significantly nonzero magnitude. It is odd that the coefficient determination scheme does not appear to work, as it is demonstrated as effective in Mandelshtam's paper. This might reflect incorrect implementation,

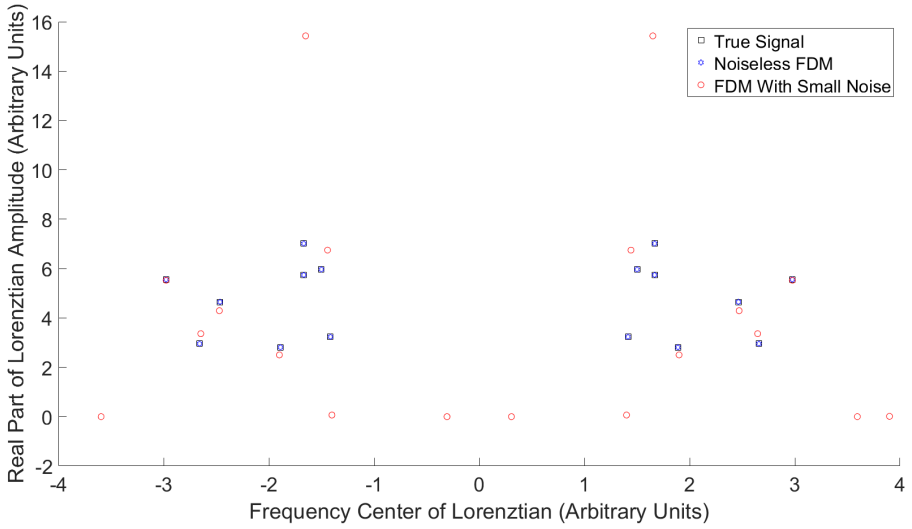

FIGURE 2.8: Real amplitudes of known Lorentzians recovered with Krylov basis diagonalization

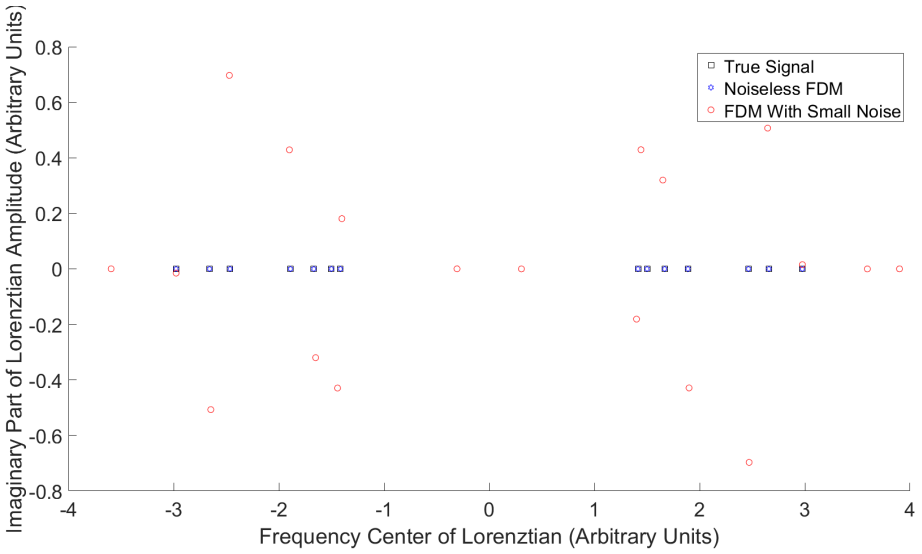

FIGURE 2.9: Imaginary amplitudes of known Lorentzians recovered with Krylov basis diagonalization

though this is unlikely due to the perfect coefficient recovery in the noiseless case. In any case, given the apparent instability, an alternative means of coefficient estimation is required.

To get around this issue, the method is modified using the introduction of the lasso. Krylov basis diagonalization provides the dictionary of functions that would be computationally expensive to generate otherwise. In addition, as the number of data points exceeds twice the true

number of basis functions, the excess eigenvalues become meaningless. The lasso eliminates these extra functions.

2.3.5.2 Filter Diagonalization

With that said, Krylov basis diagonalization is not directly applied to this problem. Before being applied, it is transformed to become the filter diagonalization method. This change is motivated partially by the desire for enhanced computational speed and partially to allow for windowing, which allows for emphasis on different parts of the frequency domain. The transformation takes places by taking the discrete Fourier transform of the matrices in the generalized eigenvalue problem. This is done by first defining a vector of frequencies $\{\phi_j\}$, defined by

$$\phi_k = \frac{4\pi k}{N\tau}, k = \{0, ..., N\} \tag{2.13}$$

where N is the length of the signal. Then, the transformed matrices $\tilde{U}_0$ and $\tilde{U}_1$ are defined as

$$(\tilde{U}_p)_{j,k} = \sum_{n=0}^{N/2-1} \sum_{m=0}^{N/2-1} e^{in\tau\phi_j} e^{im\tau\phi_k} c(\tau(m+n+p)) \tag{2.14}$$

This is relatively expensive to evaluate[2], but Mandelshtam transforms it into a more efficient form. Solving this produces the same set of eigenvalues as before, but allows for the problem to be broken into a series of 'windows' that result in more efficient computation. This is done by selecting a series of intervals in ϕ_k, evaluating Equation 2.14 for each interval individually,

[2]For the data sizes under investigation in this work, computational expense is not a significant issue. Moreover, the issue of computational expense could be mitigated by exploiting the Hankel structure of these matrices. Through permutation, these matrices can be turned into Toeplitz form. We conjecture that using the symmetry of the autocorrelation function in negative time, the eigenvalue problem could then be solved by taking the FFT of both sides. This is left to future researchers to investigate further.

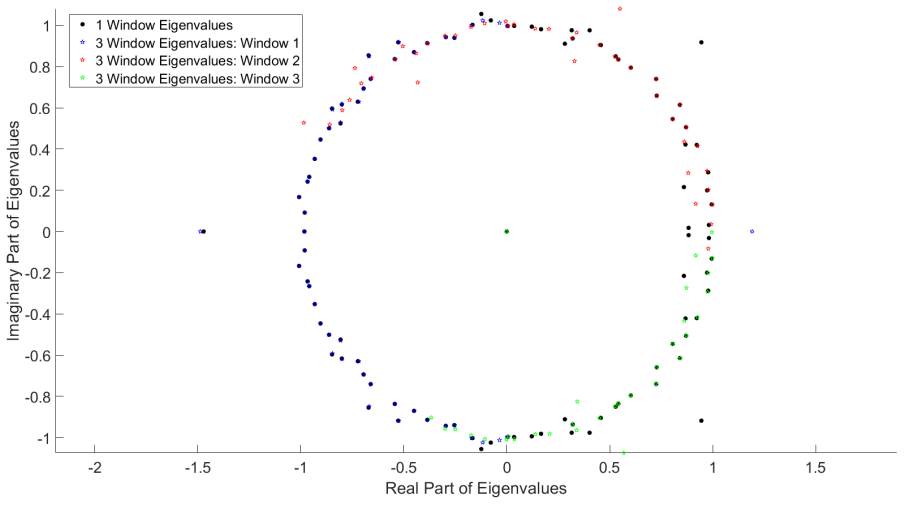

FIGURE 2.10: Eigenvalues produced from FDM using one window and three overlapping windows

solving the resulting eigenvalue problem, and combining the results. The justification for this is beyond the scope of this work, but will be demonstrated.

Figure 2.10 shows the eigenvalues produced from a subset of the methanol data with one window and with three overlapping windows. The eigenvalues produced from the three windows are nearly identical to those produced from one window alone. Interestingly, they appear to be slightly perturbed. This motivates this work's second novel modification to the FDM: rather than emphasize computational speed using windowing, the eigenvalues from multiple windowing configurations will be concatenated to form a large dictionary of functions, which is then fit using the lasso. Once the matrices $\tilde{U}_p$ have been calculated, it is relatively inexpensive to repeatedly subsample them and solve the generalized eigenvalue problem again. The effects of this modification will be explored in the results section.

2.3.5.3 Estimating Significance With FDM

Given the complexity of the FDM, there is not a straightforward way to estimate a true statistical significance of the signal that results. Still, it is useful to at least approximate the importance of various peaks for the fit solution. This section will outline a procedure to do so.

To begin, the data in the time domain should be partitioned roughly evenly and used to compute two different autocorrelation functions. The first autocorrelation goes through the modified FDM procedure, in which the data is windowed, the eigenvalues from all windowing configurations are combined, and the resulting functions are fit with the lasso. Once this is done, the functions that the lasso selected are used to fit the second autocorrelation function. This fit is supplemented with the bootstrap [13]. The autocorrelation data is sampled with replacement, with a sample size identical to the full data size. This sample is fit with the lasso, using the previously discovered functions. The sampling process is repeated many times, and the frequency of each function's appearance the fit model is recorded. While this proportion is not exactly a p-value, it provides a rough estimate as to the relevance of the basis functions in the model.

Figure 2.11 shows this process applied to the methanol data, resulting in peaks of various importances.

2.3.6 Regularized Resolvent Transform (RRT)

A final technique under consideration is the regularized resolvent transform, a nonparametric method that still relies on the Lorentzian nature of the underlying data. The precise details of the derivation of this method are omitted here, but are described in great details in

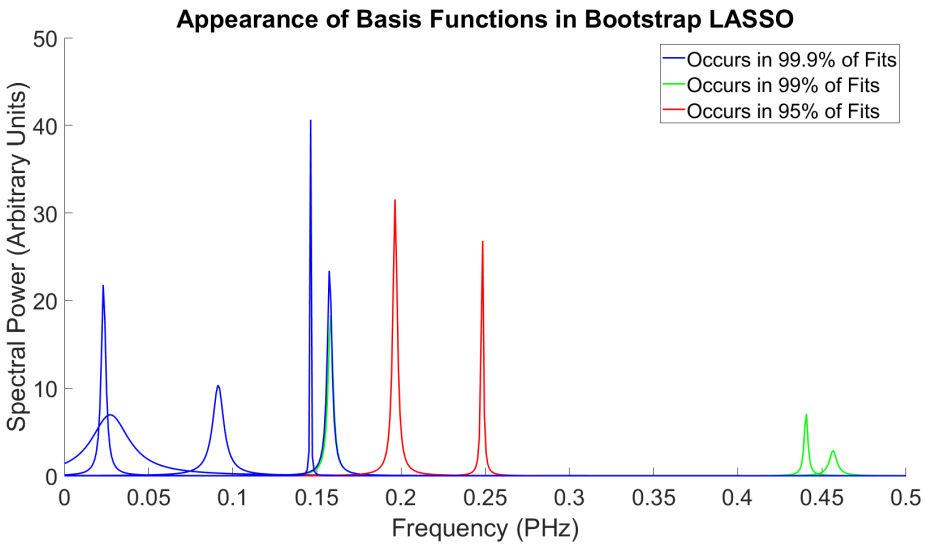

FIGURE 2.11: Rate of peaks being present from the fit model

Mandelshtam's paper [8]. The key point is that in the case of truly Lorentzian spectral data, the RRT converges to the infinite time DFT. This method is applied by first defining the 'matrix pencil', $R(s)$, with s being frequency. This term is defined as

$$R(s) = \frac{U_0 - e^{i\tau s}U_1}{i\tau} \tag{2.15}$$

with U_0 and U_1 defined in the same way as in Krylov diagonalization. Then, a regularization parameters $q \geq 0$ is introduced, and a functional representation of the spectrum is calculated as

$$I(s) = C^T(R^H R + q^2)^{-1}R^H C - \frac{i\tau c(0)}{2} \tag{2.16}$$

where C is a vector with elements $\{c(0), c(\tau), ..., c((N-1)\tau/2)\}$. For this problem, regularization negatively affects performance when added, so q is set to 0. Future work could include exploration of the effects of the regularization parameter on the performance of RRT when applied to MD simulations.

Chapter 3

Results

To test the effectiveness of various spectral estimation methods, repeated simulations are performed. For each simulation, a random starting point among the converged velocity data is selected. One hundred velocities are sampled from the next 200 velocity data points, which are used to compute a velocity autocorrelation function (VACF.) The spectral recovery methods are applied to these VACFs. Note that in the case of FDM, the modified FDM is used, including the addition of lasso, the modified windowing procedure, and weighting based on the sample size for the VACF points. The regular FDM is not tested, given the demonstrated instability of the coefficient recovery procedure.

3.1 Performance of Various Methods

Before quantitatively comparing the performance of the methods, it is useful to qualitatively examine the output compared to the converged spectra.

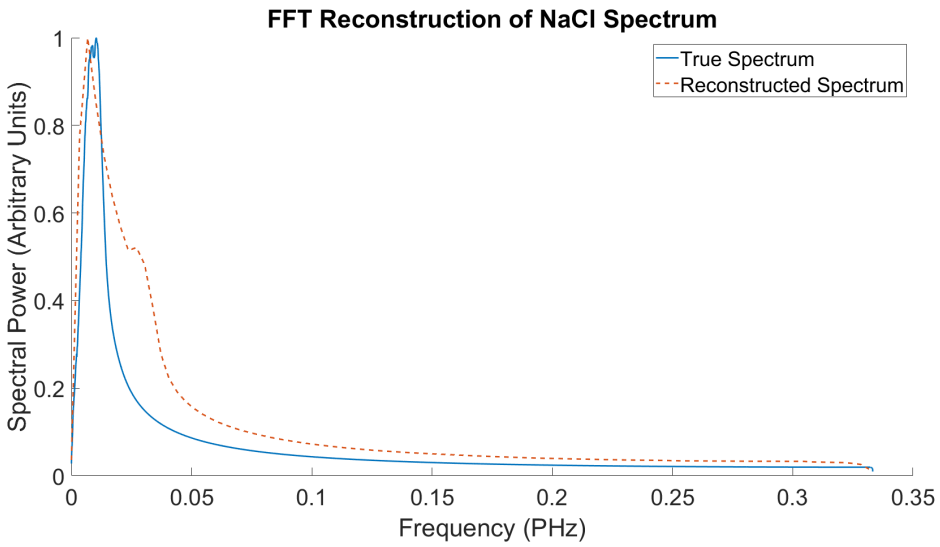

FIGURE 3.1: Reconstruction of the NaCl spectrum with the FFT

3.1.1 Discrete Fourier Transform (DFT)

Figure 3.1 shows the performance of the FFT when applied to the sodium chloride spectrum. The FFT does a decent job of describing the behavior of the spectrum, with a peak around the proper location followed by exponential decay. However, it inexplicably adds an extra shoulder as the spectrum decays.

Figure 3.2 shows the performance of the FFT when applied to the methanol spectrum. Its performance is clearly quite weak, with only general areas of density highlighted. The peak around 0.02 PHz is entirely missed, replaced with a valley in the FFT reconstruction. Virtually no peaks are correctly identified, with the peaks that do appear bearing little resemblance to the peaks of the true spectrum.

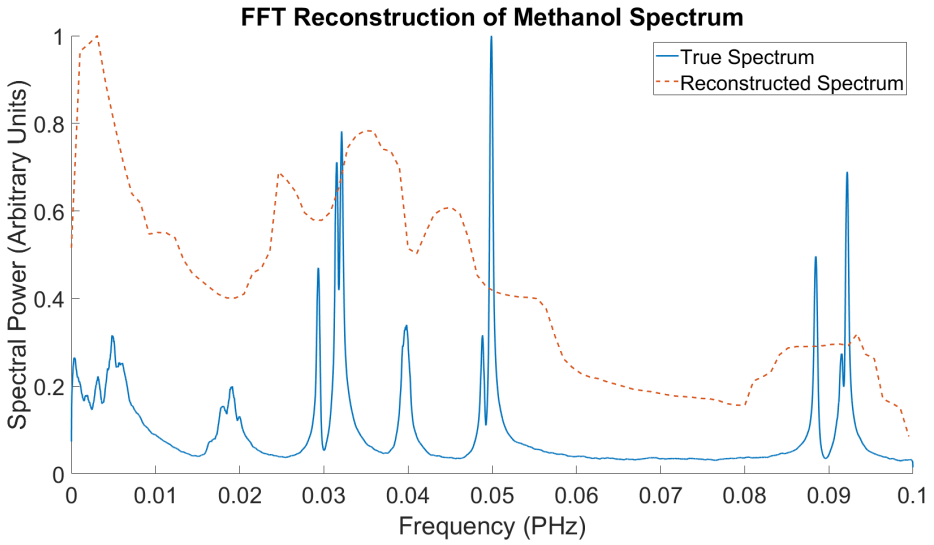

FIGURE 3.2: Reconstruction of the methanol spectrum with the FFT

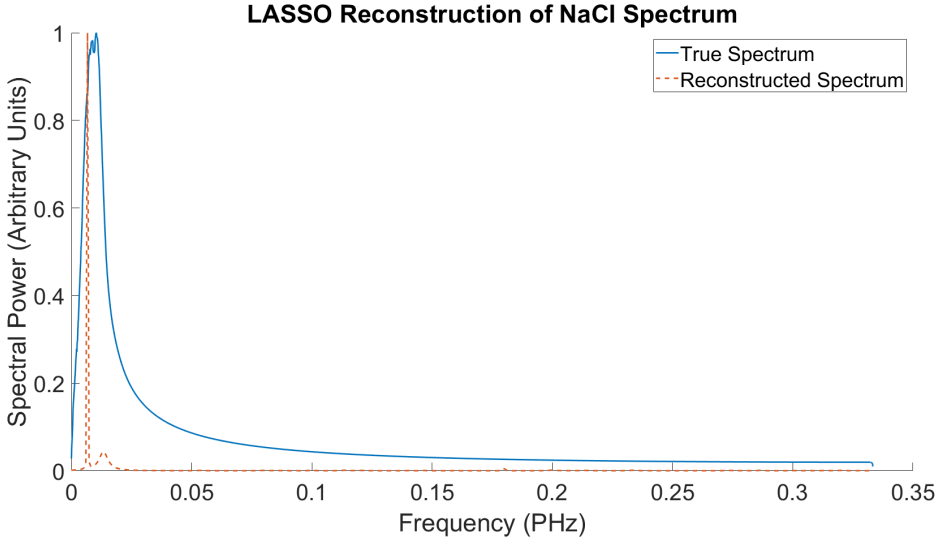

FIGURE 3.3: Reconstruction of the NaCl spectrum with the lasso

3.1.2 Regularized Regression

Figure 3.3 shows the performance of the lasso method on the NaCl spectrum. Because the lasso dictionary only contains symmetric Lorentzian functions, it is unable to correctly model the one-sided exponential decay present in the true spectrum. With that said, it does appear to place the primary peak of the spectrum in the correct location.

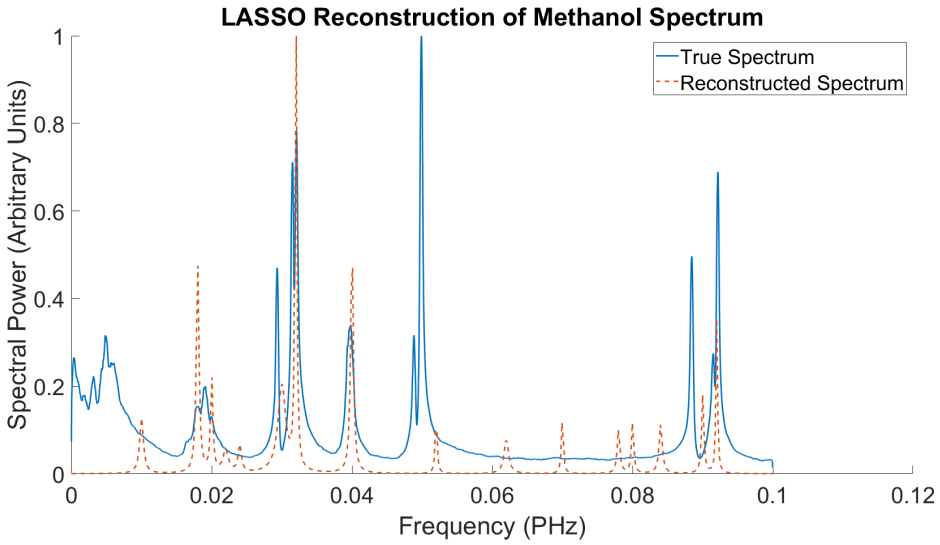

FIGURE 3.4: Reconstruction of the methanol spectrum with the lasso

Figure 3.4 shows the performance of the lasso method on the methanol spectrum. Its accuracy is very poor: it misses the spectral density between 0 and 0.01 PHz entirely, and artificially adds a number of peaks in the region between 0.05 and 0.09 PHz. Occasional peaks line up well with the true spectrum peaks, but given the randomness of the rest of the spectrum, it is likely that this occurred mainly by chance.

3.1.3 Regularized Regression With Nonlinear Optimization

Figure 3.5 shows the performance of the lasso method with subsequent nonlinear optimization on the NaCl spectrum. Its performance is much stronger than the regular lasso method, as it even appears to have captured the slightly split peak present in the main NaCl peak. Figure 3.6 shows the methanol spectrum reconstructed by the optimized lasso method. Its performance is quite strong, especially compared to the regular lasso method. Nearly every region of spectral density has at least one peak present, although some small extra peaks are predicted in the

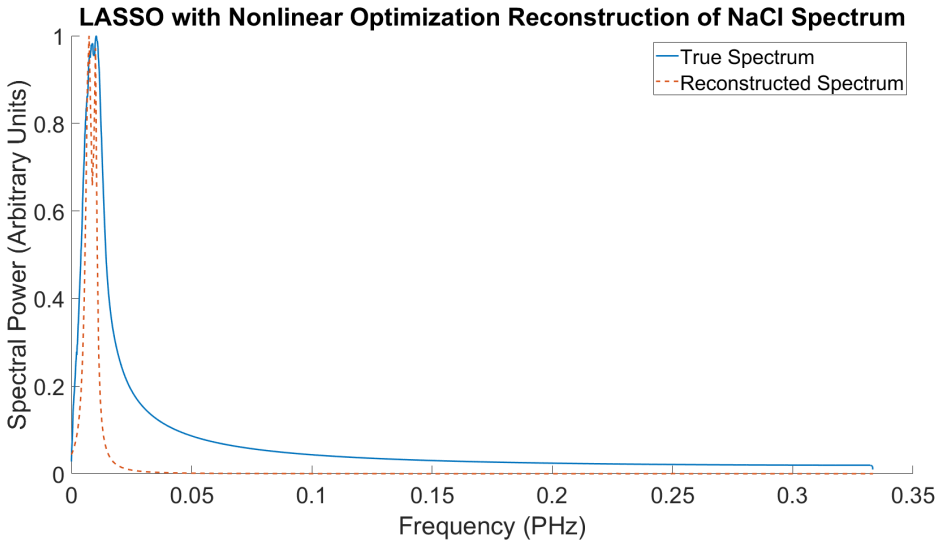

FIGURE 3.5: Reconstruction of the NaCl spectrum with the optimized lasso

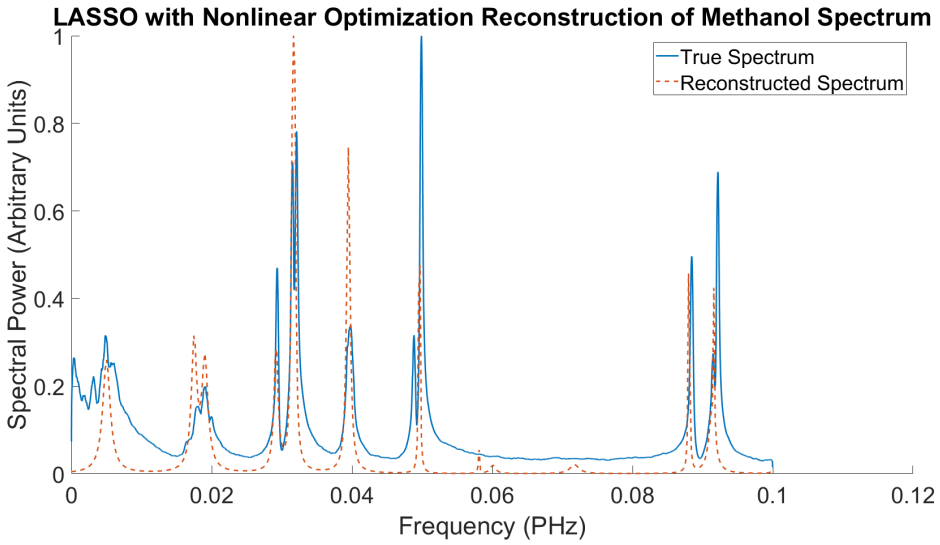

FIGURE 3.6: Reconstruction of the methanol spectrum with the optimized lasso

region from 0.06 to 0.08 PHz. Several low-frequency peaks are also absent, but many of the remaining peaks are relatively well-modeled.

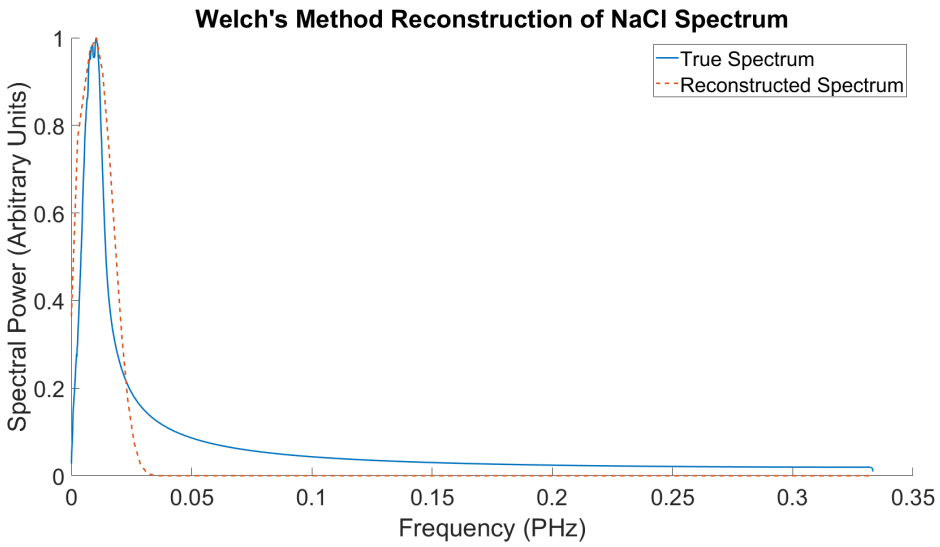

FIGURE 3.7: Reconstruction of the NaCl spectrum with Welch's Method

3.1.4 Welch's Method

Figure 3.7 shows Welch's method applied to the NaCl spectrum. The qualitative behavior appears to match the true spectrum nearly perfectly. Figure 3.8 displays the methanol spectrum reconstructed with Welch's method. In this figure, it is evident that Welch's method effectively reduces noise at the expense of resolution. All regions of spectral density are present, though their general shape appears significantly smoothed. The finer details of peaks, particularly split peaks, is absent.

3.1.5 Filter Diagonalization Method (FDM)

Figure 3.9 shows the reconstructed NaCl spectrum using the modified filter diagonalization method. Like the lasso models, its dictionary of functions only includes symmetric Lorentzians, meaning the asymmetric decay of the system is not represented in the spectral approximation. With that said, the approximation is quite strong, with the single peak correctly identified. Figure 3.10 shows the performance of the modified FDM applied to the methanol spectrum.

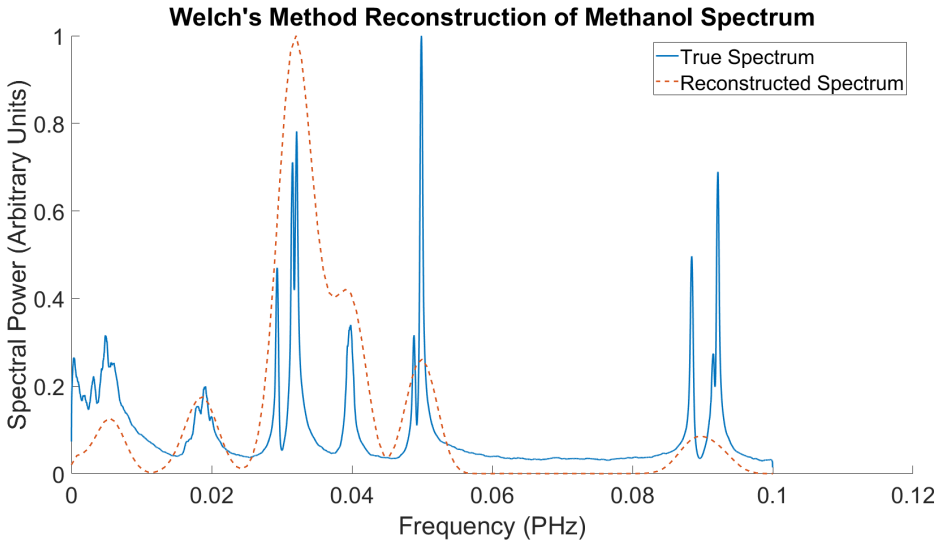

FIGURE 3.8: Reconstruction of the methanol spectrum with Welch's Method

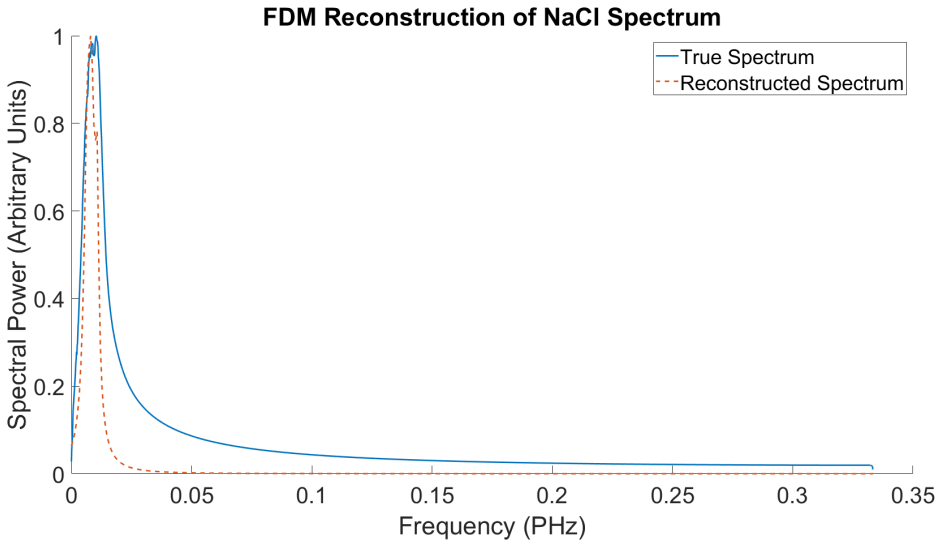

FIGURE 3.9: Reconstruction of the NaCl spectrum with the modified FDM

The performance is generally strong, with most peaks successfully identified. No excess peaks are included in the spectrum, as in the other lasso methods. The peaks at low-frequency are poorly modeled, with only the general behavior captured.

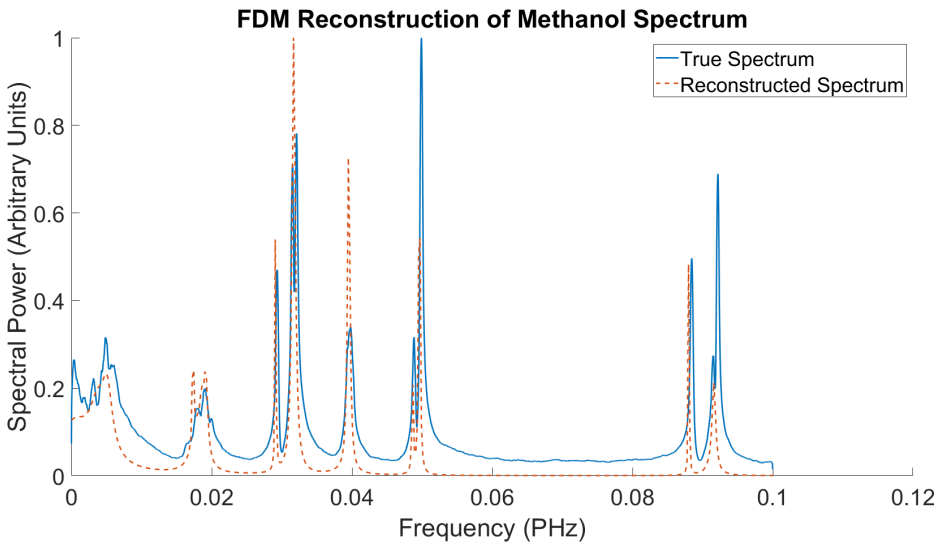

FIGURE 3.10: Reconstruction of the methanol spectrum with the modified FDM

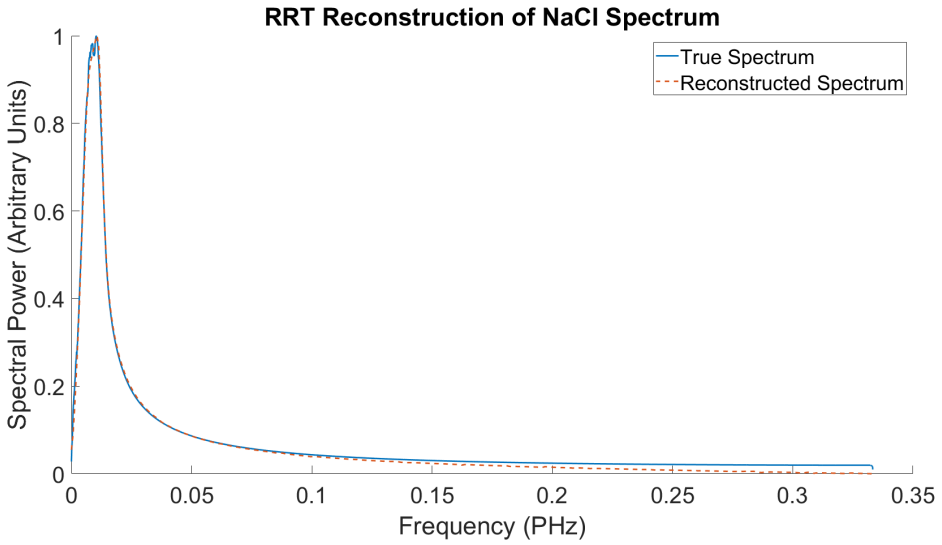

FIGURE 3.11: Reconstruction of the NaCl spectrum with the RRT

3.1.6 Regularized Resolvent Transform (RRT)

Figure 3.11 displays the NaCl spectrum reconstructed with the RRT. The true behavior is nearly perfectly captured with this method. This makes sense, given that the RRT should approximate the infinite time DFT for a truly Lorentzian signal. Figure 3.12 shows the performance of the RRT when applied to the methanol spectrum. Many of the fine details of the

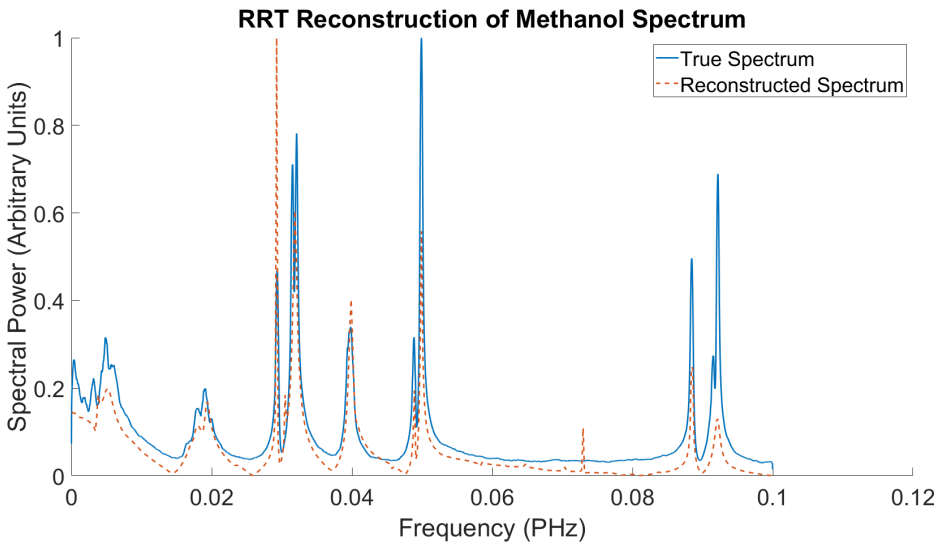

FIGURE 3.12: Reconstruction of the methanol spectrum with the RRT

system, including the three peaks present around 0.02 PHz, are captured by the RRT. Like the modified FDM, the low-frequency peaks are not included in the output. Like the lasso methods, extra peaks are present in the system in the region 0.06 to 0.08 PHz which do not appear in the true methanol spectrum.

3.2 Comparison of Method Performance

To compare spectral estimation performance, two metrics are used: the false positive rate (FPR) and false negative rate (FNR). These describe the percentage of false peaks identified by a method, and the percentage of true peaks missed by a method, respectively.

To compute these metrics, the signals output by the estimation methods are examined with the *findpeaks* command in MATLAB. To put all of these methods on an equal playing field, the minimum peak prominence is set at 1% of the difference between the largest peak and lowest trough of each output spectra. This threshold is ultimately arbitrary but treats all methods

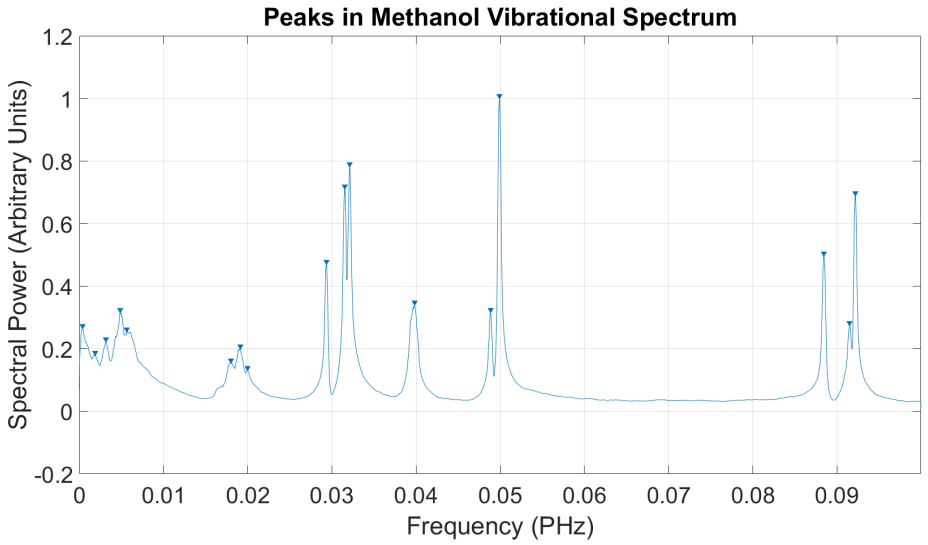

FIGURE 3.13: Converged methanol spectrum with peaks identified

equally. Figure 3.13 shows the seventeen peaks present in the converged methanol spectrum. Given the simplicity of the sodium chloride spectrum and the relatively high rate of recovery using all methods, this spectrum is excluded from comparisons.

As no method will find exactly the correct location for a given peak, another threshold must be imposed to describe a peak that is 'close enough.' This threshold is set at the smallest difference between peak locations in the converged spectrum. In addition, for a peak in the test spectrum to be counted as corresponding to a specific peak in the true spectrum, the test peak must be closer to the specific true peak than to any other true peak. Test peaks cannot be counted for multiple true peaks.

A number of arbitrary parameters must be set for the simulation, though the effects of these parameters will be explored in the next section. These include the length of the velocity interval under examination (200 points) number of velocity points sampled (100 points), the number of trials for each method (50), the windowing configuration for FDM (eigenvalues from 1 window

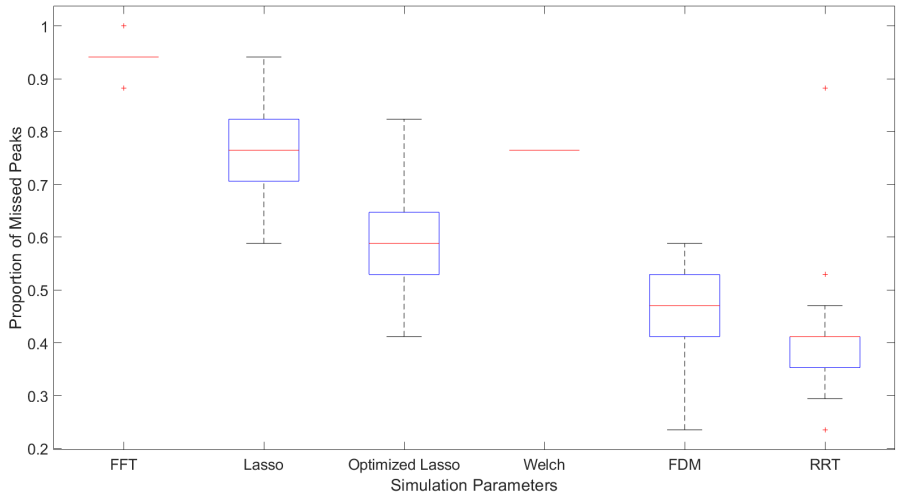

FIGURE 3.14: False negative rate of various methods applied to methanol spectrum

through 10 windows included), and the presence of weighting in the FDM method and the lasso methods (present.)

To compare the FPR and FNR of two methods, two-sample t-tests are performed with a significance of $\alpha = 0.05$ for consistency.

Figure 3.14 shows a box plot of the FNR of the various methods, while Figure 3.15 shows the FPR. Every visible difference between two methods on both plots is statistically significant, except for the difference in FNR of Welch's Method and the lasso ($p = 0.7$).

The two methods with the highest performance are the modified FDM and the RRT, with FDM having the lowest FPR of any method, and RRT having the lowest FNR. The false negative rates are, in general, quite high, likely due to the number of close peaks in the true methanol spectrum.

Every method tested outperforms the FFT, which is the current standard method used to extract spectra from MD simulations. This indicates significant room for improvement, as

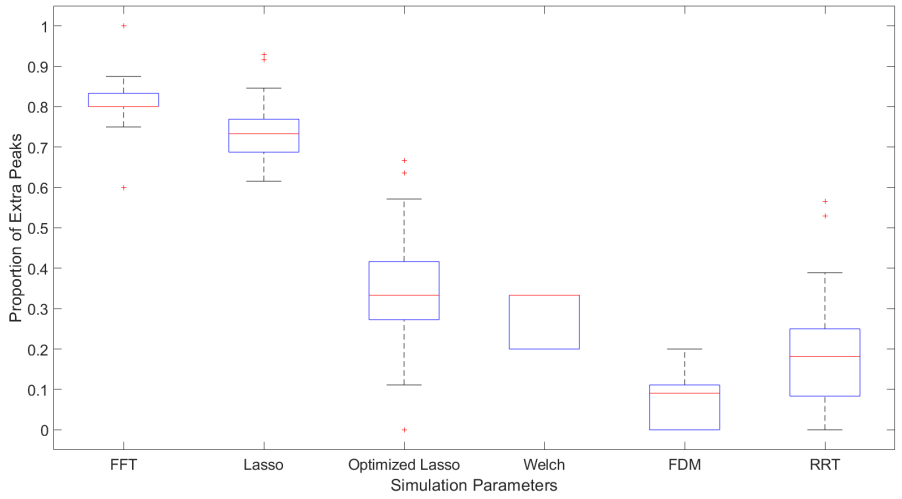

FIGURE 3.15: False positive rate of various methods applied to methanol spectrum

alternative methods could produce more precise results with fewer points.

Welch's method produces results worthy of discussion: it produces the third lowest FNR of any method, but ties for fifth in FPR. This likely occurs because Welch's method produces very smooth peaks, which would ensure that it would miss many peaks. The tolerance for Welch's method for noisy data makes it unlikely to produce erroneous peaks due to noise.

The use of nonlinear optimization in the lasso significantly improves its performance in both categories. With that said, neither method performs particularly well, especially given their relatively high computational cost.

3.3 Effects of Model Parameters and Modifications

Due to the computational expense and comparatively poor performance of the lasso and the lasso with nonlinear optimization, these methods will be left out most of the following comparisons. Instead, the remaining four methods are examined to see the sensitivity of their

performance to various factors, including the size of the sample, the rate of subsampling prior to the calculation of the autocorrelation function, and, in the case of modified FDM, the presence of weights and the windowing configuration of the signal. As before, 50 simulations are run to compute the average performance of each method.

The single worthwhile comparison of the lasso methods concerns whether or not nonlinear optimization led to higher performance than a more densely sampled grid of dictionary functions, which is discussed in section 3.3.5.

3.3.1 Sample Size

The first factor to be considered is how sample size affects the performance of the methods. To see this, the sample size is varied while keeping the subsampling rate constant. That is, the size of the interval being sampled also increases. Figure 3.16 shows how the sample size affects the false positive rate of the four methods under consideration, while Figure 3.17 shows how the false negative rate is affected.

Most of the apparent differences on the box plots are statistically significant, with a few exceptions. We cannot conclude that the mean FPR of the modified FDM is different when 100 points are used versus when 200 points are used ($p = 0.18$). Similarly, the mean FPR of the RRT with 100 points is not significantly different from the mean when 50 points are used, or 200 points are used.

The remaining differences are significant. For most methods, an increase in sample size shows a decrease in the FNR, which makes intuitive sense. The exception is the FFT, which shows a decrease in accuracy both as the sample size is decreased and is increased. There is not

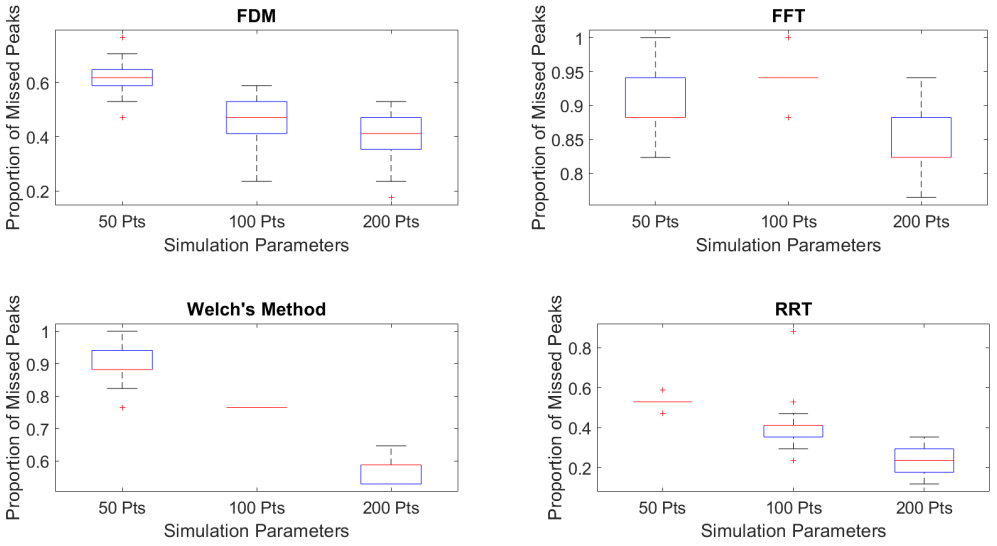

FIGURE 3.16: Effect of sample size on FNR; note that each subplot has a different scale on the y-axis.

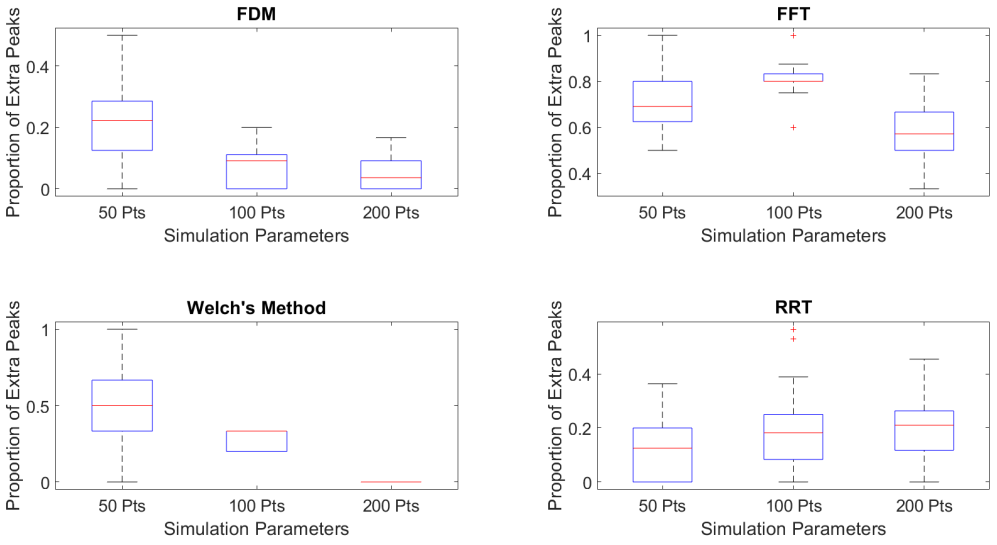

FIGURE 3.17: Effect of sample size on FPR; note that each subplot has a different scale on the y-axis.

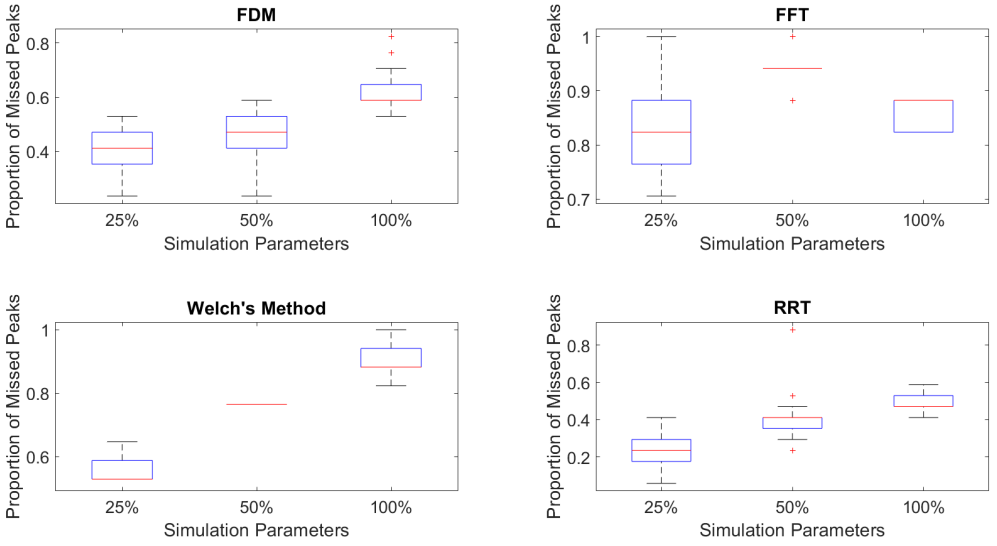

FIGURE 3.18: Effect of subsampling rate size on FNR; note that each subplot has a different scale on the y-axis.

a clear explanation for this effect. Simulations of the FFT with these various sample sizes show that the FFT acting on 50 data points tends to capture the spectral behavior around 0.02 PHz, while this behavior is absent in the 100 data point simulations. Given how few peaks the FFT correctly finds, this peak is likely the culprit. The reason the 0.02 PHz features are absent in the 100-point FFT is unclear, however. Further increase in the sample size appears to show a continual drop in both FPR and FNR.

3.3.2 Rate of Subsampling

The next parameter worth considering is the rate of subsampling. For this set of experiments, the number of samples is held constant while the interval length is modified. Figure 3.18 shows how the false negative rates of the methods change, and Figure 3.19 shows how the false positive rates change.

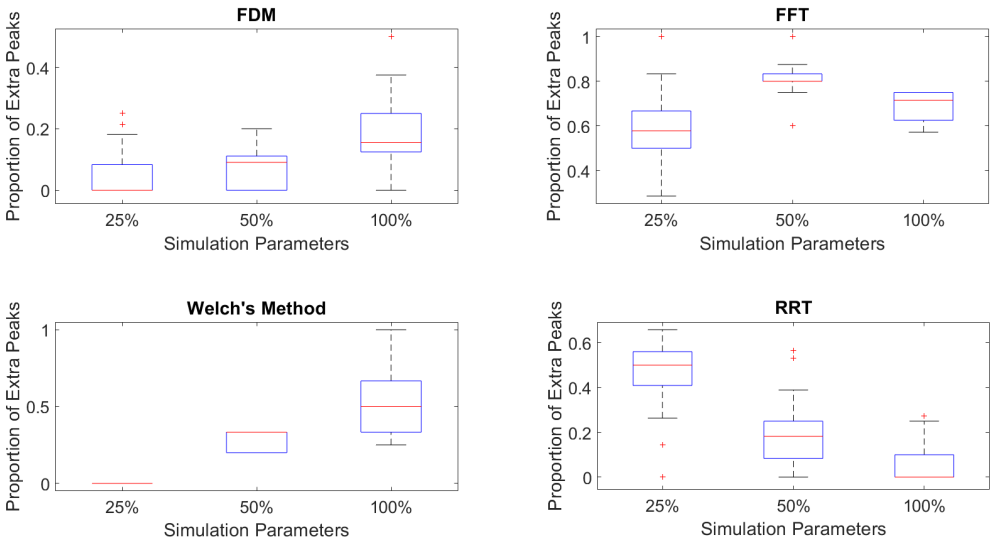

FIGURE 3.19: Effect of subsampling rate on FPR; note that each subplot has a different scale on the y-axis.

In this case, every difference is significant to $\alpha = 0.05$ except for the mean difference in FPR between the modified FDM method with 50% subsampling and with 25% subsampling ($p = 0.22$).

As in the case of sample size, the FFT performs better in both metrics if the subsampling rate is increased to 100% or decreased to 50%. Once again, this appears to stem from the disappearance of the 0.02 PHz peaks at 50% subsampling.

For the FDM and Welch's Method, an increase in the rate of subsampling generally decreases both FNR and FPR. This may not be intuitive at first, but recall that an increase in subsampling rate decreases the interval length. Thus, this indicates that a longer interval length, even one more sparsely sampled, will generally produce a more precise spectral approximation for most methods.

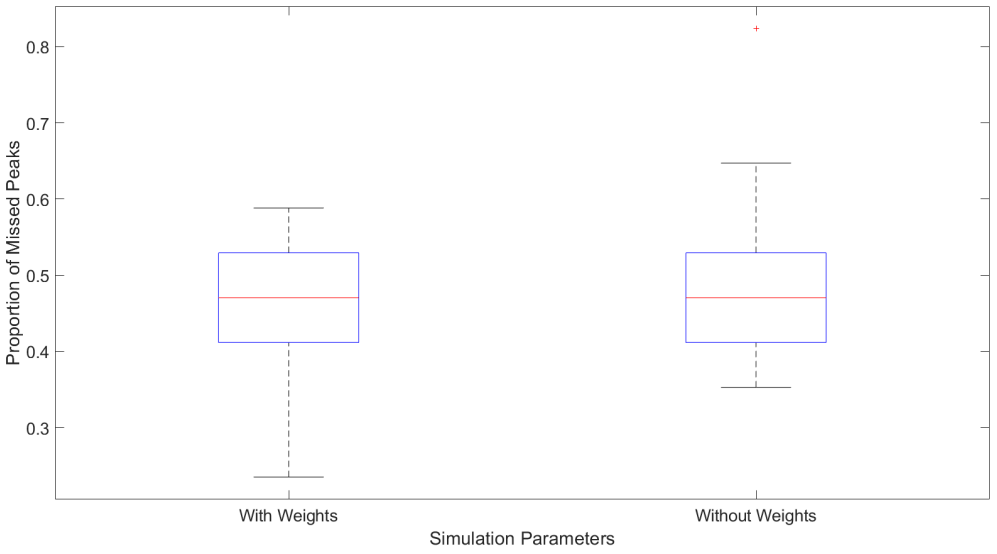

FIGURE 3.20: Effect of weighting on FNR of FDM

The RRT has a strange trend: FNR and FPR are affected in opposite ways as the subsampling rate is changed. As the rate of subsampling increases, the rate of missing peaks decreases while the rate of extra peaks decreases. This indicates a possible sensitivity of the RRT to noise that results from subsampling. This property is detrimental to its performance, as both metrics cannot be simultaneously improved by altering the windowing configuration.

3.3.3 Weighting

For the FDM, the lasso, and the optimized lasso, the observations in the autocorrelation function are weighted by the square root of the number of observations, as this should roughly indicate their certainty. As the lasso methods are generally ineffective, only the FDM is examined. Figure 3.20 shows how the FNR of the FDM changes as weighting is removed, and Figure 3.21 displays how the FPR changes.

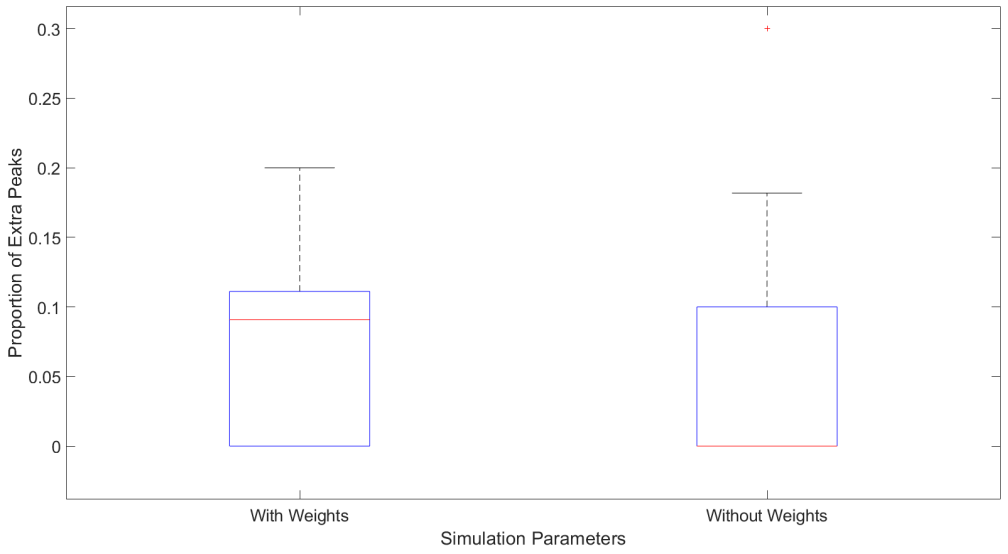

FIGURE 3.21: Effect of weighting on FPR of FDM

Though there are apparent differences in the mean FNR and FPR, these are not statistically significant, both with p-values around 10%. Thus, weighting does not have a statistically significant impact effect on the performance of the FDM.

3.3.4 FDM Windowing Configuration

The FDM used in this work is modified by using multiple windowing schemes in the frequency domain. Instead of doing just one window, as in the case of Krylov basis diagonalization, or a single set of multiple windows, as in traditional FDM, this method uses a range of overlapping window counts, with all resulting eigenvalues combined and fit with the lasso. To test whether this modification is effective, a number of configurations are examined: a single window, concatenation of all window counts between 1 and 5, a concatenation of all window counts between 1 and 20, a concatenation of all window counts between 1 and 20, 5 windows, 10 windows, and a concatenation of all window counts between 5 and 10. The FNR and FPR for these

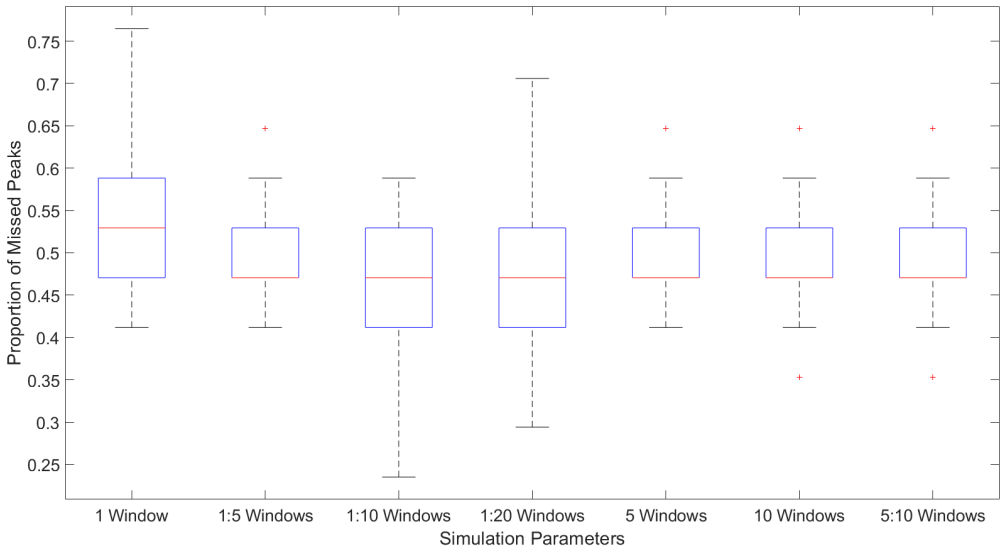

FIGURE 3.22: Effect of window configuration on FNR of FDM

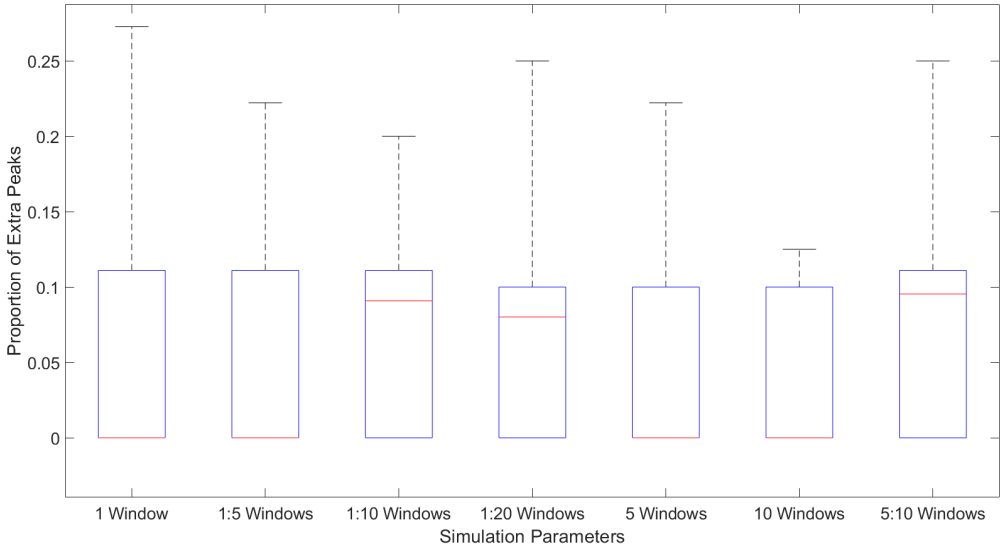

FIGURE 3.23: Effect of window configuration on FPR of FDM

configurations are shown in Figure 3.22 and Figure 3.23 respectively.

Unlike in previous cases, only a few of the differences are statistically significant. The 1:10 configuration has a higher FPR than the 5 and 10 window configurations, and a lower FNR than the 1 window, 5 window, 10 window, 1:5, and 5:10 configurations. Lastly, the single window

configuration has a higher FPR than the 10 window configuration.

From these observations, it is clear that the concatenation of multiple windowing configurations can help reduce the FNR to a point, but runs into diminishing returns. This last part is seen by the 1:20 configuration not having a statistically significant difference in FNR from the 1:10 configuration. While the FNR is reduced, the FPR is increased. This makes sense, as more basis functions are introduced. It is more likely that an incorrect function will survive the lasso, but the more correct functions will also survive. Thus, it is reasonable to use some concatenated windowing to expand the basis of Lorentzian functions.

3.3.5 Lasso Nonlinear Optimization and Grid Density

The nonlinear optimization is introduced into the lasso method to reduce the need for an extremely fine grid of dictionary functions. To determine if this is effective, the two lasso methods are run with a variety of grid sizes. The two numbers presented for the grid size describe the size of the width grid and the center grid. Figure 3.24 shows the FNR of these simulations, and Figure 3.25 shows the FPR.

Increasing the grid density for the lasso method initially decreases both the FNR and the FPR significantly, but the FPR of the finest grid (50/80) is not statistically different from the original 30/50 grid. This is likely due to excess functions being inserted into the solution. The optimized lasso significantly outperforms the regular lasso on all three grid sizes in terms of FPR. However, it is comparable to the finest lasso grid in terms of FNR ($p = 0.62$.) Therefore, the optimization step not only produces FNR values characteristic of a higher grid density, it also reduces the occurrence of false positives that occur as the dictionary size increases. Thus, nonlinear optimization offers a promising way to make the lasso more useful.

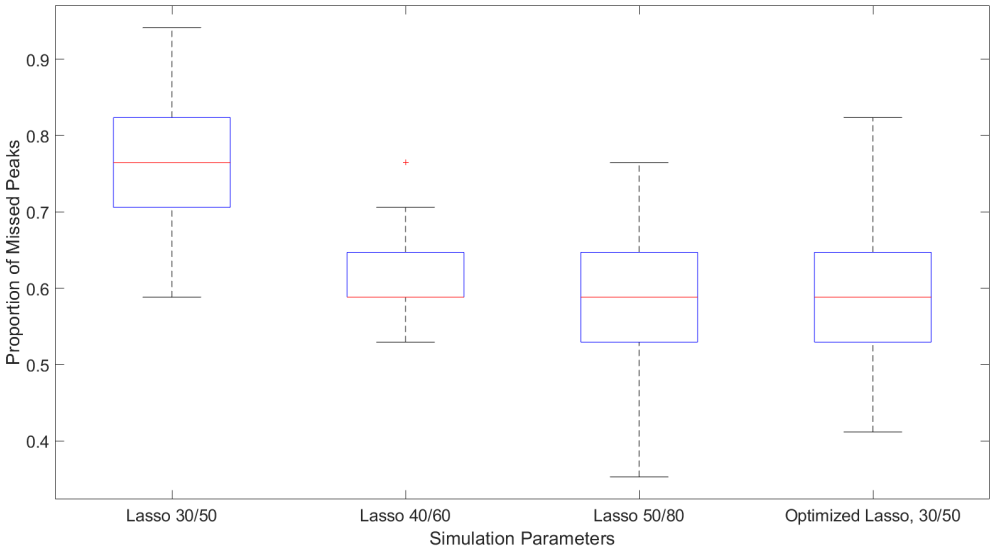

FIGURE 3.24: Effect of nonlinear optimization and grid size on FNR. The two numbers represent the grid sizes for width/center

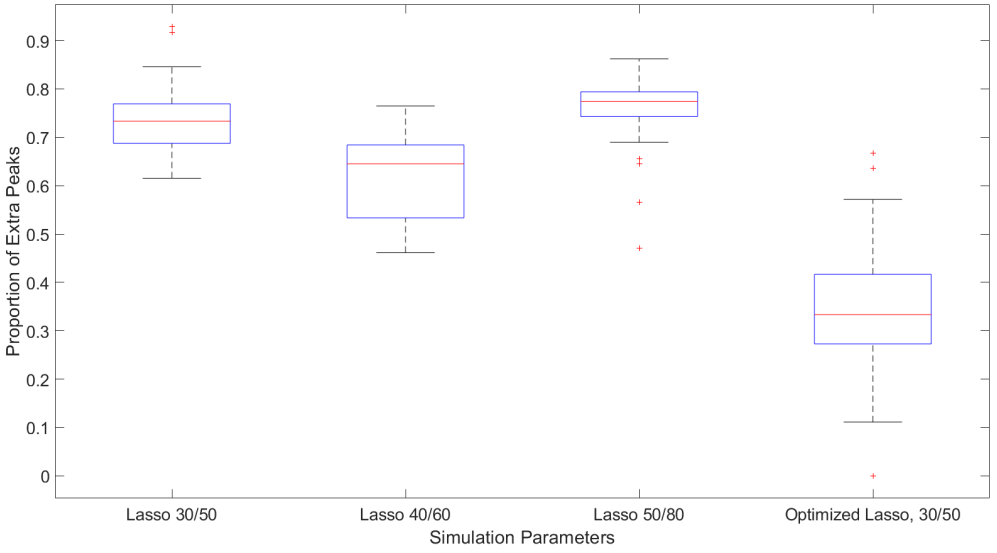

FIGURE 3.25: Effect of nonlinear optimization and grid size on FPR. The two numbers represent the grid sizes for width/center

Chapter 4

Discussion

4.1 Sources of Error

Given the defined scope of the problem, only a few aspects of this work offer the possibility of significant error. Most notably, this work relies heavily on the assumption that a method effective for methanol and sodium chloride MD simulations will be applicable to more complicated systems. It is likely that in more complicated systems, these methods will have to be modified or used with caution to ensure accurate results. Future work should be done on adapting the most successful techniques, the modified FDM and the RRT, to more complicated MD simulations.

Another possible source of error comes from the statistical techniques employed. A significance level, α is used to judge the difference between data sets. Crudely, as more than 20 t-tests are performed, it is possible that at least one null hypothesis is incorrectly rejected, and two sets of data are erroneously classified as different. This is a relatively minor source of error, as a

simple repetition of the simulations could substantiate the statistical results. Alternatively, the α value used could be adjusted to account for the multiple tests performed, as is done with the Bonferroni Correction [17]. In addition, the t-tests performed assume that the populations have the same variance, which is certainly incorrect based on the box plots in the Results chapter.

To effectively compare the data sets, several arbitrary thresholds are defined, including the minimum distance to define a peak and the minimum peak prominence. These standards are universally applied, reducing the likelihood of error; however, it would be prudent to examine how changing these thresholds might affect the rates of error. In addition, in running a large series of simulations, it is necessary to process the data en masse, rather than individually. However, this precludes conclusions that might naturally come from analyzing each trial individually. For example, suppose there are two peaks in the true spectrum separated by a small distance. If the recovery technique predicts two peaks with a slight shift, one peak from the true spectrum might line up with the other peak in the predicted spectrum. Then, only one peak might be counted as being successfully found, even though the qualitative behavior of the system is accurate.

A significant limitation to these conclusions is the small parameter space in which these methods are tested. Only a few sample sizes, subsampling rates, etc. are examined in the Results chapter, leaving open the possibility that, in other regimes, the relative performance of certain methods might flip. Further study should be done on other systems with other parameter regimes to test the consistency of the conclusions made here.

4.2 Implications of Results and Recommendations

Notwithstanding the sources of error discussed in the previous section, the results of this study show that the discrete Fourier transform is not the most effective means of constructing spectra from MD simulations. Instead, the methods with the highest performance are the regularized resolvent transform, and the filter diagonalization method with novel modifications. Depending on the application, one of these methods might be preferable over the other. FDM tends to miss more peaks than the RRT, but produces a spectrum with fewer extraneous peaks than the RRT. Using one of these methods instead of the DFT could significantly reduce the number of points necessary for a spectral approximation, leading to a speedup in MD spectroscopy.

Chapter 5

Conclusions and Future Work

In this work, the efficient recovery of chemical spectra from molecular simulation data is examined. Multiple techniques are attempted: some common, some specialized, and some novel or heavily modified. The success of these techniques is judged by their effective identification of key peaks in the spectra of NaCl and methanol. Ultimately, the RRT and the modified FDM have the strongest performance, reducing the number of points necessary to produce an accurate spectrum compared to the standard FFT. The results produced by these methods with 100 points is comparable to the results produced by the FFT with an order of magnitude more points, meaning that the computational expense could be reduced by a factor of 10 or more. The modifications to the FDM allow the method to be applied to this system, whereas the regular method suffers from significant instability to small noise.

To continue this work, these methods should be applied to more complicated MD simulations, especially systems in which classical MD is not sufficient to describe molecular interactions. This might result in a deviation from the Lorentzian pattern assumed in this work, which might

reduce the effectiveness of the highest-performing methods. In addition, the application to other systems will allow for study of the optimal parameters to use in running these techniques.

Bibliography

[1] Candès, E. J., and Fernandez-Granda, C. Towards a mathematical theory of super-resolution. *Communications on pure and applied Mathematics 67*, 6 (2014), 906–956.

[2] Henssge, E., Dumont, D., Fischer, D., and Bougeard, D. Analysis of infrared and Raman spectra calculated by molecular dynamics. *Journal of Molecular Structure 482-483* (1999), 491–496.

[3] Hiereth, M. Lifetime broadening. `http://www.pci.tu-bs.de/aggericke/PC4e_osv/Spectroscopy050119/node7.html`, 2005.

[4] in 't Veld, P., Plimpton, S., and Grest, G. Accurate and efficient methods for modeling colloidal mixtures in an explicit solvent using molecular dynamics. *Comp. Phys. Comm. 179* (2008), 320–329.

[5] James, G., Witten, D., Hastie, T., and Tibshirani, R. *An introduction to statistical learning*, vol. 112. Springer, 2013.

[6] Lamberti, V. E., Fosdick, L. D., Jessup, E. R., and Schauble, C. J. A hands-on introduction to molecular dynamics. *Journal of chemical education 79*, 5 (2002), 601.

[7] Luber, S., Iannuzzi, M., and Hutter, J. Raman spectra from ab initio molecular dynamics and its application to liquid S-methyloxirane. *The Journal of Chemical Physics 141*, 9 (2014), 094503.

[8] Mandelshtam, V. A. Cheminform abstract: FDM: The filter diagonalization method for data processing in NMR experiments. *ChemInform 32*, 20 (2001).

[9] MathWorks. Matlab 2018a, 2018.

[10] Owens, R. The discrete Fourier transform. `http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT4/node3.html`, 1997.

[11] Rudi Winter, W. Phonon spectroscopy :: Condensed matter physics :: Rudi winter's web space. `http://users.aber.ac.uk/ruw/teach/334/qns-ir.php`, 2015.

[12] Thomas, M., Brehm, M., Fligg, R., Vhringer, P., and Kirchner, B. Computing vibrational spectra from ab initio molecular dynamics. *Physical Chemistry Chemical Physics 15*, 18 (2013), 6608.

[13] Tibshirani, R., Wainwright, M., and Hastie, T. *Statistical learning with sparsity: the lasso and generalizations*. Chapman and Hall/CRC, 2015.

[14] Weisstein, E. W. Fourier transform–lorentzian function. *From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/FourierTransformLorentzianFunction.html*.

[15] Weisstein, E. W. Leakage. *From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/Leakage.html*.

[16] Weisstein, E. W. Discrete fourier transform. *From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/DiscreteFourierTransform.html* (2002).

[17] Weisstein, E. W. Bonferroni correction. *From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/BonferroniCorrection.html* (2004).

[18] Weisstein, E. W. Autocorrelation. *From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/Autocorrelation.html* (2005).

[19] Weisstein, E. W. Wiener-khinchin theorem. *From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/Wiener-KhinchinTheorem.html* (2006).

[20] Welch, P. The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics 15*, 2 (1967), 70–73.