DYNAMICAL SYSTEMS ANALYSIS USING TOPOLOGICAL SIGNAL PROCESSING

By

Audun Myers

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Mechanical Engineering – Doctor of Philosophy

2022

ABSTRACT

DYNAMICAL SYSTEMS ANALYSIS USING TOPOLOGICAL SIGNAL PROCESSING

By

Audun Myers

Topological Signal Processing (TSP) is the study of time series data through the lens of Topological Data Analysis (TDA)—a process of analyzing data through its shape. This work focuses on developing novel TSP tools for the analysis of dynamical systems. A dynamical system is a term used to broadly refer to a system whose state changes in time. These systems are formally assumed to be a continuum of states whose values are real numbers. However, real-life measurements of these systems only provide finite information from which the underlying dynamics must be gleaned. This necessitates making conclusions on the continuous structure of a dynamical system using noisy finite samples or time series. The interest often lies in capturing qualitative changes in the system's behavior known as a bifurcation through changes in the shape of the state space as one or more of the system parameters vary. Current literature on time series analysis aims to study this structure by searching for a lower-dimensional representation; however, the need for user-defined inputs, the sensitivity of these inputs to noise, and the expensive computational effort limit the usability of available knowledge especially for in-situ signal processing.

This research aims to use and develop TSP tools to extract useful information about the underlying dynamical system's structure. The first research direction investigates the use of sublevel set persistence—a form of persistent homology from TDA—for signal processing with applications including parameter estimation of a damped oscillator and signal complexity measures to detect bifurcations. The second research direction applies TDA to complex networks to investigate how the topology of such complex networks corresponds to the state space structure. We show how TSP applied to complex networks can be used to detect changes in signal complexity including chaotic compared to periodic dynamics in a noise-contaminated signal. The last research direction focuses on the topological analysis of dynamical networks. A dynamical network is a graph whose vertices and edges have state values driven by a highly interconnected dynamical system. We show how zigzag persistence—a modification of persistent homology—can be used to understand the changing structure of such dynamical networks.

ACKNOWLEDGEMENTS

I would foremost like to express my gratitude to my advisor Dr. Firas Khasawneh for the thoughtful guidance throughout my education. He has shown me how to conduct research in a robust and thorough way. I would to thank my family who have always supported me through my Ph.D by being inquisitive about my research and appreciating the work I am doing. I would like to also thank my collaborators for giving me insights into other fields of research and broadening my intellectual horizons. Lastly, this work would not be possible without generous support from Michigan State University, the National Science Foundation, and the Air Force Office of Scientific Research.

PREFACE

Dynamical systems is a term used to broadly refer to systems whose state changes in time. These systems are formally assumed to be a continuum of states whose values are real numbers. However, real-life measurements of systems only provide finite information from which the underlying dynamics must be gleaned. This necessitates making conclusions on the continuous structure of a dynamical system using noisy finite samples or time series. The interest often lies in capturing qualitative changes in the system behavior as one or more of the system parameters vary. For example, a shift in surface pressure characteristics on airfoils as a function of the angle of attack from regular to aperiodic can indicate significant loss of lift and possibly stall conditions. Recent advances in sensor technology and computer hardware has also led to a shift towards data-driven analysis and modeling of engineered and natural systems. The datasets are obtained through either numerical simulations or experiments and often contain complex dynamics hidden in some high-dimensional structure. Current literature on time series analysis aims to study this structure by searching for a lower dimensional representation; however, the need for user-defined inputs, the sensitivity of these inputs to error, and the expensive computational effort limit the usability of available knowledge, especially for in-situ signal analysis. Additionally, many current TSA methods are sensitive to additive noise, which is common in experimental data.

An emerging collection of tools breathing new life into this discipline is the nascent field of Topological Signal Processing (TSP), which leverages the power of Topological Data Analysis (TDA) [157] for analyzing complex signals [42,147,159,160,163,175–177,202,229,230,233,243, 246]. Some of the attractive features of using TDA for signal processing include its noise-robustness, compact visualization tools, and conduciveness to machine learning. Therefore, enriching signal processing using TDA has the potential to reveal information that is currently not possible by existing, standard dynamic systems methods. There has been exciting preliminary results in this field including showing empirically that the novel tools created have potential to revolutionize the field. However, despite the success shown in prior works, the fundamental science that connects

TDA to the underlying dynamic systems theory remains largely unexplored. In my work I am focused on four main chapters: (1) Implementing sublevel set persistence for parameter estimation and time series analysis, (2) choosing optimal parameters for both state space reconstruction and permutations to be used for topological signal processing, (3) the persistent homology of complex networks, and (4) applying novel tools from TDA for analyzing dynamical networks. Each of these will be introduced in the following paragraphs. In each chapter a thorough introduction to each subject is provided.

The first chapter of my research is based in sublevel set persistence of single variable time series—a tool from TDA that can be applied to the time series directly. The goal of this chapter is to use the sublevel set persistence for directly estimating damping parameters of the underlying one-dimensional oscillator from the positional output time series. While sublevel set persistence is robust to additive noise, it does have noise artifacts that need to be accounted for to accurately estimate system parameters from the signal. Therefor, my first contribution to this field was to develop a statistical analysis of the resulting persistence to separate out the significant features which hold information about the damping characteristics and parameter values of the underlying oscillator.The third contribution of this chapter is my development of methods for calculating time series complexity using sublevel set persistence and information theory. These complexity measures are shown to provide an avenue for bifurcation detection through an increased complexity of the signal's sublevel set entropy.

The second chapter of my research studies parameter selection techniques for both state space reconstruction and permutation formations. The two parameters needed are the dimension n and delay τ. Both permutations and state spaace reconstruction are vital prerequisite data processing techniques used to apply TDA to study a signal in the next chapter. In this chapter we also develop novel parameter selection methods based on a topological analysis of the data through both reconstructions from sliding windows and sublevel set persistence.

In contrast to classical tools for representing time series as a point cloud, chapter three of my work studies network representations of the underlying dynamics. One of the advantages of this approach is that the size of the representation can be better controlled as a finite set, and I can leverage graph theory to research faster methods for quantifying the topology based on the resulting network. However, the representation of time series as a graph—especially in the presence of noise—is a largely open field of research, and efficient TDA computation on the resulting graphs is still in need of a solid mathematical footing. For example, questions related to the optimal parameter choices of the representation, types of detectable bifurcations, and mathematical guarantees that govern successful bifurcation identification are all wide open. Many of these optimal parameters are associated with the parameters of both permutation entropy as a time series information measure and state space reconstruction. In chapter two (section 2) I introduce information theory and specifically permutation entropy with some of the most successful optimal parameter estimation methods for time series as well as develop several novel methods based on tools from TDA. This initial research provided the needed foundations for many of the network representation tools I later use in chapter three. I also contribute to the field of complex network analysis through TDA by investigating methods for implementing weight information and complex network formation methods that best perform for the dynamic state analysis task.

The fourth chapter of research in Section 4 focuses on novel applications of topological data analysis for studying interconnected dynamical systems represented as temporal graphs using topological data analysis. In this chapter I first show how a transportation system as a dynamical system can be represented as a temporal graph. I then develop a framework for applying zigzag persistence to detect structural changes in the temporal graph over time. This is done using zigzag persistence. I make comparisons to the persistence diagram results using standard shape summary statistics from graph theory literature. In this chapter I also develop a method for the analysis of complex dynamical systems using temporal graphs when only a one-dimensional signal is available. This is done using a sliding window approach with each window represented as a complex network (e.g., the ordinal partition network). I then show how zigzag persistence can be used to study the changing structure of these graphs to detect changes in the signal and underlying dynamical system. Specifically, I show how periodic and chaotic windows can be detected for the Lorenz system exhibiting intermittency dynamic (i.e., irregular transitions from a regular to chaotic state).

I have also included a fifth auxiliary chapter, which describes the experimental data sets and software developed in my research. Specifically, two main experimental data sets are used. The firth is the magnetic pendulum which has transitions from periodic to chaotic dynamics with a change in base excitation frequency and amplitude making it useful for testing TSP methods used for characterizing the dynamic state of a system. The second data set is from a double pendulum tracked using a high-speed camera [165]. Many of the methods developed through my research are programmed into the TSP python software teaspoon. The various available modules for teaspoon are discussed in Section 5.2.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

SUBLEVEL SET PERSISTENCE FOR TIME SERIES ANALYSIS

This chapter overviews my work on studying how sublevel set persistence, a tool from topological data analysis, can be leveraged for signal processing. The first application is to estimate damping parameters of a single degree of freedom system with a noisy time series as an input. The second application is for bifurcation and signal complexity analysis. First, in Section 1.1 I introduce sublevel set persistence and the novel and computationally efficient algorithm for applying it to one-dimensional signals, Section 1.2 develops the statistical analysis to separate sublevel sets associated to noise from signal is developed, Section 1.3 shows how sublevel set persistence can be leveraged for damping parameter estimation, and in section 1.4 I apply sublevel set persistence to signals for complexity and bifurcation detection.

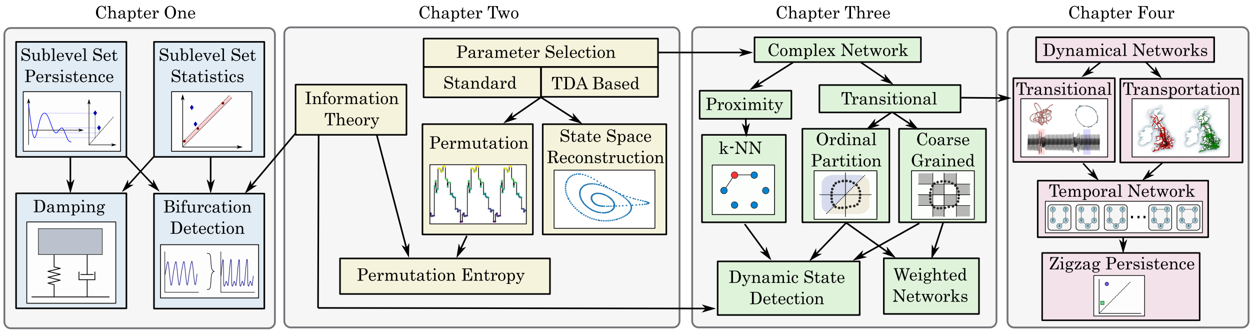


Figure 1.1: Overview of research chapters with past, current, and future works.

r

1.1 Sublevel Set Persistence

I now provide a basic introduction to sublevel set persistence so the reader has a sufficient understanding of the method. Let us begin with the single variable function $f : \mathbb{R} \to \mathbb{R}$. Given $r \in \mathbb{R}$, I define the *sublevel set below* r as $f^{-1}(-\infty, r]$. As the filtration parameter r increases, the sublevel sets may grow but remain the same (up to homology) until a local extrema (i.e., a local minimum or maximum) is reached. If the extrema is a local minima, then a new set is born at r_B; I label that set with the value r_B. On the other hand, if the extrema is a local maxima, two previously-existing sets are combined. If the two sets were labeled r_B and r'_B, with $r_B \leq r'_B$ and the maximum attained at r_D, then, by the Elder Rule [70, p. 150], I say that the component born at r'_B dies going into r_D. The pair (r'_B, r_D) is called a persistence pair. As r ranges from $-\infty$ to ∞, the persistence diagram is the collection of all n such pairs, $\text{dgm}f = \{(b_i, d_i)\}_{i=1}^n$. Any unpaired births are called essential classes and are paired with a death coordinate of ∞; thus, $\text{dgm}f$ is embedded in the extended plane $\overline{\mathbb{R}}^2$. The lifetime or persistence of a point $(b_i, d_i) \in \text{dgm}f$ is defined as $\ell_i = d_i - b_i$. In this work, the functions are only sampled on a finite domain, with the first sample at time t_a and the last sample at time t_b. I obtain a continuous function over $[t_a, t_b]$ by using a piecewise linear interpolation between consecutive samples, and extending the function to $\pm\infty$ by extending the first (resp., last) edges to rays. Doing so allows us to define a persistence diagram that does not have critical points on the boundary of the time series. As such, I study the persistence points where both coordinates are finite, and omit persistence points that contain an unbounded coordinate.

To demonstrate persistence diagrams and sublevel set persistence, I demonstrate a simple example for the function shown in 1.2. This function has thirteen sample points, two local minima, and two local maxima. The lowest critical value of the function occurs at height v_0. For all $r < v_0$, $f^{-1}(-\infty, r]$ is the ray $[f^{-1}(r), \infty)$. This connected component is labeled with $-\infty$, since it is "born" at $-\infty$. Then, at height $r = v_0$, a second connected component is born. The next topological change occurs at height $r = v_1$, where a third connected component is born. The next extrema is reached when $r = p_0$. At this extrema, the sublevel set that was born at $r = v_1$ dies, while the sublevel set born at $r = v_0$ persists based on the Elder Rule. This pair (v_1, p_0) is recorded in the

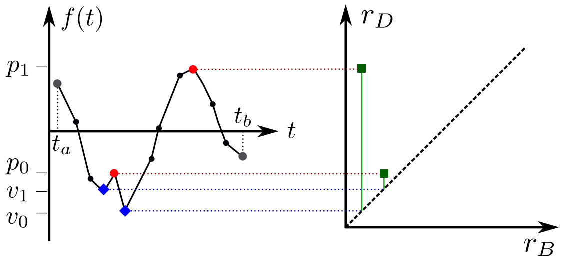


Figure 1.2: Example 0D sublevel set persistence from function $f(t)$ over finite domain $t \in [t_a, t_b]$ with the resulting persistence diagram on the right.

persistence diagram. From here, the next change happens at $r = p_1$, where the second sublevel set dies and is recorded in the persistence diagram as (v_0, p_1). Then, no further topological changes occur, but this sublevel set continues to grow as r grows. This essential class is recorded in the persistence diagram as $(-\infty, \infty)$ and is not studied in the analysis. As shown in the persistence diagram, the point (v_1, p_0) is close to the diagonal (the line $y = x$), which signifies that the sublevel set only persisted for a short range of heights (r); on the other hand, the point (v_0, p_1) is far from the diagonal, suggesting it was from a significant sublevel set.

The idea of persistence can be extended to higher dimensions allowing for the analysis of the shape of high-dimensional data sets. However, for my work, we only need to analyze the zero-dimensional features (i.e., connected components) of a one-dimensional function. A more thorough background on TDA, and persistent homology specifically, can be found in [69,157,174]. Other common ways for studying time series with a similar perspective is through merge trees or dendograms [37,46,128].

1.1.1 Sublevel Set Persistence with Additive Noise

I now investigate the stability of sublevel set persistence diagrams to additive noise for single variable functions. To illustrate the stability, I first take an example time series with additive noise as $x(t) + \epsilon$, where $x(t)$ is sampled at a uniform rate f_s and ϵ is additive noise from the noise model $\mathcal{N}$. An example of a persistence diagram from the time series with additive noise $\mathrm{dgm}x + \epsilon$ is shown in 1.3, along with the diagram without the additive noise $\mathrm{dgm}x$. This example also demonstrates how a cutoff C_α can be used to separate the significant points in the persistence diagram and those associated to the additive noise.

This example demonstrates that the addition of noise does not have a large effect on the position of significant sublevel sets in the persistence diagram with the distances between significant points (d_1 and d_2) all being relatively small. This is no surprise due to the stability theorem of the bottleneck distance for persistence diagrams [49], where the bottleneck distance is defined as the minimum distance to match two persistence diagrams. For example, if I assume $d_1 > d_2 > d_3 > d_4$,

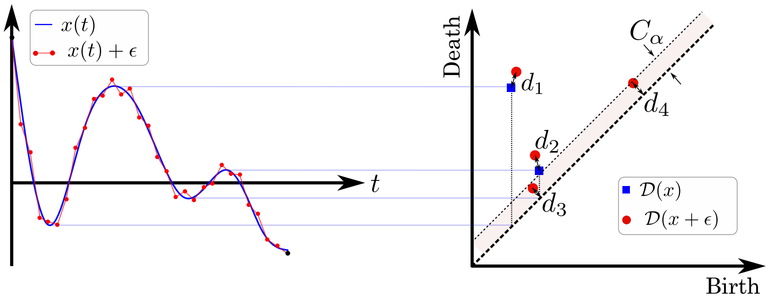


Figure 1.3: Sublevel set persistence applied to $x(t)$ of a single variable function or time series with and without additive noise ϵ from $\mathcal{N}$, shown in red and blue, respectively. This demonstrates the stability of persistent homology with the time series (left) with and without additive noise and the small effect on the resulting persistence diagrams (right). In addition, the light red region separates the significant features from those associated to additive noise.

then the bottleneck distance would be d_2. However, additive noise does introduce several points in the persistence diagram located near the diagonal with relatively small lifetimes. These noise-artifact persistence pairs are formed from the peak-valley pairs in the additive noise. This work focuses on a statistical analysis of these lifetimes to develop a method for separating the significant persistence diagram points from those of additive noise, shown in as light red region in example persistence diagram of 1.3, through a cutoff C_α with $\alpha \in [0, 1]$ as the given confidence level.

As mentioned previously, there are currently methods for developing confidence sets and associated cutoffs for persistence diagrams [40, 73]. However, these methods are specific to distance-like filtrations or require a high sampling rate. Moreover, boostrap-based techniques can be costly Additionally, methods such as persistent entropy [10] for separating noise from significant features in a persistence diagram may not properly distinguish between the noise and significant points if the number of significant data points in the persistence diagram is relatively large compared to the amount of noise. To address all of these issues, I introduce a new statistical method for developing a confidence interval and corresponding cutoff C_α.

The aforementioned statistical analysis is discussed in the proceeding sections as follows. First, in 1.2.1, I introduce my novel analysis of the statistics of the lifetimes in the persistence diagram from the sublevel set persistence of additive noise with a probability distribution $f(x)$. I then apply this analysis in 1.2.3 to several noise models commonly used or seen in real-world applications.

Following this, in 1.2.4, I introduce a method using the persistence diagram to estimate the needed distribution parameters for calculating the cutoff. Finally, in Section 1.2.5, I investigate the use of a compensation term on the distribution parameter estimation.

1.2 Statistical Analysis of Sublevel Set Persistence

1.2.1 Statistics of Additive Noise in the Persistence Diagram

Before studying a time series with additive noise, $x + \epsilon \colon \mathbb{R} \to \mathbb{R}$, I analyze the statistics of sublevel set persistence diagrams of the noise alone. Our goal is to leverage this analysis in order to generate a cutoff in the persistence diagram to separate out these noise-artifact points in the persistence diagram for $\mathcal{D}(x + \epsilon)$.

Relating Statistics Background I start with the noise, which can be thought of as a (sampled) function $\epsilon \colon \mathbb{R} \to \mathbb{R}$, where, for each $x \in \mathbb{R}$, the value $\epsilon(x)$ is a random variable sampled independently and identically distributed (iid) from some predefined noise distribution $\mathcal{N}$. In our noise model, there is no covariance structure between these random variables. The first step in developing a cutoff based on the persistence diagram statistics of additive noise $\mathcal{D}(\epsilon)$ is to determine a relationship between the descriptive additive noise distribution parameters and the distribution of the lifetimes. To do this, I develop an expression for the expected lifetime of points in $\mathcal{D}(\epsilon)$. Let $f \colon \mathbb{R} \to \mathbb{R}$ and $F \colon \mathbb{R} \to \mathbb{R}$ be the probability density function and cumulative density function of $\mathcal{N}$, respectively. Let $f_B \colon \mathbb{R} \to \mathbb{R}$ and $f_D \colon \mathbb{R} \to \mathbb{R}$ be the probability density functions for the local minima and maxima or birth and death times of the sublevel sets from $\mathcal{N}$, respectively. Let F_B and F_D be the corresponding cumulative density functions. Based on the commutative property of addition and the definition of a lifetime being the difference between the death and birth times, respectively, the expected or mean lifetime μ_L is the difference between the expected birth times $\mu_B := \mathbb{E}(B)$ and death times $\mu_D := \mathbb{E}(D)$, where B and D are the sets of birth and death values, as

$$\mu_L := \mu_D - \mu_B = \int_{-\infty}^{\infty} x \left[f_D(x) - f_B(x) \right] dx. \tag{1.1}$$

A formal proof of this relationship is provided in Theorem B.1.1 of Appendix B.1. From Eq. (1.1), I can move forward knowing that μ_L can be defined using only expressions for $f_B(x)$ and $f_D(x)$. In other words, only the distribution of birth and death times is needed, not of the lifetimes, which would require knowing how the births and deaths are paired.

I next compute the local maxima density distribution $f_D(x)$. Let $\{x_1, x_2, \ldots, x_n\} \overset{iid}{\sim} \mathcal{N}$. Ordering the samples by their index, I look at the probability of a given sample x_i being a local maximum in this sequence. Because x_{i-1}, x_i, and x_{i+1} are all iid from $f(x)$, I can state that

$$\begin{aligned} f_D(x) &= p(x_i)p(x_{i-1} < x_i)p(x_{i+1} < x_i) \\ &= f(x)F^2(x), \end{aligned} \tag{1.2}$$

where $p(x_i) = f(x)$ and $p(x_{i-1} < x_i) = p(x_{i+1} < x_i) = F(x)$ based on the definition of a cumulative probability function. Similarly, it shows that the local minima distribution is described as

$$\begin{aligned} f_B(x) &= p(x_i)p(x_{i-1} > x_i)p(x_{i+1} > x_i) \\ &= f(x)[1 - F(x)]^2, \end{aligned} \tag{1.3}$$

where $p(x_{i-1} > x_i) = p(x_{i+1} > x_i) = 1 - F(x)$. To use the expectation function $E(g(x)) = \int_{-\infty}^{\infty} xg(x)dx$ on a continuous probability density function $g(x)$, it is required that $g(x)$ is a proper density function with $\int_{-\infty}^{\infty} g(x)dx = 1$. This requirement is used to normalized both $f_B(x)$ and $f_D(x)$ as

$$\begin{aligned} \hat{f}_B(x) &= \frac{f(x)[1 - F(x)]^2}{N_B}, \\ \hat{f}_D(x) &= \frac{f(x)F^2(x)}{N_D}, \end{aligned} \tag{1.4}$$

where $N_B = \int_{-\infty}^{\infty} f(x)[1 - F(x)]^2 dx$ and $N_D = \int_{-\infty}^{\infty} f(x)F^2(x)dx$. I can further reduce N_B and N_D from Eq. (1.4) by substituting $f(x) = F'(x)$, which reduces the N_D equation to

$$\begin{aligned} N_D &= \int_{-\infty}^{\infty} F'(x)F^2(x)dx = \int_{-\infty}^{\infty} \frac{1}{3}(F^3(x))'dx \\ &= \frac{1}{3}F^3(x)|_{-\infty}^{\infty} = \frac{1}{3}, \end{aligned} \tag{1.5}$$

since it is assumed that $F(\infty) = 1$ and $F(-\infty) = 0$. Similarly,

$$N_B = \int_{-\infty}^{\infty} f(x)[1 - F(x)]^2 dx = \int_{-\infty}^{\infty} f(x)[1 - 2F(x) + F^2(x)]dx = N_D + \int_{-\infty}^{\infty} f(x)[1 - 2F(x)]dx$$
$$= N_D + \int_{-\infty}^{\infty} F'(x)dx - \int_{-\infty}^{\infty} (F^2(x))'dx = N_D + \left[F(x) - F^2(x)\right]|_{-\infty}^{\infty} = N_D = \frac{1}{3}. \tag{1.6}$$

This can now reduce Eq. (1.4) to

$$\begin{aligned} \hat{f}_B(x) &= 3f(x)[1 - F(x)]^2, \\ \hat{f}_D(x) &= 3f(x)F^2(x), \end{aligned} \tag{1.7}$$

I now assume $f(x)$ is of a Gaussian distribution to validate our expressions in Eq. (1.4). Specifically, I define the Gaussian (normal) probability distribution as

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{1.8}$$

with a cumulative distribution

$$F(x) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{x - \mu}{\sigma\sqrt{2}}\right)\right]. \tag{1.9}$$

To validate the resulting expressions for $\hat{f}_B(x)$ and $\hat{f}_D(x)$ in (1.4), a numerical simulation of a normal distribution $\mathcal{N}_n(\mu = 0, \sigma^2 = 1)$ of length $n = 10E5$ was used (see Fig. 1.4). This analysis shows a very similar result between the histograms $h(*)$ and distributions. From the numerical simulation, I also found the ratio $\bar{L} \approx 1.686$, where $\bar{L}$ is the sample mean of the lifetimes from $\mathcal{N}(0, \sigma^2 = 1)$. Additionally I found that $\bar{D} - \bar{B} \approx 1.689 \approx \bar{L}$. These results suggest that Eq. (1.1) and Eq. (1.4) are correct. I now move on to determine a suitable cutoff with unknown probability $f(x)$ and cumulative probability density $F(x)$ functions.

Now that I have shown that our expressions for the probability distribution of the minima and maxima are correct, I proceed to correlate the mean lifetime μ_L to the additive noise distribution parameters. From our results in (1.7) I can now calculate the mean lifetime as

$$\mu_L = 3\int_{-\infty}^{\infty} xf(x)\left[F^2(x) - (1 - F(x))^2\right] dx = 3\int_{-\infty}^{\infty} x\left[(F^2(x))' - F'(x)\right] dx, \tag{1.10}$$

which can then be simplified using integration by parts as

$$\mu_L = 3\int_{-\infty}^{\infty} F(x)\left[1 - F(x)\right] dx. \tag{1.11}$$

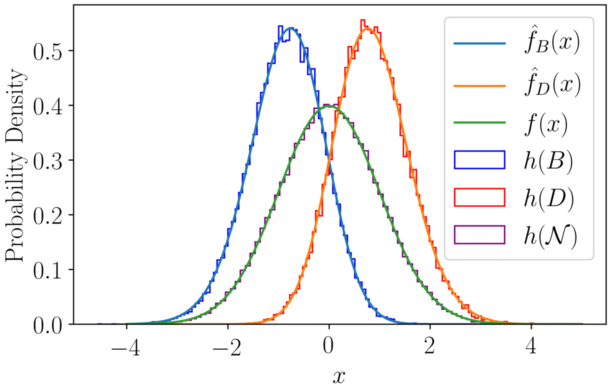


Figure 1.4: Histograms $h(*)$ of the zero mean normal distriubtion $\mathcal{N}(0, \sigma^2 = 1)$ and the resulting birth times B and death times D, which are compared to the density distributions from Eq. (1.4).

1.2.2 Cutoff Background

To determine a suitable cutoff, I again start by assuming I have n random samples from our noise distribution: $\mathbf{x} = \{x_1, x_2, \ldots, x_n\} \overset{iid}{\sim} \mathcal{N}$ with a cumulative probability function $F(x)$. The probability that the minimum of $\mathbf{x}$ is less than the value a is equivalent to

$$P(\min(\mathbf{x}) < a) = 1 - P(x_1 > a, x_2 > a, \ldots, x_n > a), \tag{1.12}$$

where $P(x_i > a) = 1 - F(a)$. If this relationship is extended to all n realizations, the probability is

$$P(\min(\mathbf{x}) < a) = 1 - (1 - F(a))^n. \tag{1.13}$$

Similarly, an expression for the probability of an element of $\mathbf{x}$ being greater than b, where $b > a$, is

$$P\left(\max(\mathbf{x}) > b\right) = 1 - (F(b))^n. \tag{1.14}$$

If I now take both of these probabilities, I can extend them to the maximum lifetime as $\max(L) \lessapprox \max(\mathbf{x}) - \min(\mathbf{x})$. we can use to generate a probability of a lifetime being greater than $b - a$ as

$$\alpha = P(\max(L) > b-a) \gtrapprox P\big(\max(\mathbf{x}) > b, \min(\mathbf{x}) < a\big) = (1-[F(b)]^n)(1-[1-F(a)]^n), \tag{1.15}$$

where α is the confidence of this event occurring. If the $f(x)$ associated to $F(x)$ of Eq. (1.15) is symmetric about some mean μ such that $c = b - \mu = \mu - a$, I can reduce Eq. (1.15) to

$$\alpha = (1 - [F(c)]^n)^2 \tag{1.16}$$

since $F(b) = 1 - F(a)$ for the symmetric case. (1.16) can be then solved for c as

$$c = F^{-1}\left[\left(1 - \sqrt{\alpha}\right)^{1/n}\right]. \tag{1.17}$$

Additionally, I know that a cutoff should be set such that $C_\alpha = b - a = 2c$ for a symmetric distribution about some mean μ, which result in a cutoff equation as

$$C_\alpha = 2F^{-1}\left[\left(1 - \sqrt{\alpha}\right)^{1/n}\right]. \tag{1.18}$$

On the other hand, if there is no symmetry in the distribution then I need a new cutoff equation. To do this, I return to our probability equation as

$$\alpha = P(\max(L) > b-a) \gtrapprox P(\min(\mathbf{x}) < a, \max(\mathbf{x}) > b) = (1-[1 - F(a)]^n)(1-[F(b)]^n), \tag{1.19}$$

However, unlike Eq. (1.18), I can not solve Eq. (1.15) for a parameter c due to their being no symmetry between a and b about a mean μ which means I must simplify Eq. (1.19) in some way. To do this, I assume that $P(\min(\mathbf{x}) < a) = P(\max(\mathbf{x}) > b)$ or $1 - [1 - F(a)]^n = 1 - [F(b)]^n = \sqrt{\alpha}$. I can then solve for a and b separately as

$$a = F^{-1}\left[1 - (1 - \sqrt{\alpha})^{1/n}\right] \tag{1.20}$$

and

$$b = F^{-1}\left[(1 - \sqrt{\alpha})^{1/n}\right]. \tag{1.21}$$

With $C_\alpha = b - a$ and the values of a and b from (1.20) and Eq. (1.21), respectively, I can solve for our general cutoff expression as

$$C_\alpha = F^{-1}\left[(1 - \sqrt{\alpha})^{1/n}\right] - F^{-1}\left[1 - (1 - \sqrt{\alpha})^{1/n}\right]. \tag{1.22}$$

For our application I want to have a high confidence level that no outliers occur and that the cutoff accurately captures all of the noise, so I suggest a confidence level of $\alpha = 0.1\%$, which is equivalent to a 0.1% chance that an outlier greater than the persistence diagram lifetime cutoff C_α (see Fig. 1.5) exists given n samples.

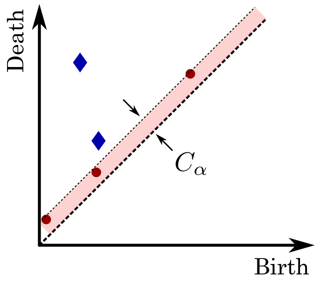


Figure 1.5: Example cutoff C_α for a persistence diagram and time order lifetimes of sublevel set persistence from $x(t)+\mathcal{N}$.

(1.18) and (1.22) are only dependent on the desired confidence α, the signal length n and the cumulative probability distribution $F(x)$ with the cumulative probability distribution having another distribution parameter (e.g. σ for the Gaussian distribution). I address how to estimate this parameter, if it is unknown, in Section 1.2.4. Before this, in Section 1.2.3 I demonstrate how to apply Eq. (1.18) and Eq. (1.22) for the Gaussian, uniform, Rayleigh, and exponential distribution.

1.2.3 Cutoff for Noise Models

For applying noise models to the confidence levels in Equations (1.15) and (1.16), I need to be either given the additive noise parameters, or estimate them from the lifetimes. However, before this can be done, I need to understand which parameters are needed given the additive noise distribution $f(x)$. I do this analysis for Gaussian (normal), Uniform, Rayleigh, and exponential distributions as shown in Fig. 1.6.

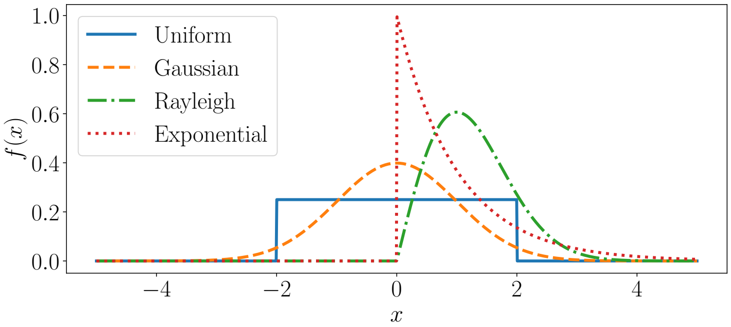


Figure 1.6: Additive noise probability distributions $f(x)$ for the four models realized in this work: uniform, Gaussian, Rayleigh, and exponential.

Cutoff for Gaussian Noise I start our analysis with the commonly used Gaussian distribution model. The Gaussian (normal) probability distribution function is defined as

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{1.23}$$

with a cumulative distribution function

$$F(x) = \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right]. \tag{1.24}$$

I start by solving for the inverse of Eq. (1.24) as

$$F^{-1}(u) = \sqrt{2}\sigma \operatorname{erf}^{-1}(2u-1) + \mu. \tag{1.25}$$

Since the mean shift μ has no effect on the sublevel set lifetimes I can ignore it and apply Eq. (1.25) with $\mu = 0$ to solve for the cutoff from Eq. (1.18) as

$$C_\alpha = 2^{3/2}\sigma \operatorname{erf}^{-1}\left[2(1-\sqrt{\alpha})^{1/n} - 1\right]. \tag{1.26}$$

With a full development of the statistics of sublevel set persistence for Gaussian (normal) additive noise I are able to determine a suitable cutoff for i.i.d. noise with only the distribution parameter σ needed.

Cutoff for Uniform Noise Let $a < b \in \mathbb{R}$. The uniform distribution over the interval $[a, b]$ has a probability density function defined as

$$f(x) = \begin{cases} \frac{1}{b-a} & x \in [a,b] \\ 0 & \text{otherwise} \end{cases} \tag{1.27}$$

with a cumulative distribution function

$$F(x) = \begin{cases} 0 & x < a \\ \frac{x-a}{b-a} & x \in [a,b] \\ 1 & x > b. \end{cases} \tag{1.28}$$

By assuming a symmetric distribution about zero (this assumptions does not influence the resulting cutoff due to the properties sublevel set persistence lifetime) such that $a = -b$ and $\Delta = b - a$. This changes $F(x)$ to

$$F(x) = \begin{cases} 0 & x < -\frac{\Delta}{2} \\ \frac{2x+\Delta}{2\Delta} & x \in [-\frac{\Delta}{2}, \frac{\Delta}{2}] \\ 1 & x > \frac{\Delta}{2} \end{cases} \tag{1.29}$$

If I now apply Eq. (1.18) to the inverse of the cumulative probability distribution in Eq. (1.29), I can calculate C_α as

$$C_\alpha = \Delta \left[2 \left(1 - \sqrt{\alpha} \right)^{1/n} - 1 \right]. \tag{1.30}$$

Equation (1.30) only requires the distribution parameter Δ as both α and n are chosen as desired and the length of the time series, respectively.

Cutoff for Rayleigh Noise The Rayleigh distribution has a probability density function over the domain $x \in [0, \infty)$ and is defined as

$$f(x) = \frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}, \tag{1.31}$$

with a cumulative distribution function

$$F(x) = 1 - e^{-\frac{x^2}{2\sigma^2}}. \tag{1.32}$$

Since this distribution is asymmetric I use Eq. (1.22) to calculate C_α as

$$C_\alpha = \sigma \left(\sqrt{-2 \ln \left([1 - \sqrt{\alpha}]^{1/n} \right)} - \sqrt{-2 \ln \left(1 - [1 - \sqrt{\alpha}]^{1/n} \right)} \right), \tag{1.33}$$

where σ is the only parameter that needs to be provided to calculate the cutoff.

Cutoff for Exponential Noise The exponential distribution has a probability density function over the domain $x \in [0, \infty)$ and is defined as

$$f(x) = \lambda e^{-\lambda x}, \tag{1.34}$$

with a cumulative distribution function

$$F(x) = 1 - e^{-\lambda x}, \tag{1.35}$$

where λ is the distribution parameter with $\lambda > 0$. This this distribution is also asymmetric, so I use Eq. (1.22) to calculate C_α as

$$C_\alpha = -\frac{1}{\lambda} \ln\left([1 - \sqrt{\alpha}]^{1/n} - [1 - \sqrt{\alpha}]^{2/n}\right), \tag{1.36}$$

where λ is the only parameter that needs to be provided to calculate the cutoff.

1.2.4 Cutoff and Distribution Parameter Estimation Method

If the distribution parameter is know (σ for Gaussian distributions, Δ for uniform distributions, σ for Rayleigh distributions, and λ for exponential distributions), then the cutoff C_α can be calculated simply with the use of the correct cutoff equation in Section 1.2.3 and the subsequent analysis may be skipped. However, in most real-world time series it is uncommon to know what this parameter is and thus it needs to be estimated. While there are some methods for estimating the additive noise parameters [54, 95, 234], I introduce a new method utilizing the relationship between the sublevel set lifetimes from both the signal and noise and the additive noise distribution parameters.

To generate a theoretical relationship between the mean lifetime μ_L and the distribution parameters, I recall Eq. (1.11):

$$\mu_L = 3 \int_{-\infty}^{\infty} F(x)\, [1 - F(x)]\, dx.$$

In the subsequent subsections, I show how this relationship is used for each of the four noise models analyzed in this work. However, when the signal is not pure noise, which would be the case for any informative time series, the mean lifetime is heavily influenced from the lifetimes associated to significant features. To address this issue, I instead calculate the median of the lifetimes as it is robust up to 50% outliers (or signal in our application) and apply a signal compensation. This brings up an assumption for this distribution parameter estimation method to function correctly: the number of persistence diagram features associated with noise N_n must be equal to or greater than the number of features from the signal N_s. Additionally, when N_n approaches N_s the cutoff becomes more conservative due to the robustness limitation of the median. To minimize this effect, in Section 1.2.5, I develop a numeric compensation multiplier which uses the persistence pairs associated to both additive noise and signal. In general, the condition for $N_s < N_n$ is met if the time series is sampled at a rate sufficiently higher than the Nyquist sampling criteria f_{Nyquist} and, of course, the time series has some additive noise. If these conditions are not met, I suggest the use of an alternative method to estimate the distribution parameter of the additive noise and apply its associated cutoff equation in Section 1.2.3.

For a symmetric distribution of the lifetimes, the median would be an accurate estimate of the mean. However, for most additive noise distributions (e.g. Gaussian), the distribution of the resulting sublevel set persistence lifetimes is not symmetric. Therefore, I resort to approximating the relationship between the mean and median numerically. While there are methods to estimate the mean using the median and Inter-Quartile Range (IQR) as described in [236]. This method is only robust for up to 25% outliers (or signal in our application) due to the Q_3 upper quartile. Therefore, I use the numerically approximated ratios of $\rho = \bar{L}/\tilde{L}$ as provided in Table 1.1 for each of the four distributions investigated, where $\bar{L}$ is the sample mean lifetime and $\tilde{L}$ is the sample median lifetime. For each of these numeric estimates a time series of length 10^5 was used. This numeric experiment was repeated ten times to provide a mean ρ with uncertainty. This ratio can be used to estimate the mean lifetime as $\bar{L} \approx \rho\tilde{L}$.

Table 1.1: Ratios $\rho = \bar{L}/\tilde{L}$ for estimating sample mean from the sample median with uncertainty as three standard deviations

Distribution	Gussian	Uniform	Rayleigh	Exponential
$\rho = \bar{L}/\tilde{L}$	1.154 ± 0.012	1.000 ± 0.010	1.136 ± 0.013	1.265 ± 0.016

Relating The distribution Statistic to the Median Lifetime I now apply Eq. (1.11) and ρ from Table 1.1 to find relationships between the median lifetime M_L and the distribution parameter used in each distribution's cutoff equation.

Normal Distribution: For estimating σ of the Gaussian distribution, I use (1.11) and the Gaussian cumulative distribution to estimate μ_L as a function of σ. Specifically, by numerically approximating the integral in Eq. (1.11) using $x \in [-10, 10]$ with $\text{len}(x) = 10^6$, I found the relationship

$$\sigma \approx \frac{\mu_L}{1.692}. \tag{1.37}$$

I then used ρ to have Eq. (1.37) as a function of the median lifetime M_L as

$$\sigma \approx \frac{\rho M_L}{1.692} \approx 0.680 M_L, \tag{1.38}$$

where M_L is the median lifetime. Applying this result to (1.26) allows for a cutoff to be calculate as

$$C_\alpha \approx 1.923 \tilde{L} \,\text{erf}^{-1}\left[2(1-\sqrt{\alpha})^{1/n} - 1\right], \tag{1.39}$$

where $\tilde{L}$ is the sample median lifetime.

Uniform Distribution Next, I apply Eq. (1.11) to the uniform cumulative distribution to estimate μ_L as a function of Δ. Substituting (1.29) into Eq. (1.11) results in

$$\mu_L = 3 \int_{-\Delta/2}^{\Delta/2} \frac{2x + \Delta}{2\Delta}\left[1 - \frac{2x + \Delta}{2\Delta}\right] dx.$$

Expanding and solving this integral results in the relationship

$$\mu_L = \frac{\Delta}{2} \implies \Delta = 2\mu_L = 2M_L. \tag{1.40}$$

Applying this result to Eq. (1.30) allows for a cutoff to be calculate as

$$C_\alpha = 2\tilde{L}\left[2\left(1 - \sqrt{\alpha}\right)^{1/n} - 1\right]. \tag{1.41}$$

Rayleigh Distribution For estimating σ of the Rayleigh distribution, I again use (1.11) with the cumulative Rayleigh distribution to numerically estimate the relationship between μ_L and σ as

$$\sigma \approx \frac{\mu_L}{1.102} \approx \frac{\rho M_L}{1.102} \approx 1.025 M_L, \tag{1.42}$$

where the integral in Eq. (1.11) was numerically approximated using $x \in [0, 20]$ with $\text{len}(x) = 10^6$. Applying this result to Eq. (1.33) allows for a cutoff to be calculate as

$$C_\alpha \approx 1.025\tilde{L}\left(\sqrt{-2\ln\left([1-\sqrt{\alpha}]^{1/n}\right)} - \sqrt{-2\ln\left(1 - [1-\sqrt{\alpha}]^{1/n}\right)}\right). \tag{1.43}$$

Exponential Distribution Next, I apply Eq. (1.11) to the exponential cumulative distribution function to estimate μ_L as a function of λ. Substituting Eq. (1.35) into Eq. (1.11) results in

$$\mu_L = 3\int_0^\infty \left(1 - e^{-\lambda x}\right) e^{-\lambda x} dx, \tag{1.44}$$

which was solved using a u-substitution as

$$\mu_L = \frac{3}{2\lambda} \rightarrow \lambda = \frac{3}{2\mu_L}. \tag{1.45}$$

By then using the appropriate ρ from Table 1.1 to use M_L instead of μ_L, I approximate λ from the median lifetime:

$$\lambda \approx \frac{1.875}{M_L}. \tag{1.46}$$

Applying this result to Eq. (1.36) allows for a cutoff to be calculate as

$$C_\alpha \approx -0.533\tilde{L} \ln\left([1 - \sqrt{\alpha}]^{1/n} - [1 - \sqrt{\alpha}]^{2/n}\right). \tag{1.47}$$

1.2.5 Signal Compensation for the Cutoff and Distribution Parameter

In this section, I discuss the effects of signal on the cutoff estimation methods described. In Section 1.2.4 I assumed that the time series was of the form $x(t) = \{x_1, x_2, \ldots, x_n\} \overset{iid}{\sim} \mathcal{N}$, however in practice, I typically have some underlying informative signal $s\colon \mathbb{R} \rightarrow \mathbb{R}$ and have a time series of the form $x(t) = s(t) + \epsilon$ with a finite domain as $t \in [t_a, t_b]$. The resulting sublevel sets from $s(t) + \epsilon$ are assumed to have some lifetimes from $s(t)$ with the slope of the signal having an effect on the lifetimes associated with $\mathcal{N}$. Because of these effects, I attempt to compensate the cutoff calculation and distribution parameter estimations for these effects for a general signal. Since a general signal is, in practice, rather subjective, I move away from a theoretical analysis of the signal and rather analyze the effects of the signal experimentally. I have partially addressed this issue of signal compensation by implementing the median lifetime M_L instead of the mean lifetime μ_L with the median being an outlier (signal in our case) robust statistic for up to 50% outliers. Even with the use of the median, I need to further develop a signal compensation procedure to improve the accuracy of the suggested cutoff.

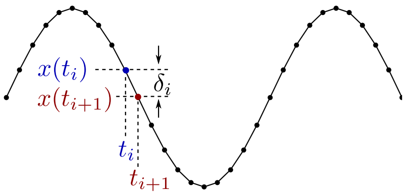


Figure 1.7: Example time series showing sample δ_i.

To fully understand the effects of signal on estimating the cutoff, I do a numeric study to develop a method for adjusting the median lifetime such that $M_L(s(t)+\epsilon) \approx M_L(\mathcal{N})$. This analysis requires a new variable which I term δ as simply the median of step sizes defined as $\delta_i = x(t_{i+1}) - x(t_i)$ as shown in Fig. 1.7, where $x(t)$ is a discretely and uniformly sampled signal with a constant sampling rate f_s. I now experimentally approximate the effects of signal on the median lifetime by using three “generic" signals suggested by [241] as

$$f_1(t) = t - t^3/3, \tag{1.48}$$

with $t \in [3.1, 20.4]$ and sampling rate $f_s = 20$ Hz,

$$f_2(t) = \sin(t) + \sin(2t/3), \tag{1.49}$$

with $t \in [3.1, 20.4]$ and sampling rate $f_s = 20$ Hz, and

$$f_3(t) = -\sum_{i=1}^{5} \sin((i+1)t + i), \tag{1.50}$$

with $t \in [-10, 10]$ and sampling rate $f_s = 20$ Hz. Additionally, additive noise is included in the signal with $s(t) = Af(t) + \epsilon$ with the additive noise distribution parameter set to one (e.g. $\sigma = 1$ for Gaussian) and signal amplitude A increment by unit steps starting from zero such that the δ is also incremented until reaching a value $\delta/\sigma = 2$. At each δ I calculate the median lifetime $\tilde{L}$ for 100 trials to provide a mean $\tilde{L}$ with uncertainty u_L as one standard deviation (see Fig. 1.8 for the Gaussian additive noise example).

To find a function fitting to approximate this relationship between δ and $\tilde{L}$ for each distribution type. By observation of the median lifetimes in Fig. 1.8, I experimentally found an approximate functional template:

$$\tilde{L}^* = \tilde{L}_0 e^{-c_1\left(\frac{\delta}{\delta+\tilde{L}}\right)^{c_2}}, \tag{1.51}$$

where $\tilde{L}_0$ is the median lifetime when $\delta = 0$ or when the signal is just additive noise $\mathcal{N}$.

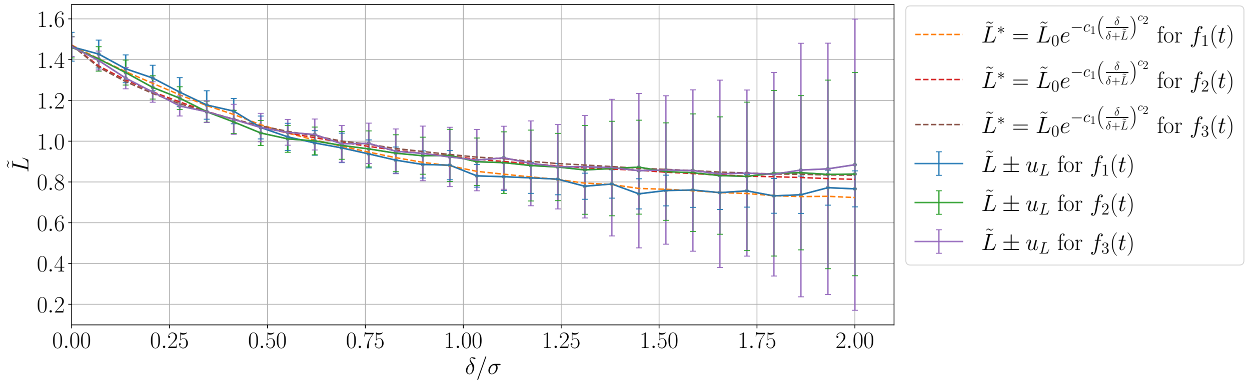


Figure 1.8: Numeric function fitting of Eq. (1.51) to the mean of the median lifetime $\tilde{L}$ of $f_i(t)$ for $i \in [1, 3]$ where $\mathcal{N}$ is unit variance Gaussian additive noise with $\delta \in [0, 2]$ being incremented to understand the effects of signal on the median lifetime.

As shown in Fig. 1.8, the fitted function shows a very similar quality to the numerically simulated means of the median lifetimes when the two constants in Eq. (1.51) were set to $c_1 \approx 0.845$ and $c_2 \approx 0.809$ for a Gaussian additive noise, which were chosen using the BFGS minimization of the ℓ_2 norm cost function on the residuals when fitting to $\tilde{L}$ for all three generic functions. Another characteristic of these constants is that they are approximately independent of the additive noise distribution parameter, sampling frequency, and time series, which makes them global constants. The two constants from Eq. (1.51) are provided in Table 1.4 for the four distributions investigated in this work. With these constants, I calculate a multiplication compensation term for the signal as R, which is calculated from Eq. (1.51) as

$$R = \frac{\tilde{L}_0}{\tilde{L}^*} = e^{c_1\left(\frac{\delta}{\delta+\tilde{L},}\right)^{c_2}} \tag{1.52}$$

which is used to compensate for the effects of signal with $C^*_\alpha = RC_\alpha$ and $\sigma^* = R\sigma$.

Table 1.2: Constants of Eq. (1.51) for each distribution type investigated in this work with associated uncertainty from ten trials.

Distribution	Gussian	Uniform	Rayleigh	Exponential
c_1	0.845 ± 0.029	0.880 ± 0.017	0.726 ± 0.026	0.436 ± 0.036
c_2	0.809 ± 0.061	0.639 ± 0.026	0.605 ± 0.054	0.393 ± 0.075

Unfortunately, when $s(t)$ is unknown, the δ parameter used in (1.80) can no longer be directly calculated from the time series or sublevel set persistence diagram. To approximate δ I use the lifetimes greater than the initial uncompensated cutoff C_α as

$$\delta \approx \frac{2}{n} \sum L_{C_\alpha}, \tag{1.53}$$

where L_{C_α} are the lifetimes greater than C_α.

To validate the accuracy of Eq. (1.80) with δ approximated from (1.81) I estimate σ with and without the signal compensation R from Eq. (1.80), I use a new time series $x(t) = A\sin(\pi t) + \epsilon$ with $\mathcal{N}$ being a Gaussian distribution with unit variance and A incremented to change $\delta \in [0, 2]$ for 100 trials at each δ. As shown in Fig. 1.9, the true $\sigma = 1$ and the estimated σ without compensation from (1.38) shows an underestimate as δ increases until plateauing around $\delta/\sigma \approx 1$, which would cause for a cutoff that may not capture all of the lifetimes associated with noise. However, the signal compensated distribution parameter σ^* shows an accurate estimation of σ even as δ becomes significantly large. This example demonstrates the importance of signal compensation for an accurate cutoff and distribution parameter estimation.

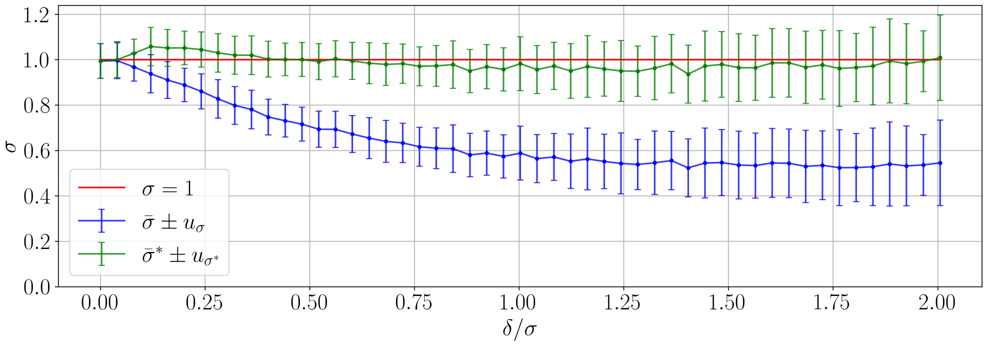


Figure 1.9: Demonstration of distribution parameter σ estimation of Gaussian additive noise in $x(t) = A\sin(\pi t) + \mathcal{N}$ using the median lifetime with and without signal compensation as σ and σ^*, respectively.

1.3 Damping Parameter Identification Using Sublevel Set Persistence

The study of damping mechanisms in the field of vibrations has always been a critical aspect of understanding the way dynamical systems behave and has been leveraged for many real-world applications. While there have been data analysis methods for estimating these system parameters, the ability of engineers and scientisits to use signal processing techniques to determine these parameters is ever improving as new and more sophisticated data analysis techniques are discovered.

The identification of these damping mechanisms and their assocaited damping parameters in a real-world dynamical system is a critical tool for analyzing and predicting the dynamics [29, 84, 191, 192]. Specifically, methods for estimating the damping parameters have been used extensively in signal processing engineering with application in structural health monitoring [32], improved predictions of mechanical response [136], biological system analysis [88, 155], and the analysis of Micro and Nano Electromechanical Systems (MEMS and NEMS) [187].

A common method for damping parameter identification is through a time domain analysis of the amplitude decrement (i.e. the damping envelope). This form of analysis is often implemented for viscous damping estimation through the logarithmic decrement of peaks. Unfortunately, many systems do not have damping of this nature or they have some non-linearity causing the log decrement method to not be suitable. Additionally, when significant noise is present in the signal, the estimation of peak values takes a degree of expertise and human evaluation, which makes damping parameter identification difficult to implement in an automatic scenario. These common issues have pushed researchers to develop automatic, noise robust methods for estimating damping parameters [102, 151]. In the past decade several of these methods have been developed for identifying systems parameter for a single degree of freedom system, including damping constants. These methods are typically based on either a time domain or frequency analysis (i.e. modal analysis) of the oscillator. The time domain response methods for damping parameter estimation are typically based on analyzing the envelope of the free response decrement or through an energy balance approach. The envelope of the free response is commonly used for estimating either viscous damping through an exponential envelope (viscous damping) or a constant decrement envelope (Coulomb or dry friction damping). Additionally, systems with both coulomb and viscous damping can be simulataneously analyzed using vibration decrements through the time domain [132].

As an alternative to analyzing the envelope, the energy loss can be studied to estimate the damping parameters through least squares fitting for forced vibrations [133, 134]. However, this approach does require a method of forcing the oscillator to estimate the damping parameters, which is not always available or feasible. There are also energy-balance technqiues for parameter identification that do not require forcing, but rather both the position and velcoity signals [142]. However, for this technique to function properly a filtration is needed (cubic spline fitting in [142]), which is inherently computationally cumbersome. Another approach is to use the institaneous energy dissipation [150], but this method requires a lightly damped system which is a significant, yet common, limitation. There are also several other time domain methods including a method based on areas [96], which requires viscous damping but could possible be extended to other damping mechanisms. This method implements a numeric integration of the signal and zero crossing making it noise robust and only requiring the position signal of the oscillator. While this could seem like an easy solution for damping parameter estimation, the task of finding zero-crossings is not trivial and typically requires a filtration method which can be computationally expensive. Another commonly used method for parameter identification is to fit a function to the time series response based on tuning parameters, but this requires an initial guess on all parameters and an optimization algorithm. Another possible method for damping parameter estimation is based on solving a parabolic-type partial differential equation for analysis of the inverse vibration problem to estimate both stiffness and damping [138]. However, this method requires both the position and velocity data and is only resilient to moderate amounts of noise.

As an alternative to a time domain analysis, frequency response methods are typically done by externally forcing the oscillator and measuring the phase and amplitude of the response at resonance (e.g. half-power method [172]). However, this assumes that the range of operation is within the linear region of oscillations or that the damping mechanisms are ampltidue independent. This method also requires a method of forcing the oscillator at multiple frequencies, which is not always feasible. An alternative to this option is to analyze the frequency response of a damped oscillation through the Fourier spectrum [245]. This method has been shown to be robust to some degree of additive Gaussian noise [193]. However, it requires a least-squares estimation algorithm applied to the frequency domain of the signal, which is an additional computational expense.

To estimate the damping model that is most suitable, I have developed a new method that implements zero-dimensional (0D) sub-level set persistence, a tool from topological data analysis, to analyze the time domain response of a free vibration single degree of freedom oscillator with viscous, coulomb, or quadratic damping. This novel method provides an extension of envelope analysis methods through a unique and noise robust analysis of the time domain response. This sublevel set persistnce analysis method also holds an advantage of not requiring a zero-mean for low damping parameters and is robust to non-stationarity in the signal. This is in comparison to many common damping parameter estimation techniques requiring these conditions [38]. I show that this technique is robust for a wide range of damping parameters (including very high damping up to a critically damped response), low sampling frequencies, and a high degree of noise contamination. Additonally, the algorithm for calculating the sublevel set persistence for one dimensional signals has a low computational cost with it being faster than the fast fourier spectrum [114].

Sublevel set persistence has recently been shown as a robust data analysis tool through applications ranging from step detection [114] to cancer histology [127]. One of the most attractive features of sublevel set persistence is its robustness to perturbations (see stability theorem [49]). Additionally, by using sublevel set persistence to analyze the time domain of the free responses of a damped oscillator I will later be able to analyze the full domain (including non-lineary responses) of the system similar to the work done in analyzing MEMS [187].

The results in this work will be generated from both experimental data and numerically simulated single-degree-of-freedom spring-mass system with three common forms of damping (see Fig. 1.10): Coulomb, viscous, and quadratic. The forces caused by each of the damping mechanisms are applied to Newton's law to generate an equation of motion as

$$m\ddot{x} = -kx - \mu_c N|\dot{x}|\text{sgn}(\dot{x}) - \mu_v\dot{x} - \mu_q|\dot{x}|\dot{x}, \tag{1.54}$$

with a mass m, spring constant k, and normal force $N = mg$. Here the normal force is constant, but in many applications this will not be the case, which can leave $N = f(\ddot{x}, \dot{x}, x)$ where x is the position of the system and $\dot{}$ are its time derivatives.

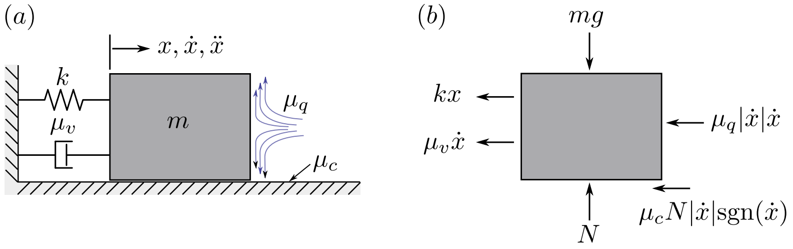


Figure 1.10: Single degree of freedom oscillator with multiple modes of energy dissipation. Energy dissipation mechanisms include Coulomb μ_c, viscous μ_v, and quadratic μ_q damping.

This work is ordered as follows. First, in Section 1.3.1 the closed form solutions (if applicable) and background information for viscous, Coulomb and quadratic damping are summarized. Section 1.3.1 also leverages the solutions to the damped responses for use with sublevel set persistence for damping identification. With an introduction to the damping mechanisms and sublevel set persistence, in Section 1.3.2 I begin an analysis of the effects of noise on damping parameter identification using sublevel set persistence. This analysis will introduce two methods for minimizing the effects of noise. The first is based on a statistical analysis of additive noise in the persistence domain and the second is based on a function fitting approach. In Section 1.3.5 I provide three examples demonstrating each damping mechanism. Finally, in the results section (Section 4.3), the method is applied to a wide range of damping parameters, noise levels, and sampling frequencies to determine the limitations of the method. To make replicating this work easier for readers, the Python code for automatically calculating the damping parameters and constants has been made publicly available through GitHub (*github.com/Khasawneh-Lab*).

1.3.1 Sublevel Set Persistence of Damping Mechanisms

In this section I introduce three damping mechanisms commonly used: Coulomb, viscous, and quadratic. For each form of damping, a theoretical relationship between the theoetical consecutive persistence pairs is formulated and used to determine the underlying damping parameter of the system.

Viscous Damping If the system being analyzed is assumed to be dominated by viscous damping then the system model is reduced from Eq. (1.54) to $m\ddot{x} + kx + \mu_v\dot{x} = 0$. This linear differential equation has the closed form solution as

$$x(t) = Ae^{-\zeta\omega_n t}\cos(\omega_d t - \phi), \tag{1.55}$$

where the viscous damping can be summarized using the damping ratio $\zeta_v = \mu_v/(2\sqrt{mk})$, the natural frequency $\omega_n = \sqrt{k/m}$, the damped natural frequency $\omega_d = \omega_n\sqrt{1-\zeta^2}$, the phase shift ϕ, and the initial amplitude of the time series A. Typically, ζ is estimated using local maxima and the log decrement method as

$$\zeta_v = \sqrt{\frac{1}{1+\left(\frac{2\pi n}{\ln\left(\frac{p_{i+n}}{p_i}\right)}\right)^2}}, \tag{1.56}$$

where p_{i+n} and p_i denote the $(i+n)^{\text{th}}$ and i^{th} peaks, respectively. Unfortunately, this method for estimating ζ_v is difficult to implement in an automatic way when noise is present as the selection of peaks becomes difficult. Additionally, if the time series is non-stationary or does not have a zero-mean, the standard logarithmic decrement method will not provide accurate damping parameter estimates. To help combat these issues I will implement sub-level set persistence to show how ζ_v can be calculated from the resulting persistence diagram.

Let us begin with a toy example of the time series and the resulting persistence diagram for viscous damping as shown in Fig. 1.11. The x and y coordinates in the persistence diagram correspond to the local minima v_n and maxima p_{n+1} in the time series $x(t)$. From the known,

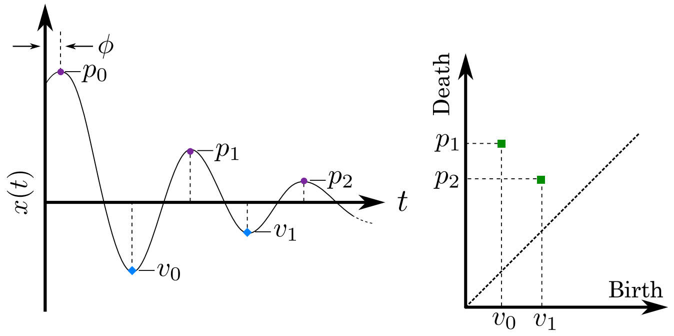


Figure 1.11: Example 0D sub-level set persistence from the viscously damped free response time series $x(t)$.

closed-form solution in Eq. (1.55), the values of the peaks and valleys are solved for as

$$p_i = Ae^{-\zeta_v(2i\pi+\phi)/\sqrt{1-\zeta_v^2}} \tag{1.57}$$

and

$$v_i = -Ae^{-\zeta_v(2i\pi+\pi+\phi)/\sqrt{1-\zeta_v^2}}, \tag{1.58}$$

respectively. From the peaks and valleys or births and deaths for the persistence pairs, their lifetimes are calculated as

$$L_i = p_{i+1} - v_i = Ae^{\frac{-i\zeta_v 2\pi}{\sqrt{1-\zeta_v^2}}}\left(e^{\frac{-\zeta_v 2\pi}{\sqrt{1-\zeta_v^2}}} + e^{\frac{-\zeta_v \pi}{\sqrt{1-\zeta_v^2}}}\right), \tag{1.59}$$

where L_i is a lifetime of the sub-level set persistence pair (v_i, p_{i+1}). repeating this lifetime calculation for the $(i+n)^{\text{th}}$ peak-valley pair results in another lifetime ${}_{i+n}$, which is used to find the ratio between lifetimes as

$$\frac{L_{i+n}}{L_i} = e^{-n\zeta_v 2\pi/\sqrt{1-\zeta_v^2}}. \tag{1.60}$$

By taking this ratio, the amplitude A cancels out, which allows for Eq. 1.60 to be used to calculate ζ_v as

$$\zeta_v = \sqrt{\frac{1}{1+\left(\frac{2n\pi}{\ln(L_{i+n}/L_i)}\right)^2}}, \tag{1.61}$$

From the damping ratio I can also calculate the viscous damping constant as $\mu_v = 2\zeta_v\sqrt{km}$ if the other system parameters m and k are known. Another benefit of using sublevel set persistence for estimating the damping ratio is that only a single lifetime is needed. The standard method for estimating ζ_v in Eq. (1.56) needs atleast two peaks to estimate the damping ratio, while only a single lifetime is needed for estimating the damping constant with a slight variation of Eq. (1.61). Specifically, if I assume the time series $x(t)$ is centered about zero such that $\lim_{t\to\infty} x(t) = 0$, then I use the v_0 and p_1 to calculate the damping ratio as

$$\zeta_v = \sqrt{\frac{1}{1+\left(\frac{\pi}{\ln(-p_1/v_0)}\right)^2}}, \tag{1.62}$$

It should be noted that this method does require a first valley, which results in a damping ratio $\zeta_v < 1$. If $\zeta_v > 1$, then the damping is considered over-damped and the method will not work to estimate the damping ratio.

Coulomb Damping To determine a method for relating the lifetimes to the Coulomb damping constant μ_c and coulomb damping parameter ζ_c, I must first determine a theoretical expression for the response of a spring mass damper with only Coulomb damping. To do this, I will implement the method defined in [100]. However, I do acknowledge other methods for analyzing Coulomb damping (i.e. an energy approach [74]). Let us begin by defining the equation of motion from Eq (1.54) with $\mu_v = \mu_q = 0$, resulting in $m\ddot{x} = -kx - \mu_c N \text{sgn}(\dot{x})$. The solution to this differential equation is solved by breaking the system into two different states: (1) $\dot{x} > 0$ or (2) $\dot{x} < 0$, which each result in a unique (linear) differential equation. By “stitching" these solutions together I can get the solution as

$$x(t) = (A - \frac{2\mu_c N \omega_n}{\pi k} t) \cos(\omega_n t - \phi), \tag{1.63}$$

which has a linear amplitude decrement while $|A(1 - \frac{2\mu_c N\omega_n}{\pi k} t)| > \mu_s N/k$ with the phase shift ϕ introduced from other initial conditions. If the inequality is broken at sticking time t_s, then

$$x(t > t_s) = (A - \frac{2\mu_c N \omega_n}{\pi k} t_s) \cos(\omega_n t_s - \phi) \tag{1.64}$$

An example of this linear decrement and the sticking condition are shown in Fig. 1.12.

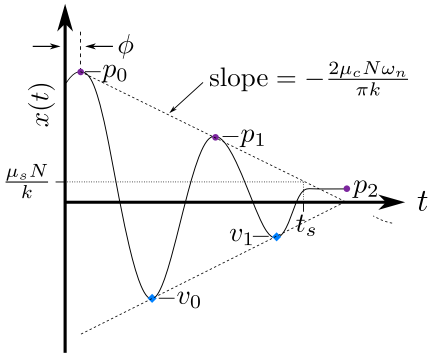


Figure 1.12: Example free vibration response of system with Coulomb damping.

I now leverage this closed form solution to be used with sublevel set persistence. To do this I start by shifting Eq. 1.63 to have $t = \tau$, where τ is the time at the first valley or $\tau = (\phi - 1)/\omega_n$, which results in the shift form of the equation of motion as

$$x(\tau) = (A - \frac{2\mu_c N \omega_n}{\pi k} \tau) \cos(\omega_n \tau). \tag{1.65}$$

From Eq. (1.65), the peaks p_i occur at $\tau = 2i\pi/\omega_n$ and have values of $p_i = A - 4i\mu_c N/k$, and the valleys v_i occur at $\tau = \pi(2i+1)/\omega_n$ with values of $v_i = 2(2i+1)\mu_c N/k - A$. The lifetime of the resulting persistence pairs are calculated as

$$L_i = p_{i+1} - v_i = 2A - \frac{(8i+6)\mu_c N}{k}. \tag{1.66}$$

Extending Eq. (1.66) to a second persistence pair results in the lifetime L_{i+n}, which is used to cancel the amplitudes with $L_{i+n} - L_i = \frac{-8n\mu_c N}{k}$. This difference is then used to solve for the Coulomb damping constant as

$$\mu_c = \frac{k(L_i - L_{i+n})}{8nN}. \tag{1.67}$$

With an expression for μ_c, the coulomb damping parameter ζ_c is estimated since it is independent of other system parameters (N and k). This parameter is the magnitude of the slope of the decrement and is solved for using Eq. (1.67) as

$$\zeta_c = \frac{2\mu_c N\omega_n}{\pi k} = \frac{\omega_n(L_i - L_{i+n})}{4n\pi} = \frac{(L_i - L_{i+n})}{2(t_{B_{i+n}} - t_{B_i})}, \tag{1.68}$$

where t_{B_i} is the time when L_i was born or at the time indice of the local minima. Similar to Viscous damping, I can also use a single lifetime to estimate both μ_c and ζ_c. To do this, I again assume that the time series $x(t)$ is zero centered. If so, the damping constant and parameter are calculated as

$$\mu_c = -\frac{k(v_0 + p_1)}{2N} \tag{1.69}$$

and

$$\zeta_c = \frac{2\mu_c N\omega_n}{\pi k} = -\frac{\omega_n(v_0 + p_1)}{\pi} = \frac{v_0 + p_1}{t_{v_0} - t_{p_1}}, \tag{1.70}$$

where t_{p_1} and t_{v_0} are the time indices at the local maxima and minima, respectively.

If the damping mechanism of a system is dominated by both viscous and coulomb damping I suggest implementing the amplitude decrement described by Liang and Feeny [132] in combination with sublevel set persistence.

Quadratic Damping For quadratic damping, Eq. (1.54) is reduced to $m\ddot{x} = -kx - \mu_q \mathrm{sgn}(\dot{x})\dot{x}^2$, which is a non-linear differential equation that does not have a closed form solution. However, there is a solution for calculating the turning points of the solution $x(t)$ [75]. For estimating the damping constant μ_q and the associated parameter ζ_q I use these turning points, which are determined by first splitting the equation of motion into two states as

$$0 = \begin{cases} \ddot{x} + \frac{\mu_q}{m}(\dot{x})\dot{x}^2 + \frac{k}{m}x, & \dot{x} > 0 \\ \ddot{x} - \frac{\mu_q}{m}(\dot{x})\dot{x}^2 + \frac{k}{m}x, & \dot{x} < 0. \end{cases} \tag{1.71}$$

Similar to the solution method for coulomb damping, quadratic damping requires the solution to be solved Iteratively between the two possible equations of motion in Eq. (1.71) as $\mathrm{sgn}(\dot{x})$ alternates. Fay [75] uses an an integration multiplier to show that the differential equation

$$\ddot{x} + p(x)\dot{x}^2 + f(x) = 0 \tag{1.72}$$

has the solution form

$$\mu(x)\frac{y^2}{2} + \int_{x_0}^{x} \mu(\epsilon)f(\epsilon)d\epsilon = \frac{y_0^2}{2}\mu(x_0), \tag{1.73}$$

where $\mu(x) = e^{\int 2p(x)dx}$. By applying this solution to the equation of motion with $p(x) = \pm\mu_q$ (the ± represents the two possible conditions with + if $\dot{x} > 0$), $\mu(x) = e^{\pm 2\mu_q x/m}$, and $f(x) = kx$, I solve the equation as

$$\frac{\dot{x}^2}{2}e^{\frac{\pm 2\mu_q}{m}x} + \frac{k}{m}\int_{x_0}^{x} e^{\frac{\pm 2\mu_q}{m}\epsilon}\epsilon d\epsilon = \frac{\dot{x}_0^2}{2}e^{\frac{\pm 2\mu_q}{m}x_0}. \tag{1.74}$$

The integral is then solved using the method of integration by parts as

$$\left(\dot{x}^2 + \frac{k\left(x - \frac{m}{\pm 2\mu_q}\right)}{\pm\mu_q}\right)e^{\frac{\pm 2\mu_q}{m}x} = \left(\dot{x}_0^2 + \frac{kx_0}{\pm\mu_q} - \frac{km}{2\mu_q^2}\right)e^{\frac{\pm 2\mu_q}{m}x_0}. \tag{1.75}$$

Equation (1.75) is then numerically solved iteratively as the solution goes through $\dot{x} = 0$. However, I would like to use an expression for the relationship between a valley and the following peak to understand how the lifetimes decrease due to the quadratic damping mechanism. To do this I first assume any initial condition $[|x_0|, |\dot{x}_0|] \neq \mathbf{0}$, which will yield a solution $x(t)$ that will eventually go to a valley. I then consider the new initial condition $\mathbf{x}_0' = [v_0, +0]$ at this first valley v_0 (see fig. 1.13 for a sample response with non-zero initial conditions). The velocity is positive ($\dot{x} > 0$) between

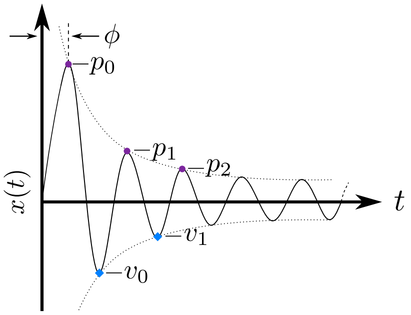


Figure 1.13: Example free vibration response of system with quadratic damping.

this first valley v_0 and until the next peak p_1. Therefor, I can use Eq. (1.75) with $+\mu_q$ to solve for the relationship between any valley and peak pair as

$$e^{\frac{2\mu_q}{m}p_{i+1}}\left(p_{i+1}-\frac{m}{2\mu_q}\right)=e^{\frac{2\mu_q}{m}v_i}\left(v_i-\frac{m}{2\mu_q}\right). \tag{1.76}$$

This relationship can be rearranged as

$$L_i = p_{i+1} - vi = \frac{m}{2\mu_q}\ln\left(\frac{2\mu_q v_i - m}{2\mu_q p_{i+1} - m}\right). \tag{1.77}$$

After applying sublevel set persistence and generating a persistent diagram, values for the lifetimes, valleys, and peaks are known, which allows for the numerical estimation of μ_q. This is done by minimizing the cost function

$$C(\mu_q) = \left[L_i - \frac{m}{2\mu_q}\ln\left(\frac{2\mu_q v_i - m}{2\mu_q p_{i+1} - m}\right)\right]^2. \tag{1.78}$$

where $C(\mu_q)$ is the cost as a function of μ_q. I can now also introduce the quadratic damping parameter $\zeta_q = \mu_q/m$. Applying ζ_q to Eqs. (1.78) results in

$$C(\zeta_q) = \left[L_i - \frac{1}{2\zeta_q}\ln\left(\frac{2\zeta_q v_i - 1}{2\zeta_q p_{i+1} - 1}\right)\right]^2. \tag{1.79}$$

which now has no needed system parameters m and k. Equation (1.79) can also be numerically minimized to estimate ζ_q.

1.3.2 Noise Compensation

While I have already developed expressions for estimating the damping parameters and constants from sublevel set persistence in Section 1.3.1, I need to develop an automatic framework for the method to be applied to real-world signals with inherent noise. To illustrate the effects of noise, let us return to the example sublevel set persistence from Fig. 1.2, but with additive noise as $x(t) + \mathcal{N}$. The resulting persistence diagrams from sublevel set persistence from the time series without $\mathcal{D}(x)$ and with additive noise $\mathcal{D}(x + \mathcal{N})$ are shown in Fig. 1.14 as well as the resulting time ordered lifetimes. This example shows that the addition of noise does not have a large effect on the position of signficant sublevel sets in the persistence diagram with the distances between signficant points (d_1, d_2, d_3, d_4) all being relatively small. This is no surprise due to the stability theorem of persistence diagrams [49]. However, additive noise does introduce several points in the persistence diagram located near the diagonal with relatively small lifetimes. These noise-artifact persistence pairs are formed from the peak-valley pairs in the additive noise. For the method of damping parameter estimation to function correctly, I needed to develop a method for dealing with these noise-artifact persistence pairs.

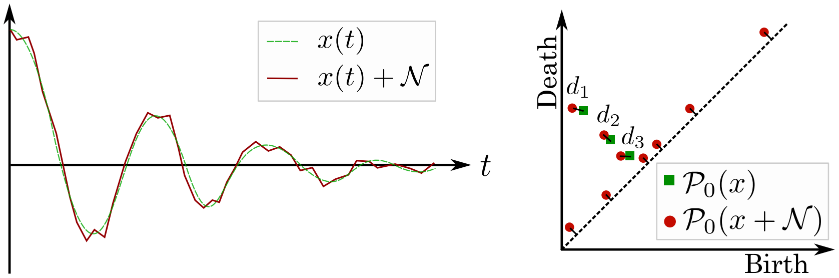


Figure 1.14: Sub-level set persistence applied to sample time series $x(t)$ with and without additive noise $\mathcal{N}$. This demonstrates the robustness of persistent homology with the time series (top left) with and without additive noise and the small effect on the resulting persistence diagrams (top right) and the corresponding time ordered lifetimes (bottom left).

One way of removing the noise-artifact persistence pairs is to seperate signficant and insignficant lifetimes through a confidence interval or cutoff. While there are methods for developing cutoffs based on a confidence set for persistence diagrams [40, 73], these methods often require that the time series sampling frequency is significantly higher than the highest dominate frequency of the time series or that the persistence diagram is generated from persistent homology and not sublevel set persistence. Both of these issues make implementing these methods difficult for persistence diagrams generated from sublevel set persistence. Additionally, methods such as persistent entropy [10] for separating noise from significant features in a persistence diagram may not properly distinguish between the noise and significant points if the number of significant data points in the persistence diagram is relatively large compared to the amount of noise. To combat both of these issues, I will introduce two methods for estimating the damping constant with additive noise using sublevel set persistence.

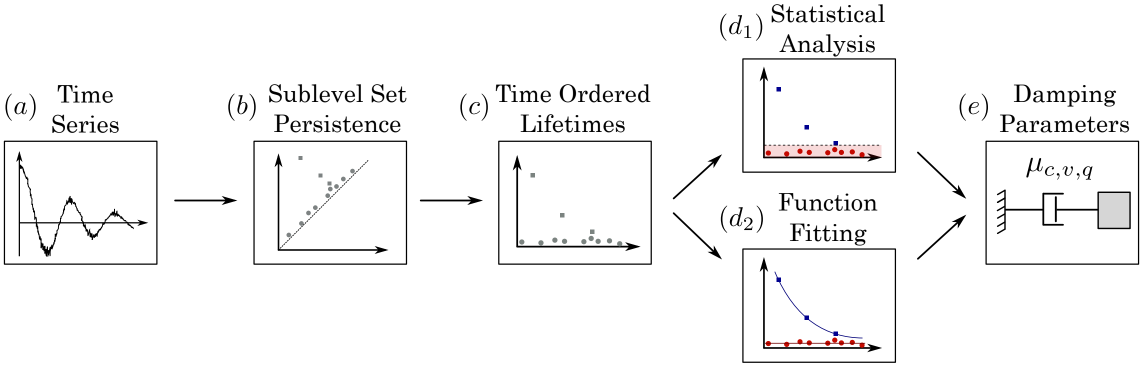

(a) Time Series
(b) Sublevel Set Persistence
(c) Time Ordered Lifetimes
(d1) Statistical Analysis
(d2) Function Fitting
(e) Damping Parameters
$\mu_{c,v,q}$

Figure 1.15: Overview of method: starting with a time series, the sublevel set persistence is calculated. The lifetimes from the persistence diagram are then plotted as a function of their birth time. The resulting diagram is analyzed from both a statistical and function fitting perspective to estimate the damping parameters.

The first method is based on generating a confidence level based cutoff for the persistence diagram for sublevel set persistence, which is founded on the assumed theoretical probability distribution $f(x)$ of noise in the persistence diagram developed in [11]. This assumed distribution allows for an accurate cutoff separating noise from features based on a desired confidence level α.

The second method uses a dual function fitting algorithm applied to the time ordered lifetimes diagram.Specifically one curve is fit to the damping envelope of the lifetimes while the second is fit to the additive noise lifetimes. However, this method is only viable for viscous and Coulomb damping as the envelope function is unknown for quadratic damping.

The aforementioned methods will be developed and discussed in the proceeding sub-sections as follows. First, in Section 1.3.3 I will provide an overview of the recently developed and novel analysis of the statistics of the lifetimes in the persistence diagram [11] and how its resulting cutoff can be used to separate significant persistence pairs from those associated to noise in the persistence diagram. These significant persistence pairs can then be used to estimate the damping parameters as discussed in Section 1.3.1. In Section 1.3.4, I will introduce the method based on a dual curve fitting procedure in the time ordered lifetimes diagram to estimate the damping parameters.

1.3.3 Method 1: Persistence Diagram Cutoff

The first method is based on calculating a suitable cutoff to seperate persistence pairs associated to additive noise from those of signal. To do this, I implement the recently published work on estimating a suitable cutoff for the persistence diagram (and time ordered lifetimes diagram) by assuming an additive noise distribution [11]. I overview the key results from this work in Section 1.3.3. I additionally develop a noise floor compensation term to minimize the effects additive noise has on the accuracy of the estimated damping parameters in Section 1.3.3. Finally, in Section 1.3.3 I show how the cutoff and noise floor are used to estimate the damping parameters.

Cutoff Equations For the method developed in [11], the cutoff equations require an assumed probability distribution function for the additive noise. Due to this constraint, I have provided four, commonly assumed probability distributions as Gaussian, uniform, Rayleigh, and Exponential distributions with their associated cutoff equations and approximated distribution parameters as shown in Table 1.3. From Table 1.3, $\tilde{L}$ is the median lifetime, n is the number of samples in the signal, α is the confidence level (this is usually chosen as 0.001), and σ, Δ, σ, and λ are the distribution parameters for the Gaussian, uniform, Rayleigh, and exponential distributions, respectively.

To compensate for the effects of signal on the cutoff and parameter estimation equations, I suggest the use of the multiplication compensation term for the signal as R. This term is used to compensate for the effects of signal with $C_\alpha^* = RC_\alpha$ and $\sigma^* = R\sigma$ and is calculated as

$$R = e^{c_1\left(\frac{\delta}{\delta+\tilde{L}}\right)^{c_2}}, \tag{1.80}$$

where the two constants c_1 and c_2 are provided in Table 1.4 and δ is approximated as

$$\delta \approx \frac{2}{n}\sum L_{C_\alpha}, \tag{1.81}$$

Table 1.3: Cutoff and parameter estimation equations for the Gaussian, uniform, Rayleigh, and exponential probability distribution functions.

Distribution	**Cutoff Equation C$_\alpha$**	**Parameter Estimation**
Gaussian	$1.923\tilde{L}\,\mathrm{erf}^{-1}\left[2(1-\sqrt{\alpha})^{1/n}-1\right]$	$\sigma \approx 0.680\tilde{L}$
Uniform	$2\tilde{L}\left[2\left(1-\sqrt{\alpha}\right)^{1/n}-1\right]$	$\Delta \approx 2\tilde{L}$
Rayleigh	$1.025\tilde{L}\left(\sqrt{-2\ln\left([1-\sqrt{\alpha}]^{1/n}\right)}-\sqrt{-2\ln\left(1-[1-\sqrt{\alpha}]^{1/n}\right)}\right)$	$\sigma \approx 1.025\tilde{L}$
Exponential	$-0.533\tilde{L}\ln\left([1-\sqrt{\alpha}]^{1/n}-[1-\sqrt{\alpha}]^{2/n}\right)$	$\lambda \approx \frac{1.875}{\tilde{L}}$

with L_{C_α} as the lifetimes greater than C_α.

Table 1.4: Constants of (1.80) for each distribution type investigated in this work with associated uncertainty from ten trials.

Distribution	Gussian	Uniform	Rayleigh	Exponential
c_1	0.845 ± 0.029	0.880 ± 0.017	0.726 ± 0.026	0.436 ± 0.036
c_2	0.809 ± 0.061	0.639 ± 0.026	0.605 ± 0.054	0.393 ± 0.075

Noise Floor A secondary effect on the lifetimes associated to signal from additive noise is the increase in the lifetimes, which I term as the “noise floor" F_β. For example, consider the sample peak-valley pair shown in Fig. 1.16, which illustrates $x(t)$ as the original time series without noise (blue dashed line), $x(t)+\mathcal{N}$ (red dot data points), and an increase and decrease in the local maxima and minima by approximately ϵ_{p_i} and ϵ_{v_i}, respectively. Additionally, from Fig. 1.16, I can approximate the original noise-free lifetimes

$$L_i^{'} \approx L_i - \epsilon_{L_i}, \tag{1.82}$$

where L_i is the lifetime associated to the signal with additive noise and ϵ_{L_i} is the uncertainty in the lifetime associated to signal from additive noise.

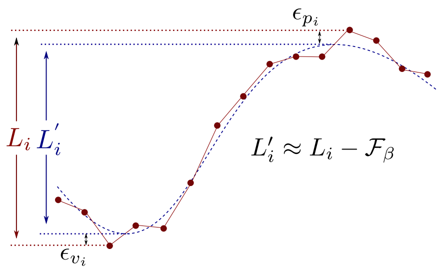


Figure 1.16: Example section of sampled time series $x(t)$ with (black dots) and without (green dashed line) additive noise to demonstrate effect of additive on increasing the lifetime of sublevel set persistence by approximately $L_i - L_i^{'} = \epsilon_{v_i} + \epsilon_{p_i} \approx \mathcal{F}_\beta$.

I attempt to approximate the increase in the lifetime from this uncertainty as the noise floor $F_\beta \approx \epsilon_{L_i}$. This uncertainty will generally increase the lifetime associated to signal and will consequently alter the calculations for the damping constants. Therefore, I will attempt to approximate $\mathcal{F}_\beta$ and reduce the measure lifetimes accordingly as $L_i - \mathcal{F}_\beta$.

It is straightforward to realize that ϵ_L is distributed the same as the lifetimes associated to additive noise. Therefore, the goal will be to approximate, on average, what the increase in L_i from additive noise using the previously derived statistics and resulting cutoff equations. Specifically, the goal is to represent the value of F_β as a function of the number of points near the local extrema n_e, the assumed additive noise model, and the approximate distribution parameter from the median lifetime with signal compensation (e.g. σ^* for Gaussian additive noise). To estimate F_β I will recycle the previously derived expressions from [11] in Table 1.3 as shown in Table 1.5. However, I must first develop a method to estimate n_e and an appropriate confidence level β.

I first choose an appropriate confidence level β. To determine β I consider the goal of the calculation: estimate the average increase in the lifetimes associated to signal from the additive noise near the extrema. Here, the key word is average. In comparison to the cutoff with $\alpha = 0.01$, I need a much higher confidence level for β due to the goal not being to provide a cutoff greater than the max of lifetimes associated to noise, but rather the average max itself. Therefore, I chose to set the probability as 50% or $\beta = 0.5$ such that there is an equal probability of increase in the lifetime being greater or less than the floor $\mathcal{F}_\beta$.

With β assumed as 0.5, I now need to determine n_e as the average number of points near

Table 1.5: Cutoff and parameter estimation equations for the Gaussian, uniform, Rayleigh, and exponential probability distribution functions.

Distribution	**Noise Floor F$_\beta$**
Gaussian	$2^{3/2}\sigma^* \operatorname{erf}^{-1}\left[2(1-\sqrt{\beta})^{1/n_e}-1\right]$
Uniform	$\Delta^*\left[2\left(1-\sqrt{\beta}\right)^{1/n_e}-1\right]$
Rayleigh	$\sigma^*\left(\sqrt{-2\ln\left([1-\sqrt{\beta}]^{1/n_e}\right)}-\sqrt{-2\ln\left(1-[1-\sqrt{\beta}]^{1/n_e}\right)}\right)$
Exponential	$-\frac{1}{\lambda^*}\ln\left([1-\sqrt{\beta}]^{1/n_e}-[1-\sqrt{\beta}]^{2/n_e}\right)$

the extrema of a lifetime associated to signal. I do not use the total number of data points n as only the points near the extrema have a significant probability of increasing L_i. Since I are working with signals of the underlying form $x(t) = A\sin(t+\phi)e(t)$ for damped oscillators with a damping envelope $e(t)$, I develop an expression for the number of samples near an extrema using the approximate response of the signal for a lifetime $L_i > C_\alpha^*$ as

$$f(x) = -\frac{L_i}{2}e(t_{B_i})\sin(t) \tag{1.83}$$

with the lifetime L_i born at t_{B_i} and $t \in [0, 2\pi]$. I consider points near an extrema when

$$|\sin(t)| \geq 1 - 2\frac{C_\alpha^*}{L_i}, \tag{1.84}$$

where $t \in [0, 2\pi]$. I now calculate the ratio between all $t \in [0, 2\pi]$ and the t that satisfy Eq. (1.84) as

$$r_i = \frac{\{\max(t) - \min(t),\ |\sin(t)| \geq 1 - 2\frac{C_\alpha^*}{L_i}\}}{2\pi}, \tag{1.85}$$

where $r_i \in [0, 1]$ and $t \in [0, 2\pi]$. r_i is estimated for each L_i with the average approximated as

$$r = \mathrm{median(r_i)}. \tag{1.86}$$

The total number of points in the signal with the damped sinusoidal function satisfying $Ae(t) > C_\alpha^*$ is estimated as

$$N = f_s(\max(t_B) - \min(t_B)), \tag{1.87}$$

where f_s is the sampling frequency and t_B is the set of birth times associated to lifetimes with $L_i > C_\alpha^*$. Using the total number of points associated to signal N and the ratio of those points near the extrema, I now estimate the number of points near the extrema for a lifetime as

$$n_e = \frac{rN}{n_L}, \tag{1.88}$$

where n_L is the number of lifetimes with $L_i > C_\alpha^*$.

I can now implement the results for n_e, β, and the distribution parameter into the cutoff equations from Table 1.3 as shown in Table 1.5 to calculate a noise floor $\mathcal{F}_\beta$. As a note, the noise floor compensation does not have a major effect for relatively low levels of noise (e.g. SNR > 30 dB). However, for higher levels of noise the compensation can be critical for calculating an accurate estimate of the damping constant. The importance of the noise floor compensation will be shown in Section 4.3.

Damping Parameter Estimation The damping parameters are estimated using the cutoff and noise floor as follows:

1. Calculate the lifetimes from the persistence diagram $L = \alpha_D - \alpha_B$ and match them with the time indices of the lifetime minima as t_B. This allows for the time ordered lifetimes plot as shown in Fig. 1.15.

2. With the cutoff C_α known, separate the lifetimes and birth times based on the $L > C_\alpha$. Adjust the lifetime above the cutoff using the noise floor by substituting L_i with $L_i - F_\beta$, p_i with $p_i - F_\beta/2$, and v_i with $v_i + F_\beta/2$.

3. Using the noise floor adjusted lifetimes above the cutoff and their time indices t_B, use the appropriate equation for estimating the damping constant for Coulomb, viscous, or quadratic

damping (see equation reference in Table 1.6). Additionally, I suggest using $i = 0$ and n as the lifetime closest to $0.3211\max(L)$ to minimize the effect of additive noise as shown in [137].

Table 1.6: Quick reference to equations (or cost functions) for using sublevel set persistence to estimate damping parameters and constants.

	Coulomb	**Viscous**	**Quadratic**
Parameter ζ:	$\frac{(L_i - L_{i+n})}{2(t_{B_{i+n}} - t_{B_i})}$	$\sqrt{\frac{1}{1+\left(\frac{2n\pi}{\ln(L_{i+n}/L_i)}\right)^2}}$	$C(\zeta_q) = \left[L_i - \frac{1}{2\zeta_q}\ln\left(\frac{2\zeta_q v_i - 1}{2\zeta_q p_{i+1} - 1}\right)\right]^2$
Constant μ:	$\frac{k(L_i - L_{i+n})}{8nN}$	$\frac{2\zeta_v k}{\omega_n}$	$C(\mu_q) = \left[L_i - \frac{m}{2\mu_q}\ln\left(\frac{2\mu_q v_i - m}{2\mu_q p_{i+1} - m}\right)\right]^2$

1.3.4 Method 2: Function Fitting to the Persistence Space

The second method is based on function fitting to the time ordered lifetimes. As mentioned previously, when calculating the sublevel set persistence diagram for a time series with additive noise, the persistence pairs associated to noise populate the region near the diagonal. Similarly, for the time-ordered lifetime plot the lifetimes of the persistence pairs associated to noise L_N will be near the x-axis and the lifetimes from the persistence pairs associated to signal L_S will capture the damping envelope as shown in the example signal in Fig. 1.17. I leverage this result as a method of filtering the noise in time series such that I can apply a function fitting to the lifetimes associated to signal.

Using all the lifetimes $L = L_N \cup L_S$, I fit two functions of the form $f_N(t) = b$ and $f_S(t) = e(t) + b$, where $e(t)$ is the envelope function for the lifetimes based on the damping type (Coulomb or viscous) with b as a constant to account for the noise offset. f_N and f_S are fit to L_N and L_S, respectively. In Fig. 1.17 I demonstrate this dual function fitting method. Based on this methodology, I can use the fitting parameters of $e(t)$ to determine the damping constants.

For viscous damping I estimate the envelope function as $e(t) = a\mathrm{e}^{-ct}$, where a and c are constant parameters. The exponent parameter c correlates with Eq. (1.59) and Eq. (1.55) with

$$\zeta_v = c/\omega_n = \frac{nc(t_{B_{i+n}} - t_{B_i})}{2\pi}. \tag{1.89}$$

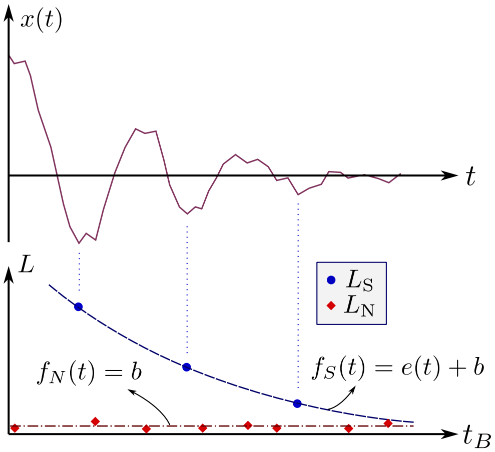


Figure 1.17: Example demonstrating process of going from a time series $x(t)$ with amplitude decrement and additive noise to the time ordered lifetimes of the persistence diagram with dual function fitting.

Additionally, $\mu_v = 2mc$. For Coulomb damping I estimate the envelope function as $e(t) = -at + d$, where a is the magnitude of the slope of the linear function and d is the intercept. I use the relationship in Eq. 1.65 to calculate the coulomb damping ratio as $\zeta_c = a/2$, which is extended to

$$\mu_c = \frac{\zeta_c \pi k}{2N\omega_n} = \frac{a\pi k}{4N\omega_n}. \tag{1.90}$$

I have now demonstrated how the function fitting method can easily be used to estimate the damping parameters from the lifetime plot (example illustrated in Fig. 1.17). This methods has the benefit of not needed a statistical analysis of the noise in the persistence diagram. However, the method does require an extra computational step of function fitting.

For function fitting I use a unique cost function for fitting two curves simultaneously, which is defined as

$$C = \sum_{i=0}^{T} \min\left([L_i - f_N(t_{B_i})]^2, [L_i - f_S(t_{B_i})]^2 \right), \tag{1.91}$$

where the cost function C is a function of the parameters a, b, c for viscous damping and a, b, d for Coulomb damping. Additionally, the subscript i of L_i and t_{B_i} denote the i^{th} sublevel set lifetime of all T lifetimes such that $i \in [1, T]$. I minimize Eq. (1.91) using Python's *scipy.optimize.minimize* implementation of the Broyden-Fletcher-Goldfarb-Shanno (BFGS) minimization algorithm.

A required input for the BFGS algorithm is an initial guess of the unknown parameter values. For viscous damping, I suggest the following estimations: $a = \max(L)$, $b = max(L)/100$, and $c =$

$\ln(1/0.3299)/t_{\text{opt}}$, where t_{opt} is the birth time of the lifetime nearest to $0.3299\max(L) \neq \max(L)$. For Coulomb damping, I make the following estimations: $b = 0.1\max(L), m = \max(L)/t_opt$, and $d = \max(L)$. Through simulations I have found that these initial guesses yield accurate results for a wide range of parameter values as demonstrated in the Section 4.3.

1.3.5 Examples

I will now implement the method for three examples. The first example is a simulated viscously damped oscillator, the second is an experimental single pendulum with damping dominated by the Coulomb damping mechanism, and the third is a simulated quadratically damped oscillator.

Example 1: Viscously Damped Oscillator For the first example, the system analyzed is the free response of the viscously damped oscillator described by $m\ddot{x} + kx + \mu_v\dot{x} = 0$, where $m = 1$ kg, $k = 20$ N/m, and $\mu_v = 0.5$ Ns/m. This system is solved as Eq. (1.92) with initial conditions $x_0 = 1$ m and $\dot{x}_0 = 0$ m/s as

$$x(t) = e^{-\zeta\omega_n t}\cos(\omega_d t), \tag{1.92}$$

where $\omega_n = \sqrt{k/m} \approx 4.472$ rad/s, $\zeta = 0.05590$, and $\omega_d = 4.465$ rad/s.

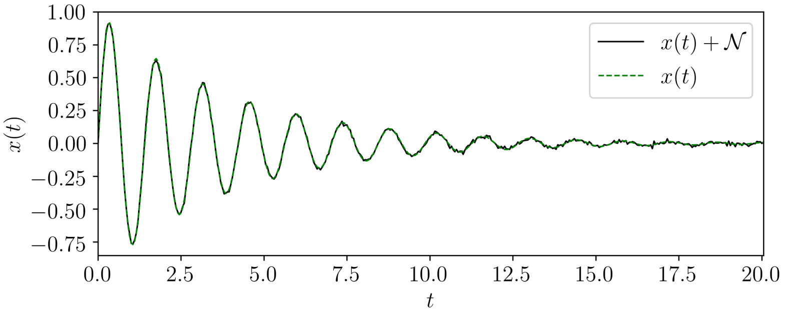


Figure 1.18: Time series $x(t)$ sampled at 20 Hz from Eq. (1.92) with and without additive noise $\mathcal{N}$ from a normal distribution with standard deviation $\sigma = 0.01$.

The simulation was sampled at a rate of $f_s = 20$ Hz for 20 seconds with additive noise $\mathcal{N}$ from a Gaussian distribution with a standard deviation $\sigma \approx 0.01$ m as shown in Fig. 1.18.

Sub-level set persistent homology is applied to the time series with and without additive noise as $\mathcal{P}_0(x + \mathcal{N})$ and $\mathcal{P}_0(x)$. The lifetimes L and their time indices t_B are then calculated from the persistence diagram and time series, respectively. As mentioned previously, the persistence diagrams with and without additive noise show only slight differences for the significant lifetimes. I can then apply both the statistics based analysis (see left side of Fig. 1.19) and function fitting analysis (see right side of Fig. 1.19) to the resulting lifetimes and time indices.

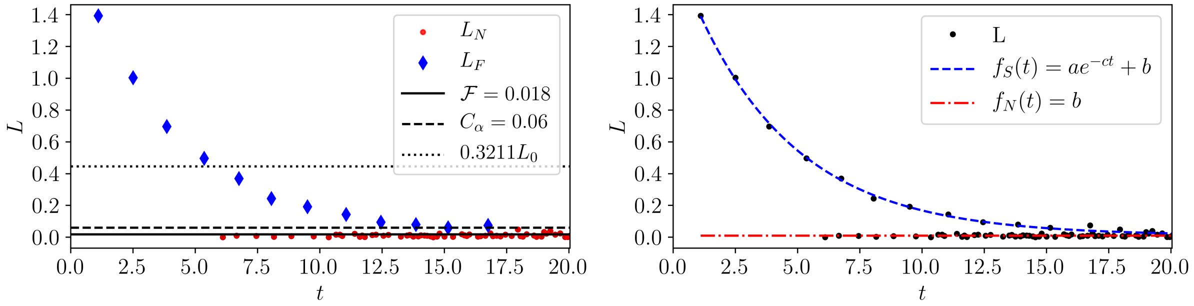


Figure 1.19: Resulting time-ordered lifetimes plot for the viscous damping mechanism example in Fig. 1.18 with (left) the statistical analysis and (right) function fitting.

Using the lifetimes from the persistence diagram, a cutoff $C_\alpha = 0.119$ is calculated using $\alpha = 1\%$. To calculate the damping constant, the lifetime indices are chosen as $i = 0$ and $n = 3$ so that $L_{n+i}/L_i \approx 0.3211$ ($L_3/L_0 \approx 0.583/1.542 \approx 0.378$) as suggested in [137]. Using these lifetimes, ζ_v is calculated from Eq. (1.61) as

$$\zeta_v = \sqrt{\frac{1}{1 + \left(\frac{2n\pi}{\ln(L_{i+n}/L_i)}\right)^2}} = \sqrt{\frac{1}{1 + \left(\frac{6\pi}{\ln(L_3/L_0)}\right)^2}} \approx 0.05480.$$

Using ζ_v I can then calculate $\mu_v = 2\zeta_v\sqrt{km} \approx 0.4901$. As noticed, both of these values a slightly below the theoretical values of $\zeta_v = 0.05590$ and $\mu_v = 0.5$. To improve the estimation, I can account for a noise floor in the calculation of ζ_v as

$$\mu_v = 2\sqrt{\frac{1}{1 + \left(\frac{6\pi}{\ln(L_3 - \mathcal{F}/L_0 - \mathcal{F})}\right)^2}} \approx 0.05611,$$

where $\mathcal{F} \approx 0.018$ was calculated as described in Section 1.3.3. I then calculate $\mu_v = 2\zeta_v\sqrt{km} \approx 0.5019$, which is significantly closer to the actual $\mu_v = 0.5$. Accounting for the noise floor becomes more critical as the noise level increases, which will be investigated more thoroughly in section 4.3.

For the second method, I implement the dual function fitting analysis as shown on the right side of Fig. 1.19. This analysis results in the constant $c \approx 0.2475$, which is used in Eq. (1.89) to calculate $\zeta \approx 0.4955$. I then calculate $\mu_v \approx 0.05540$. This shows that the dual function fitting method also work wells for estimating the damping constants, but the statistics based method with a noise floor compensation is slightly more accurate.

Example 2: Experimental Single Pendulum The second example uses data collected from a free drop experiment of a bench top pendulum within the linear range of oscillations. The pendulum used has CAD and design documentation provided through GitHub[1] with uncertainty analysis [183]. This single pendulum has an approximate system model of the form $I\ddot{\theta} = -\mu_c \mathrm{sgn}(\dot{\theta}) - mgr_{\mathrm{cm}}\theta$, where $I = I_{\mathrm{cm}} + mr_{\mathrm{cm}}^2$ with r_{cm} as the radius to the center of mass and I_{cm} as the inertia about the center of mass. This equation can be compared to Eq. (1.54) with $\mu_v = \mu_q = 0$. This comparison results in equivalence of $m = I$ and $k = mgr_{\mathrm{cm}}$.

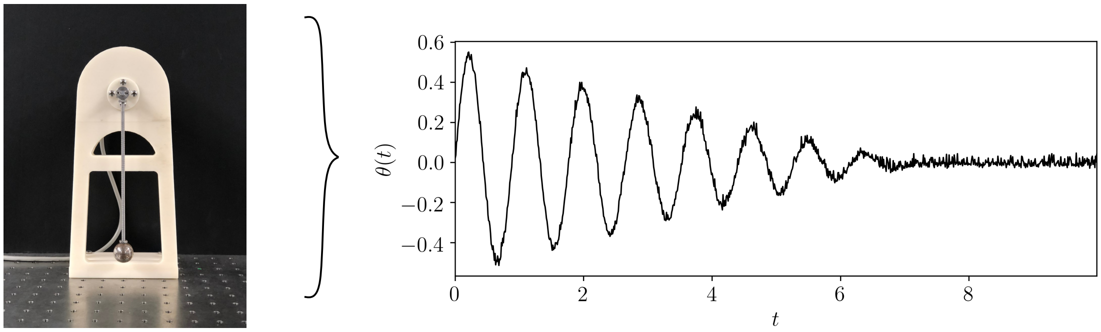


Figure 1.20: Time series $x(t)$ sampled at 20 Hz from Eq. (1.92) with and without additive noise $\mathcal{N}$ from a normal distribution with standard deviation $\sigma = 0.01$.

For the pendulum model it is assumed that the other damping mechanisms are negligible in comparison to the Coulomb damping. To validate this assumption, I implemented the BFGS algorithm for fitting a simulation of the model to collected free drop data, where the three damping constants μ_c, μ_v, and μ_q were the only unknowns. This required acurate estimates for the m, r_{cm}, and I. These parameters were estimated with either a direct measurement or through SolidWorks' *mass properties* tool with an accurate CAD model, which resulted in values of $m \approx 0.1231$ kg, $I \approx 0.00295$ kg m^2, and $r_{\mathrm{cm}} \approx 0.128$ m. From 5 free drops, the model fitting resulted in estimated average damping parameters with uncertainties (one standard deviation) of $\mu_c = (2.56 \pm 0.09) \times 10^{-3}$,

[1] https://github.com/Khasawneh-Lab/simple_pendulum

$\mu_v = (1.20 \pm 0.32) \times 10^{-4}$, and $\mu_q = (6.0 \pm 2.2) \times 10^{-6}$. These parameter values show that a large majority of the damping occurred through Coulomb damping, which substantiates the reduced model for the pendulum.

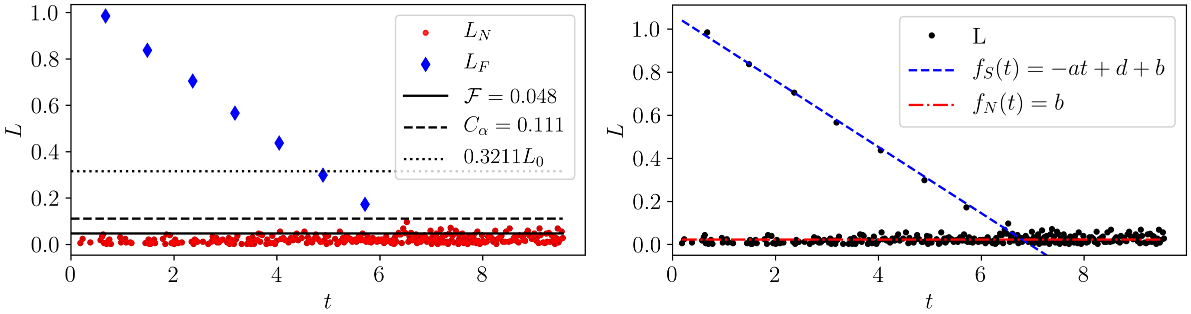


Figure 1.21: Resulting time-ordered lifetimes plot for the experimental pendulum data (see Fig. 1.20) having an approximate Coulomb damping mechanism in the linear range with (left) the statistical analysis and (right) function fitting.

The collected angular data (in radians) is shown in Fig. 1.20. Next, similar to the first example, the time ordered lifetimes are calculated using sublevel set persistence. I can then apply both the statistics based analysis (see left side of Fig. 1.21) and function fitting analysis (see right side of Fig. 1.21) to the resulting lifetimes and time indices.

I can now estimate the damping parameter (slope of decrement envelope) as $\zeta_c = (L_i - L_{i+n})/2(t_{B_{i+n}} - t_{B_i})$, where $i = 0$ and $n = 5$. This calculation results in $\zeta_c \approx 0.07909$. Similarly, I can use the function fitting method resulting in $a \approx 0.1538$, which is used to calculate $\zeta_c = a/2 \approx 0.07690$. Using ζ_c from the two methods, I can now calculate the damping constants as $\mu_c \approx 2.65 \times 10^{-3}$ and $\mu_c \approx 2.58 \times 10^{-3}$ for the statistics and function fitting methods, respectively. Both of these results fall within the uncertainty of the parameter estimated from model fitting ($\mu_c = (2.56 \pm 0.09) \times 10^{-3}$), which suggests that this method for damping estimation is viable for experimental data.

Example 3: Quadratically Damped Oscillator For the last example, and completion of damping types, I will again simulate a time series. However, I will now use quadratic damping as the mechanism of energy dissipation. To do this, I simulated a response of $m\ddot{x} + kx + \mu_q \dot{x}^2 \mathrm{sgn}(\dot{x}) = 0$ with initial conditions $x_0 = 1$ m and $\dot{x}_0 = 0$ m/s and parameters $m = 1$ kg, $k = 20$ N/m, and $\mu_q = 0.5$

Ns^2/m^2. The solution was sampled for 20 seconds at a sampling rate of 20 Hz. Additionally, I included additive noise $\mathcal{N}$ to the time series $x(t)$ from a Gaussian distribution with a standard deviation $\sigma \approx 0.01$ m as shown in Fig. 1.22.

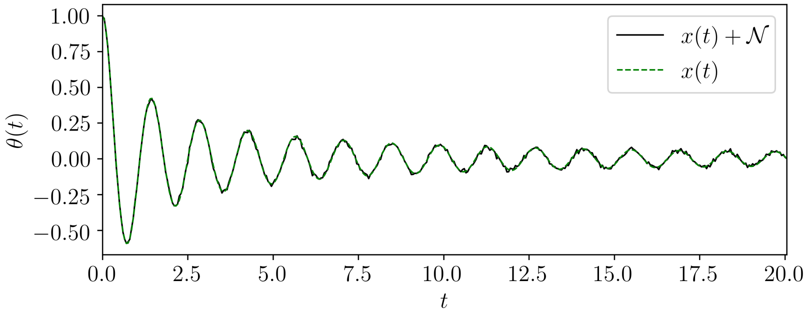


Figure 1.22: Time series $x(t)$ sampled at 20 Hz from the simulation of a quadratically damped oscillator with and without additive noise $\mathcal{N}$ from a normal distribution with standard deviation $\sigma = 0.01$.

Next, sublevel set persistence was applied to the time series with additive noise, and the corresponding birth times t_B and lifetimes L were recorded. A statistical analysis of the lifetimes was used to calculate a noise floor and cutoff as shown in Fig. 1.23. By minimizing the cost function

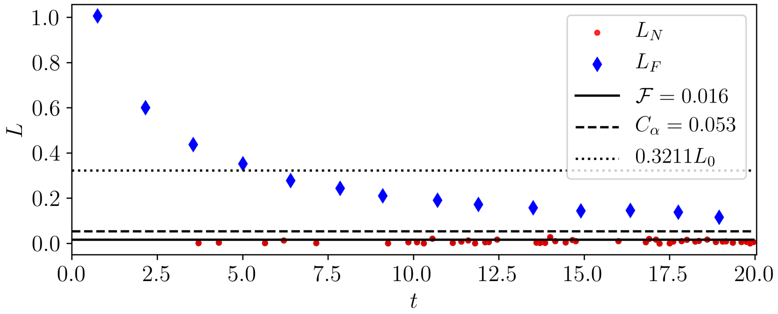


Figure 1.23: Resulting time-ordered lifetimes plot for the quadratic damping mechanism example in Fig. 1.22 with (left) the statistical analysis and (right) function fitting.

in Eq. (1.78) and Eq. (1.79), I calculate the damping constant and parameter as $\mu_q \approx 0.513$ and $\zeta_q \approx 0.513$, respectively. By comparing these values to the actual $\mu_q = 0.5$ and $\zeta_q = 0.5$, I can see that sublevel set persistence is an accurate an automatic method for estimating quadratic damping parameters.

1.3.6 Results

In this section I provide three main results of sublevel set persistence for damping parameter identification: noise robustness, functionality at low sampling frequencies, and applicability for a wide range of damping parameters. All three of these analyses are based on estimating damping parameters from the three different damping mechanisms with damping parameters of $\mu_c = 0.05$ N, $\mu_v = 0.5$ Ns/m, $\mu_q = 0.5$ Ns^2/m^2 for Coulomb, viscous, and quadratic damping, respectively. The other system parameters are set as $m = 1$ kg and $k = 20$ N/m with initial conditions $x_0 = 1$ m and $\dot{x}_0 = 0$ m/s. These systems are simulated for 20 seconds at a rate of 20 Hz unless specified otherwise.

Noise Robustness For analyzing noise robustness, I implement a sweep of a Signal-to-Noise-Ratio (SNR) from 15 to 40 dB, where a low SNR signifies a high level of noise. The SNR is defined as

$$\mathrm{SNR} = 20 \log_2 \left(\frac{A_{\mathrm{signal}}}{A_{\mathrm{noise}}} \right), \tag{1.93}$$

where $A_{\mathrm{signal}} = 1$ m as the maximum value of the signal (based on initial conditions), and $A_{\mathrm{noise}} = \sigma\sqrt{2}$ with σ as the standard deviation of the additive Gaussian noise. In signal processing an SNR of 15 dB is considered the limit for extracting useful information from a time series. At each SNR, I add Gaussian (normal distribution) noise with the specified SNR and estimate the damping constant using all three methods: single lifetime, optimal lifetime ratio, and function fitting. I compute these estimates for 100 samples at each SNR, which provide a mean and standard deviation and is represented as a data point with standard deviation error bars as $\mu \pm \sigma_\mu$ (see Fig. 1.24). I also ran two variations of the parameter estimations: one with and one without noise compensation. By noise compensation I are referring to a compensation of the noise floor in the damping parameter estimation as described in Section 1.3.3.

the goal with this noise robustness analysis is to determine the functional limits of each method with additive noise. On the left side of Fig. 1.24 I show the results with the automated noise

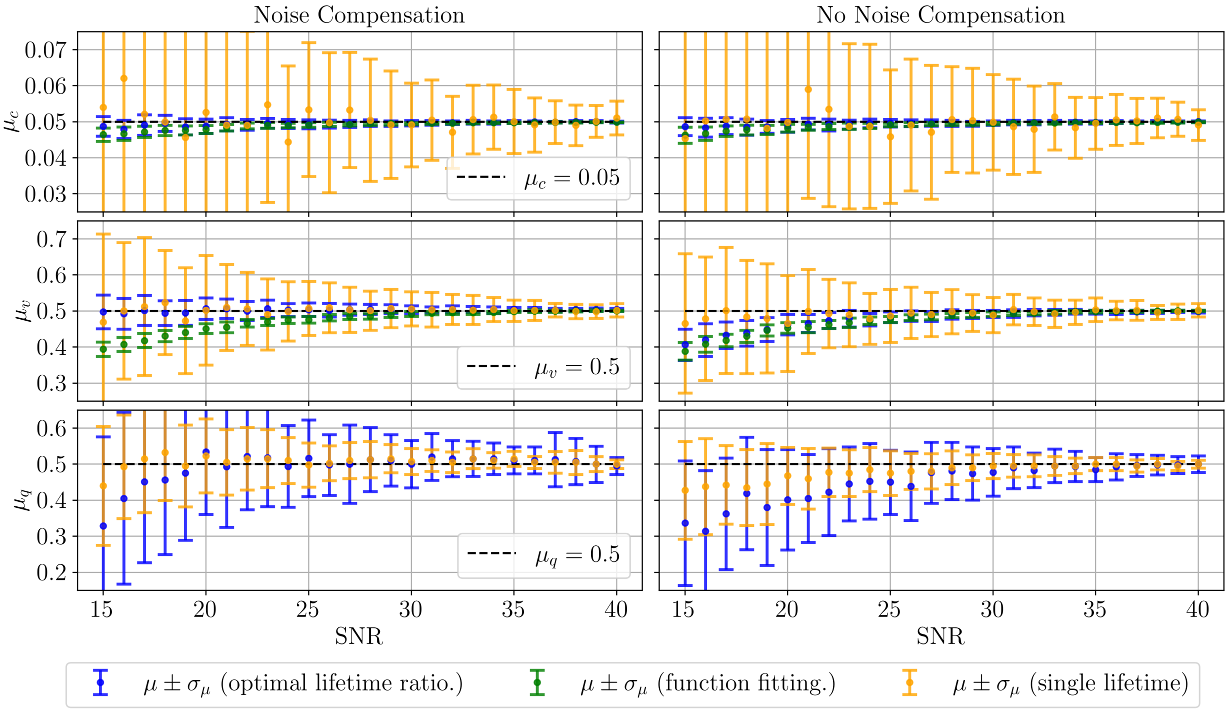


Figure 1.24: Analysis of the noise robustness of sublevel set persistence for damping parameter estimation of an oscillator with (top) coulomb, (middle) viscous, and (bottom) quadratic damping mechanisms with (left) and without (right) noise compensation. For each damping mechanism I estimate the damping parameters using a single lifetime (One), and optimal lifetime ratio (Opt.), and function fitting (Fit.).

compensation with the parameter estimation for Coulomb, viscous, and quadratic damping, from top to bottom.

On the top left I have the estimated coulomb parameters (actual $\mu_c = 0.05$ N), which shows that both the function fitting and optimal ratio methods accurately estimate the damping parameter all the way down to an SNR of 15 dB. However, the damping estimation has a large uncertainty when using only a single lifetime. This suggests that the single lifetime method should only be used for low noise levels or a high SNR. Additionally, on the top right of Fig. 1.24 I see almost no difference between the noise compensation and no noise compensation results suggesting that it is unnecessary to do noise compensation for Coulomb damping parameter estimation. This is most likely due to the approximately even increase in the lifetimes associated to additive noise on the lifetimes associated to signal has a minimal effect on the slope of the damping envelope.

For the middle row of figures in Fig. 1.24 the results for viscous damping are shown. These results demonstrate that only the optimal noise ratio with noise compensation accurately estimates the damping parameter ($\mu_v = 0.5$ Ns/m) at high levels of noise. At slightly lower levels of noise (SNR > 25), all three methods accurately estimate the damping parameters, but the function fitting method shows parameter estimation with higher accuracy. Similarly, for the no noise compensation case on the right, all three methods show accurate results when for SNR > 25 dB.

For the last damping parameter $\mu_q = 0.5$ Ns^2/m^2 on the bottom row, there is no function fitting method as there is currently no closed-form solution for the damping envelope function for quadratic damping. This means only the single and optimal lifetime methods can be used. Additionally, for improved accuracy I see that noise compensation is necessary for SNR values less than approximately 30 dB. I also notice that quadratic damping estimation is more sensitive to additive noise than Coulomb and viscous damping and only has a relatively high precision for low noise levels with the SNR greater than approximately 30 dB.

Effects of Sampling Frequency The second analysis of sublevel set persistence for damping parameter identification is based on the effects of sampling rate to determine the minimum sampling rate at which the method will continue to function accurately. To do this analysis I scaled the sampling frequency (originally 20 Hz) from 2 to 20 Hz. At frequencies lower than 4 Hz I are approaching the Nyquist sampling rate with $f_{\text{Nyquist}} = 2\omega_n \approx 1.42$ Hz and expect the method to fail. Additionally, I expect the accuracy will only improve for frequencies greater than 20 Hz. The additive noise level was left at 50 dB. At each frequency an uncertainty was added to the sampling frequency and the damping parameters were calculated for 100 samples. This allows for a mean and standard deviation on the parameter values (See Fig. 1.25).

This analysis shows that for all three damping mechanisms low sampling frequencies approaching the Nyquist sampling rate reduce the accuracy and precision of the parameter estimation. I also conclude that both the function fitting and the optimal lifetime ratio methods have similar results. However, both Coulomb and quadratic damping estimation show a a significantly higher level of uncertainty for sampling rates less than 4 Hz, which suggests that the time series should be sampled

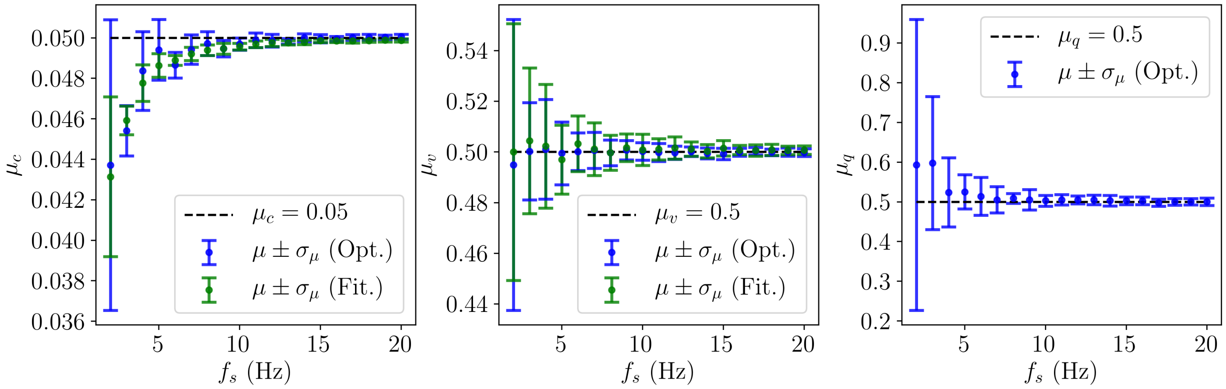


Figure 1.25: Effect of low sampling frequencies for the damping parameter identification methods based on sublevel set persistence for Coulomb (left), viscous (middle), and quadratic (right) damping mechanisms. Analysis shows accurate results for sampling rate $f_s > 2f_{\text{Nyquist}}$, where $f_{\text{Nyquist}} \approx 1.42$ Hz is the Nyquist sampling rate.

at rates greater than twice the Nyquist rate. On the other hand, the viscous damping parameter estimation showed accurate results up to the Nyquist sampling rate.

Effects of Damping Parameter Variation The last result and analysis is the effect of damping parameters to determine at what parameters the methods fail. To do this analysis, there is no additive noise and I will only consider significantly high damping parameters as small damping should not decrease the accuracy of the optimal lifetime ratio and function fitting methods. However, at low damping parameters and high noise levels, the accuracy of the method based on the first, single lifetime will become inaccurate (I do not show this result).

For Coulomb damping the range of damping constants μ_c will vary from 0.001 to 0.55 N, where at $\mu_c \approx 0.4N$ the sticking effect has a significant influence on the damping parameter estimation and causes the method to fail. For viscous damping I consider damping constants that result in damping parameters up to $\zeta_v = 1.0$ or $\mu_v \in [0.01, 8.5]$ Ns/m (i.e. critically damped). At $\zeta_v = 1.0$ the response has no oscillations, which results in no lifetimes and an upper limit for viscous damping. Finally, for quadratic damping the damping constant does not have a large influence on the accuracy, which is why I chose a large damping constant range $\mu_q = [0.01, 8.0]$ Ns^2/m^2.

Figure 1.26 shows the resulting damping constant estimates over the range of damping constants. For Coulomb damping both the function fitting and optimal lifetime ratio begin to lose accuracy

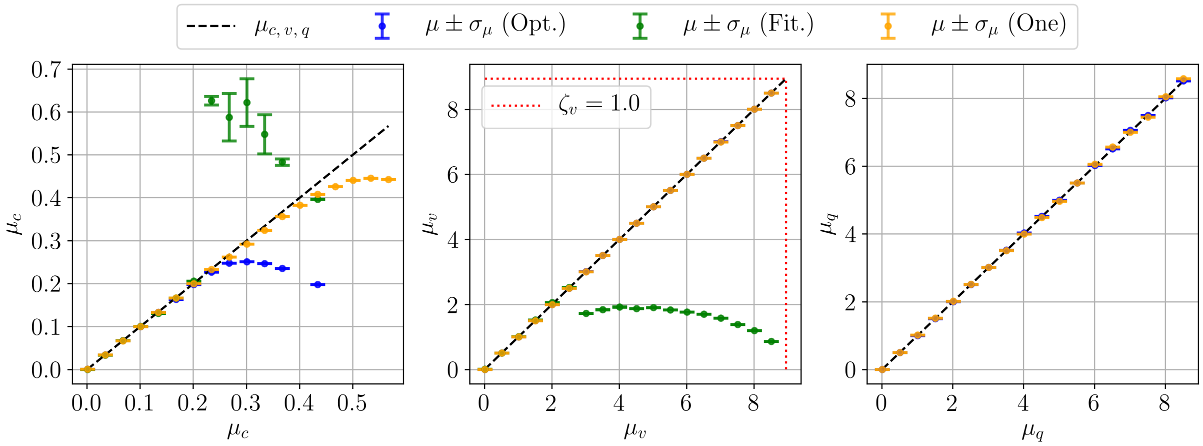


Figure 1.26: Effects of damping parameters of (left) Coulomb, (middle) viscous, and (right) quadratic damping. These parameter values are ranged from very low damping to high or critical damping values.

when the number of lifetimes decreases to one. This occurs at approximately $\mu_c = 0.2$ N. Additionally, at $\mu_c \approx 0.4$ N, the sticking effect of coulomb damping is effecting the single lifetime, which reduces the accuracy of the method based on a single lifetime. For viscous damping in the middle of Fig. 1.26, the function fitting method (Fit.) loses accuracy at approximately $\mu_v = 2.5$ Ns/m or $\zeta_v \approx 0.3$, the optimal lifetime ratio method loses accuracy at $\mu_v \approx 6$ Ns/m, and, finally, the method based on a single lifetime accurately estimates the damping constant almost all the way to $\zeta_v = 1.0$. For quadratic damping on the right, the damping estimation method functions accurately for the entire damping constant range. I theorize that the function fitting method loses accuracy for high levels of damping due to a lack of data points or lifetimes associated to signal for the function to fit to. This result shows the benefit of using the statistics based method for estimating the damping ratio since it is functional for higher damping levels.

1.4 Sublevel Set Entropy

1.4.1 Information Entropy Statistics

Entropy is used as a summary statistic for measuring the predictability of a data source based on the probability distribution across a set of discrete states. Information entropy was first realized as Shannon entropy, which was introduced in 1948 [210]. Since then, several new forms of entropy have been popularized for time series analysis. Some examples include approximate entropy [185], sample entropy [198], and permutation entropy [14]. Additionally, information entropy can be applied to transition probability matrices from a Markov chain representation through the entropy rate or conditional entropy. However, this application of entropy requires the time series to be represented as a sequence of discrete states.

In this work we show how each of these entropy statistics can be used to analyze the sublevel set persistence of a time series. The following paragraphs provide a brief introduction to each entropy measurement.

Shannon Entropy Shannon entropy [210] is calculated using the probability distribution of a set of possible states $\mathcal{A}$ from the sequence of states $\mathcal{S}$. Each state has its probability calculated based on frequency with state a_i having probability $p(a_i)$. The Shannon entropy is calculated as

$$H(\mathcal{S}) = -\sum_{i \in N} p(a_i) \log(p(a_i)), \tag{1.94}$$

where N is the number of possible states. Shannon entropy can be normalized as

$$h(\mathcal{S}) = -\frac{\sum_{i \in N} p(a_i) \log(p(a_i))}{\log(N)}, \tag{1.95}$$

with $h \in [0, 1]$. If each state is equiprobable over all possible states then the underlying dataset has a high level of uncertainty and $h = 1$. Conversely, if $h = 0$ then only one state has probability $p(a_i) = 1$, while all others have zero probability representing a perfectly regular dataset. A major issue for Shannon entropy, as described in the introduction, is that it does not account for the order in which the data is received. To alleviate this issue approximate entropy was created.

Approximate Entropy Unlike Shannon entropy, which can measure predictability using a probability distribution among the states, approximate entropy [185] measures the regularity of a signal based on the sequence of states. Additionally, it does not require distinct states with the use of the uncertainty or filtering level parameter r when comparing sequence segments. Unfortunately the choice of an appropriate r value is not trivial and is dependent on the application. Therefor, using a

```
Input: Signal $x = [x(0), x(1), \ldots, x(N-1)$ with $N$ as length of signal, filter level $r$, and data comparison length $m$.
Output: Approximate entropy $h_a$
Form collection of vectors $V_m = [\boldsymbol{v}_m(i), \ldots, \boldsymbol{v}_m(N-m)]$ with $\boldsymbol{v}_m(i) = [x(i), x(i+1), \ldots, x(i+m-1)] \in \mathbb{R}^m$ for each $i \in [0, N-m]$.
Calculate
    $C_i^m(r) = \frac{\#\{\boldsymbol{v}_m(j) \in V_m \mid d(\boldsymbol{v}_m(j), \boldsymbol{v}_m(i)) \leq r\}}{N-m+1}$,
  which measures the percent of vectors within a distance $r$ of vector $v(i)$ with the Chebyshev (or $L_\infty$) distance function as $d(\boldsymbol{a}, \boldsymbol{b}) = \max_i(|a_i - b_i|)$ where $a_i \in \boldsymbol{a}$ and $b_i \in \boldsymbol{b}$.
Define
    $\Phi^m(r) = \frac{1}{N-m+1} \sum_{i=0}^{N-m} \log(C_i^m(r))$.
Calculate the approximate entropy as
    $h_a(x) = \Phi^m(r) - \Phi^{m+1}(r)$.
```

Algorithm 1.1: Approximate Entropy

sequence of states makes the choice of $r = 0$ simple. The approximate entropy is calculated using the Algorithm 1.1 as follows.

The algorithm calculates the regularity of a sequence of states by comparing how many unique (with r filtering level) sequences of states of length m there are. For a periodic signal there would be relatively few unique sequences and thus a low approximate entropy. In comparison, a chaotic or patternless signal would have many unique sequences and a high approximate entropy.

Two major drawbacks exist for approximate entropy. The first is its high sensitivity to parameter selection [152] and second is its need for sufficiently long data. To alleviate the latter, sample entropy was devised.

Sample Entropy Sample entropy [198] is similar to approximate entropy in that it compares sequences of length m with filtration level r. However, sample entropy h_s has the benefit of having data length independence. Sample entropy is typically used for measuring signal complexity with applications in physiological time-series data [198], and it is calculated as

$$h_s(x) = -\log\frac{A}{B}, \tag{1.96}$$

where

$$A = \#\{[\boldsymbol{v}_m(i), \boldsymbol{v}_m(j)] \in [V_m, V_m] \mid d(\boldsymbol{v}_m(i), \boldsymbol{v}_m(j)) \leq r\}$$

and

$$B = \#\{[\boldsymbol{v}_{m+1}(i), \boldsymbol{v}_{m+1}(j)] \in [V_{m+1}, V_{m+1}] \mid d(\boldsymbol{v}_{m+1}(i), \boldsymbol{v}_{m+1}(j)) \leq r\}.$$

In this work we use $m = 3$ by default unless otherwise stated. Sample entropy is unfortunately still sensitive to the filtering level parameter r and is computationally demanding for large signals as demonstrated in Section 1.4.5. For more details on approximate and sample entropy we guide the reader to [61]

Permutation Entropy Permutation entropy [14] was developed as a more computationally efficient method for calculating the complexity of a sequence in comparison to approximate and sample entropies. Permutations are the ordinal partitions of sequences of time series data. Specifically, the sequences (or state space reconstruction vectors) are defined as $v_{n,\tau}(i) = [x(i), x(i+\tau), \ldots, x(i+(n-1)\tau)]$, where the signal x is discretely sampled from a data source that can be either continuous or discrete, n is the permutation dimension, and τ is the spacing between points in the signal. Each vector $v_{n,\tau}(i)$ can be categorized as one of $n!$ possible permutations. Applying this procedure over all $v_{n,\tau}(i)$ allows calculating the probability of each permutation. The Shannon entropy from Eq. (1.94) can then be used to calculate the permutation entropy as

$$h_p(\pi) = -\frac{\sum_{i\in n!} p(\pi_i)\log(p(\pi_i))}{\log(n!)}, \tag{1.97}$$

which is normalized to the range $[0, 1]$ by using the number of possible states $n!$. While computationally efficient, permutation entropy does not account for amplitude as is done with sample and approximate. It can therefor be sensitive to additive noise.

Markov Chain Entropies: Entropy Rate and Average Conditional Entropy Markovian based entropy statistics are calculated using a transition probability matrix. To create the transition probability matrix, a sequence of states is used to track transitions through an adjacency matrix A. For each transition from state a_i to state a_j, $A(i, j)$ is incremented by one. The adjacency matrix A is $|V| \times |V|$, where $|V|$ is the number of states observed. The adjacency matrix is used to form a one-step transition probability matrix according to

$$P(i, j) = \frac{A(i, j)}{\sum_{k=0}^{|V|-1} A(i, k)}. \tag{1.98}$$

The probability matrix now represents the probability of transitioning from state a_i to state a_j in one step. This transition probability matrix serves as a stochastic model of the time series dynamics. The goal is to then quantify the predictability of this stochastic Markov chain model to calculate its complexity.

The first tool for calculating its predictability is the average condition entropy $\bar{h}_c$, which measures the average normalized Shannon entropy of transitions for each states. It is calculated as

$$\bar{h}_c(\mathcal{S}) = -\frac{1}{\log(N)|V|} \sum_{i=0}^{|V|-1} \sum_{j=0}^{|V|-1} P(i, j) \log(P(i, j)). \tag{1.99}$$

The conditional entropy measures the models' predictability and complexity by quantifying the predictability of each state transition. If there is only one transition direction possible (e.g. from state s_i to state s_j) then the conditional entropy of state s_i would be zero. However, if it is possible to transition from s_i to many other states then the corresponding conditional entropy would be higher.

Similar to the average conditional entropy, the entropy rate h_r is calculated as the normalized Shannon entropy of the transition probabilities for all states but with a weighting of each state's entropy based on its stationary distribution μ. The entropy rate is calculated as

$$h_r(\mathcal{S}) = -\frac{1}{\log(N)} \sum_{i=0}^{|V|-1} \sum_{j=0}^{|V|-1} \mu_i P(i, j) \log(P(i, j)), \tag{1.100}$$

where we estimate μ based on the probability distribution over the states such that $\sum_i \mu_i = 1$. If the distribution is equiprobable, then the average conditional entropy is equivalent to the entropy rate. A drawback of the entropy rate and conditional entropy is that only the single step transition probability is investigated. In comparison, sample, approximate, and permutation entropy can analyze sequences of larger dimensions.

1.4.2 Method

In this work, we implement the entropy tools discussed in Section 1.4.1 to the sublevel set persistence diagram. Our method is outlined in the pipeline shown in Fig. 4.2. We begin with an oscillatory signal in Fig. 4.2 (a) and calculate the 0D sublevel set persistence diagram in Fig. 4.2 (b). Additionally, we separate persistence pairs associated with noise using a cutoff as described in Section 1.2. Next, we calculate and bin the chronologically ordered lifetimes (based on the birth times t_B) as shown in Fig. 4.2 (c). The chronological lifetimes are sorted based on the time index at which the persistence pair was born. At this stage, the approximate and sample entropy methods can be directly applied to the chronologically ordered lifetimes above the cutoff C_α with $r = 0.1 \max(L)$. A benefit of using sublevel set persistence and associated lifetimes to apply approximate and sample entropy is that it eliminates the need to analyze the multi-scale aspects of the signal. This is due to the sublevel sets naturally partitioning the data using the critical points of the signal. Additionally, using sublevel set persistence provides a much more compact representation of possibly lengthy time series. This reduces the computational demand of approximate and sample entropy thus enabling in-situ analysis even for long signals.

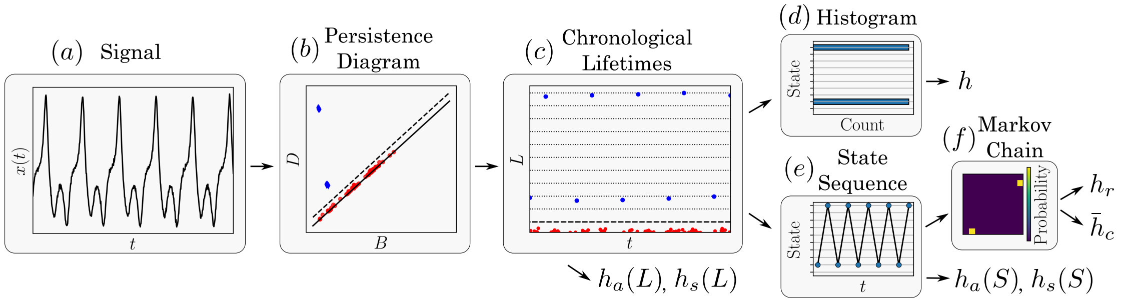


Figure 1.27: Pipeline for applying entropy metrics to the sublevel set persistence homology. The sublevel set persistence diagram in (b) is calculated from the signal in (a), which is used to calculate the lifetimes that are ordered chronologically based on their birth index in (c). The lifetimes can either be used to directly calculate the approximate and sample entropy as $h_a(L)$ and $h_s(L)$ or are then digitized into states based on the binning procedure in (d) and (e) with bin edges shown in (c). The probability of each state can be found to calculate the information entropy h. Additionally, the chronologically ordered states in (e) can be used to calculate the approximate and sample entropies $h_a(\mathcal{S})$ and $h_s(\mathcal{S})$, where $\mathcal{S}$ is the state sequence composed of states $a_i \in \mathcal{A}$. The entropy rate h_r and average conditional entropy $\bar{h}_c$ can also be calculated from the Markov chain matrix in (f).

A procedure for mapping the lifetimes L to a state sequence $\mathcal{S}$ is needed to implement the remaining entropy statistics. We use an equi-sized partitioning of the lifetimes within $[C_\alpha, \max(L)]$ into $\mathcal{B}$ bins. This method allows us to represent a signal with a small set of discrete states. Using the corresponding state sequence and each state's abundance, the information entropy $h(\mathcal{S})$ is calculated using the probabilities of each state as shown in Fig. 4.2 (d).

The state sequence $\mathcal{S}$ can also be used to calculate the approximate entropy $h_a(\mathcal{S})$ and sample entropy $h_s(\mathcal{S})$ as shown in Fig. 4.2 (e). The benefit of applying approximate and sample entropies to the state sequence is the simplicity of parameter selection with $r = 1$, which works well for $\mathcal{B} < 30$. For larger $\mathcal{B}$ values we suggest setting $r = 0.1\mathcal{B}$. While not shown in Fig. 4.2, approximate and sample entropy can also be applied directly to the signal in subfigure (a). However, this is significantly more computationally demanding which will be demonstrated in Section 4.3.

We can also use the state sequence $\mathcal{S}$ from the ordered lifetimes L to create a transition probability matrix P shown in Fig. 4.2 (f). The transition probability matrix or Markov chain matrix is then used to calculate the entropy rate h_r and average conditional entropy $\bar{h}_c$.

1.4.3 Example

To demonstrate the functionality of sublevel set entropy for dynamic state detection we use the popular Lorenz dynamical system

$$\frac{dx}{dt} = \sigma(y - x), \ \frac{dy}{dt} = x(\rho - z) - y, \ \frac{dz}{dt} = xy - \beta z, \tag{1.101}$$

with $\sigma = 10$, $\beta = 8/3$, and $\rho = 100$ for periodic while $\rho = 105$ for chaotic dynamics. For our analysis, we only use the x-solution to Eq. (4.3) which was simulated for 100 seconds at a sampling rate of 100 Hz. Only the last 20 seconds were used to avoid the transient response. The left column of Fig. 1.28 shows the resulting periodic (top panel) and chaotic (bottom panel) time series.

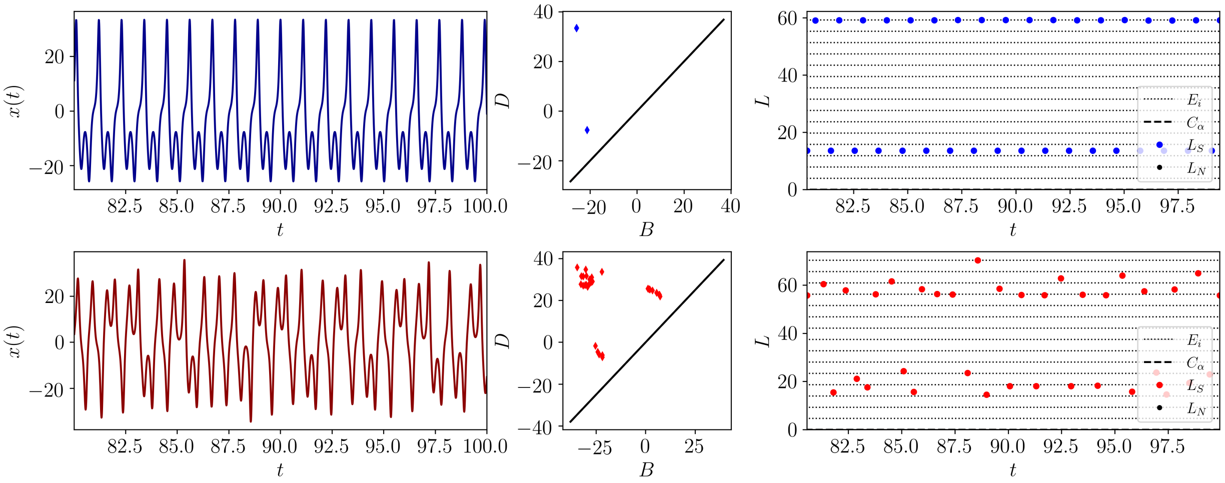


Figure 1.28: Example demonstrating sublevel set persistence of periodic (top row of figures) and chaotic (bottom row of figures) simulations of the Lorenz system. Each row shows the time series $x(t)$ (left), sublevel set persistence diagram (middle), and binned lifetimes (right).

In this example our goal is to identify periodic from chaotic dynamics through the sublevel set persistence. The persistence diagram on the top row of Fig. 1.28 shows that periodic signals tend to cluster points (persistence pairs) in a few locations on the persistence diagram (two points for this example). The goal is to then quantify this regularity in the persistence diagram. To do this we use the time ordered lifetimes which can easily be binned into $\mathcal{B}$ states that are above the cutoff C_α. For periodic signals we would expect the clustering of points in the persistence diagram to translate to a periodic sequence of states from the binned lifetimes as shown in the top-right subfigure of

Fig. 1.28. On the other hand, the chaotic signal will not have the same properties in the resulting state sequence as shown in the bottom-right subfigure of Fig. 1.28. The non-periodic behavior of a chaotic signal causes the points in the persistence diagram to not repeat. However, there may still be clusters in the persistence diagram from a chaotic signal due to the dynamics (strange attractor) of the system as shown in the bottom-center subfigure of Fig. 1.28.

We now implement our entropy statistics to the resulting periodic and chaotic persistence diagrams through the time-ordered lifetimes. For this example, we set $\mathcal{B} = 15$ based on the binning analysis done in Section 1.4.4. The information entropy is calculated from the probability distribution of states in the state sequence $\mathcal{S}$ derived from the binned lifetimes as shown in the frequency plot (see left column of Fig. 1.29). The entropy and associated probabilities result in $h \approx 0.2559$ for periodic and $h \approx 0.6522$ for chaotic dynamics. The periodic entropy is not zero since it is distributed over two states (4 and 15) and the chaotic is not one because it is not equiprobable over all states. However, the large difference between the two scores shows that entropy distinguishes between the two dynamic states.

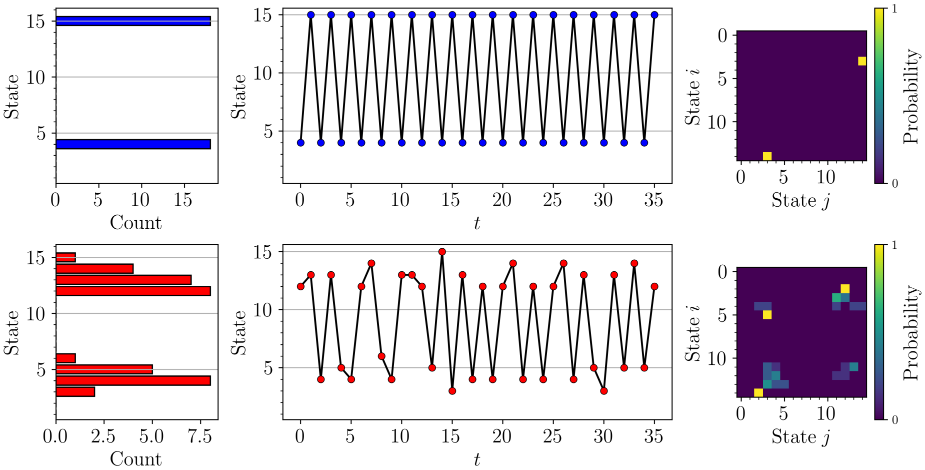


Figure 1.29: Further diagrams for entropy analysis of example signals in Fig. 1.28. The top row is again for the periodic signal and bottom for chaotic. The left column is the distribution of states, the middle is the state sequence, and the right is the 1-step transition probability matrix.

The approximate and sample entropies are calculated using the state sequence shown in the middle column of Fig. 1.29 as $h_a \approx 0.0004$ and $h_s \approx 0.0308$ for periodic and $h_a \approx 0.4921$ and $h_s \approx 0.7864$ for chaotic dynamics. The approximate and sample entropy are near zero due to

Table 1.7: Tabulated results for sublevel set entropy of Lorenz example

Entropy	**Periodic**	**Chaotic**
Information Entropy h	0.2559	0.6522
Approximate Entropy h_a	0.0004	0.4921
Sample Entropy h_s	0.0308	0.7864
Entropy Rate h_r	0	0.3832
Average Condition Entropy $\bar{h}_c$	0	0.3321

the regularity in the state sequence, while the chaotic signal results in significantly higher entropy values. In Section 4.3 we will demonstrate typical approximate and sample entropy values for a variety of chaotic systems.

The Markov chain transition probability matrix models the dynamics of the signal as a stochastic system. This modeling approach allows periodic signals with very high transition probabilities between specific states to have low state entropies and resulting entropy rate. Conversely, the chaotic signal has a distribution of transition probabilities between multiple states with lower probabilities. For example, the right column of Fig. 1.29 shows the transition probability matrix for the periodic and chaotic time series, where the periodic signal results in only two non-zero transitions with unit probability and the chaotic signal has transitions between multiple states with lower probabilities. The resulting entropy rate and average conditional entropy are $h_r = 0$ and $\bar{h}_c = 0$ for periodic and $h_r \approx 0.3832$ and $\bar{h}_c \approx 0.3321$ for chaotic dynamics. The entropy rate of 0 for the periodic signal is due to each state having unit probability.

The results for all of the entropy statistics are summarized in Table 1.7. Comparing the two columns of Table 1.7 illustrates that the entropy statistics based on the sublevel set persistence can identify periodic and chaotic dynamics for the Lorenz system.

In the next section, we further challenge our method using a large number of flows and maps, and we show that the ability of the sublevel set entropy to distinguish periodic from chaotic dynamics is evident for a variety of dynamical systems.

1.4.4 Analysis on the Number of Bins

The first result needed is an analysis of the effects of the number of bins on the entropy values for periodic and chaotic dynamics. To gain a universal understanding of these effects, we used 21 continuous dynamical systems and 15 maps (see Table C.1 in appendix), with each having periodic and chaotic dynamics. We ranged the number of bins $\mathcal{B}$ from 2 to 50, which demonstrated several characteristics as shown in Fig. 1.30.

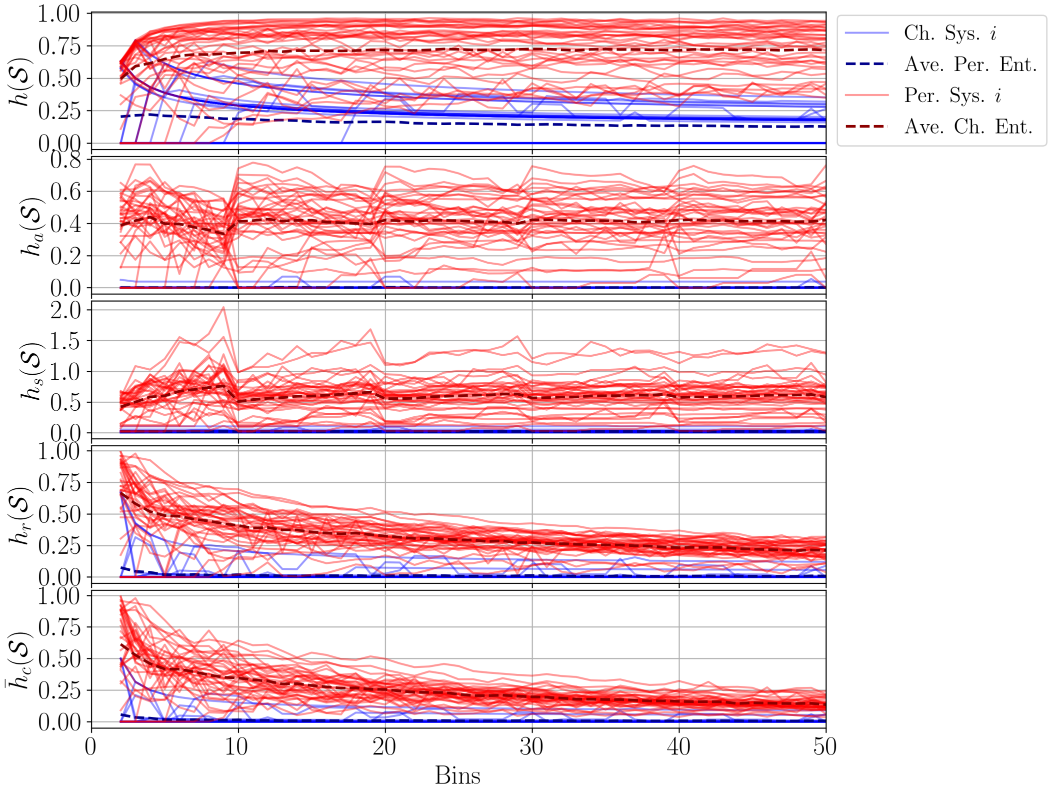


Figure 1.30: Analysis on effect of number of bins or states on entropy values for 18 continuous and 12 discrete dynamical systems.

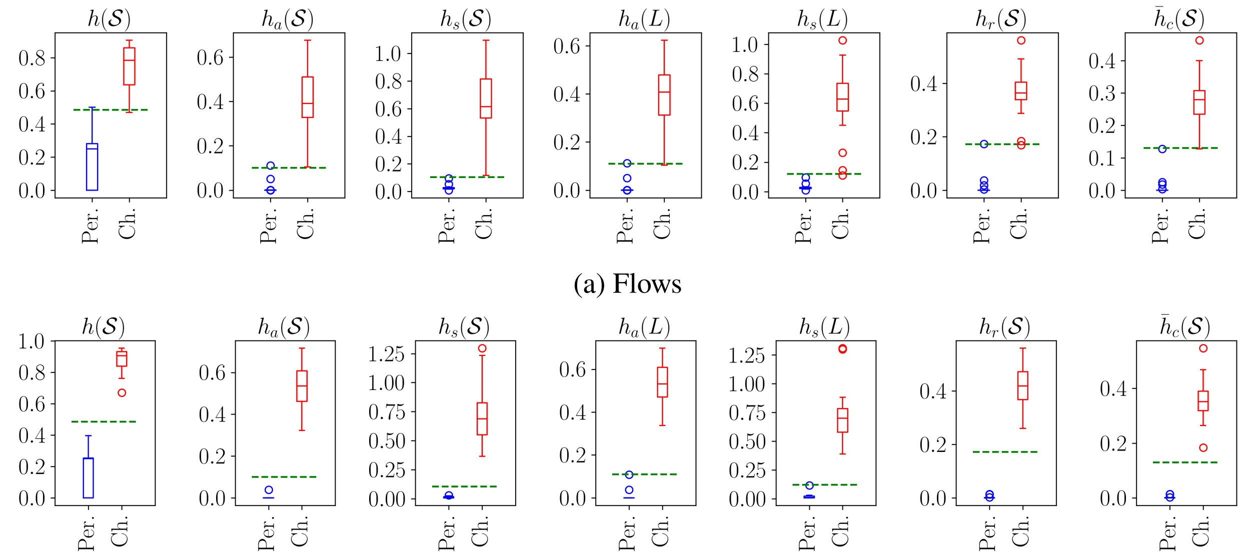
First, the separation between periodic and chaotic dynamics based on the entropy values tends to plateau at approximately $\mathcal{B} = 15$ bins. We also note that there seems to be very little differentiation between the entropy distributions for maps and flows, suggesting that 15 bins are appropriate for both. As such, we will use 15 bins when calculating the entropy statistics throughout the manuscript.

1.4.5 Results

The main focus of this work is on dynamic state detection, including a bifurcation analysis and robustness to noise. The first example in Section 1.4.3 uses the Lorenz system to demonstrate typical entropy values for a system with periodic and chaotic behavior. However, a global understanding of the typical distribution of entropy values for each statistic is necessary to draw conclusions on the dynamic state detection abilities. To gain a better understanding of the distribution of entropy values for identifying the dynamic state, we use box plots with no additive noise in Section 1.4.5 for both continuous (flows) and discrete (maps) binned states. For each system, we used $\mathcal{B} = 15$ bins. For approximate and sample entropy we set $m = 3$ and $r = 0.1\mathcal{B}$ or $r = 0.1 \max L$. To understand the noise robustness characteristics of the entropy statistics, in Section 1.4.5 we empirically demonstrate the effects of additive noise on the Lorenz system. This was done for each entropy value with Signal to Noise Ratios (SNRs) ranging from 10 dB to 60 dB. Note that in signal processing, 15 dB is typically considered the SNR limit below which it becomes challenging to extract any useful information from the signal. In Section 1.4.5 we provide a bifurcation analysis to determine how well the entropy statistics can detect changes in a system as parameters change. We show this bifurcation analysis for the logistic map and the Lorenz system. Lastly, in Section 1.4.5 we provide a computation speed analysis for the Lorenz system and logistic map to demonstrate the benefits of applying the entropy statistics to the sublevel set persistence diagram in comparison to directly applying them to the signal.

Dynamic state detection analysis Figures 1.31a and 1.31b use box-plots to demonstrate the distributions of the entropy statistics for periodic and chaotic behavior, respectively. The analysis was performed using 18 continuous and 12 discrete dynamical systems, which were simulated using the MakeData module in the python package `teaspoon` [161] with the default parameters.

The box-plot distribution results show that the entropy statistics perform better for discrete continuous systems than the flows with less overlap between distributions. However, there is still a very clear distinction between the periodic and chaotic dynamics for both maps and flows. Further,



(a) Flows

(b) Maps

Figure 1.31: Spread of entropy values for periodic and chaotic dynamics using 15 bins for 12 discrete dynamical systems (maps) and 18 continuous dynamical systems (flows). The green dashed line seperates periodic and chaotic entropy sttistics based on a maximized accuracy for both flows and maps.

the distributions for maps and flows align closely and are distributed over a specific range which allow a cutoff parameter separating periodic from chaotic dynamics to be chosen for both maps and flows. Based on the distributions we set cutoffs as 0.485 for $h(\mathcal{S})$, 0.100 for $h_a(\mathcal{S})$, 0.105 for $h_s(\mathcal{S})$, 0.110 for $h_a(L)$, 0.120 for $h_s(L)$, 0.172 for $h_r(\mathcal{S})$, and 0.130 for $\bar{h}_c(\mathcal{S})$ which are marked in Fig. 1.31a and 1.31b using green dashed lines. These cutoffs were chosen to maximize the accuracy of the dynamic state detection for each entropy statistic.

It can also be noted that the approximate or sample entropy applied to the lifetimes L or state sequence $\mathcal{S}$ make little difference in the entropy values. As such, there is no advantage in applying approximate or sample entropy to either the lifetimes or state sequence from a performance standpoint.

Robustness to Additive Noise The initial analysis in Section 1.4.5 provided a starting point for dynamic state analysis through a cutoff based on the distribution of entropy statistics. However, noise-robustness must be considered to apply the sublevel set entropy statistics to real-world data.

In this subsection, we determine how well these cutoffs perform for dynamic state detection in the presence of additive white noise.

To test the noise robustness, we use the Lorenz system with additive noise SNRs ranging from 10 dB (high noise) to 60 dB (low noise). Figure 1.32 shows $h_a(\mathcal{S})$, $h_s(\mathcal{S})$, $h_a(L)$, and $h_s(L)$ all being the most noise robust down to an SNR of 20 dB. $h(\mathcal{S})$ is also moderately noise robust with accurate separation between periodic and chaotic dynamics based on the cutoff down to an SNR of 23 dB. The Markov chain statistics, h_r and $\bar{h}_c$, are the least noise robust and only correctly separate periodic from chaotic dynamics with SNR values greater than 26 dB.

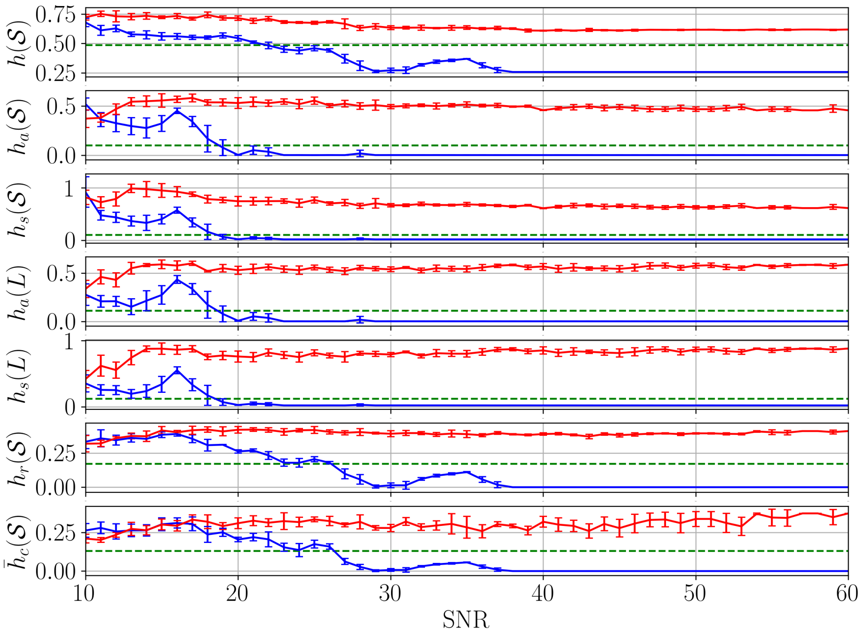


Figure 1.32: Resilience of entropy statistics to additive noise for SNR values from 10 to 50 dB for the periodic and chaotic Lorenz system simulation described in Eq. (4.3). Uncertainties are reported as the standard deviation for each SNR repeated 20 times.

We observed that these noise robustness results hold for the other dynamical systems with similar levels of noise robustness. We speculate that the noise robustness of these methods is mainly due to the stability theorem for sublevel set persistence [49]. This theorem states that the persistence diagram of a function with and without additive noise will only change linearly proportional to the additive noise level. Therefore, if the noise-artifact persistence pairs are removed using the cutoff C_α, then the entropy statistics on the resulting persistence pairs should only be based on the noise robustness of the entropy statistics.

Bifurcation Analysis In our initial dynamic state analysis in Figures 1.31a and 1.31b we only looked at a single realization of chaotic and periodic signals from each system. However, it is often of interest to analyze the bifurcation behavior as one parameter varies. To determine the viability of the sublevel set entropy statistics for bifurcation analysis, we study the bifurcations in the Logistic map and the Lorenz system.

Logistic Map Our first bifurcation analysis uses the logistic map as an example discrete dynamical system. The logistic map is defined as

$$x_{n+1} = rx_n(1 - x_n). \tag{1.102}$$

For this system we increment the r parameter from 3.2 to 4.0 in 10^{-3} step sizes. At each step, the system is solved for 1000 map iterations but we only retain the last 300 iterations to avoid transients. Figure 1.33 shows each of our sublevel set entropy statistics for each r value, and it contrasts them to permutation entropy $h(\pi)$, sample entropy $h_s(x)$, and approximate entropy $h_a(x)$ computed directly from the simulated signals. The permutations used in calculating the permutation entropy were of dimension $n = 6$ with time delay $\tau = 1$. The sample and approximate entropy used dimension $m = 3$ with filtering level of 0.2σ, where σ is the standard deviation of the signal.

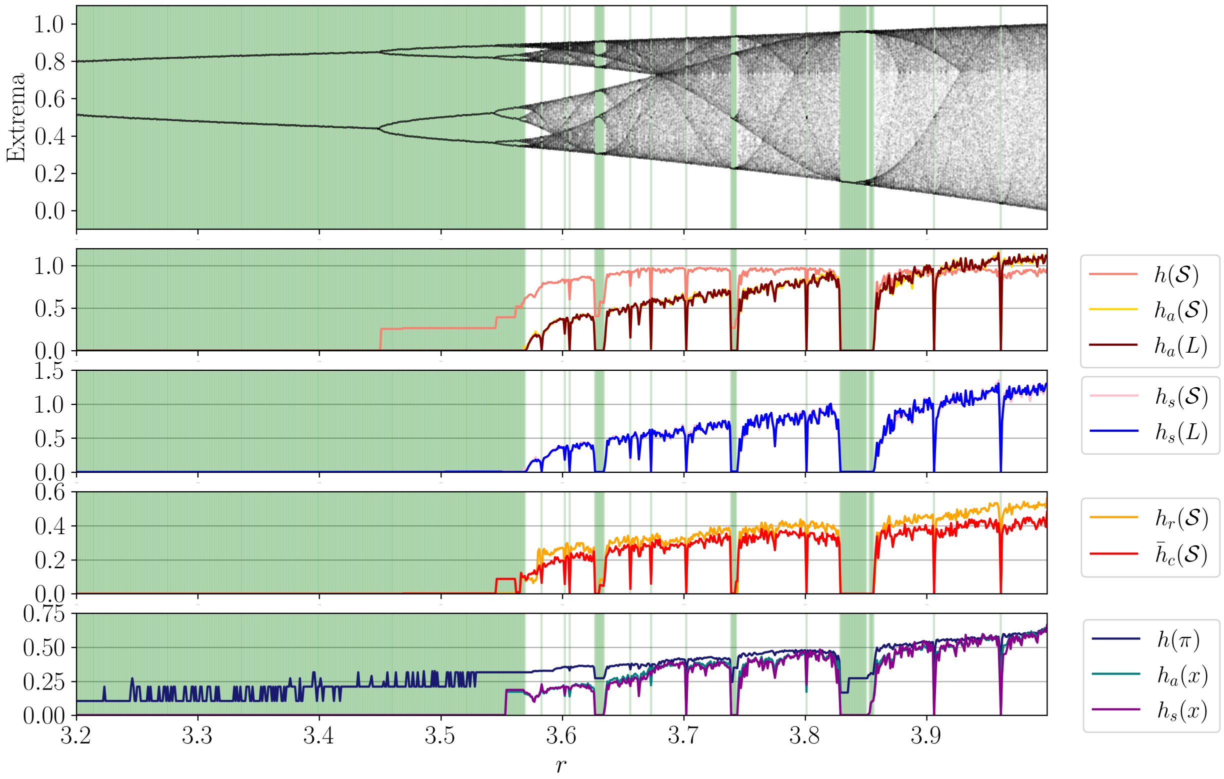


Figure 1.33: Bifurcation analysis of entropy statistics for the logistic map dynamical system with $r \in [80, 190]$ with step sizes of $\Delta r = 0.001$. Green highlighted regions are periodic.

Figure 1.33 demonstrates that the sublevel set entropy statistics outperform the standard entropy tools. Specifically, all sublevel set entropies can locate the small periodic window at approximately $r \approx 3.67$, which is not identified by the standard tools. Further, permutation entropy does not provide clear drops in its value for periodic windows, where the sublevel set entropy statistics are at approximately zero for periodic dynamics.

When comparing the sublevel set entropy statistics, there is no clear distinction in performance. The sample entropies are almost identical, suggesting that there is little benefit in applying it to the lifetimes or the state sequence besides bypassing the complexity involved in parameter selection when directly applying them to the sequences. It is also important to note that the Shannon entropy of $\mathcal{S}$ provides more information in regards to signal complexity. Specifically, it more clearly shows bifurcations. For example, at $r \approx 3.45$, there is a period-doubling bifurcation which increases $h(\mathcal{S})$, while the other entropy statistics do not show any change.

Lorenz System Our second bifurcation analysis used the Lorenz System defined in Eq. (4.3) where the ρ parameter was incremented from 80 to 190 in step sizes of 0.1. The same entropy statistics from the logistic map bifurcation analysis were used for the Lorenz example. As shown in Fig. 1.34, the entropy statistics can show bifurcations in the system with periodic dynamics having low entropy values. Similar to the logistic map bifurcation, the standard entropy tools did not identify all of the periodic windows (e.g., at $\rho \approx 112$ and $\rho \approx 182$). At the same time, the sublevel set entropy methods show this as a periodic window. This example demonstrates the viability of sublevel set entropy statistics to detect periodic from chaotic windows and bifurcations for both maps and flows.

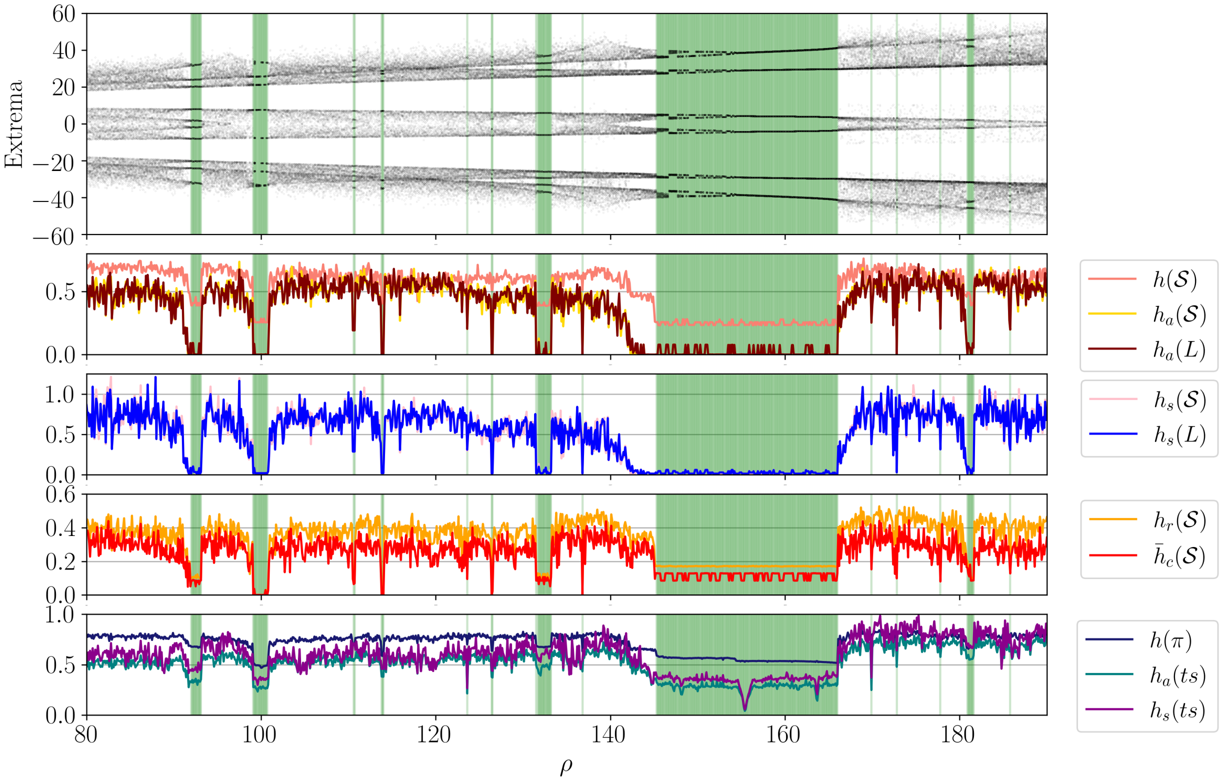


Figure 1.34: Bifurcation analysis of entropy statistics for the Lorenz dynamical system with $\rho \in [3.2, 4.0]$ with step sizes of $\Delta\rho = 0.1$ and $\sigma = 10$ and $\beta = 8/3$. Green highlighted regions are periodic.

Computation Time We now investigate the computational speed benefits of using the sublevel set persistence when calculating the sample entropy compared to its direct application to signals.

When approximate, sample, and permutation entropy are applied directly to a signal of length N, all $N - m$ sequences are used. However, the computational demand is significantly decreased when using the sublevel set persistence diagram. This is due to the new lifetimes being shorter than the original signal. Additionally, the length N is proportionally increased by increasing the signal's sampling rate. However, the number of persistence pairs in the sublevel set persistence diagram remains constant. We demonstrate the computational demand of each entropy statistic for both the Lorenz system and logistic map in Fig. 1.35.

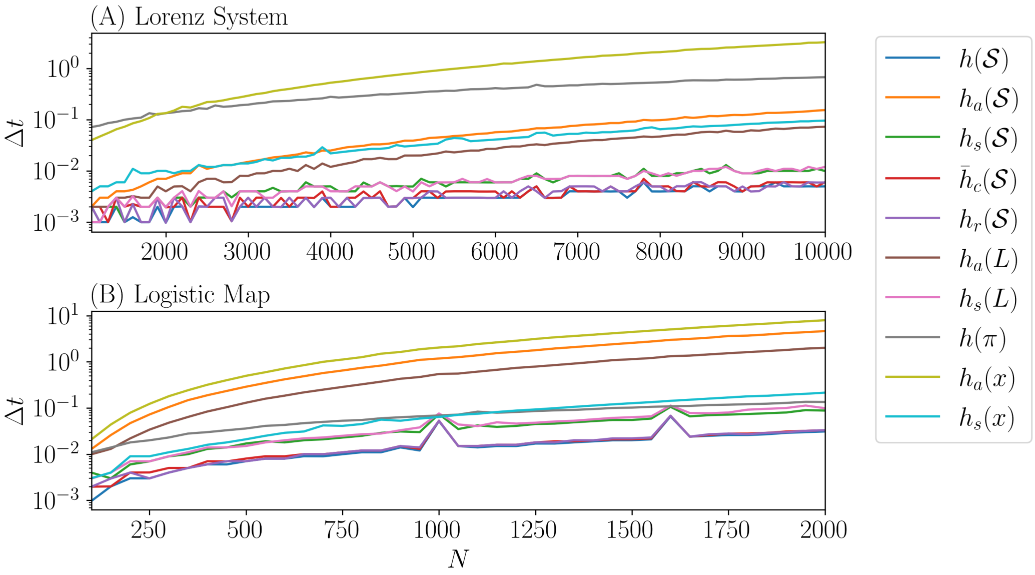


Figure 1.35: Computation Time Example for Lorenz system (A) and logistic map (B) for each entropy statistic.

Our computation speed analysis shows that, as expected, approximate and sample entropy applied directly to the signal as $h_a(x)$ and $h_s(x)$ are faster than when applied to the sequence $\mathcal{S}$ and lifetimes L. Specifically, for the Lorenz system with $N = 10^3$, $h_a(L)$ is approximately 45 times faster than $h_a(x)$ and $h_s(L)$ is approximately 9 times fast than $h_s(x)$. Further, $h_a(L)$ is approximately twice as fast as $h_a(\mathcal{S})$ and $h_s(L)$ and $h_s(\mathcal{S})$ are approximately equivalent in computational speed. For the logistic map, the computational times are generally larger for the same signal length N of a flow due to oscillations occurring more frequently with maps. We would also like to note that the average conditional $\bar{h}_c$, entropy rate h_r and Shannon entropy $h(\mathcal{S})$ have the fastest computational speed making them the most suitable for in-situ applications. The computational benefit of the sublevel set entropy statistics most likely stems from the $O(N\log(N))$ algorithmic complexity of the zero-dimensional sublevel set persistence of one-dimensional signals [164].

CHAPTER 2

PARAMETER SELECTION FOR PERMUTATION ENTROPY AND STATE SPACE RECONSTRUCTION

This chapter of my research is focused on choosing the optimal delay and dimension parameters for both permutation entropy and state space reconstruction. This section will begin by introducing information entropy and then specifically permutation entropy as a time series analysis tool. Following this introduction several delay and dimension parameter selection algorithms are introduced and then compared in Section 2.3.4 to choose an optimal method. This work is based on my publication “On the Automatic Parameter Selection for Permutation Entropy" [161]. The future work section is based on work that will soon be published on relating permutation entropy to state space reconstruction to allow for tools from TDA to be used for delay parameter selection in permutation entropy.

2.1 Permutation Entropy

Permutation Entropy (PE) has its origins in information entropy, which is a tool to quantify the uncertainty in an information-based system. Information entropy was first introduced by Shannon [212] in 1948 as Shannon Entropy. Specifically, Shannon entropy measures the uncertainty in future data given the probability distribution of the data types in the original, finite dataset. Shannon entropy is calculated as $H_s(n) = -\sum p(x_i)\log p(x_i)$, where x_i represents a data type, and $p(x_i)$ is the probability of that data type. In recent years information entropy has been heavily applied to the time series of dynamical systems. Several new variations of information entropy have been proposed to better accommodate these applications, e.g. approximate entropy [186], sample entropy [199], and PE [15] with a timeline shown in Fig. 2.1. These methods measure the predictability of a sequence through the entropy of the relative data types. However, PE considers the ordinal position of the data through permutations, which has been shown to be effective for analyzing the dynamic state and complexity of a time series [6, 16, 33, 62, 80, 81, 145]. PE is also

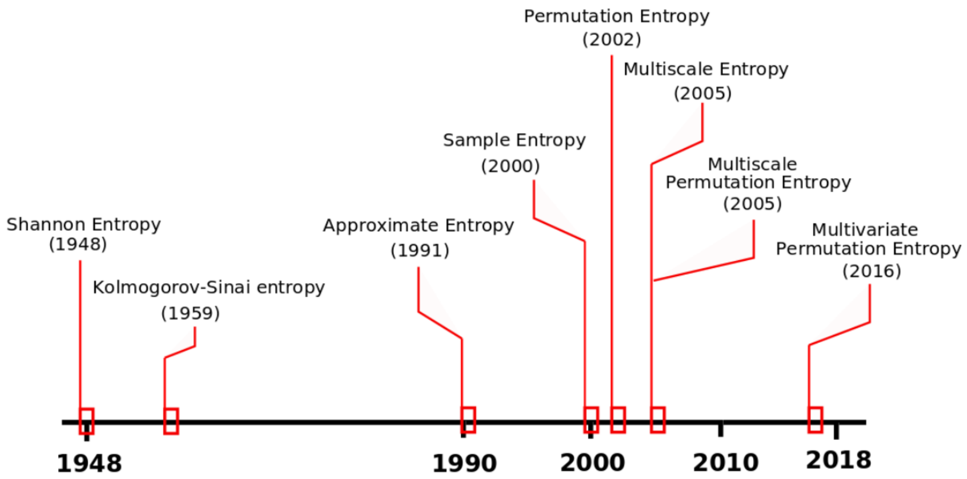

Permutation Entropy (2002)
Multiscale Entropy (2005)
Sample Entropy (2000)
Multiscale Permutation Entropy (2005)
Shannon Entropy (1948)
Approximate Entropy (1991)
Multivariate Permutation Entropy (2016)
Kolmogorov-Sinai entropy (1959)
1948
1990
2000
2010
2018

Figure 2.1: Timeline of entropy measurements for time series analysis.

noise robust for time series of sufficient length and relatively high signal-to-noise ratios, which is the ratio between useful signal and background noise. Alternatively, if the time series is relatively short or has a low signal-to-noise ratio, it is suggested to use a different entropy measurement such as coarse-grained entropies [190]. PE is quantified in a similar fashion to Shannon entropy with only a change in the data type to permutations (see Fig. 2.3), which I symbolically represent as π_i. PE has two parameters: the permutation dimension n and embedding delay τ, which are used when selecting the permutation size and spacing, respectively. PE is sensitive to these parameters [131, 201, 221] and there is no accurate selecting approach for all applications. This introduces the motivation for this paper: investigate automatic methods for selecting both PE parameters. There are currently three main methods for selecting PE parameters: (1) parameters suggested by experts for a specific application, (2) trial and error to find suitable parameters, or (3) methods developed for phase space reconstruction. I will now overview a simple example to better understand these parameters.

Bandt and Pompe [15] defined PE according to

$$H(n) = -\sum p(\pi_i) \log p(\pi_i), \tag{2.1}$$

where $p(\pi_i)$ is the probability of a permutation π_i and $H(n)$ is the permutation entropy of dimension n with units of bits when the logarithm is of base 2. The permutation entropy parameters τ and n are used when selecting the motif size, with τ determining the time difference between two consecutive points in a uniformly sub-sampled time series and n as the permutation length or motif dimension. To form a permutation, begin with with an element x_i of the series X. Using this element, the dimension n, and delay τ, define the vector $v_i = [x_i, x_{i+\tau}, x_{i+2\tau}, \ldots, x_{i+(n-1)\tau}]$. The corresponding permutation π_i of this vector is determined using its ordinal pattern. For example, consider the third degree $n = 3$ permutation shown in Fig. 2.2. The permutation type, which categorizes the permutation, is found by first ordering the n values of the permutation smallest to largest, and then accounting for the order received. For the given permutation in Fig. 2.2, the resulting permutation is categorized as the sequence $\pi_i = (1, 0, 2)$, which is one of $n!$ possible permutations for a dimension n, see Fig. 2.3 for the other possible permutations of $n = 3$.

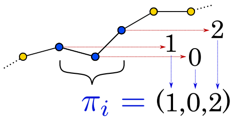


Figure 2.2: Sample permutation formation for $n = 3$ and $\tau = 1$.

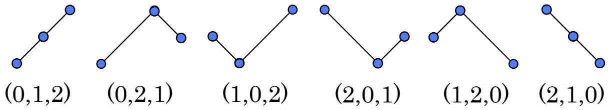


Figure 2.3: All possible permutation configurations for n = 3.

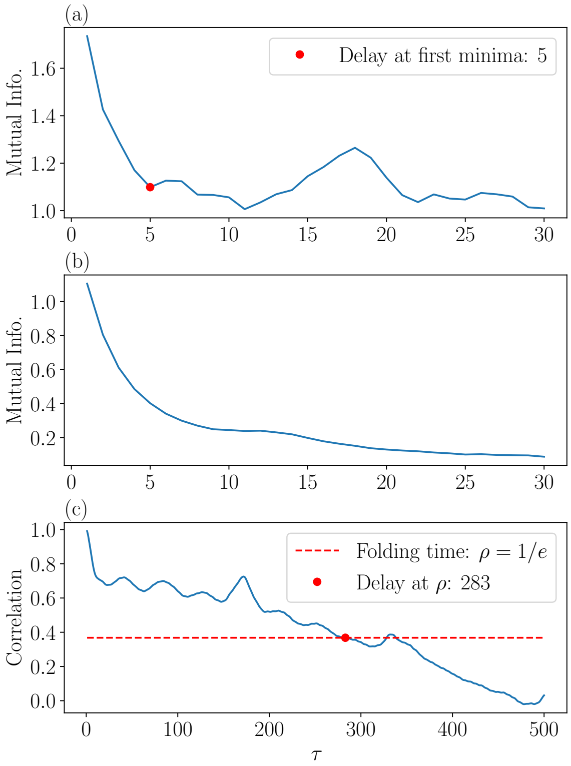
I can normalize PE using the maximum possible PE value, which occurs when all $n!$ possible permutations are equiprobable according to $p(\pi_1) = p(\pi_2) = \ldots = p(\pi_{n!}) = \frac{1}{n!}$. The resulting normalized PE is

$$h_n = -\frac{1}{\log_2 n!} \sum p(\pi_i) \log_2 p(\pi_i). \tag{2.2}$$

Many domain scientists who apply PE make general suggestions for n and τ [76, 248], which can be impractical for some applications. As an example, Popov et al. [189] showed the influence of the sampling frequency on the proper selection of τ. As for the dimension n, there are general suggestions [201] on how to choose its value based on the vast majority of applications having an appropriate permutation dimension in the range $3 < n < 8$. Additionally, Bandt and Pompe [15] suggest that $N \gg n$, where N is the length of the time series. However, these general outlines for the selection of n (and τ) do not allow for an application specific suggestions.

If I assume that suitable PE parameters correspond to optimal phase space reconstruction parameters, then a common approach for selecting τ and n is to implement one of the existing methods for estimating the optimal Takens' embedding [225] parameters. Hence, some of the common methods for determining τ include the mutual information function approach [77], the first folding time of the autocorrelation function [25, 86], and phase space methods [30]. Additionally, some common phase space reconstruction methods for determining n include box-counting [22], correlation exponent method [86], and false nearest neighbors [110]. Although the parameters in PE have similar names to their delay reconstruction counterpart, there are innate differences between ordinal patterns and phase space reconstruction which can also lead to inaccurate n or τ values. In spite of these differences, permutations can be viewed as symbolic representation of regions in the phase space through a binning process. Permutations partition the phase space based on the ordinal rankings of the embedded vectors. This relationship between phase space and permutations opens up the potential for some of the classic phase space reconstruction methods for selecting both n and τ to be a plausible solution for selecting the same parameters for PE.

Even with the possibility that phase space reconstruction methods for selecting τ and n may work for choosing synonymous parameters of PE, there are a few practical issues that preclude using parameters from time series reconstruction for PE. One issue stems from many of the methods (e.g. false nearest neighbors and mutual information) still requiring some degree of user input through either a parameter setting or user interpretation of the results. This introduces issues for practitioners working with numerous data sets or those without enough expertise in the subject area to interpret the results. Another issue that arises in practice is that the algorithmic implementation of existing time series analysis tools is nontrivial. This hinders these tools from being autonomously applied to large datasets. For example, the first minimum of the MI function is often used to determine τ. However in practice there are limitations to using mutual information to analyze data without the operator intervention to sift through the minima and choose the first 'prominent' one. This is due



h

Figure 2.4: Some possible modes for failure for selecting τ for phase space reconstruction using classical methods: (a) mutual information registering false minima as suitable delay generated from a periodic Lorenz system, (b) mutual information being mostly monotonic and not having a distinct local minimum to determine τ generated from EEG data [7], and (c) autocorrelation failing from a moving average of ECG data provided by the MIT-BIH Arrhythmia Database [154].

to possibility that the mutual information function can have small kinks that can be erroneously picked up as the first minimum. Figure 2.4a shows this situation, where the first minimum of the mutual information function for a periodic Lorenz system is actually an artifact and the actual delay should be at the prominent minimum with $\tau = 11$. Further, the mutual information function approach may also fail if the mutual information is monotonic. This is a possibility since there is no guarantee that minima exist for mutual information [13]. An example of this mode of failure is shown in Fig. 2.4b, which was generated using EEG data [7] from a patient during a seizure.

A mode of failure for the autocorrelation method can occur when the time series is non-linear or has a moving average. In this case, the autocorrelation function may reach the folding time at an unreasonably large value for τ. As an example, Fig. 2.4c shows the autocorrelation not reaching the folding time of $\rho = 1/e$ until a delay of $\tau = 283$ for electrocardiogram data provided by the

MIT-BIH Arrhythmia Database [154]. The last mode of failure concerns choosing the permutation dimension n to be equal to the embedding dimension optimized using delay embedding from time series analysis. This can lead to an overly large embedding dimension [47] ($n \gg 8$), which would make the calculation of PE impractical because the number of possible permutations $n!$ would become too large. All of these possible modes of failure can make using classical phase space methods for selecting τ and n unreliable thus necessitating new tools or modifications to make selecting τ and n for PE more robust and less user-dependent.

These shortcomings lead us to the problem that I address in this chapter: Given a sufficiently sampled/oversampled and noisy time series $X = \{x_t\}_{\mathcal{R}^+}$, how can I reliably and systematically define appropriate dimension n and time delay τ values for computing the corresponding PE?

The first contribution towards answering this question is detailed in Section 2.2, which addresses the automatic selection of the time delay τ. In Section 2.2.1 I combine the Least Median of Squares (LMS) approach for outliers detection with Fourier transformation theorem to derive a formula for the maximum significant frequency in the Fourier spectrum, with the assumption that X is contaminated by Gaussian measurement noise. This formula allows obtaining a cutoff value where the only input, besides the time series, is a desired percentile from the Probability Density Function (PDF) of the Fourier spectrum. Once this value is obtained, Nyquist's sampling theorem is used to compute an appropriate τ value.

The second contribution is through an approach that I develop in Section 2.2.2, which uses Multi-scale Permutation Entropy (MPE) for finding τ. I show how MPE can be used to find the main period of oscillation for a time series derived from a periodic system. Building upon this, I show how the method can be extended to find τ for a chaotic time series by using the first maxima in the MPE as it satisfies the Nyquist's sampling theorem.

The third contribution to the automatic selection of τ is through the analysis of Permutation Auto-Mutual Information [135] (PAMI). PAMI is an existing method for measuring the mutual information of permutations. However, I tailor this method to specifically select τ for PE.

The final contribution towards answering the posited question is our evaluation of the ability

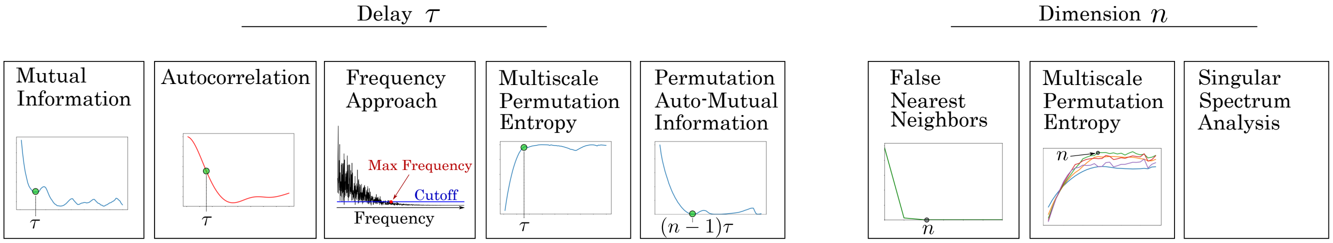


Figure 2.5: Overview of methods investigated for automatically calculating both the delay τ and dimension n for permutation entropy.

of existing tools for computing an embedding dimension to provide an appropriate value for the PE parameter n. I compare dimension n values computed from False Nearest Neighbors (FNN—Section 2.3.1), Singular Spectrum Analysis (SSA—Section 2.3.2), and MPE (Section 2.2.2). While I use existing methods for performing the FNN and the SSA analyses, for the MPE-based approach I use a criteria established in prior works [201], which requires finding τ first. I made this process automatic through the selection of τ from our second contribution.

This chapter is organized as follows. I first go into detail on some existing methods for selecting both τ and n. Specifically, in Section 2.2 I provide a detailed explanation for selecting τ using existing, automatic methods such as autocorrelation in Section 2.2.3 and Mutual Information (MI) in Section 2.2.4. Additionally, I modify and develop/tailor methods to automatically select τ. These methods include a frequency approach in Section 2.2.1, MPE in Section 2.2.2, and PAMI in Section 2.2.5. In Section 2.4.3 I expand on the process for selecting n using False Nearest Neighbors (FNN) in Section 2.3.1 and Singular Spectrum Analysis in Section 2.3.2. In Section 2.3.3, I explain our algorithm for automatically selecting n using MPE. After introducing each method, in Section 2.3.4 I contrast all of these methods and make conclusions on their viability by comparing the resulting parameters to those suggested by PE experts. An overview of the methods that will be investigated for automatically calculating both τ and n are shown in Fig. 2.5. All the functions used and developed in this work are available in Python through GitHub [161].

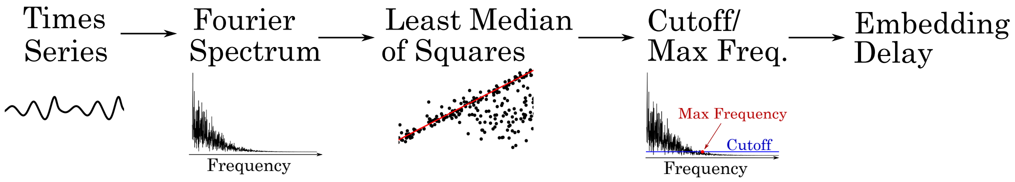


Figure 2.6: Overview of our frequency domain approach for finding the maximum significant frequency f_{max} using LMS for a signal contaminated with GWN.

2.2 Embedding Delay Parameter Selection Methods

The delay embedding parameter τ is used to uniformly subsample the original time series. To elaborate, consider the time series $X = \{x_i \mid i \in \mathbb{N}\}$. By applying the delay $\tau \in \mathbb{N}$, a new subsampled series is defined as $X(\tau) = [x_0, x_\tau, x_{2\tau}, \ldots]$. In order to obtain a stable and automatic method for estimating an optimal value for τ I investigate: a novel frequency-based analysis that I describe in Section 2.2.1, Multi-scale Permutation Entropy (MPE) (Section 2.2.2), autocorrelation (Section 2.2.3), and Mutual Information function (MI) (Section 2.2.4). I recognize, but do not investigate, some other methods for finding τ such as diffusion maps [20] and phase space expansion [30].

2.2.1 Frequency Approach for Embedding Delay

In this section we develop a method for finding the noise floor in the Fourier spectrum using Least Median of Squares (LMS) [143]. We then use the noise floor to find the maximum significant frequency of a signal contaminated with additive Gaussian white noise (GWN). Our method is based on finding the maximum significant frequency in the Fourier spectrum and the Nyquist sampling frequency criteria. To motivate the development of this approach, I begin by working with the frequency criteria developed by Melosik and Marszalek [148], which agrees with Nyquist sampling theorem [124] for choosing a suitable sampling frequency f_s as

$$2f_{\text{max}} < f_s < 4f_{\text{max}}, \tag{2.3}$$

where f_{max} is the maximum significant frequency in the signal. Melosik and Marszalek [148] showed that a sampling frequency within this range is appropriate for subsampling an oversampled signal, thus mitigating the effect of temporal correlations of neighboring points in densely sampled signals. However, the automatic identification of $f_{\max}$ from an oversampled signal is not trivial. Melosik and Marszalek [148] selected a maximum significant frequency by inspecting the normalized Fourier spectrum and using a threshold cutoff of approximately 0.01 for a noise-free chaotic Lorenz system. This made *visually* finding the maximum frequency significantly easier but did not provide guidance on how to algorithmically find $f_{\max}$. Further, attempting to algorithmically adopt the approach suggested by Melosik and Marszalek [148] resulted in large errors especially in the presence of a low signal to noise ratio. This motivated the search for an automatic and data-driven approach for identifying the noise floor which could then be used to find the maximum significant frequency. To do this I develop a method based on 1-D least median of squares applied to the Fourier spectrum. The assumptions inherent to our method are

1. The time series is not undersampled. The purpose of the methods is to determine a suitable delay parameter for subsampling the signal, which would be meaningless if the time series is undersampled.

2. The Fourier transform of the time series needs to have less than 50% of the points with significant amplitudes. This requirement stems from the limitations of the least median of squares regression.

3. The noise in the signal is approximately GWN; otherwise, the ensuing statistical analysis becomes inapplicable. Violating this assumption can yield false peak detections, which would lead to an incorrect delay parameter.

We find suitable cutoffs for obtaining $f_{\max}$ of the signal by using the noise floor determined from the 1-D least median of squares, and compute a suitable embedding delay according to

$$\tau = \frac{f_s}{\alpha f_{\max}}, \tag{2.4}$$

where I set $\alpha = 2$, thus agreeing with the range in Eq. (2.3) and the Nyquist sampling criterion.

Figure 2.6 summarizes the frequency approach for τ with the use of our 1-D LMS method for finding a noise floor in the Fourier spectrum. This process begins with computing the Fourier spectrum of the signal, which is followed by fitting an 0-D LMS regression line to the noise in the Fourier spectrum. This provides statistical information about the Probability Distribution Function (PDF) of the noise level. The PDF is used to determine the Cumulative Distribution Function (CDF), which I use determine a meaningful noise cutoff in the Fourier spectrum. However, it is assumed that the noise is approximately GWN for this method to hold statistical significance. This cutoff is used to separate the highest significant frequency in the Fourier spectrum $f_{\max}$, which is used to find a suitable embedding delay τ based on the frequency criteria in Eq. (2.4). In the following paragraphs I review our use of the LMS and the derivation of the PDF of the Fourier spectrum of GWN. I then show how to combine the LMS method with the resulting PDF expression to find a suitable noise floor cutoff and the corresponding maximum significant frequency.

Least Median of Squares: LMS [143] is a robust regression technique used when up to 50% of the data is corrupted by outliers. Outliers will be considered as anything other than noise in the fourier spectrum for our application. In comparison to the widely used least sum of squares (LS) algorithm, the LMS replaces the sum for the median which makes LMS resilient to outliers. The difference between LS and LMS is highlighted as

$$\begin{aligned} LS &: \min \sum_i r_i^2, \\ LMS &: \min \left(\text{median}_i(r_i^2)\right), \end{aligned} \tag{2.5}$$

where r is the residual. Similar to the i subscript in $\sum_i$, the i in median$_i$ signifies that the median is of all residuals. Figure 2.7 shows an example application of the linear LMS regression.

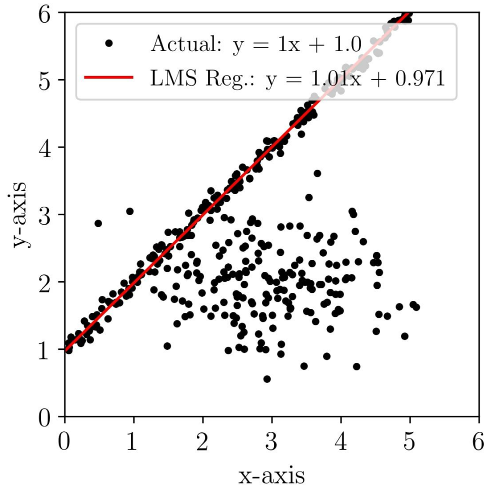


Figure 2.7: LMS linear regression with 45% outliers. Results match those found in [143].

Specifically, this figure shows 110 data points drawn from the line $y = x + 1$ with added GWN of zero mean and 0.1 standard deviation. The data is corrupted with 90 outliers centered around $(3, 2)$ with a normal distribution of 1.0 along x and 0.6 along y. Figure 2.7 shows that the linear regression results closely match the actual trend line with the fitted line being $y = 0.998x + 1.012$ in comparison to the actual $y = x + 1$.

PDF and CDF of the magnitude of the Fast Fourier Transform of GWN: This section reviews the probability distribution function (PDF) and cumulative density function (CDF) for the Fourier Transform (FT) of white noise. Additionally, this section derives the location of the theoretical maximum of the PDF. The FT distribution of GWN [197] is described as

$$P_{|X|}(|X|) = \frac{2|X|}{E_w \sigma_x^2} e^{\frac{-|X|^2}{E_w \sigma_x^2}}, \tag{2.6}$$

where $|X|$ is the magnitude of the FT of GWN, $P_{|X|}$ is the probability density function of $|X|$, σ_x is the standard deviation of the GWN, and E_w is the window energy or number of discrete transforms taken during the FT. By setting the first derivative of $P_{|X|}$ with respect to $|X|$ equal to zero, the theoretical maximum of the PDF is

$$|X|_{\max} = \sqrt{\frac{E_w \sigma_x^2}{2}}. \tag{2.7}$$

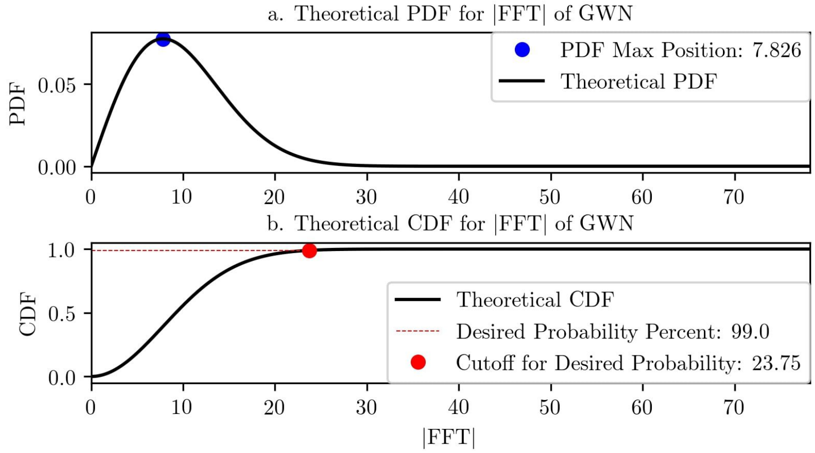


Figure 2.8: (a) Theoretical PDF for GWN. (b) CDF for GWN with an example cutoff at the 99% CP.

We calculate the CDF corresponding to the PDF described in Eq. (2.7) by combining the PDF in Eq. (2.6) with the CDF for a Rayleigh distribution as [173]

$$CP_{|X|}(|X|) = 1 - e^{\frac{-|X|^2}{E_W \sigma_x^2}}, \tag{2.8}$$

where $CP_{|X|}$ is the cumulative probability of $|X|$.

Finding the Noise Floor: Our approach for finding the noise floor combines LMS with Eqs. (2.6) and (2.7). Specifically, I utilize LMS to obtain a 0-D fit of the Fast Fourier Transform (FFT) of the signal, which results in an approximate value of $|X|_{\max}$, which is $|X|$ at the maximum of $P_{|X|}$. Using $|X|_{\max}$ from the LMS fit, I then find the standard deviation of the distribution σ_x from Eq. 2.7, which is used to find a cutoff based on a set cumulative probability in Eq. (2.8).

We begin by showing the accuracy of the LMS fit for finding $|X|_{\max}$. Our example uses GWN with a mean of zero and standard deviation of 0.035 with 1000 data points. Taking the FFT of the GWN (see Fig. 2.9A) results in the distribution shown in Fig. 2.9B. The distribution shows a 1-D LMS fit of 8.215 compared to the theoretical maximum of the PDF from Eq. 2.7 of 7.826, which is approximately 4.67% greater. This shows that the 1-D LMS fit accurately locates $|X|_{\max}$. Additionally, the theoretical shape of the PDF in Fig. 2.9B is shown to be very similar to the actual distribution.

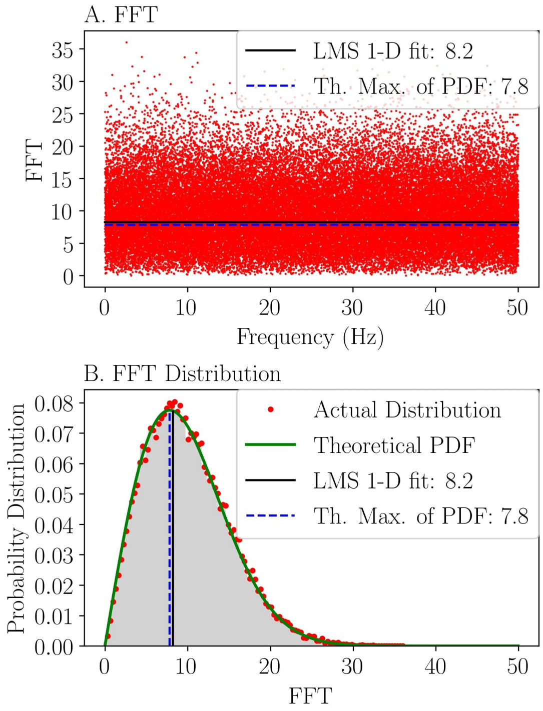


Figure 2.9: (A) FFT of GWN with 0.035 standard deviation and zero mean with the location of the theoretical maximum of the PDF and one-dimensional LMS regression value. (B) Distribution of GWN in the Fourier Spectrum with overlapped theoretical PDF and location of the theoretical maximum of the PDF and one-dimensional LMS regression value.

Next, our approach utilizes Eq. (2.8) and σ_x derived from Eq. (2.7) for finding the cutoff value $|X|_{\text{cutoff}}$. The $|X|_{\text{cutoff}}$ for a desired cumulative probability CP is found by solving Eq. (2.8) for $|X|$ as

$$|X|_{\text{cutoff}} = \sqrt{-E_w \sigma_x^2 \ln(1 - CP)}. \tag{2.9}$$

In order to make $|X|_{\text{cutoff}}$ robust to normalization and scaling of the FFT I define the ratio C between the suggested cutoff from Eq. (2.9) and the maximum of the PDF from Eq. (2.7) as

$$C = \frac{|X|_{\text{cutoff}}}{|X|_{\text{max}}} = \sqrt{-2\ln(1 - CP)}. \tag{2.10}$$

Example Cutoff: An example of how Eqs. (2.7) and (2.9) are used is shown in Fig. 2.8, where the maximum of the PDF and the cutoff for $CP = 99\%$ are marked in Fig. 2.8a and 2.8b, respectively. For this example, I find the ratio C to be approximately 3.03 for a 99% probability. In addition, I suggest a cutoff ratio $C = 6$ to be used for signals with less than 10^4 data points. This yields an expected probability of $\approx 10^{-8}\%$ for a point in the FFT of the GWN attaining a magnitude greater than $|X|_{\text{cutoff}}$. Alternatively, Eq. (2.10) can be used to calculate a different value of C based on the desired probability and length of the signal.

2.2.2 Multi-scale Permutation Entropy for Selecting Delay

In this section I develop a method based on Multi-scale Permutation Entropy (MPE) to find the periodicity of a signal, which is then used to find a suitable delay parameter. MPE is a method of applying permutation entropy over a range of delays for analyzing physiological time series [51]. Zunino et al. [252] showed how the first maxima in the MPE plot arises when τ matches the characteristic time delay τ_r. Furthermore, the periodicity can be captured by the first dip in the MPE plot as shown in Fig. 2.10 at the location d$_2$ when the delay τ matches the characteristic time delay τ_r.

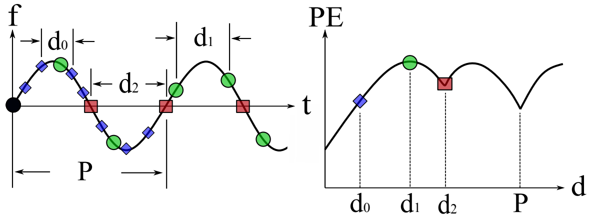


Figure 2.10: (right) Resulting MPE plot for (left) 2P periodic time series with example embedding delays d$_0$, d$_1$, and d$_2$.

Figure 2.10 shows embedding delays d_0, d_1, and d_2 calculated as $d = \frac{\tau}{f_s}$ as well as their corresponding locations on a normalized MPE plot. This toy MPE plot shows that the normalized MPE reaches its first maximum when the delay is roughly d_1, which corresponds to approximately an even distribution of permutations. A second observation, as mentioned previously, is that at d_2 (or the first dip in the MPE plot) there is a resonance or aliasing effect caused by $\tau \approx \tau_r$, which can be used to determine the period of the time series. This is based on the embedding delay size at d_2 causing the embedding vector size $V = d(n-1)$ to be approximately half the of the periodicity P, which can be expressed as

$$d_2 = \frac{1}{2}P = \frac{1}{f_s}\tau_r = \frac{1}{2f}, \tag{2.11}$$

where P is the main period of oscillation, f is the main frequency of the time series corresponding to P, and f_s is the sampling frequency. The reason for the dip in the permutation entropy (PE) when the condition from Eq. (2.11) is met is caused from an aliasing effect, which reduces PE through more regularity in the permutation distribution.

We use the criteria of Melosik and Marszalek [148] to determine a suitable delay from the location of the first dip at d_2. Their criteria states that the sampling frequency must fall within the range shown in Eq. (2.3). This range led to Eq. (2.4), which is used to calculate τ. However, for MPE, I substitute f_s and $f_{\max}$ in Eq. (2.3) with $f_s = 2f\tau_r$ from Eq. (2.11) and $f_{\max} = f$. These substitutions allow Eq. (2.4) to reduce to

$$\tau = \frac{2}{\alpha}\tau_r, \tag{2.12}$$

where $\alpha \in [2,4]$. These simplifications show that τ is only dependent on the delay which causes resonance τ_r when applying MPE. However, for a chaotic time series, the dip at τ_r may not be present due to non-linear trends. To address this issue, I will first investigate the three dominant regions of the MPE plot, which will also be located for a chaotic time series example. I will then propose a new, automatic method for selecting τ that agrees with the frequency criteria stated in Eq. (2.12). Additionally, in Section A.1 of the appendix I investigate the robustness of the method to noise and in Section C.1 of the appendix I provide the algorithm for finding τ using MPE.

MPE Regions Riedl et al. [201] showed that the MPE plot can be separated into three distinct regions as described below and shown in Fig. 2.11. Region A shows a gradual increase in the permutation entropy until reaching a maxima at the transition between regions A and B. Oversampling or a low value of τ causes the motif distribution corresponding to the permutation entropy to be heavily weighted on just increasing or decreasing motifs (motifs (0,1,2) and (2,1,0) for $n = 3$ from Fig. 2.3). This effect was coined as the “Redundancy Effect" by De Micco et al. [58], which means sufficiently low values of τ result in redundant motifs. However, as τ increases, the motif distribution becomes more equiprobable. Additionally, when the motif probability reaches a maximum equiprobability, the permutation entropy is at a maxima, which is the point of transitions from region A to B. Region B shows a slight dip to the first minima. This reduction in permutation entropy is caused by the aliasing or resonance from the value of d approaching half the main period length. At the transition from B to C, the resonance is reached, which provides information on the main frequency and period of the time series. Region C has possible additional minima and maxima from additional alignment of the embedding vector d with multiples of the main period. This region was referred to as the “Irrelevant Region" by De Micco et al. [58] due to effectively large values of τ forcing the delayed sampling frequency to fall below the Nyquist sampling rate as described by the lower bound in Eq. (2.3).

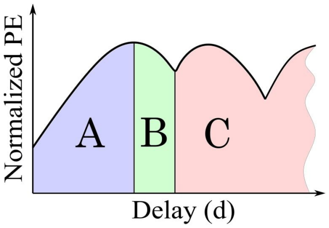


Figure 2.11: The three regions of the MPE plot for a periodic signal: (A) redundant, (B) resonant, and (C) irrelevant.

MPE Example with Chaotic Time Series In Sections 2.2.2 and 2.2.2, I used a periodic time series to show and explain the regions developed in an MPE plot as well as an MPE-based method for determining a suitable embedding delay τ. In this section I further show the applicability of this approach to chaotic signals using the x-coordinate of the Lorenz System as an example. I simulate the Lorenz equations

$$\frac{dx}{dt} = \sigma(y - x),\ \frac{dy}{dt} = x(\rho - z) - y,\ \frac{dz}{dt} = xy - \beta z, \tag{2.13}$$

with a sampling rate of 100 Hz and using the parameters $\rho = 28.0$, $\sigma = 10.0$, and $\beta = 8.0/3.0$. This system was solved for 100 seconds and only the last 15 seconds from the time series are used. Figure 2.12 shows the result of applying MPE to the simulated Lorenz system.

Figure 2.12 shows similarities to Fig. 2.11 with a clear maxima at the boundary between regions A and B, albeit with no obvious minima. Therefore, a new distinct feature needs to be used to

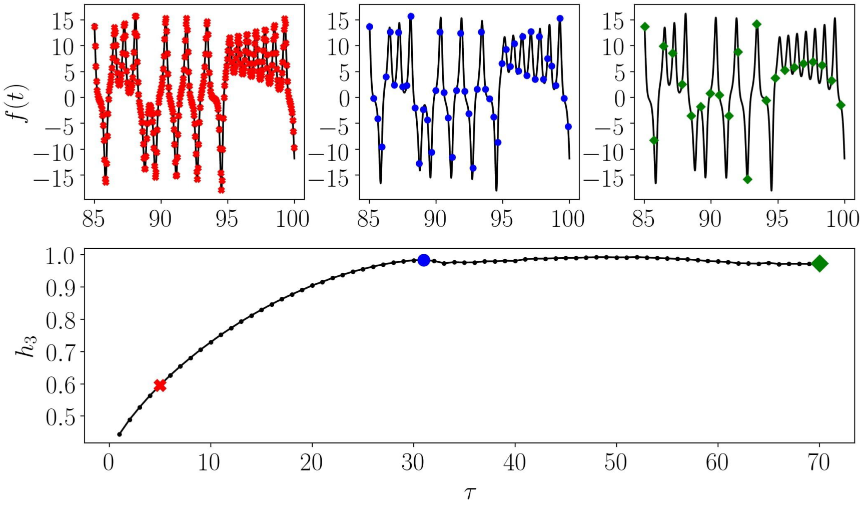


Figure 2.12: MPE plot for the x coordinate of the Lorenz system. Additionally, points in the MPE plot with their corresponding subsampled time series are shown for the redundant, resonant, and irrelevant regions as described in Section 2.2.2.

determine τ_r. I suggest using the first maxima to find τ because this delay is likely to fall within the region described by Eq. (2.12).

2.2.3 Autocorrelation for Embedding Delay

Autocorrelation is a traditional method for selecting τ for phase space reconstruction by using the correlation coefficient between the time series and its τ-lagged version. This method was first introduced by Box et al. [25]. Typically, the autocorrelation function is computed as a function of τ and, as a rule of thumb, a suitable delay τ is found when the correlation between $x(t)$ and $x(t+\tau)$ reaches the first folding time, i.e., when $\rho \leq 1/e$ [106]. The two prominent correlation techniques that are commonly used when implementing an autocorrelation-based approach for finding τ are Pearson Correlation (see Section A.2 of appendix) and Spearman's Correlation (see Section A.2 of appendix). Additionally, an example demonstrating how to calculate τ using autocorrelation and the difference between the two correlation methods is provided in Section A.2 of the appendix.

2.2.4 Mutual Information for Embedding Delay

Mutual information (MI) can be used to select the embedding delay τ based on a minimum in the joint probability between two sequences. The mutual information between two discrete sequences was first realized by Shannon et al. [211] as

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}, \tag{2.14}$$

where X and Y are the two sequences, $p(x)$ and $p(y)$ are the probability of the element x and y separately, and $p(x,y)$ is the joint probability of x and y. Fraser and Swinney [77] showed that for a chaotic time series the MI between the original sequence $x(t)$ and and delayed version $x(t+\tau)$ will decrease as τ increases until reaching a first minimum. At this minima, the delay τ allows for the individual data points to share a minimum amount of information, which indicates sufficiently separated data points. While this delay value was specifically developed for phase space reconstruction, it is also used for the selection of the PE parameter τ. We would like to point out that, in general, there is no guarantee that local minima exist in the mutual information, which is a serious limitation for computing τ using this method. All MI methods can be applied to either ranked or unranked data. We investigate four methods for estimating τ for PE using MI. These methods include MI with equal-sized partitions, adaptive partitions, and two permutation-based MI estimation methods. For details on these methods please reference the appendix in Section A.3.

To determine the optimal MI approximation method for selecting τ for PE, Fig. 2.13 shows a comparison between the τ values computed from each of the MI methods and the corresponding values suggested by experts. The table shows that the adaptive partitioning method of Section A.3 results in an accurate selection of τ for the majority of systems. We will use the adaptive partitioning estimation method when making comparisons to other methods. For the exact values of τ from each of the MI methods please reference Table A.1 in the appendix.

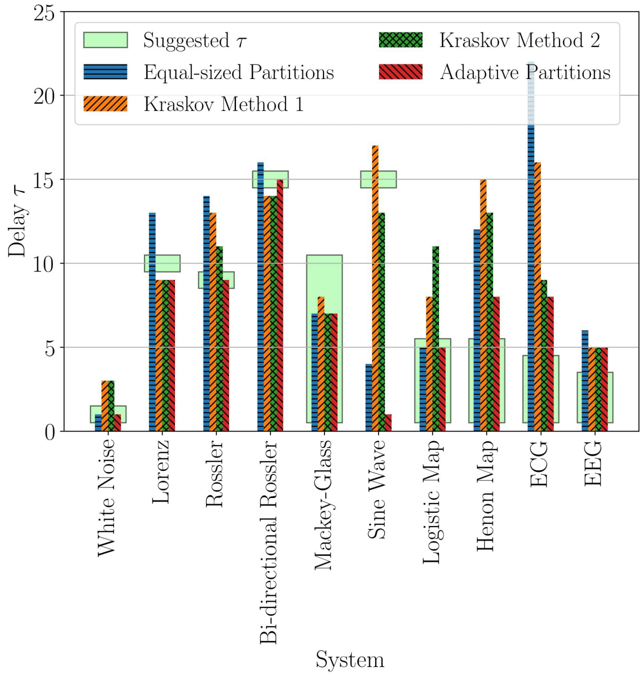


Figure 2.13: A comparison between the calculated and suggested values for the delay parameter τ for multiple MI approximation methods. The methods investigated were equal-sized partition method, Kraskov et al. methods 1 and 2, and the adaptive partitioning approach.

2.2.5 Permutation Auto-mutual Information for Selecting Delay

As shown in Section 2.2.4, Mutual information (MI) is a useful method for selecting τ for phase space reconstruction. However, it does not account for the permutation distribution when selecting τ, which can lead to inaccuracies in computing the PE. To circumvent this issue, we develop a new method for selecting τ using Permutation Auto-Mutual Information (PAMI) [135], which was developed to detect dynamic changes in brain activity. We are tailoring PAMI for its application in the selection of the permutation entropy parameter τ for the first time. This is done by measuring the joint probability between the original permutations formed when a delay of $\tau = 1$ is used and to the permutations when τ is incremented. PAMI is defined as

$$I_p(\tau, n) = H_{x(t,n)} + H_{x(t+\tau,n)} - H_{x(t,n),x(t+\tau,n)}, \tag{2.15}$$

where H is the permutation entropy described in Eq. (2.1). We suggest an optimal delay τ for a given dimension n when PAMI is at a minimum. This delay corresponds to minimum shared information between the original permutations with $\tau = 1$ and its time lagged permutations. By applying this method for the simple sinusoidal function, we can form Fig. 2.14 with $n \in [2, 5]$ and $\tau \in [1, 50]$. As shown, the window size is approximately independent of the dimension n,

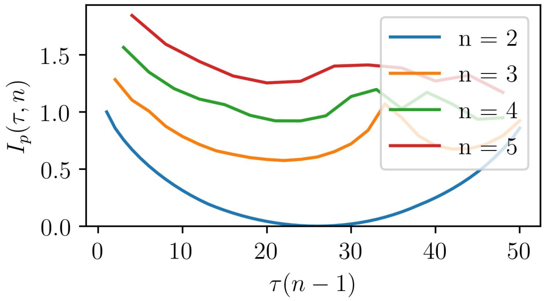


Figure 2.14: PAMI results for the sinusoidal function with $n \in [2, 5]$ and $\tau \in [1, 50]$. The figure shows an optimal window size $\tau(n-1) \approx 25$.

with an optimal window $\tau(n-1) \approx 25$ for the example. Through our analysis of the minimum PAMI as a function of the window size, we have developed a new method for selecting the optimal embedding window. However, we need the embedding dimension to suggest an optimal delay. Hence, we implement the common choice for n ranging from $4 \leq n \leq 6$ for PE [201]. To reduce the computational demand, we suggest using permutation dimensions $n = 2$ to find an optimal window size. In addition to the reduced computational demand of using $n = 2$, we found that $I_p(n = 2) \approx 0$ at the first minima. This also helps making this first minima even more simple.

2.3 Embedding Dimension Parameter Selection Methods

The second parameter for permutation entropy that needs to be automatically identified is the embedding dimension n. The methods for determining n fall into one of two categories: (1) independently determining n and τ, and (2) simultaneously determining n and τ based on the width of the embedding window. For the first category, we investigate using the method of False Nearest Neighbors (FNN) [110] in Section 2.3.1, and Singular Spectrum Analysis (SSA) [26] in Section 2.3.2. For the second category, we contribute to the selection of n by developing an automatic method using MPE from Section 2.3.3. This method combines the results for finding τ through MPE in Section 2.2.2 with the work of Riedl et al. [201]. We acknowledge that our work does not include other commonly used methods for independently calculating n such as box-counting [48], largest Lyapunov exponent [240], and Kolmogorov–Sinai entropy [182].

2.3.1 False Nearest Neighbors for Embedding Dimension

False Nearest Neighbors (FNN) is one of the most commonly used methods for geometrically determining the minimum embedding dimension n for state space reconstruction [110]. For this method the time series is repeatedly embedded into a sequence of m-dimensional Euclidean spaces for a range of increasing values of m. The idea is that when the minimum embedding dimension m is reached or $m \geq n$, the distance between neighboring points does not significantly change as we keep increasing m. In other words, the Euclidean distance $d_m(i, j)$ between the point $\mathbf{P}_i \in \mathbb{R}^m$ and its nearest neighbor $\mathbf{P}_j \in \mathbb{R}^m$ minimally changes when the embedding dimension increases to $m + 1$. If the dimension m is not sufficiently high, then the points are false neighbors if their pairwise distance significantly increases when incrementing m. This ratio of change in the distance between nearest neighbors embedded in $\mathbb{R}^m$ and $\mathbb{R}^{m+1}$ is quantified using the ratio of false nearest neighbors

$$R_i = \sqrt{\frac{d^2_{m+1}(i, j) - d^2_m(i, j)}{d^2_m(i, j)}}. \tag{2.16}$$

R_i is compared to the tolerance threshold R_{tol} to distinguish false neighbors when $R_i > R_{\text{tol}}$. In this paper, we select $R_{\text{tol}} = 15$ as used by Kennel et al. [110]. By applying this threshold over all points, we can find the number of false neighbors as a percent FNN P_{FNN}. If there is no noise in the system, P_{FNN} should reach zero when a sufficient dimension is reached. However, with additive noise present, P_{FNN} may never reach zero. Thus, it is commonly suggested to use a percent FNN cutoff for finding a sufficient dimension n. We use the typically chosen cutoff $P_{\text{FNN}} < 10\%$, which is suitable for most applications when moderate noise is present.

2.3.2 Singular Spectrum Analysis for Embedding Dimension

The singular spectrum analysis method was first introduced as a tool to find trends and prominent periods in a time series [26]. Leles et al. [129] summarized the SSA procedure as (1) immersion,

(2) Singular Value Decomposition (SVD), (3) grouping, and (4) diagonal averaging. Specifically, immersion embeds the time series into a dimension L to form a Hankel matrix, SVD factors all the matrices, grouping combines the matrices that are similar in structure, and diagonal averaging reconstructs the time-series using the combined matrices. The needed embedding dimension is determined from the SVD by calculating the ratio D

$$D = \frac{g_L}{g_r} \tag{2.17}$$

of the sum of the Lth diagonal entries g_L to the sum of the total diagonal entries g_r. When D exceeds 0.9, we consider the dimension to be high enough and set $n = L$, which can then be used as the embedding dimension for permutation entropy.

2.3.3 Multi-scale Permutation Entropy for Permutation Dimension

Riedl et al. [201] showed how MPE can be used to determine an embedding dimension n. This method requires the embedding delay τ to be set to the length of the main period of the signal as shown in Section 2.2.2. The theory behind the method is based on normalizing the MPE according to

$$h'_n = \frac{-1}{n-1} H(n), \tag{2.18}$$

where h'_n is the PE normalized using the embedding dimension, and H_n is the PE calculated from Eq. (2.1). Riedl et al. [201] determine the embedding dimension by incrementing n to find the largest corresponding normalized PE h'_n with an embedding delay τ heuristically determined from the main period length. They concluded that the h'_n with the highest entropy accurately accounts for the needed complexity of the time series, and therefore suggests a suitable embedding dimension. Rield et al. [201] show how this method provides an accurate embedding dimension for the Van-der-Pol-oscillator, Lorenz system, and the logistic map. However, the method is not automatic due to the reliance on a heuristically chosen τ.

To make the process automatic, we introduce an algorithm based on Section 2.2.2 to automatically select the correct τ, which we then use in conjunction with Eq. (2.18) to find n corresponding to the maximum h'_n. Additionally, we suggest scaling n from 3 to 8 as we have not yet found a system requiring $n > 8$ using this method.

2.3.4 Method Comparisons and Conclusions

To make conclusions about the described methods for determining τ and n, we made comparisons to values suggested by experts. The majority of the suggested parameters are taken from the work of Riedl et al. [201], while parameters for the Rossler system and sine wave are from Tao et al. [227]. Figures 2.15 and 2.16 show the calculated and suggested values for τ and n, respectively. For the exact values of τ and n from each of the parameter estimation methods please reference Tables A.2 and A.3 in the appendix, respectively. Additionally, script for reproducing the results found in this paper are provided through the Mendeley.

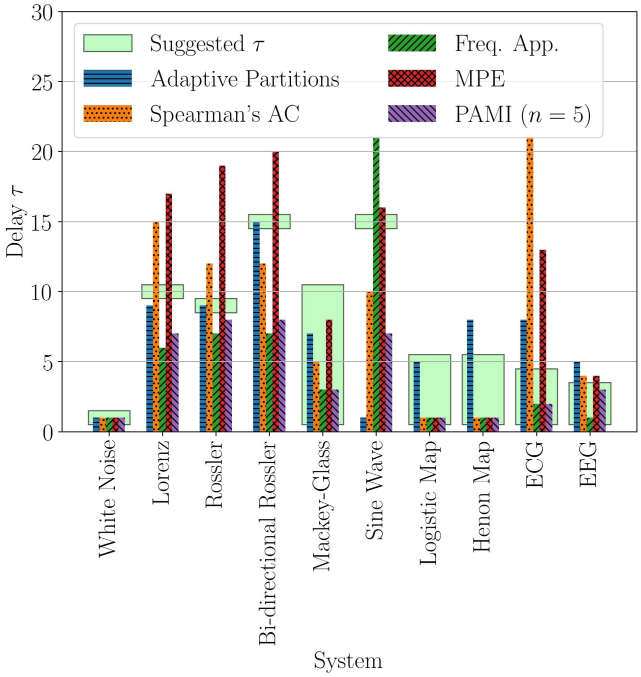


Figure 2.15: A comparison between the calculated and suggested values for the delay parameter τ. The methods investigated were MI with adaptive partitions, Spearman's Autocorrelation (AC), the frequency analysis, Multi-scale Permutation Entropy (MPE), and Permutation Auto-mutual Information (PAMI) with $n = 5$.

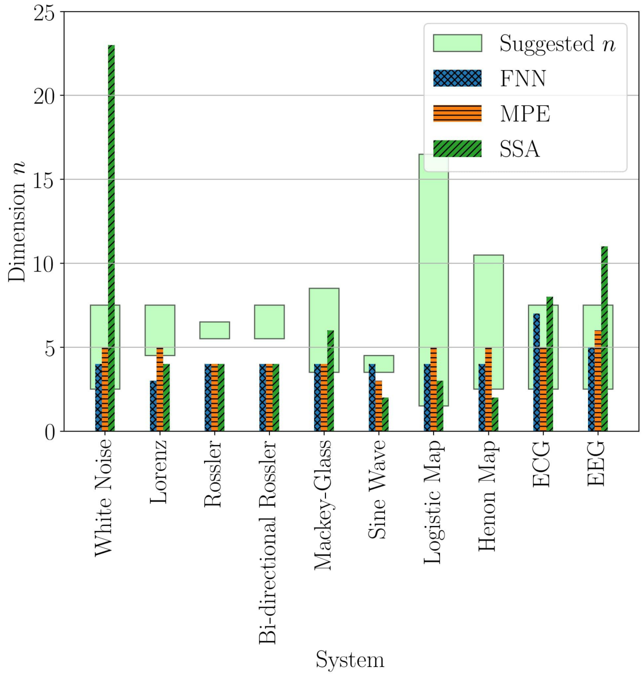


Figure 2.16: A comparison between the calculated and suggested values for the embedding dimension n. The methods investigated were False Nearest Neighbors (FNN), Multi-scale Permutation Entropy (MPE), and Singular Spectrum Analysis (SSA).

Embedding Delay Figure 2.15 shows the automatically computed τ in comparison to the expert-identified values for a variety of systems. These systems fall within several categories including the following: noise, chaotic differential equations, periodic systems, nonlinear difference equations, and medical data. The methods presented in Fig. 2.15 include PAMI from Section 2.2.5, MI calculated using adaptive partitioning from Section A.3, Spearman's Autocorrelation from Section 2.2.3, MPE from Section 2.2.2, and the frequency approach from Section 2.2.1. For the noise category we only investigated Gaussian white noise, and all the methods accurately suggest an embedding delay. For the second category of chaotic differential equations, Mutual Information approximated using adaptive partitions accurately provided suitable delay values. However, there are possible modes of failure for MI. To validate that MI is accurately selecting a value for τ, we recommend also calculating τ using the frequency approach. For the third category, periodic systems, we only investigated a simple sinusoidal function. This resulted in both MPE and the Frequency approach providing accurate suggestions. Therefor, we suggest using both of these methods to calculate τ

for periodic systems. Additionally, we do not suggest the use of MI for periodic systems as it can have early false minima resulting in inaccurate delay selection. For difference equations we found that PAMI, autocorrelation, MPE, and the frequency approach provide accurate suggestions for the delay. Finally, when testing each method on medical data with intrinsic noise, we found that the noise-robust frequency approach yielded the optimal parameter selection for τ. As a generalization of the results found, we suggest the use of MI with adaptive partitioning when selecting τ for chaotic differential equations. For periodic systems, nonlinear difference equations, and ECG/EEG data we suggest the use of the frequency approach that we developed in this paper. However, when applying the frequency approach to quasiperiodic time series with multiple harmonics of decreasing amplitude, the method may fail due to the delay being selected based on an insignificant high frequency. The use of either Spearman's autocorrelation or MPE may be more suitable under this condition. In general, multiple methods should be used for each system to validate that an accurate delay is selected due to the possible modes of failure of each method. Specifically, The frequency approach may fail if the noise does not have a Gaussian distribution, MI can fail if a false minima occurs or the relationship is monotonic, and autocorrelation can fail if the time series being analyzed does not oscillate about a fixed value.

Embedding Dimension Figure 2.16 shows the automatically computed parameter n in comparison to the expert-identified values. It can be seen that both MPE and FNN commonly had parameters within the range specified for all categories. However, SSA failed to provide a consistently suitable embedding dimension n. This leads to the conclusion that either MPE or FNN are sufficient methods for determining the embedding dimension for the majority of the considered applications. However, FNN may fail if the effects of noise are not correctly accounted for, which can lead to overly large embedding dimensions. These results also show that the dimension $n = 6$ works well for almost all applications.

2.4 Topological Methods for Delay Parameter Selection

The main thrust for this work is on parameter selection for permutation entropy and state space reconstruction using topological methods. To do this, a goal of this work is to relate the distribution of permutations formed from a given delay τ to the state space reconstruction with the same delay τ. This connections will show the time delay for both permutations and state space reconstruction are related. Establishing this relationship allows for tools from TDA to be used for delay parameter selection.

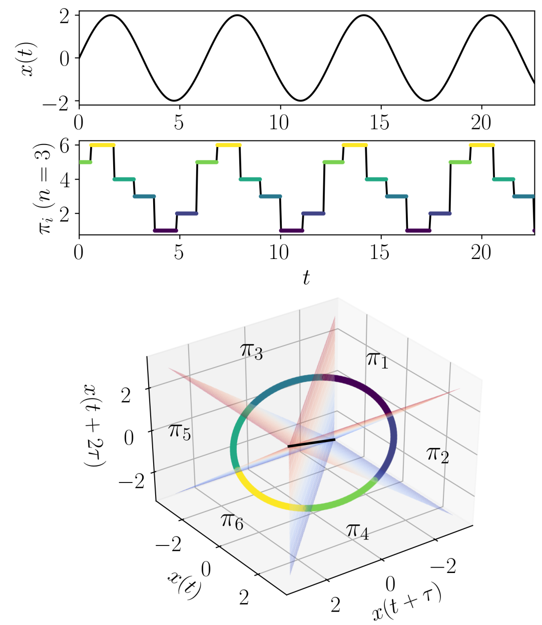


Figure 2.17: Example formation of a permutation sequence from the time series $x(t) = 2\sin(t)$ with sampling frequency $f_s = 20$ Hz, permutation dimension $n = 3$ and delay $\tau = 40$. The corresponding time-delay embedded vectors from $x(t)$ with the permutation binnings $(\pi_1, \ldots, \pi_6)$ in the state space are shown in the bottom figure.

Let me first start by redescribing the process for state space reconstruction and its similarity to permutations. As described by Takens' [226], I can reconstruct an attractor that is topologically equivalent to the original original attractor of a dynamical system by embedding a 1-D signal into $\mathbb{R}^n$ by forming a cloud of delayed vectors as $v_i = [x(t_i), x(t_{i+\tau}), x(t_{i+2\tau}), \ldots, x(t_{i+(n-1)\tau})]$ for $i \in [0, L - n\tau]$, where L is the length of the discretely and uniformly sampled signal. Permutation are formed in a very similar fashion where I take our vectors v_i and find its symbolic representation based on its ordinal ranking as explained in Section 2.1. The different permutations types can be viewed as a inequality-based binning of the $\mathbb{R}^n$ vector space of the reconstructed dynamics as shown in Fig. 2.17 for dimension $n = 3$. This provides a first intuitive understanding of the connection between permutation and state space reconstruction, however, I need to determine some connection between the optimal τ parameter used in v_i and determine if it is also an optimal delay τ PE.

Takens' embedding theorem explains that, technically, any delay τ would be suitable for reconstructing the original topology of the attractor, however, this has the requirement of unrestricted signal length and no additive noise in the signal [226]. Since this is rarely a condition found in real-world signals, a τ is chosen to unfold the attractor such that the effects of noise have a minimal effect on the topology of the reconstructed dynamics.

Let us now explain what I mean by the correspondence between τ and the unfolding of the dynamics and what effect this has on the corresponding permutations. If the delay τ is too small (e.g. $\tau = 1$ for a continuous dynamical system with a high smapling rate) the delay embedded reconstructed attractor will be clustered around the hyper-diagonal in $\mathbb{R}^n$ space. Additionally, the corresponding permutations will be overwhelmingly dominated by the permutation types π_1 and $\pi_{n!}$ with these two permutations being of the all increasing and decreasing ordinal patterns, respectively. The dominance of these two permutations for a delay τ that is too small was termed by De Micco et al. [59] as the "redundancy effect." For an example of this see the permutation distribution and clustering about the hyper-diagonal in $\mathbb{R}^3$ as shown in Fig. 2.18. This example is based on the x-solution to the periodic Rossler dynamical system as described in Section C.1. As the delay increases pass the redundancy effect, the reconstructed attractor begins to unfold to have a similar shape and topology as the true attractor. Corresponding with this unfolding, as the delay increase the permutation distribution tends towards a more equiprobable distribution (See Fig. 2.18 at $\tau \approx 14$). A way of summarizing the permutation probability distribution is actually through PE itself and more specifically the analysis of Multi-scale Permutation Entropy (MsPE). Riedl et al. [200] showed how after the redundancy effect there is a suitable delay for PE, which I related to the first maxima of the MsPE plot [161]. The MsPE plot for our periodic Rossler example is shown in Fig. 2.18. Let us also look at the MI plot as a comparison. The theory behind MI states that at the first minima of the mutual information between $x(t_i)$ and $x(t_{i+\tau})$ the delay τ accurately provides a suitable delay for state space reconstruction. By a quick investigation of the MI function I can observe that there is a high degree of correlation between the MI function and the MsPE function with the first maxima of MsPE being approximately at the same τ as the first minima of MI. When the delay becomes signficantly larger than the first minima of MI or maxima of MsPE, the permutation distribution begins to fluctuate as shown in Fig. 2.18. This effect was termed as the “irrelevance" effect by De Micco et al. [59]. This increasing of τ beyond the the first minima also correlates with, as described by Kantz and Schreiber [105], the reconstruction filling an overly large space with the vectors already being independent. Additionally, at a minima beyond the first minima, Fraser and Swinney [78] showed how the reconstructed attractor shape will no longer qualitatively match the shape of the true state space.

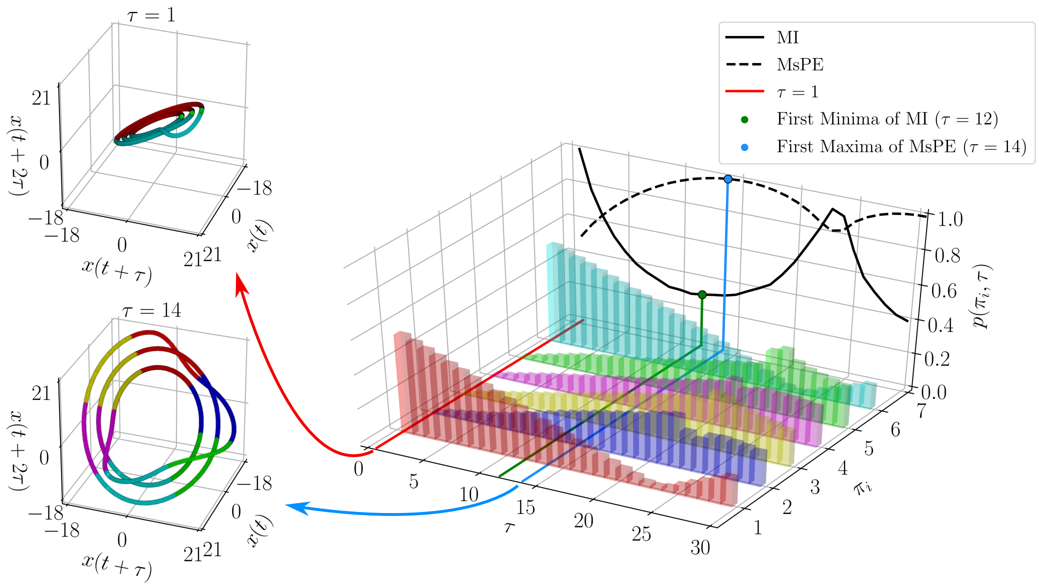


Figure 2.18: Example comparing first minima of mutual information and first maxima of multi-scale permutation entropy, which demonstrates the correspondance between the two. On the left are the $n = 3$ time delayed state space reconstructions with an inaccurately chosen $\tau = 1$ and appropriate $\tau = 14$. On the right shows the permutation distribution as τ increases and the associated multi-scale permutation entropy and mutual information plots.

I have now shown with both an example and a qualitative analysis that the optimal τ for permutation entropy and the state space reconstruction are correlated with the unfolding of the reconstructed attractor. While I do not provide a proof that PE and state space reconstruction use the same τ, it has recently been shown that there is a connection between co-homology, information theory, and probability does exist [18], which strengthens our qualitative analysis of this connection. In the future sections I will leverage tools from TDA to determine the optimal τ associated with the unfolding of the attractor. Some of the methods that I have researched are an adaptation of SW1PerS [177] for the delay parameter selection and two methods to estimate the dominate frequency in a signal using sublevel set persistence which can be used for delay parameter selection.

2.4.1 Finding τ Using SW1PerS

In this section I develop a novel method implementing persistent homology for estimating an appropriate delay for permutations and state space reconstruction. Specifically, we investigate the effects of varying $\tau \in [1, \tau_{\text{max}}]$ on the calculation of the maximum persistence and the periodicity score from SW1PerS [180]. Perea and Harer developed SW1PerS as a TDA method for measuring periodicity in a time series; however, our goal is to leverage this method for use in determining a suitable selection of τ for permutation entropy and state space reconstruction based on the unfolding of an attractor and the associated 1-D persistent homology.

SW1PerS uses 1-D persistent homology to measure how periodic or significant the circular shape of an embedded time series (point cloud) is as τ increases, which corresponds to the embedding window size increasing as $w = m\tau$ with m as the embedding dimension of the sliding window vector. Specifically, the sliding window SW for SW1PerS is defined as

$$SW_{m,\tau} f(N)(t) = [f(N)(t), f(N)(t+\tau), \ldots, f(N)(t+m\tau)], \tag{2.19}$$

where $f(N)(t)$ is a truncated Fourier series of the signal and τ and m are, respectively, SW1PerS' embedding delay and dimension. Applying Eq. (2.19) to a sliding window of width w across the domain of the time series results in a collection of vectors known as a point cloud, which live in an m-dimensional Euclidean space. However, it may not be desirable to use all of the embedded

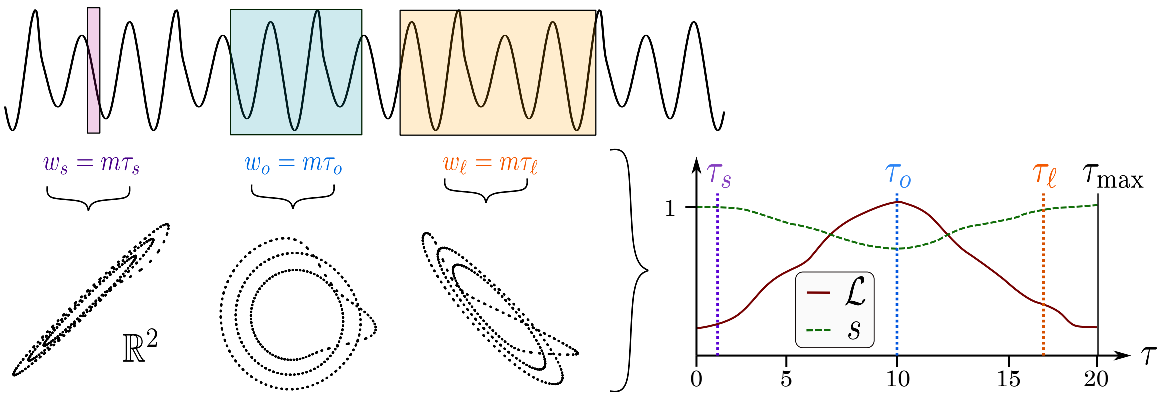


Figure 2.19: Example showing three sample windows with $m = 2$ of increasing size, which is slid across the entire time series (periodic Rossler system) resulting in the embedded time series in $\mathbb{R}^2$. The window size is defined as $w = m\tau$ with (left) $w_s m\tau_s$ being too small with $\tau_s = 1$ and an embedding shape concentrated on the diagonal line and a high periodicity score s and low $\mathcal{L}$, (middle) w_o is properly sized and results in a minimum periodicity score s and maximum $\mathcal{L}$ suggesting an optimal delay $\tau_o = 10$, and (right) w_ℓ with $\tau = 17$ is too large and results in a high periodicity score s and low $\mathcal{L}$.

vectors from (2.19) due to the $O(n^3)$ time complexity of calculating the persistent homology of a point cloud via the Vietoris-Rips complex. To improve the calculation time we chose to use a sparse version of the point cloud through a subsampling to have n_T windows from the original point cloud. We chose to set the number of sliding windows as $n_T = 200$ to be sufficiently high to detect circular structure in the embedding.

For SW1PerS, m is determined based on the theory developed by Perea et al. [180], which showed the necessary value of m for reconstruction is bounded by $m \geq 2N$ (here we use $m = 2N$), where N is the number of Fourier terms necessary for reconstructing the signal to some desired accuracy. In this work we automate choosing N by approximating the Fourier series using the discrete Fourier transform. To do this we compute the normalized ℓ_2 norm between the reconstructed time series from the truncated Fourier series and the original signal. The ℓ_2 norm is used to obtain the value of N that yields an error within a desired threshold of $\ell_2(N) < 0.25$. Specifically, if we let the time series X be a discrete time sampling of a piece-wise smooth signal $x(t)$, then the N-partial sum of the Fourier series of $x(t)$ can be approximated according to

$$f(N)(t) = \frac{1}{|X|}\sum_{k=0}^{N}\Bigg(\sum_{j=0}^{|X|} X(j)e^{-2\pi ijk/T}\Bigg)e^{2\pi ikt/T}, \tag{2.20}$$

where X is the original signal that has been point-wise centered and normalized with $|X|$ as the length of the signal. As a rule of thumb $N \approx |X|/8$ yields an accurate reconstruction of $x(t)$ [8], which we use as an upper bound of N. The relative ℓ_2 norm that measures the error between time series X and its reconstruction $f(N)(t)$ is given by

$$\ell_2(N) = \frac{\left(\sum_{j=0}^{|X|}\left[X(j) - f(N)(j)\right]^2\right)^{1/2}}{\left(\sum_{j=0}^{|X|} X^2(j)\right)^{1/2}}. \tag{2.21}$$

For our application, we consider $f(N)(t)$ as sufficiently close to $x(t)$ when we find a value of N for which $\ell_2(N) < 0.25$. We chose 0.25 as it provides dimension m which are not overly large ($m < 10$ typically) and it deals with the possibility of moderate additive noise in the signal. Using the truncated Fourier series we are also able to determine an upper bound for τ using the Nyquist sampling criteria as

$$\tau_{\max} = \frac{f_s}{2\min(f_{\text{sig}})}, \tag{2.22}$$

where f_{sig} are the N significant frequencies from the truncated fast Fourier transform. We now have all the components we need to apply SW1PerS.

To determine the optimal delay using persistent homology, we investigated two summaries of the resulting persistence diagrams from SW1PerS: (1) the maximum lifetime as

$$\mathcal{L} = \max(\text{pers}(\tilde{\mathcal{D}}_1) \tag{2.23}$$

with $\tilde{\mathcal{D}}$ as the SW1PerS persistence diagram and (2) the periodicity score, which was defined in [178] as

$$s = 1 - \frac{r_B^2 - r_D^2}{3}, \tag{2.24}$$

where r_B and r_D are the birth and deaths times associated to $\max(\text{pers}(\tilde{\mathcal{D}}_1)$. We then calculate these point summaries for each τ as we vary $\tau \in [1, \tau_{\max}]$ to generate $\vec{s}$ and $\vec{\mathcal{L}}$ for the periodicity scores s and persistence maximums $\mathcal{L}$, respectively.

To demonstrate the functionality of this method, let us implement a simple example using the periodic Rossler system (see Fig. 2.19). This example shows three different window sizes for embedding dimension $m = 2$ (this dimension was chosen for visualization purposes)and $\tau = 1$, 10, and 17, to show the resulting scores for a small, optimal and overly large window size, respectively. Figure 2.19 shows that the optimal window size at $\tau_o = 10$ results in a maximum $\mathcal{L}$ and minimum s over the range $\tau \in [1, \tau_{\text{max}}]$, where $\tau_{\text{max}} = 20$ from the truncated Fourier spectrum. This suggests that an appropriate delay for both state space reconstruction via Takens' embedding and permutation entropy is $\tau = 10$.

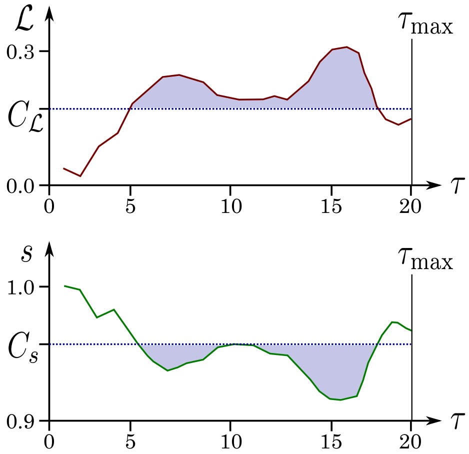


Figure 2.20: Example periodicity s and max persistence $\mathcal{L}$ plots for the chaotic Rossler system with associated cutoffs to determine the average τ.

For a chaotic time series, choosing τ from the minimum or maximum of s and $\mathcal{L}$ is not as trivial as the example shown in Fig. 2.19. Specifically, due to the non-linear behavior of a chaotic time series there may not always be a clear, single minima as shown in the example periodic Rossler system, but rather two or more local minima with similar prominence. To accurately approximate the average minima and select an associated delay τ, we will use heuristic cutoffs C_s and $C_{\mathcal{L}}$, where these cutoffs are defined as $C_s = \frac{1}{2}[\max(s) + \min(s)]$ and $C_{\mathcal{L}} = \frac{1}{2}[\max(\mathcal{L}) + \min(\mathcal{L})]$. Specifically, we will choose τ based on the average τ such that $s \geq C_s$ or $\mathcal{L} \geq C_{\mathcal{L}}$. To demonstrate this method we use a chaotic response of the Rossler system and calculate the two cutoffs as shown in Fig. 2.20. This example results in an average delay greater than $C_{\mathcal{L}}$ as $\tau = 12$ and less than C_s as $\tau = 12$. This example demonstrated that the method of selecting the average τ greater or less than the cutoffs results in a similar τ for both periodic and chaotic time series.

2.4.2 Finding τ Using Sublevel Set Persistence

In this section our goal will be to leverage sublevel set persistence for the selection of τ for both state space reconstruction and permutation entropy. Specifically, our goal is to automate the frequency analysis method [149] for selecting τ for state space reconstruction by analyzing both the time and frequency domain of the signal using sublevel set persistence. The method developed by Melosik and Marszalek [149] uses the maximum significant frequency $f_{\max}$ and the sampling frequency f_s to select an appropriate τ as

$$\tau = \frac{f_s}{\alpha f_{\max}}, \tag{2.25}$$

where $\alpha \in [2, 4]$ with an $\alpha = 2$ associated to the Nyquist sampling rate and $\alpha > 4$ producing an oversampling. Since this method was developed using the Nyquist sampling rate, we will first include its associated assumptions as a continuous, bandlimited signal. This frequency based approach was founded on the requirements for suitable delays for the 0/1 test on chaos and the a heuristic comparison between the Lorenz attractor and a delay-reconstruction of the Lorenz attractor. The heuristic comparison showed that this frequency approach actually provided more accurate delay parameter selections for state space reconstruction than the mutual information function when trying to replicate the shape of the attractor. Unfortunately, a major drawback of this method is the non-trivial selection of $f_{\max}$. In Melosik's and Marszalek's original work [149] the maximum frequency was manually selected using a normalized, such that $\in [0, 1]$, Fast Fourier Spectrum (FFT) cutoff of approximately 0.01, which does not address the possibility of additive noise.

In our previous work [161] we approximated the maximum "significant" frequency in a time series using the FFT and defining a power spectrum cutoff based on the statistics of additive noise in the FFT. An issue with this method for non-linear time series is that the Fourier spectrum does not easily yield itself to selecting the maximum "significant" frequency for chaotic time series even with an appropriately selected cutoff to ignore additive noise. Additionally, the method was only developed for Gaussian White Noise (GWN) contamination of the original time series.

To improve the selection of the maximum frequency in this section we developed two novel methods based on 0-D sublevel set persistence. We chose to use 0-D sublevel set persistence due to its computational efficiency and stability for true peak selection [49, 115]. The first method is based on a time domain analysis of the sublevel set lifetimes (see Section 2.4.2) and the second implements a frequency domain analysis using sublevel set persistence and the modified z-score (see Section 2.4.2).

Time Domain Approach The first approach we implement for estimating the maximum significant frequency of a signal is based on a time domain analysis of the sublevel set persistent. This process uses the time ordered lifetimes from sublevel set persistence diagram. We previously introduced time ordered lifetimes and a cutoff separating the sublevel sets associated with noise in [11]. Here we use those methods and results to find the time t_B in which all the significant sublevel sets are born. Fig. 2.21 shows a resulting time order lifetimes plot where the time between two adjacent lifetimes is defined as T_{B_i}. If we use T_{B_i} as an approximation of a period in the time series, then

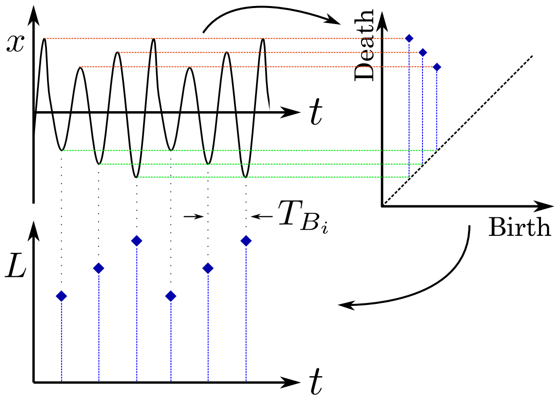


Figure 2.21: Example demonstrating process from time series x (periodic Rossler system) to sublevel set persistence diagram to time ordered lifetimes on the bottom left. Additionally, on the bottom left shows a sample time periodic between sublevel sets as T_{B_i}.

we can calculate the associated frequencies as $f_i = 1/T_{B_i}$ Hz. If we then look at the distribution of f_i, the maximum “significant" frequency can be approximated using the 75% quantile of the distribution of the frequencies as $f_{\max} \approx Q_{75}(f)$. This quantile allows for a few outlying high frequencies to occur without having a significant effect on the estimate of the maximum frequency.

Applying this method to the periodic Rossler system results in a $\tau = 10$ with the corresponding

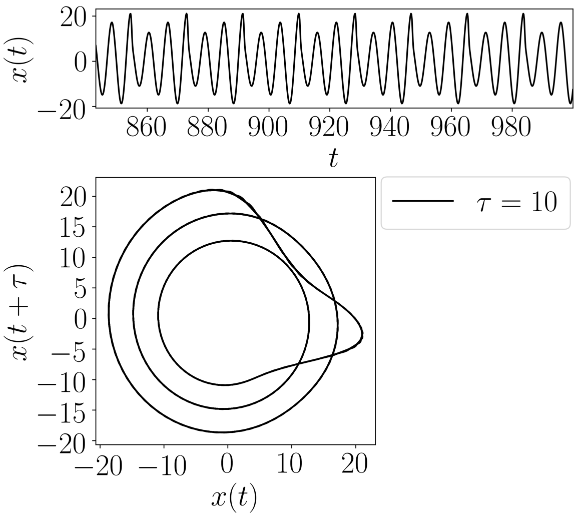


Figure 2.22: Example demonstrating the time delay $\tau = 10$ result for the periodic Rossler example time series shown in the top figure and the resulting $n = 2$ Takens' embedding.

state space reconstructions for $n = 2$ shown in Fig. 2.22. This suggested delay is very similar to that of mutual information ($\tau = 12$). This result suggests that the time-domain analysis for selecting the maximum frequency and corresponding delay functions should accurately suggest an appropriate delay for permutation entropy and state space reconstruction.

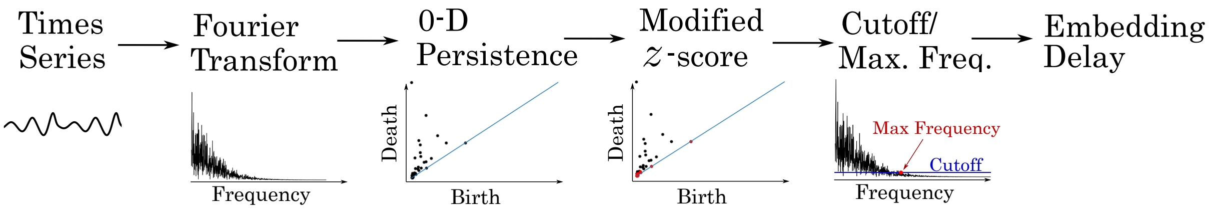


Figure 2.23: Overview of procedure for finding maximum significant frequency using 0-dimensional sublevel set persistence and the modified z-score for a signal contaminated with noise.

Fourier Spectrum Approach In this section we present a novel TDA based approach for finding the noise floor in the Fourier spectrum for selecting the maximum significant frequency f_{max} to be used for selecting τ for PE through Eq. (2.25). Specifically, we show how the 0-dimensional sublevel set persistence, a tool from TDA discussed in Section 1.1, can be used to find the significant lifetimes and associated frequencies the frequency spectrum. while it would then be ideal to analyze the theoretical distribution of the sublevel set lifetimes of the FFT of a random process, this would not be a trivial task. There have been studies on pushing forward probability distributions into the persistence domain [3, 4, 104], but it is difficult to obtain a theoretical cutoff value in persistence space. Therefor, without doing an in depth statistical analysis of the distributions, we will calculate a instead use the modified z-score. Specifically, we separate the noise lifetimes from significant lifetimes through the use of the modified z-score, which allows us to find the noise floor and maximum significant frequency via a cutoff. This process for finding the cutoff and associated maximum frequency is illustrated in Fig. 2.23. The following paragraphs give an overview of the modified z-score and cutoff analysis.

Modified z-score The modified z-score z_m is essential to understanding the techniques used for isolating noise from a signal [209]. The standard score, commonly known as the z-score, uses the mean and the standard deviation of a dataset to find an associated z-score for each data point and is defined as

$$z = \frac{x - \mu}{\sigma}, \tag{2.26}$$

where x is a data point, μ is the mean, and σ is the standard deviation of the dataset, respectively. The z-score value is commonly used to identify outliers in the dataset by rejecting points that are above a set threshold, which is set in terms of how many standard deviations away from the mean are acceptable. Unfortunately, the z-score is susceptible to outliers itself with both the mean and the standard deviation not being robust against outliers [130]. This led Hampel [91] to develop the modified z-score as an outlier detection method that is robust to outliers. The logic behind the modified z-score or median absolute deviation (MAD) method is grounded on the use of the median instead of the mean. The MAD is calculated as

$$\text{MAD} = \text{median}(|x - \tilde{x}|), \tag{2.27}$$

where x is a data set and $\tilde{x}$ is the median of the dataset. The MAD is substituted for the standard deviation in Eq. (2.26). To complete the modified z-score, Iglewicz and Hoaglin [99] suggested to additionally substitute the mean with the median. The resulting equation for the modified z-score is then quantified as

$$z_m = 0.6745\frac{x - \tilde{x}}{\text{MAD}}, \tag{2.28}$$

where the value 0.6745 was suggested from [99]. We can now use the modified z-score z_m for evaluating the “significance" of each point in the sublevel set persistence diagram of the Fourier spectrum. A threshold for separating noise in the persistence domain is discussed in the following paragraph.

Threshold and Cutoff Analysis To determine the noise floor in the normalized Fast Fourier Transform (FFT) spectrum, we compute the 0-dimensional persistence of the FFT. This provides relatively short lifetimes for the noise, while the prominent peaks, which represent the actual signal, have comparatively long lifetimes or high persistence. To separate the noise from the outliers we calculate the modified z-score for the lifetimes in the persistence diagram. We can then determine if the lifetime is associated to noise or signal based on a z_m cutoff as D, where we can label a lifetime as signficant (an outlier) if $z_m > D$. Iglewicz and Hoaglin [99] suggest a z_m threshold of $D = 3.5$ based on an analysis of 10,000 random-normal observations. However, we apply both the FFT and 0-D sublevel set persistence to the original signal so it would be appropriate to determine if this cutoff is suitable for our application. To do this we used a signal of 10,000 random-normal observations and applied a FFT and then calculated 0-D sublevel set lifetimes as our signal to analyze using the modified z-score z_m. For an accurate cutoff we would expect to label all of the lifetimes as noise with $z_m < D$ since the signal is observations are composed of pure noise. As shown in Fig. 2.24, a threshold of approximately $D = 4.8$ labels all of the lifetimes as noise. This threshold was rounded up to 5 for simplicity. We can now simply define a cutoff based on the labeling of of each lifetime from the modified z-score with Cutoff $= \max(\text{lifetime}_{\text{noise}}$.

We can now find the maximum significant frequency $f_{\max}$ as the highest frequency in the Fourier spectrum with an amplitude greater than the specified cutoff. For this method to accurately function, it is required that there is some additive noise in the time series. To accommodate this, additive Gaussian noise with Signal-to-Noise Ratio of 30 dB is added to the time series before

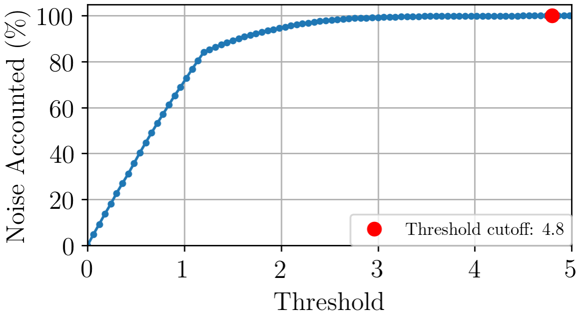


Figure 2.24: Percent of the persistence points from 0-D sublevel set persistence of the FFT of GWN using the modified z-score with the provided threshold ranging from 0 to 5.

calculating the FFT. If we apply this method to the example periodic Rossler system time series we find a suggest delay of $\tau = 5$. In comparison to mutual information this delay is approximately half as large as it should be. However, we will investigate its accuracy on several other systems in Section 4.3 to make conclusions on the functionality of this method for selecting τ.

2.4.3 Permutation Dimension

In this section we will show that, contrary to the delay selection, the dimension for permutation entropy is not related to that of Takens' embedding. Additionally, we will provide a simple method for selecting an appropriate permutation dimension based on the permutation distribution.

The goal of permutation entropy is to differentiate between the complexity of a time series when there is a dynamic state change (e.g. periodic compared to chaotic), so the dimension should be chosen such that it is large enough to capture these changes. To accomplish this we suggest that permutations of the time series do not occupy all of the possible permutations, but rather only a fraction of the permutations when an appropriate delay is selected. This criteria is set so that a change can be captured by an increase/decrease in the number of permutations and their associated probabilities. Because of this, we suggest a dimension where, at most, only 50% of the permutations are used. However, it may be more suitable to select a dimension where a lower percent are used (e.g. 10%).

To begin this method for determining if the dimension is high enough to capture the time series complexity we will define N_π as the number of permutation types where the probability of that permutation type is significant. Specifically, we will consider the probability of that permutation to be significant if the number of occurrences of permutation π is greater than 10 percent of the maximum number of occurrences of any permutation type from dimension n. The permutation delay τ was selected from the expert suggested values provided in [161, 200]. We can now express our needed dimension as the ratio and inequality

$$\frac{N_\pi}{n!} \leq R, \tag{2.29}$$

where $R = 0.50$ for the suggested maximum 50% criteria.

To compare this dimension to stand Takens' embedding tools for selecting n we will implement four examples:

$$\begin{aligned} x_1(t) &= \frac{t}{10} \\ x_2(t) &= \sin(t) \\ x_3(t) &= \sin(t) + \sin(\pi t) \\ x_4(t) &= \mathcal{N}(\mu = 0, \sigma^2 = 1), \end{aligned} \tag{2.30}$$

where $t \in [0, 100]$ with a sampling rate of 20 Hz and $\mathcal{N}$ is Gaussian additive noise. By applying Eq. (2.29) to the time series in Eq. (2.30), we can suggest dimensions of 2, 4, 6, and 7 for time series $x_i(t)$ with $i \in [1, 4]$ as shown in Fig. 2.25, respectively.

In comparison to Takens' embedding, for time series $x_2(t)$ dimension $n = 2$ would be sufficient, but if this was used for permutation entropy, no increase in complexity could be detected. Additionally, this result suggests an upper bound on the dimension for permutation entropy as $n \approx 9$ as the ratio in Eq. (2.29) is approximately 0 for dimensions $n > 9$. As a rule of thumb from this result, a dimension of 8 would be suitable for almost all applications, but it would be optimal to minimize the dimension to reduce the computation time of PE. In Section 4.3 we will show the resulting suggested dimensions using this method for a wide variety of dynamical systems.

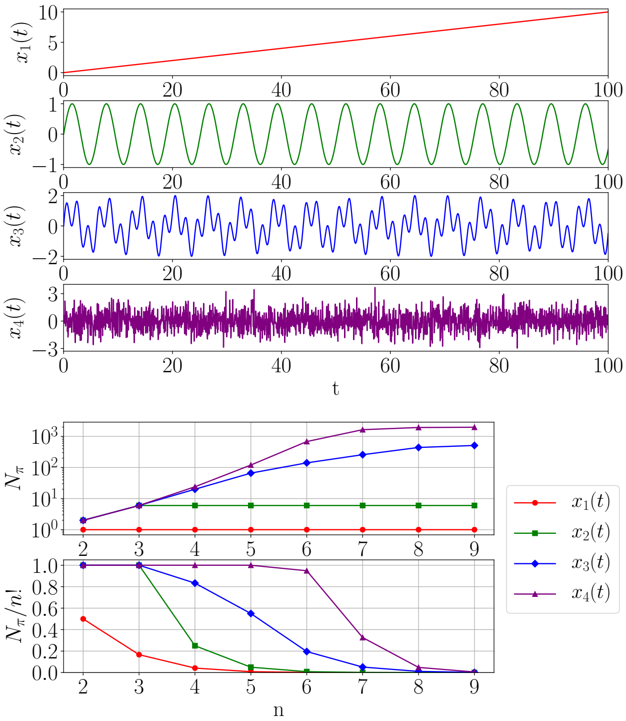


Figure 2.25: Percent of permutations used $R = N_\pi/n!$ for each example time series (see Eq. (2.30)) as the dimension is incrmented.

2.4.4 Results for Topological Data Analysis Methods

In this section we will provide the results of the parameter selection methods. First, in Section 2.4.4, we calculate the delay parameter for a wide variety of dynamical systems and data sets using mutual information and the the automatic TDA-based methods described in this manuscript. Unfortunately, the optimal parameters can not be decided based on a simple entropy value comparison since there is no direct equivalence between PE and other entropy approximations of a signal such as Kolmogorov-Sinai (KS) entropy with only a bounding between the two as KS ≤ PE [107]. Therefor, to determine the accuracy of the automatically selected PE parameters we implement two other methods of comparison. The first comparison is to expert suggested parameters for a wide variety of systems (see Section 2.4.4). The second approach is a comparison to optimal parameters based on having a significant difference between the PE of two different states for each system. Of course the second method has the requirement that we have a system model or data set with two different states for comparison, which is not typically the case, but does allow for an approximation of optimal PE parameters for these systems. These comparisons are discussed in Section 2.4.4.

The second half of the results, in Sections 2.4.4 and 2.4.4, is based on analyzing the robustness of the automatic TDA-based PE parameter selection methods to additive noise contamination and signal length requirements, respectively.

Parameter Value Comparison for Common Dynamical systems To determine a range of approximately optimal PE parameters we will quantify the difference between PE values for a wide range of delays and dimensions with the difference for a given τ and n calculated as

$$\Delta h_n(\tau) = h_n^{(\mathrm{Ch.})}(\tau) - h_n^{(\mathrm{Pe.})}(\tau), \tag{2.31}$$

where the superscripts Ch. and Pe. represent the PE calculation on the chaotic and periodic time series for the given dynamical system. The specific parameters used to generate periodic and chaotic responses for each system are described in the Appendix Section C.1. If we apply Eq. (2.31) to the Rossler system for $\tau \in [1, 15]$ and $n \in [3, 10]$ we find that $\Delta h_n(\tau)$ is significant when $\tau \in [9, 15]$ and $n \in [6, 10]$ as shown in Fig. 2.26. However, as mentioned previously in section 2.4.3, dimensions greater than 8 can be computationally expensive. We consider this range where $\Delta h_n(\tau)$ is relatively large as the range of optimal PE parameters to be compared to. We

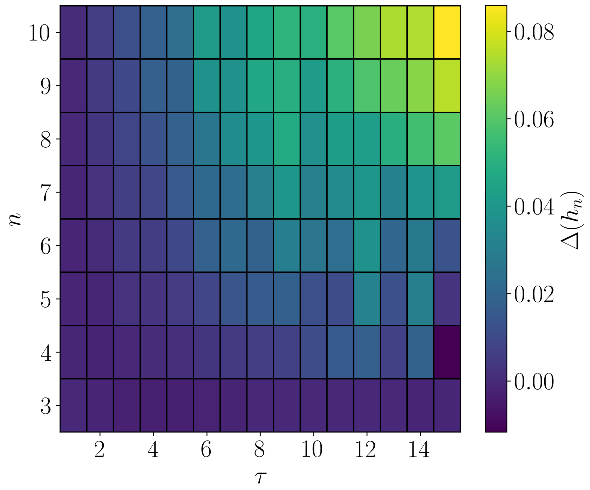


Figure 2.26: Example showing difference in PE (see Eq. (2.31)) for periodic and chaotic dynamic states of the Rossler system for a wide range of PE parameters.

repeated this process for finding the optimal parameter ranges for PE using a similar procedure to this Rossler example as shown in Table A.2.

Table 2.1: A comparison between the calculated and suggested values for the delay parameter τ. The shaded (red) cells highlight the methods that failed to provide a close match to the suggested delay.

Cat.	system	State	Delay					Dim.		Exp. Sugg. Parameters			Opt. Param. Range	
			1-D Pers. Homol.		Sublevel Set Pers.		MI	R						
			s	L	t	f		0.5	0.1	τ	n	Ref.	τ	n
Noise Models	Gauss.	-	1	1	1	1	3	7	8	1	3-6	[200]	-	-
	Uniform	-	1	1	1	1	3	7	8	-	-	-	-	-
	Rayleigh	-	1	1	1	1	2	7	8	-	-	-	-	-
	Expon.	-	1	1	1	1	2	7	8	-	-	-	-	-
Cont. Flows	Lorenz	Per.	13	11	11	7	11	5	6	10	5-7	[200]	8-17	5-10
		Cha.	12	13	12	9	12	5	7					
	Rossler	Per.	10	10	10	8	11	5	6	9	6	[228]	9-15	6-10
		Cha.	12	12	12	10	12	5	6					
	Bi-direct. Rossler	Per.	19	17	16	9	15	5	6	15	6-7	[200]	11-22	6-10
		Cha.	18	16	16	15	17	5	6					
	Mackey Glass	Per.	7	7	6	3	8	5	6	10	4-8	[253]	6-12	4-8
		Cha.	7	7	7	4	9	5	7					
	Chua Circuit	Per.	16	17	17	11	19	5	6	20	5	[213]	16-24	5-10
		Cha.	37	52	17	19	19	5	7					
	Coupled Ross.-Lor.	Per.	8	8	8	7	9	4	6	8	3-10	[222]	5-11	4-9
		Cha.	12	10	8	5	10	5	7					
	Double Pendul.	Per.	16	16	17	11	18	4	5	-	-	-	8-20	5-10
		Cha.	13	12	10	8	14	6	7					
Period. Funct.	Periodic	-	12	12	13	24	16	4	5	15	4	[228]	-	-
	Quasi	-	45	46	25	49	26	6	7	-	-	-	-	-
Maps	Logistic	Per.	1	1	1	1	3	4	5	1-5	4-7	[200]	1-4	3-6
		Cha.	1	1	1	1	16	4	6					
	Henon	Per.	2	2	1	1	3	4	5	1-2	2-16	[200]	1-5	5-8
		Cha.	1	1	1	1	16	6	7					
Med. Data	ECG	Cont.	9	9	22	7	17	5	6	10-32	3-7	[139]	6-23	5-7
		Arrh.	13	13	15	6	15	5	6					
	EEG	Cont.	19	18	1	3	6	8	8	1-3	3-7	[200]	2-6	4-7
		Seiz.	10	4	12	4	10	5	7					

To verify our TDA-based methods for determining τ, Table A.2 compares our results to the values from a wide variety of systems for both the first minima of the mutual information function and from expert suggestions, including several listed by Riedl et al. [200]. The table also shows the resulting permutation dimensions suggested from the permutation statistics as described in Section 2.4.3 for both $R = 0.1$ and $R = 0.5$ from Eq. (2.29). For these systems we have also included, where applicable, the delay and dimension parameter estimates for both periodic and chaotic responses to validate each methods robustness to chaos and non-linearity. However, for the medical data section we instead we included a healthy/control and unhealthy (arrhythmia for ECG and seizure for EEG) as a substitute for a periodic and chaotic response, respectively. A detailed description of each dynamical system or data set used, including parameters for periodic and chaotic responses, is provided in the Appendix.

In table A.2 we have highlighted the methods that failed to provide an accurate delay τ in red. We will now go through the methods and highlight the advantages and drawbacks as well as general suggestions for which method to use based on the category.

Noise Models: We only have one expert suggestion of parameters for the noise models category, which is for Gaussian white noise (Gauss.) as $\tau = 1$ and $n \in [3, 6]$. In regards to the delay, all TDA based methods show an accurate selection of $\tau = 1$, however the suggestion of $\tau = 3$ from Mutual Information (MI)is slightly higher than suggested. We found that the expert suggested dimensions of 3 to 6 is significantly lower than the minimum dimension suggested by our permutation statistics method of $n = 7$. As mentioned in Section 2.4.3, we believe it is necessary to have the number of permutation used to be atleast less than 50% of all the permutation available, which corresponds to a dimension $n = 7$ for Gaussian noise. From this logic we can conclude that a suitable dimension should actually be atleast $n = 7$ if any increase in the time series complexity is expected. If only decreases in complexity are expected, then a dimension of $n = 6$ may be suitable.

Continuous Flows: The next category is of continuous flows described by systems of non-linear differential equations. As shown in Table A.2, both the time domain analysis via sublevel set persistence and mutual information provide accurate delay suggestions for all of the examples. However, the 1-D persistent homology methods discussed in Section 2.4.1 also provide an accurate delay for every systems besides for the chaotic Chua circuit. This failure was most likely due to an inaccurate selection of the maximum significant frequency and associated $\tau_{\max}$. We can also conclude that the frequency domain analysis using sublevel set persistence consistently provided delays that were too small. In regards to the dimension, the suggested dimensions from the permutation statistics agreed with the delay suggested by experts for all of the continuous flow systems. This suggests that the method of selecting a dimension for permutation entropy using the method described in Section 2.4.3 is accurate for simulations of continuous differential equations.

Periodic Functions: For periodic functions, including a simple sinuisodal function (periodic) and two incommensurate sinuisoidal functions (quasiperiodic), our results in Table A.2 show that all methods, including mutual information, provide accurate selections of τ except the Fourier spectrum analysis via sublevel sets. This method results in a significantly high suggestion for τ. In regards to the dimension selection, our results using the permutation statistics method described in Section 2.4.3 agree with the expert suggested minimum dimension of $n = 4$.

Maps: When selecting the delay parameter for permutations and takens' embedding for maps we found that all of the topological methods suggested accurate delay parameters, while the standard mutual information methods selected overly large delay parameters when the maps are exhibiting a chaotic state. Therefor, we suggest the use of one of the topological methods when estimating the delay parameter for maps. For the permutation dimension we found a suggested dimension from $n \in [4, 7]$, in comparison to the expected suggested dimension ranging from 2 to 16. While the range suggested from the permutations statistics as described in Section 2.4.3 falls within the range suggested by experts, their range is too broad. Specifically, a dimension greater than 9 can be computationally cumbersome, and a dimension lower than 4 would not show significant differences for dynamic state changes. Therefor, we suggest the user of our narrower range of dimension from $n \in [5, 6]$ for maps which agrees with our optimal PE parameter range.

Medical Data: The medical data used in this study Inherently has some degree of additive noise, which provides a first glimpse into the noise robustness of the delay parameter selection methods investigated. However, a more thorough investigation will be provided in Section 2.4.4. From our analysis, we disagree with the delay from experts suggested as $\tau \in [1, 3]$, but rather the delay selected from either mutual information or the time domain analysis of sublevel set persistence. The general selection for delays between 1 and 3 does not account for the large variation in possible sampling rates. If the small delay is used in conjunction with a high sampling rate, an inaccurate delay could be selected resulting in indistinguishable permutation entropy values as the dynamic state changes. In regards, to the permutation dimension n, we believe that a more appropriate dimension, in comparison to the values suggested by experts, should range between 5 and 7 for medical data applications.

Robustness to Additive Noise To determine the noise robustness of the delay parameter selection methods investigated in this work we will use an example time series. Specifically, we will use the x solution to the periodic Rossler system. We will use additive Gasussian noise $\mathcal{N}(\mu = 0, \sigma^2)$, where σ is determined from the Signal-to-Noise Ratio (SNR). The SNR is a measurement of how much noise there is in the signal with units of decibels (dB)and is calculated as

$$\mathrm{SNR}_{\mathrm{dB}} = 20 \log_{10}\left(\frac{A_{\mathrm{signal}}}{A_{\mathrm{noise}}}\right), \tag{2.32}$$

where A_{signal} and A_{noise} are the Root-Mean-Square (RMS) amplitudes of the signal and additive noise, respectively. If we manipulate Eq. (2.32) we can solve for A_{noise} as

$$A_{\mathrm{noise}} = A_{\mathrm{signal}} 10^{-\frac{\mathrm{SNR}_{\mathrm{dB}}}{20}}. \tag{2.33}$$

Because $x(t)$ is a discrete sampling from a continous system with $t = [t_1, t_2, \dots, t_N]$, we calculate A_{signal} as

$$A_{\mathrm{signal}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} [x(t_i) - \bar{x}]^2}, \tag{2.34}$$

where $\bar{x}$ is the mean of x and is subtracted from $x(t)$ to center the signal about zero. with A_{noise} calculated, we set the additive noise standard deviation as $\sigma = A_{\mathrm{noise}}$.

We applied a sweep of the SNR from 1 to 40 in increments of 1 with each SNR being repeated for 30 unique realizations of the noise $\mathcal{N}(0, A_{\mathrm{noise}}^2)$. For each realization of $x(t) + \mathcal{N}(0, A_{\mathrm{noise}}^2)$ the delay parameters were calculated using all 5 methods: sublevel set persistence of the frequency domain $\tau_{\mathrm{SL_f}}$, sublevel set persistence of the time domain $\tau_{\mathrm{SL_t}}$, the minima of SW1PerS score $\tau_{\mathrm{PH_s}}$, the maxima of the maximum persistence $\tau_{\mathrm{PH}_{\mathcal{L}}}$, and mutual information τ_{MI}. The mean and standard deviation of the 30 trials at each SNR were calulcated for each method as shown in Fig. 2.27.

Figure 2.27 shows that the sublevel set persistence methods fail to provide an accurate delay τ in comparison to the expert suggested delay $\tau_{\mathrm{exp.}} = 9$ when SNR < 10 dB. While this does show a limit

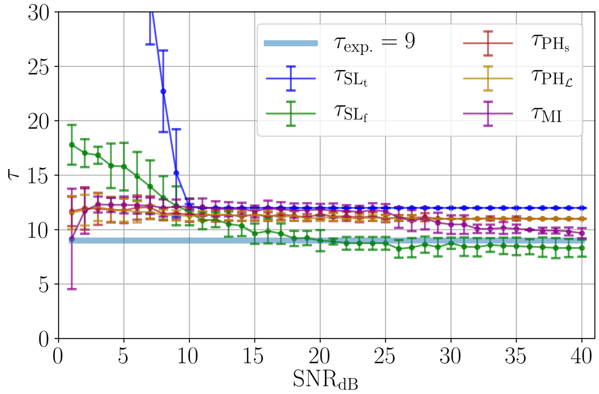


Figure 2.27: Noise robustness analysis of the delay parameter selection using the Rossler system with incriminating additive noise. The mean and standard deviation as error bars of the delay parameters from 30 trials at each SNR were calculated using sublevel set persistence of the frequency domain $\tau_{\text{SL}_\text{f}}$, sublevel set persistence of the time domain $\tau_{\text{SL}_\text{t}}$, the minima of SW1PerS score $\tau_{\text{PH}_\text{s}}$, the maxima of the maximum persistence $\tau_{\text{PH}_\mathcal{L}}$, and mutual information τ_{MI}.

for the sublevel set persistence methods, SNR values below 10 dB are uncommon since this level of noise contamination is not considered acceptable for a signal with a rule-of-thumb requirement of SNR > 15 dB. However, the 1-D Persistent homology methods and mutual information provide accurate delay parameter selection down to an SNR of 2 dB.

Robustness to Signal Length A common issue with signal processing and time series analysis methods is their limited functionality with smaller sets of data available, which has been used to analyze the sentitivty of the delay parameter selection [63]. Here we will investigate the limitations of these methods in the face of short time series. We will do this analysis by incrementing the length of the time series with the PE parameters calculated at each increment. For our analysis we will again use the Rossler system. Specifically, we incremented the length of the signal from $L = 75$ to 1000 in steps of 25 (see Fig. 2.28. However, if this type of analysis is not available for the data set being analyzed, for time series analysis applications it is commonly suggested to have a data length of $L = 4000$ for continuous dynamical systems and and $L = 500$ for maps [250].

In Fig. 2.28 we see that all of the methods reach an accurate value of τ, in comparison to the expert suggested $\tau = 9$ when the time series is atleast 125 data points long. An important note to make is that this result is not general for all continuous dynamical systems. The required length

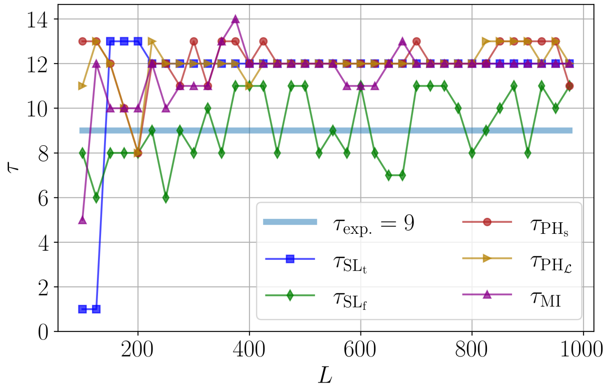


Figure 2.28: Signal length robustness analysis of the delay parameter selection using the Rossler system with incrementing signal length from 75 to 1000 in steps of 25. The delay parameters were calculated at each L using set persistence of the frequency domain $\tau_{\mathrm{SL_f}}$, sublevel set persistence of the time domain $\tau_{\mathrm{SL_t}}$, the minima of SW1PerS score $\tau_{\mathrm{PH_s}}$, the maxima of the maximum persistence $\tau_{\mathrm{PH}_\mathcal{L}}$, and mutual information τ_{MI}.

of the signal is going to vary depending on the sampling rate of the time series. To determine a general requirement for the methods we repeated this analysis method for all of the systems shown in Table A.2. Our result from this analysis found that, in general, $L \geq 15\tau$ for selecting an appropriate PE and state space reconstruction delay τ using the TDA-based methods described in this manuscript.

CHAPTER 3

PERSISTENT HOMOLOGY OF COMPLEX NETWORKS

This chapter of my research investigates methods for mapping time series data in discrete complex networks whose topology can be used to infer meaningful characteristics about the underlying dynamics of the system. The topology of these complex networks is measured using persistent homology.

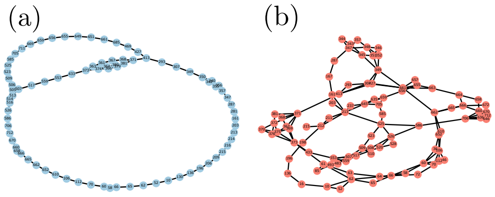


Figure 3.1: Comparison between ordinal partition networks generated from x-solution of Rössler system for both periodic (a) and chaotic (b) time series.

These networks have the potential to provide new insights into the systems driving the time series outputs. For instance, periodic time series tend to create transitional networks with overarching circular structure, while those arising from chaotic systems have a seemingly unorganized state transition entanglement (see, for example, the OPNs in 3.1). Further, networks can provide an efficient approach for approximating topological entropy of low-dimensional chaotic systems [205]. However, practitioners often only have access to standard network analysis tools to quantify the resulting outputs such as centrality measures or average path length, and these measures can only do so much to quantify the overarching structure of the graph. The power of combining network approaches to signal processing with TDA is that there is the potential for novel methods for encoding the overall structure of the network in a quantifiable, robust manner.

My work is the first to bring the tools of TDA to these networks. My work [162] provides a novel combination of persistent homology and network methods to yield a compressed, multi-scale representation of complex networks that can distinguish between dynamic states such as periodic and chaotic behavior. Applying a filtration of the simplicial complex enables us to track the changes in homology classes over the course of the filtration through a persistence diagram. The persistence diagram encodes information about the loop structures and corresponding periodicity of the signal. I then extract existing as well as new geometric and entropy based point summaries from the persistence diagram. I can also make direct comparisons between persistence diagrams using distance measures and multi-scale projections. In [162], I showed that persistence-based point summaries yield a clearer distinction, compared to traditional statistics, of the dynamic behavior for a variety of simulated dynamical systems and electrocardiogram and electroencephalogram data sets. Additionally, I showed that the persistence-based point summaries are more robust to noise than existing graph-based scores.

In section 3.1 I introduce the field of complex network representation of signals and the complex networks I use. Section 3.2 overviews how persistent homology is applied to the resulting networks including the various distances that can be used as well as summary statistics. Several examples are provided in Section 3.3 to demonstrate the procedure for forming the complex networks as well as the correct application of persistent homology per application. In section 3.4, I provide the results from analyzing the complex networks using persistent homology.

3.1 Complex Networks

Network representations of time series generally fall within three categories: proximity networks, visibility graphs, and transitional networks. These types of complex networks are discussed in the following paragraphs.

Proximity networks are formed from proximity conditions in the reconstructed state space. Examples include the k-Nearest Neighbors (k-NN) [118] and recurrence networks [68] (which are essentially the network underlying the Vietoris-Rips complex of the point cloud). For proximity networks, the graph representation includes all points in the state space reconstruction as part of the vertex set. When studying the shape of these networks with TDA based tools, careful consideration is needed in the selection of k or ϵ to generate a graph with the expected topology. Additionally, due to each point in the state space serving as a vertex, there are no speed gains in computing persistent homology in comparison to the original state space reconstruction since the size of the simplicial complex remains the same in both representations. While proximity networks encode the dynamics of the signal into its structure, they do not store temporal information.

Transitional networks partition a time series $\{x(t)\}$ such that it has a vertex set of states $\{s_i\}$ for each visited state and an edge for temporal transitions between states. The resulting transitional network constitutes a finite state space $K = \{s_i\}_{i \in \mathbb{N}}$, where K is compact and every map $\phi : K \longrightarrow K$ is continuous. One interpretation of a topological system on a finite state space is as a finite graph where the edges describe the action of ϕ, i.e., if there is a directed edge from vertex i to vertex j, then $\phi(i) = j$. Therefore, the transitional networks I obtain from a time series are topological systems, and they yield themselves to further analysis within the framework of topological dynamics. The two most common transitional networks for time series analysis are the ordinal partition network (OPN) [146] and the Coarse Grained State Space Network (CGSSN) [31, 237, 239]. Both of these transitional networks are formed by first reconstructing the state space through Takens' embedding as $\chi = \{\mathbf{X_i} = (x_i, x_{i+\tau}, \ldots, x_{i+(d-1)\tau})\} \subset \mathbb{R}^d$. The OPN is generated by defining states from the lexicographic order of the ordinal ranking of $\mathbf{X_i}$. This method of partitioning the state space results in the vertex set of states as the $d!$ possible permutations $\Pi = (\pi_1, \ldots \pi_{d!})$ representing the regions of $\mathbb{R}^d$ separated by hyperplanes; see the example across the top of 3.7. Similarly, using the same example signal, the CGSSN is shown along the bottom of 3.7. The CGSSN is formed by defining a set of states as d-orthotopes that partition the state space occupied by $\mathbf{X_i}$ in a data-driven manner. For the example shown in Fig. 3.7 I defined 8 equal sized cubes (3-orthotopes) that represent the possible states, where the temporal transitions between states are tracked to add edges in the corresponding network. Both of these examples demonstrate the periodic structure of the embedding being encoded into a cyclic network structure.

The visibility graph [5, 89, 120–123, 140, 141, 167, 242, 249], an idea taken from computational geometry [57], is defined by including a vertex for each data point, and including an edge between vertices if a line can be drawn between the two which does not pass below any other data point; see [168] for a review. The visibility graph is closely related to the sublevelset persistence computed directly on the time series rather than on the Takens embedding. As my focus for this work is related to building upon the strong theory developed for the Takens embedding, I do not expect to utilize these constructions at this stage of the work. Additionally, visibility graphs, unfortunately, do not yield themselves well to be analyzed with persistent homology due to the lack of periodic cycle structure (e.g., loops) associated to regular dynamics. As such, I will not be investigating them.

3.1.1 Background

State Space Reconstruction Takens' theorem forms one of the theoretical foundations for the analysis of time series corresponding to nonlinear, deterministic dynamical systems [226] and is often used to form complex networks. It basically states that in general it is possible to obtain an embedding of the attractor of a deterministic dynamical system from one-dimensional measurements of the system's evolution in time. The embedding of the signal is commonly known as the State Space Reconstruction (SSR).

An embedding is a smooth map $\Psi : M \to N$ between the manifolds M and N that diffeomorphically maps M to N. Specifically, assume that the state of a system is described for any time $t \in \mathbb{R}$ by a point $\mathbf{x}$ on an m-dimensional manifold $M \subseteq \mathbb{R}^d$. The flow for this system is given by a map $\phi^t(\mathbf{x}) : M \times \mathbb{R} \to M$ which describes the evolution of the state $\mathbf{x}$ for any time t. In reality, I typically do not have access to $\mathbf{x}$, but rather have measurements of $\mathbf{x}$ via an observation function $\beta(\mathbf{x}) : M \to \mathbb{R}$. The observation function has a time evolution $\beta(\phi^t(\mathbf{x}))$, and in practice it is often a one-dimensional, discrete and equi-spaced time series of the form $\{\beta_n\}_{n \in \mathbb{N}}$.

Although the state $\mathbf{x}$ can lie in a higher dimension, the time series $\{\beta_n\}$ is one-dimensional. Nevertheless, Takens' theorem states that by fixing an embedding dimension $d \geq 2m+1$, where m is the dimension of a compact manifold M, and a time lag $\tau > 0$, then the map $\Phi_{\phi,\beta} : M \to \mathbb{R}^d$ given by

$$\begin{aligned}\Phi_{\phi,\beta} &= (\beta(\mathbf{x}), \beta(\phi(\mathbf{x})), \ldots, \beta(\phi^{d-1}(\mathbf{x})))\\ &= (\beta(\mathbf{x}_t), \beta(\mathbf{x}_{t+\tau}), \beta(\mathbf{x}_{t+2\tau}), \ldots, \beta(\mathbf{x}_{t+(d-1)\tau})),\end{aligned}$$

is an embedding of M, where ϕ^{d-1} is the composition of ϕ $d-1$ times and $\mathbf{x}_t$ is the value of $\mathbf{x}$ at time t.

Theoretically, any time lag τ can be used if the noise-free data is of infinite precision; however, in practice, the choice of τ is important in the delay reconstruction. The other component in Takens' embedding is the embedding dimension d, which must be large enough to unfold the attractor. If this dimension in not sufficient, then some points can falsely appear to be neighbors at a smaller dimension due to the projection of the attractor onto a lower dimension. The appropriate method for selecting both of these parameters is thoroughly described in Chapter 2.

3.1.2 Graphs

A graph $G = (V, E)$ is a collection of vertices V with edges $E = \{uv\} \subseteq V \times V$. In this paper, I assume all graphs are simple (no loops or multiedges) and undirected. The complete graph on the vertex set V is the graph with all edges included, i.e. $E = \{uv \mid u \neq v \in V\}$.

I will reference a few special graphs. The cycle graph on n vertices is the graph $G = (V, E)$ with $V = \{v_1, \cdots, v_n\}$, and $E = \{v_i v_{i+1} \mid 1 \leq i < n\} \cup \{v_n v_1\}$; i.e. it forms a closed path (cycle) where no repetition occurs except for the starting and ending vertex. The complete graph on n vertices is the graph $G = (V, E)$ with $V = \{v_1, \cdots, v_n\}$, and $E = \{v_i v_j \mid i \neq j\}$. That is, it is the graph with n vertices and all possible edges are included.

I will also work with weighted graphs, $G = (V, E, \omega)$ where $\omega : E \longrightarrow \mathbb{R}$ gives a weight for each edge in the graph. In this paper, I assume all weights are non-negative, $\omega : E \longrightarrow \mathbb{R}_{\geq 0}$. Given an ordering of the vertices $V = \{v_1, \cdots, v_n\}$, a graph can be stored in an adjacency matrix $\mathbf{A}$ where entry $\mathbf{A}_{ij} = 1$ if there is an edge $v_i v_j \in E$ and 0 otherwise. This can be edited to store the weighting information by setting $\mathbf{A}_{ij} = \omega(v_i v_j)$ if $v_i v_j \in E$ and 0 otherwise.

A path γ in a graph is an ordered collection of non-repeated vertices $\gamma = u_0 u_1 \cdots u_k$ where $u_i u_{i+1} \in E$ for every i. The length of the path is the number of edges used, namely $\mathrm{len}(\gamma) = k$ in the above notation. The distance between two vertices u and v is the minimum length of all paths from u to v and is denoted $d(u, v)$. Given an ordering of the vertices, this information can be stored in a distance matrix $\mathbf{D}$ where $\mathbf{D}_{ij} = d(v_i, v_j)$. Thus an unweighted graph $G = (V, E)$ gives rise to a weighted complete graph on the vertex set V by setting the weight $\omega(uv) = d(u, v)$.

3.1.3 Proximity and Transition Networks

Proximity Network: k-Nearest Neighbor Graph Given a collection of points in $\mathbb{R}^d$, the k-nearest neighbor graph, or k-NN, is a commonly used method to build a graph. Fix $k \in \mathbb{Z}_{\geq 0}$. The (undirected) k-NN graph has a vertex set in 1-1 correspondence with the point cloud, so I abuse notation and write v_i for both the point $v_i \in \mathbb{R}^d$, and for the vertex $v_i \in V$. An edge v_iv_j is included if v_i is among the kth nearest neighbors of v_j. When required, I can give a weighting for this graph by setting $\omega(v_iv_j) = \|v_i - v_j\|$.

Transition Networks: Ordinal Partition and Coarse Grained State Space Networks For a graph $G = (V, E)$ given an ordering of the vertices $V = \{v_1, \cdots, v_n\}$, the graph can be stored in an adjacency matrix $\mathbf{A}$ where the weighting information is stored by setting $\mathbf{A}_{ij} = w_{(v_i,v_j)}$ if $v_iv_j \in E$ and 0 otherwise.

Transitional networks are generated from a graph formation technique for time series data. They are formed through a chronologically ordered sequence of symbols or states. For time series analysis, these states are mapped from the measurement signal. Specifically, I first use a state space reconstruction and then assign a symbolic representation for each vector in the SSR. Our definition of the state space reconstruction is slightly different for discretely sampled time series data $x = [x_1, x_2, \ldots, x_L]$ with L as the number of samples from the signal assuming the signal was sampled at uniform time stamps $t = [t_1, t_2, \ldots, t_L]$ with sampling frequency f_s. An SSR vector of a discrete sampled signal is defined as

$$X_i = [x_i, x_{i+\tau}, x_{i+2\tau}, \ldots, x_{i+\tau(n-1)}] \tag{3.1}$$

with $i \in \mathbb{Z} \cap [1, L - \tau(n-1)]$, $\tau \in \mathbb{Z}$.

To form a symbolic sequence from the time series data we implement a function to map the SSR to a set of symbols or an alphabet $\mathcal{A}$ of possible symbols as $f : v_i \to s_j$, where $s_j \in \mathcal{A}$ is a symbol from the alphabet. In this work we consider the symbols from the alphabet as integers such that $s_i \in \mathcal{A} = \mathbb{Z} \cap [1, N]$, where N is the number of possible symbols. Applying this mapping over all embedding vectors we get a symbol sequence as $S = [s_1, s_2, \ldots, s_{L-\tau(n-1)}]$.

The symbol sequence S forms a transitional network by considering a graph $G = (V, E)$. We represent the graph using the adjacency matrix $\mathbf{A}$ data structure of size $N \times N$. We add edges to the graph via the symbolic transitions with an edge between row s_i and column s_j when there is a transition from s_i to s_j. This is represented in the adjacency matrix structure by incrementing the value of $\mathbf{A}_{s_i,s_j}$ by one for each transition between s_i and s_j, where $\mathbf{A}$ begins as a zero matrix. We set the total number of transitions between two nodes s_i and s_j as the edge weight $w_{(s_i,s_j)}$. To better illustrate this process take the example of a simple cycle shown in Fig. 3.2. In this example we take the symbol or state sequence S on the left side of Fig. 3.2 with symbols in the alphabet $\mathcal{A} = [1, 2, 3, 4]$ and create a network in the middle of Fig. 3.2. This network is represented as a directed and weighted adjacency matrix as shown on the right side of Fig. 3.2. With an understanding of transitional networks and their formation. I next introduce two commonly used method for assigning symbolic representations to the SSR vectors.

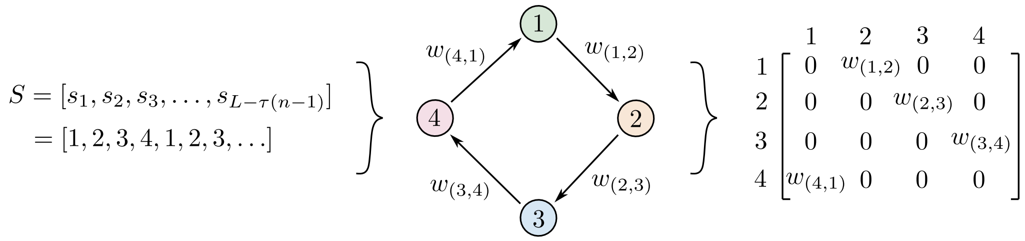

$S = [s_1, s_2, s_3, \ldots, s_{L-\tau(n-1)}]$
$= [1, 2, 3, 4, 1, 2, 3, \ldots]$
1
2
3
4
$w_{(4,1)}$
$w_{(1,2)}$
$w_{(2,3)}$
$w_{(3,4)}$
$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & w_{(1,2)} & 0 & 0 \\ 2 & 0 & 0 & w_{(2,3)} & 0 \\ 3 & 0 & 0 & 0 & w_{(3,4)} \\ 4 & w_{(4,1)} & 0 & 0 & 0 \end{array}$

Figure 3.2: Example formation of a weighted transitional network as a graph (middle figure) and adjacency matrix (right figure) given a state sequence S (left figure).

The ordinal partition network [146,216] provides a relatively simple method to assign symbolic representations for the SSR vectors to form a transition network. This construction arose as a generalization of the concept of permutation entropy [14]. The basic idea of the OPN construction is to replace each SSR vector X_i with a permutation π where the vector X_i is assigned to a permutation based on the sorted order of its coordinates. Specifically, the permutation π is the one in the set of $n!$ possible permutations for which $x(t+\pi(0)\tau) \leq x(t+\pi(1)\tau) \leq \cdots \leq x(t+\pi(n-1)\tau)$, where $\pi(i)$ is the permutation value at index i; see the top row (OP) of Fig. 3.3 for an example. Then the OPN is built with a vertex set of encountered permutations in the sequence S with an edge included if the ordered point cloud passes from one permutation to the other.

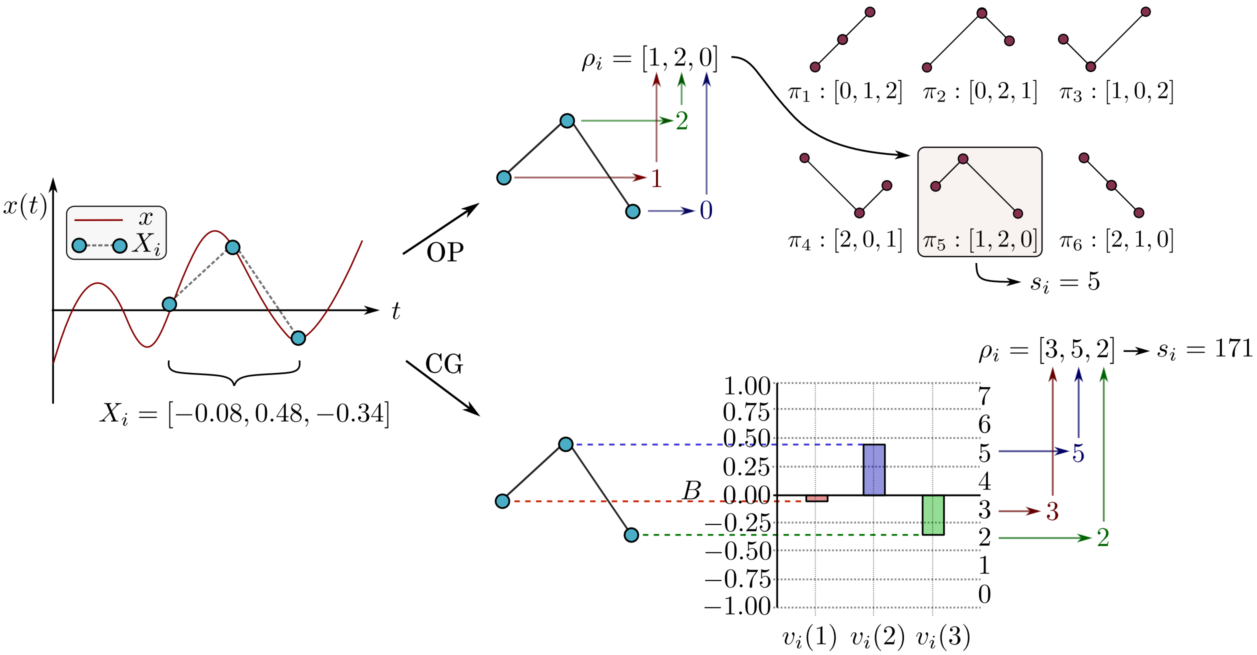


Figure 3.3: Assignment of Ordinal Partition (OP) or Coarse Grained (CG) state for example dimension 3 SSR vector.

The coarse grained state space network is created by partitioning the space occupied by the SSR into discrete n-dimensional hypercubes. This is done by first digitizing the SSR vectors using a digitization function $\psi(\chi_i, B)$, where $B = [B(1), B(2), \ldots, B(b-1)]$ is the monotonically increasing discrete binning of the vector's coordinates into b bins. We do this using an equal sized binning method. Specifically, the binning B is a vector of bin edges needs to encapsulate the entire range of signal values such that $\max(x) \le B(b)$ and $\min(x) \ge B(1)$. Let us assume our binning scheme has a total of b bins such that our digitized χ_i is defined as

$$p_i = \psi(\delta_i, B) = [p_i(1), p_i(2), \ldots, p_i(n-1)], \tag{3.2}$$

where $p_i(j)$ is the bin index that $\chi_i(j)$ is bounded by with $B(p_i(j)) < \chi_i(j) \le B(p_i(j)+1)$. We now have our digitized SSR vectors p_i which can be assigned a unique symbolic representation as

$$s_i = \sum_{j=1}^{n-1} (\chi_i(j) - 1) b^{n-1-j}, \tag{3.3}$$

where $s_i \in [0, b^n - 1]$ for a total of b^n possible states. This symbolic assignment is computationally efficient since it does not require a comparison to a bank of possible states as is required with ordinal partition networks. An example assignment is shown in the bottom CG row of Fig. 3.3 with $b = 8$ and $n = 3$.

3.2 Topological Analysis of Complex Networks

In order to analyze the shape of the constructed graphs, we turn to a generalization of the graph known as a *simplicial complex*, and a measurement tool known as *persistent homology*. We direct the interested reader looking for a more in depth discussion to [65,92,158,170]. I will first introduce persistent homology in how it applies to a distance matrix, which is easily extended to graphs. I will follow this introduction to some methods for generating a distance matrix from a graph.

3.2.1 Persistent Homology of Complex Networks

Simplicial complexes A simplicial complex can be thought of as a generalization of the concept of a graph to higher dimensions. Given a vertex set V, a simplex $\sigma \subseteq V$ is simply a collection of vertices. The dimension of a simplex σ is $\dim(\sigma) = |\sigma| - 1$. The simplex σ is a face of τ, denoted $\sigma \leq \tau$ if $\sigma \subseteq \tau$. A simplicial complex K is a collection of simplices $\sigma \subseteq V$ such that if $\sigma \in K$ and $\tau \leq \sigma$, then $\tau \in K$. Equivalently stated, K is a collection of simplices which is closed under the face relation. The dimension of a simplicial complex is the largest dimension of its simplices, $\dim(K) = \max_{\sigma \in K} \dim(\sigma)$. The d-skeleton of a simplicial complex is all simplices of K with dimension at most d, $K^{(d)} = \{\sigma \in K \mid \dim(\sigma) \leq d\}$.

Given a graph $G = (V, E)$, I can construct the clique complex

$$K(G) = \{\sigma \subseteq V \mid uv \in E \text{ for all } u \neq v \in \sigma\}.$$

This is sometimes called the flag complex. The clique complex of the complete graph on n vertices is called the complete simplicial complex on n vertices.

A filtration is a collection of nested simplicial complexes

$$K_1 \subseteq K_2 \subseteq \cdots \subseteq K_N.$$

See the bottom row of 3.4 for an example of a filtration. A weighted graph gives rise to a filtration I will make use of extensively. Given a weighted graph $G = (V, E, \omega)$ and $a \in \mathbb{R}$, I set

$$K_a = \{\sigma \in K(G) \mid \omega(uv) \leq a \text{ for all } u \neq v \in \sigma\}.$$

Since $K_a \subseteq K_b$ for $a \leq b$, this can be viewed as a filtration

$$K_{a_1} \subseteq K_{a_2} \subseteq \cdots \subseteq K_{a_N}$$

for any collection $a_1 \leq a_2 \leq \cdots \leq a_N$.

In particular, for this paper, I will build a filtration from an unweighted graph G by the following procedure. First, construct the pairwise distance matrix for the vertices of G using shortest paths. This can be viewed as a weighting on the complete graph with the same vertex set as G. Thus, it induces a filtration on the complete simplicial complex K where the 1-skeleton of K_a includes edges between any pair of vertices u and v for which $d(u, v) \leq a$. See 3.4 for an example.

Homology Traditional homology [92, 158] counts the number of structures of a particular dimension in a given topological space, which in our context will be a simplicial complex. In this context, the structures measured can be connected components (0-dimensional structure), loops (1-dimensional structure), voids (2-dimensional structure), and higher dimensional analogues as needed.

For the purposes of this paper, I will only ever need 0- and 1-dimensional persistent homology so I provide the background necessary in these contexts. Further, as a note for the expert, I always assume homology with $\mathbb{Z}_2$ coefficients which removes the need to be careful about orientation.

I start by describing homology. Assume I are given a simplicial complex K. Say the d-dimensional simplices in K are denoted $\sigma_1, \cdots, \sigma_\ell$. A d-dimensional chain is a formal sum of the

d-dimensional simplices $\alpha = \sum_{i=1}^{\ell} a_i \sigma_i$. I assume the coefficients $a_i \in \mathbb{Z}_2 = \{0,1\}$ and addition is performed mod 2. For two chains $\alpha = \sum_{i=1}^{\ell} a_i \sigma_i$ and $\beta = \sum_{i=1}^{\ell} b_i \sigma_i$, $\alpha + \beta = \sum_{i=1}^{\ell} (a_i + b_i)\sigma_i$. The collection of all d-dimensional chains forms a vector space denoted $C_d(K)$. The boundary of a given d-simplex is

$$\partial_d(\sigma) = \sum_{\tau < \sigma, \dim(\tau) = d-1} \tau.$$

That is, it is the formal sum of the simplices of exactly one lower dimension. If $\dim(\sigma) = 0$, that is, if σ is a vertex, then I set $\partial_d(\sigma) = 0$. The boundary operator $\partial_d : C_d(K) \to C_{d-1}(K)$ is given by

$$\partial_d(\alpha) = \partial_d \left(\sum_{i=1}^{\ell} a_i \sigma_i \right) = \sum a_i \partial_d(\sigma_i).$$

A d-chain $\alpha \in C_d(K)$ is a cycle if $\partial_d(\alpha) = 0$; it is a boundary if there is a $d+1$-chain β such that $\partial_{d+1}(\beta) = \alpha$. The group of d-dimensional cycles is denoted $Z_d(K)$; the boundaries are denoted $B_d(K)$.

In particular, any 0-chain is a 0-cycle since $\partial_0(\alpha) = 0$ for any α. A 1-chain is a 1-cycle iff the 1-simplices (i.e., edges) with a coefficient of 1 form a closed loop. It is a fundamental exercise in homology to see that $\partial_d \partial_{d+1} = 0$ and therefore that $B_d(K) \subseteq Z_d(K)$. The d-dimensional homology group is $H_d(K) = Z_d(K)/B_d(K)$. An element of $H_d(K)$ is called a homology class and is denoted $[\alpha]$ for $\alpha \in Z_d(K)$ where $[\alpha] = \{\alpha + \partial(\beta) \mid \beta \in C_{d+1}(K)\}$. I say that the class is represented by α, but note that any element of $[\alpha]$ can be used as a representative so this choice is by no means unique.

In the particular case of 0-dimensional homology, there is a unique class in $H_0(K)$ for each connected component of K. For 1-dimensional homology, I have one homology class for each "hole" in the complex.

Persistent homology We next look to a more modern viewpoint of homology which is particularly useful for data analysis, persistent homology. In this case, we study a *changing* simplicial complex and encode this information via the *changing* homology. In explaining persistence, we will follow the example of Fig. 3.4 for the setting used in this work where the input data is a weighted network.

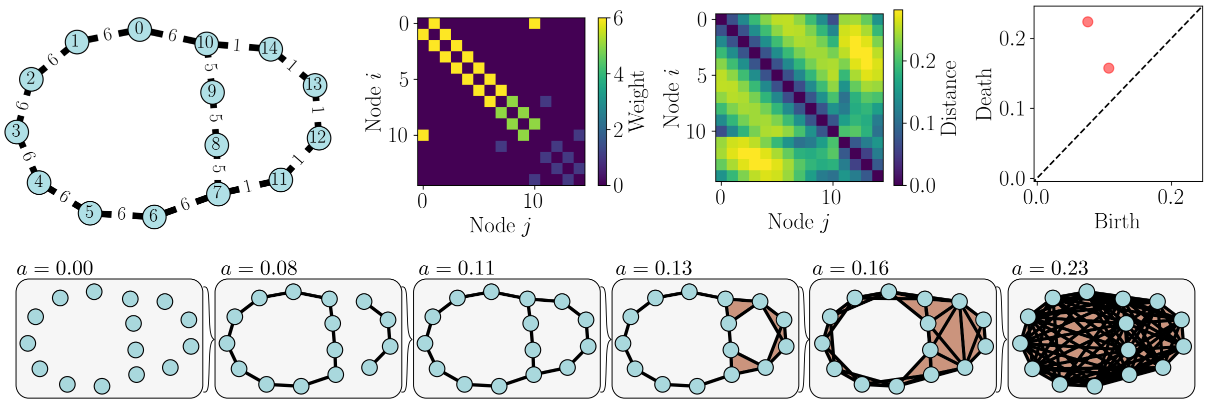


Figure 3.4: Persistent homology of weighted complex network. Top left shows the weighted network with corresponding adjacency matrix to its right. Third is the distance matrix and then at the top right is the persistence diagram of one-dimensional features. The bottom row shows the filtration at critical values.

A filtration of a simplicial complex K is a collection of nested simplicial complexes

$$K_1 \subseteq K_2 \subseteq \cdots \subseteq K_N = K.$$

See the bottom row of Fig. 3.4 for an example of a filtration. In this work, we will be focused on the following filtration which arises from finite metric space; in our case, this is given as a pairwise distance matrix $\mathbf{D} \in \mathbb{R}^{n\times n}_{\geq 0}$, obtained from a weighted graph as described in Sec. 3.2.2. Set the vertex set to be $V = [1, \cdots, n]$ and for a fixed $a \in \mathbb{R}$, let

$$K_a = \{\sigma \subset V \mid \mathbf{D}(u, v) \leq a \text{ for all } u \neq v \in \sigma\}.$$

This can be thought of as the clique complex on the graph with edges given by all pairs of vertices with distance at most a. Further, since $K_a \subseteq K_b$ for $a \leq b$, this construction gives rise to a filtration

$$K_{a_1} \subseteq K_{a_2} \subseteq \cdots \subseteq K_{a_N}$$

for any collection $a_1 \leq a_2 \leq \cdots \leq a_N$.

Fix a dimension d. For any inclusion of one simplicial complex to another $L \hookrightarrow K$, there is an induced map on the d-chains $\iota : C_d(L) \to C_d(K)$ by simply viewing any chain in the small complex as one in the larger. Less obviously, this extends to a map on homology $\iota_* : H_d(L) \to H_d(K)$ by sending $[\alpha] \in H_d(L)$ to the class in $H_d(K)$ with the same representative. That this is well defined is a non-trivial exercise in the definitions [92]. Putting this together, given a filtration

$$K_{a_1} \subseteq K_{a_2} \subseteq \cdots \subseteq K_{a_N}$$

there is a sequence of linear transformations on the homology

$$H_d(K_{a_1}) \to H_d(K_{a_2}) \to \cdots \to H_d(K_{a_N}).$$

A class $[\alpha] \in H_d(K_{a_i})$ is said to be born at a_i if it is not in the image of the map $H_d(K_{a_{i-1}}) \to H_d(K_{a_i})$. The same class dies at a_j if $[\alpha] \neq 0$ in $H_d(K_{a_{j-1}})$ but $[\alpha] = 0$ in $H_d(K_{a_j})$. In the case of 0-dimensional persistence, this feature is encoding the appearance of a new connected component at K_{a_i} that was not there previously, and which merges with an older component entering K_{a_j}. For 1-dimensional homology, this is the appearance of a loop structure that likewise fills in entering K_{a_j}.

The persistence diagram encodes this information as follows. For each class that is born at a_i and dies at a_j, the persistence diagram has a point in $\mathbb{R}^2$ at (a_i, a_j). Because several features can appear and disappear at the same times, we allow for repeated points at the same location. For this reason, a persistence diagram is often denoted as a multiset of its off-diagonal points, $D = \{(b_1, d_1), \cdots, (b_k, d_k)\}$. See the top right of Fig. 3.4 for an example. Note that the farther a point is from the diagonal, the longer that class persisted in the filtration, which signifies large scale structure. The *lifetime* or *persistence* of a point $x = (b, d)$ in the diagram in a persistence diagram D is given by $\text{pers}(x) = |b - d|$. It is often of interest to investigate only a specific subset of d dimensional features from a persistence diagram, which we represent as D_d.

3.2.2 Distance Measures for Graphs

We next look at four different ways to define a distance between pairs of vertices given an input (weighted) graph. In each case, we generate a distance matrix $\mathbf{D}$ where entry $\mathbf{D}(a, b)$ gives the associated distance between vertices a and b.

Shortest Unweighted Path Distance The first method, the shortest unweighted path distance, ignores the weighting information entirely, using only the number of edges to get from vertex a to vertex b. Specifically, $\mathbf{D}(a, b)$ is the number of steps it takes to transition from a to b through the shortest path. See the example of Fig. 3.5. The shortest path distance is calculated using the `NetworkX` implementation of Dijkstra's algorithm [66] with the unweighted adjacency matrix.

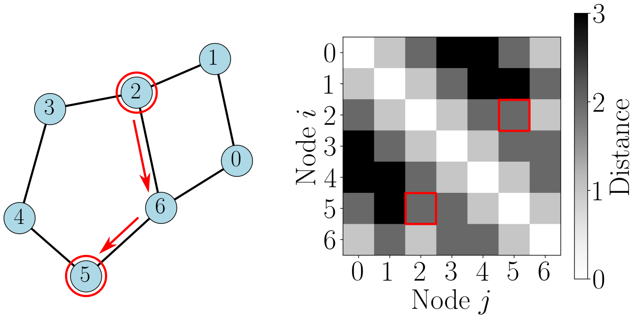


Figure 3.5: Example basic graph with corresponding shortest path distance matrix. Highlighted in red is an example shortest path from node 2 to 5 with shortest path distance 2.

Shortest Weighted Path Distance The second method, the shortest weighted path, similarly only uses the number of edges between vertex a and b as the path distance. However, the weighted information is incorporated through the choice of the path. This is done by choosing the path with the lowest summed weight of all paths between a and b. To make it such that the path with the largest weights is used, the inverse of the edge weights is used when calculating the shortest path. Again, this distance is calculated using the `NetworkX` implementation of Dijkstra's algorithm [66] but with the inverse of the weighted adjacency matrix.

Weighted Shortest Path The third method, the weighted shortest path is very similar to the second method. The only variation is that the sum of the edge weights along the path is used as the distance. The path used is found using the inverse of the edge weights similar to the second method.

The fourth method for computing distances is the diffusion distance; for more details we direct the reader to [50]. This is computed using the transition probability distribution matrix $\mathbf{P}$ of the graph, where $\mathbf{P}(a, b)$ is the probability of transitioning to vertex b in the next step given you are currently at a. Given the weighted, undirected adjacency matrix $\mathbf{A}$, the transitional probability matrix is calculated as

$$\mathbf{P}(i,j) = \frac{\mathbf{A}(i,j)}{\sum_{k=1}^{|V|} \mathbf{A}(i,k)}.$$

This formulation of the probability matrix only has transition probabilities greater then zero for one step neighbors of i. However, the transition probabilities for non-adjacent neighbors of node i can be calculated using the random walk and the diffusion process. A random walk is the sequences of nodes visited $(a_1, a_2, \ldots)$ in t steps, where the selection of the next node is based on the transition probabilities. It is a classic exercise to show that, given $\mathbf{P}$, the probability distribution for transitioning to vertex b from vertex a in t random walk steps is $\mathbf{P}^t(a,b)$.

Diffusion Distance The diffusion distance is a measure of the degree of connectivity of two nodes in a connected graph after t steps using the lazy transition probability $\tilde{\mathbf{P}}^t$ based on the possible random walks of length t and is calculated as

$$d_t(a,b) = \sqrt{\sum_{c \in V} \frac{1}{\mathbf{d}(c)} \left[\tilde{\mathbf{P}}^t(a,c) - \tilde{\mathbf{P}}^t(b,c)\right]^2} \tag{3.4}$$

where $\mathbf{d}$ is the degree vector of the graph with $\mathbf{d}(i)$ as the degree of node i and $\tilde{\mathbf{P}}$ is the lazy transition probability matrix, where the initial zero diagonal of P is set such that $\tilde{\mathbf{P}} = 1/2(\mathbf{I} + \mathbf{P})$. In other words, there is an equal probability of staying and leaving at node i in a single step. Applying the diffusion distance to all node pairs results in the distance matrix $\mathbf{D}_t$.

Consider the diffusion distance with two nodes having a connected path with high transition probability edges or many random walk paths connecting the two, then the diffusion distance between them will be low. However, if two vertices are only connected through a single, low probability edge transition from a possible perturbation in the graph, then their diffusion distance will be large. A common example implementing the diffusion distance is based on assigning $\mathbf{P}$ as a function of the proximity of nodes. Using this formulation of the transition probability, it is possible to cluster the data based on the distances as demonstrated in [50]. However, due to the natural transitions that occur in transitional complex networks, the diffusion distance is a natural solution for incorporating edge weight data into the distance measurement.

It is important to mention the sensitivity of the diffusion distance $\mathbf{D}_t$ to the selection of the number of walk steps t. We used an empirical study of 23 continuous dynamical systems to determine the optimal t such that a periodic signal creates a significant point in the persistence diagram representing the cycle. More details on this analysis are available in the appendix in Section D.2. We found an optimal value of $d < t < 3d$, where d is the diameter of the graph. Specifically, the diameter is measured as the maximum shortest unweighted path between any two vertices. Intuitively, this value of t seems suitable since it allows for a transition probability between all nodes in the graph. I.e., if $t \geq d$ then there is a probability of transitioning between every node pair in a random walk of length t.

3.2.3 Point summaries of persistence diagrams

A common issue with persistence diagrams is that they are notoriously difficult to work with as a summary of data. While they are quantitative in nature, determining differences in structure such as "has a point far from the diagonal" is often a qualitative procedure. Metrics for persistence diagrams exist, namely the bottleneck and p-Wasserstein[1] distances, however these objects are not particularly easy to work with in a statistical or machine learning context. Thus, I will pass to working with the simplest of featurizations, namely point summaries of a given diagram, which I also call *scores*.

Maximum persistence The first very simple but extremely useful point summary is maximum persistence. Given a persistence diagram D, the maximum persistence is simply

$$\mathrm{maxpers}(D) = \max_{x \in D} \mathrm{pers}(x).$$

While this is obviously a very lossy point summary for a persistence diagram, it is quite useful in that, particularly for applications where the existence of a large circle is of interest, it often does what I need. See, e.g., [112, 232].

Periodicity Score

[1] This metric is closely related to but not the same as the eponymous metric from probability theory.

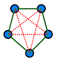
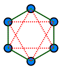
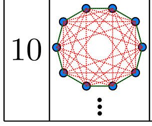
n	Cycle	r_B	r_D	L_n
3		1	1	0
4		1	2	1
5		1	2	1
6		1	2	1
	⋮			
10		1	4	3
	⋮			

Figure 3.6: Table of examples showing the lifetime L_n of the single class (r_B, r_D) in the persistence diagram for the pipeline applied to a cycle with n nodes.

Next, I set out to build a point summary which I can use to measure the similarity of our weighted graph to a cycle graph which is independent of the number of nodes. If G' is an unweighted cycle graph with n vertices, then following the procedure of Fig. 3.4 using the shortest path metric, I have that there is exactly one cycle which is born at 1, and fills in at $\lceil \frac{n}{3} \rceil$. See the examples of 3.6. This means the persistence diagram D' has exactly one point $(1, \lceil \frac{n}{3} \rceil)$, and so I denote the maximum persistence of this diagram as

$$L_n = \text{maxpers}(D') = \left\lceil \frac{n}{3} \right\rceil - 1.$$

Then, assume I are given another unweighted graph G with $|V| = n$ and persistence diagram D. I define the network periodicity score

$$P(D) = 1 - \frac{\text{maxpers}(D)}{L_n}. \tag{3.5}$$

This score is an extension of the periodicity score in [177] to unweighted networks, and it has the property that $P(D) \in [0, 1]$, with $P(D) = 0$ iff the input graph G is a cycle graph.

The ratio of the number of homology classes to the graph order The next point summary I define is

$$M(D) = \frac{|D|}{|V|}, \tag{3.6}$$

which is the reciprocal of the ratio between the number of vertices in the network $|V|$, i.e., the order of the graph, and the number of classes in the persistence diagram $|D|$.

I can think of this number as an approximation of the reciprocal of the number of vertices in each class, however, this is only an approximation because some classes in 1-D persistence diagram may share vertices in the network. Note that for a network with n nodes, the 0-dimensional persistence diagram will always have $n - 1$ points, and so this metric is not particularly useful. In this paper, I only use this summary for 1-dimensional persistence diagrams.

The logic behind this heuristic is that for a periodic signal I would expect to see a small number of 1-D homology classes in comparison to a chaotic time series. Therefore, for two networks of similar order but with different dynamic behavior, i.e., one is chaotic and one is periodic, the ratio $M(D)$ for the periodic time series will be smaller than its chaotic counterpart.

Normalized Persistent Entropy Persistent entropy is a method for calculating the entropy from the lifetimes of the points in a persistence diagram, inspired by Shannon entropy. This summary function, first given by Chintakunta et al. [45], is defined as

$$E(D) = -\sum_{x \in D} \frac{\mathrm{pers}(x)}{\mathscr{L}(D)} \log_2 \left(\frac{\mathrm{pers}(x)}{\mathscr{L}(D)} \right), \tag{3.7}$$

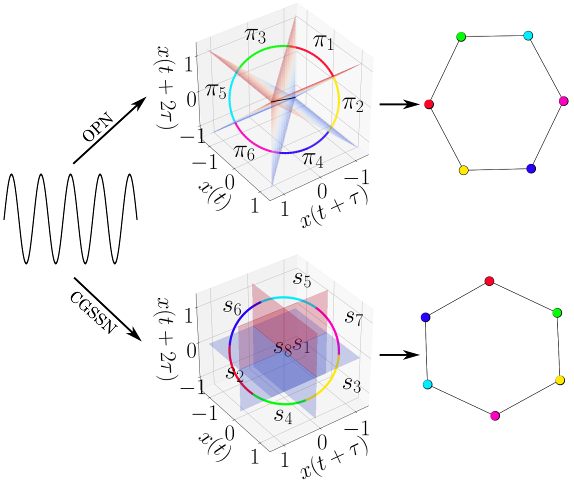
where $\mathscr{L}(D) = \sum_{x \in D} \mathrm{pers}(x)$ is the sum of lifetimes of points in the diagram. I cannot easily compare this value across different diagrams with different numbers of points. To deal with this issue, I provide the following normalization heuristic. Specifically, I normalize E as

$$E'(D) = \frac{E(D)}{\log_2 \left(\mathscr{L}(D) \right)}. \tag{3.8}$$

This normalization allows for an accurate measurement of the entropy even when there are few significant lifetimes.

3.3 Examples

This section overviews several examples applying transitional networks to time series data. Namely, I provide applications of both ordinal partition and coarse grained state space networks that highlight the limitations and benefits of each. Further, I show the benefits of incorporating weight information. Lastly, I show how these networks can capture the topology of the underlying state space of the time series. This is done for both synthetic and experimental data.



r[h]

Figure 3.7: Example formation of the ordinal partition (top) and coarse grained state space (bottom) networks for $x(t) = \sin(t)$ embedded into $\mathbb{R}^3$.

In this work we choose τ using the method of multi-scale permutation entropy as suggested in [160] since we are forming permutations to construct the OPN. While an appropriate embedding dimension n for the state space reconstruction may be sufficient, it may not be a high enough dimension to capture the complexity of the time series. To alleviate this issue, Bandt and Pompe [14] suggested using higher dimensions (e.g. $n \in [4, 10]$) to allow for $n!$ different states to better capture the complexity of the time series. In this work we will use a dimension $n = 6$ unless otherwise stated.

3.3.1 First Example: Ordinal Partition and Coarse Grained State Space Network Comparison

This first example compares the ordinal partition and coarse grained state space networks in terns of noise robustness.

Let us first start with a simple demonstrative example showing in Fig. 3.7 showing how the ordinal partition and coarse grained state space networks are related. The example is from embedding a simple sinusoidal function into dimension $n = 3$ creating a circle structure in the state space reconstruction.

Both network are created by covering the space occupied in the state space reconstruction. For ordinal partition networks, the set of all permutations of dimension n gives a cover of $\mathbb{R}^n$ with permutation π_i representing a subspace of $\mathbb{R}^n$ given by the intersection of $\binom{d}{2}$ inequalities. An example of these inequality planes and their intersections for a three-dimensional embedding is shown on the top OPN route of Fig. 3.7. Coarse grained state space networks create a cover using a set of n-dimensional hypercubes. These eight cubes are equal-sized for the example in the bottom of Fig. 3.7. Both network formation techniques capture the periodic structure of the state space reconstruction with resulting cycle graphs.

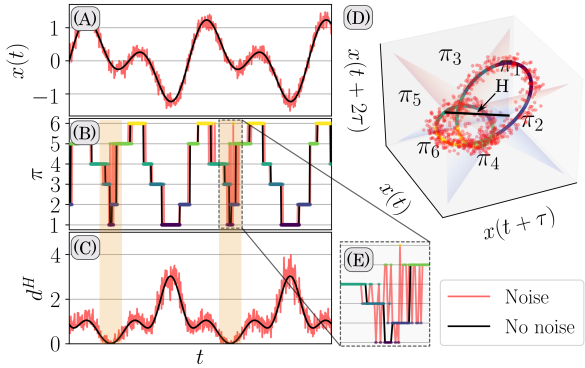


Figure 3.8: Example illustrating issue with erraneous permutation transitions when there is additive noise and a tracjectory close to the hyperplane intersection H. The three dimensional state space reconstruction (D) from the signal $x(t)$ with and without additive noise (A) demonstrate that as the distance to the hyperdiagonal d^H (C) becomes small, undesired permutation transitions (B)–with zoomed in section shown in (E)–occur as shown in the orange highlighted regions.

Robustness to Noise During my work with ordinal partition networks I discovered that they are not particularly resilient to noise. Indeed, one can think of the ordinal partition network as being the 1-skeleton of the nerve of a particular closed cover of the state space, delineated by the hyperplanes $x_i \leq x_j$. Consequently, when noise is injected into the system, there are superfluous transitions when nearing one of these boundaries. This effect becomes even more prominent near an intersection of multiple hyperplanes. For example, consider the signal and its embedding into $\mathbb{R}^3$ in Fig. 3.8.

As the distance to the hyperdiagonal d^H becomes small, there is a significant increase in seemingly superfluous transitions between permutations π (highlighted in orange in Fig. 3.8). This issue is even more exaggerated when the embedded signal is consistently close to the hyperdiagonal, which causes networks where no useful network topology can be extracted (e.g. see signal and far right OPN in Fig. 3.9). This issue can be partly alleviated by including the weight information as the most probable transition between permutations should still have the highest weight. However, these superflous transitions can become to severe when the state space reconstruction passes near the hyperdiagonal. For example, Fig. 3.9 shows the OPN and CGSSN for the signal with and without noise. This example clearly demonstrates that the CGSSN is the best choice for this signal with only very minor changes in its shape, while the OPN loses all resemblance of the noise free network. This loss in structure is due to the nature of the signals reconstructuction passing along the hyperdiagonl. While the OPN loses its structure, the ordinal partition network does not. This stability is due to their being no hyperdiagonal between states. At most there are only 8 possible states that intersection t a single point and no along an edge. This helps preserve the structure of the network when there is additive noise as it is not possible for the state to superfluously transition more than 8 states away if the amplitude of the noise is smaller than the edge length of the hypercube states.

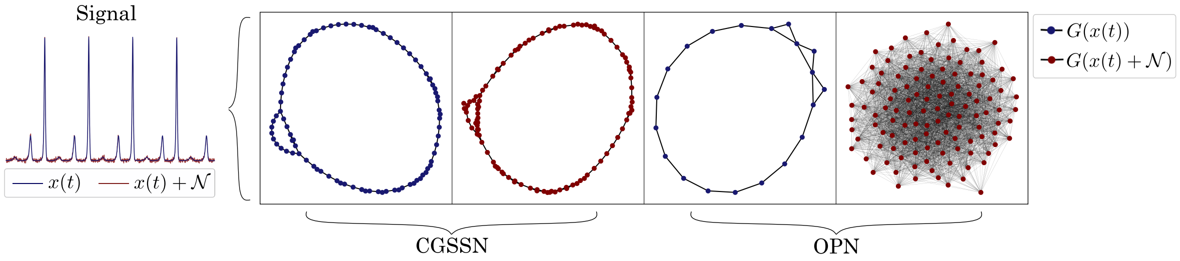


Figure 3.9: Example demonstrating importance of choosing an appropriate network formation method when there is additive noise in the signal. The CGSSN retains the graph structure when additive noise, but the OPN network quickly loses all resemblance of the noise free topological structure even with a small amount of additive noise. $x(t)$ is the signal, $\mathcal{N}$ is additive noise and $G(x)$ is the graph formation function of the signal x.

While this example highlighted a limitation of the OPN, the OPN does have benefits over the coarse grained state space network. Specifically, the ordinal partition network does not need to be adaptive to the amplitude of the data as does the CGSSN. Additionally, it has fewer parameters with only n and τ being selected with the CGSSN requiring an additional number of bins b parameter.

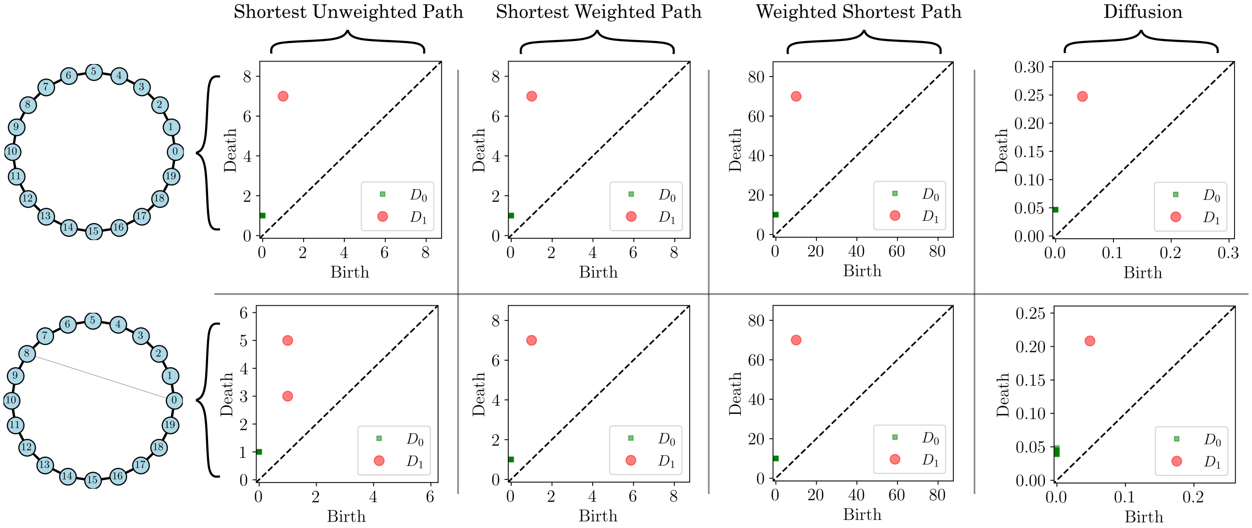


Figure 3.10: Two example weighted cycle graphs of weight 10 with the bottom row having an additional edge of weight one connecting nodes 0 and 8. The persistence diagram associated to each of the four distance methods are shown by column both both graphs.

3.3.2 Second Example: Distance Method Comparison

To compare my original [162] work done using the naive shortest unweighted path distance to the weight incorporating shortest path and diffusion distances, let us look at a simple example that highlights the issue previously mentioned with the unweighted shortest path not accounting for weight information. In Fig. 3.10 there are two graphs: on the top is a cycle graph with edge weights of 10 and on the bottom is the same cycle graph but with an additional single perturbation edge added between nodes 0 and 8 with a weight of 1. This edge could be caused by additive noise, a perturbation to the underlying dynamical system, or simply a falsely added state transition in the OPN formation procedure. If we implement the shortest unweighted path distance for calculating the persistent homology of the cycle graph we get a single significant point in the resulting persistence diagram as shown in the top left persistence diagram of Fig. 3.10. However, adding the single, low-weighted edge splits the graph with the persistence diagram using the shortest unweighted path distance having two significant points in the persistence diagram (see bottom left diagram of Fig. 3.10). This is due to the edge weight information being discarded when using the shortest path distance.

In comparison to the shortest unweighted path distance, the second, third, and fourth columns of

Fig. 3.10 show the persistence diagrams for both graphs using the shortest weighted path, weighted shortest path, and diffusion distances, respectively. For all three of these distance methods there is only a single one-dimensional point in the persistence diagrams for both graphs. Additionally, both the shortest weighted path and weighted shortest path have identical persistence diagrams for both graphs. This is due to the shortest weighted path between any two vertices never using the edge between vertices 0 and 8. For the diffusion distance we also only have a single point in the persistence diagram for one-dimensional features. This is caused by the weighted information being used in the diffusion distance calculation where the change in distance from the nodes 0 and 8 is not significantly changed from the addition of the perturbation edge connecting them since it has a low weight relative to the cycle and the transition probability distributions between vertices 0 and 8 are dissimilar. For calculating the diffusion distance in this example we used $t = 2d$ walk steps with d as the shortest path diameter of the graph.

This example demonstrates the importance of incorporating weight information when calculating the persistent homology of a complex network. The possibility of these low weight edges is evident as shown in Fig. 3.8 where there are noise associated edge state transitions when near state intersections.

3.3.3 Third Example: Periodic and Chaotic Dynamics

The third example qualitatively demonstrates that persistence of OPNs (similar results can be shown for CGSSNs) can detect the dynamic state of a signal as either periodic or chaotic. The example signal used here is from the Lorenz system defined as

$$\frac{dx}{dt} = \sigma(y - x),\ \frac{dy}{dt} = x(\rho - z) - y,\ \frac{dz}{dt} = xy - \beta z. \tag{3.9}$$

The system was simulated with a sampling rate of 100 Hz and system parameters $\sigma = 10.0$, $\beta = 8.0/3.0$, and $\rho = 180.1$ for a periodic response or $\rho = 181.0$ for a chaotic response. This system was solved for 100 seconds with only the last 20 seconds used to avoid transients.

Figure 3.11 shows the resulting Lorenz system simulation signals $x(t)$ for periodic (top row of

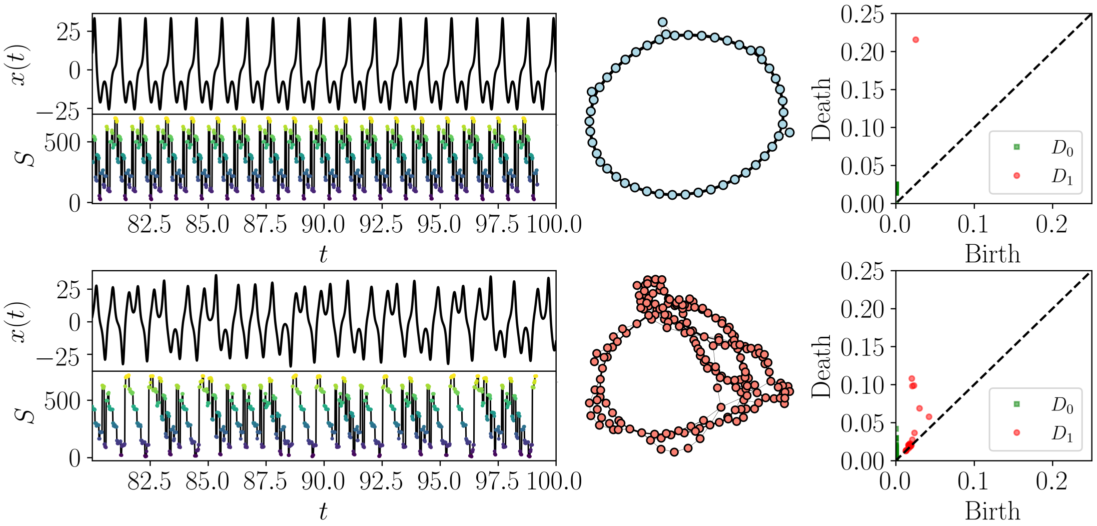


Figure 3.11: A comparison of the resulting persistence diagrams for an OPN formed from a periodic and chaotic signal from the Lorenz system.

figure) and chaotic (bottom row of figure) dynamics with the corresponding ordinal partition state sequence S using dimension $n = 6$ and $\tau = 17$ selected using multi-scale permutation entropy [160], OPN, and persistence diagram. For this example I used the diffusion distance with $t = 2d$ walk steps. This example result demonstrates that the persistence diagram for a periodic signals tend to have one or few significant points in the persistence diagram of one dimensional features D_1 representing the cyclic nature of the signal. On the other hand, the D_1 for chaotic signal has many significant points representing the entanglement of the OPN. The other distance methods also demonstrate similar behavior when comparing the resulting persistence diagrams from periodic and chaotic dynamics.

3.3.4 Fourth Example: The Magnetic Pendulum

To demonstrate the method applied to experimental data, I will be using a time series obtained from the angular position $\theta(t)$ of the magnetic pendulum experiment shown in Fig. 5.1 described in Section 5.1 with base excitation amplitude $A = 0.08$ m and frequency $\omega = 1.5$ Hz. This forcing amplitude results in the periodic time series shown in Fig. 3.12-(a). The resulting permutation sequence as well as the unweighted, undirected network are shown in Figs. 3.12-(b) and (c),

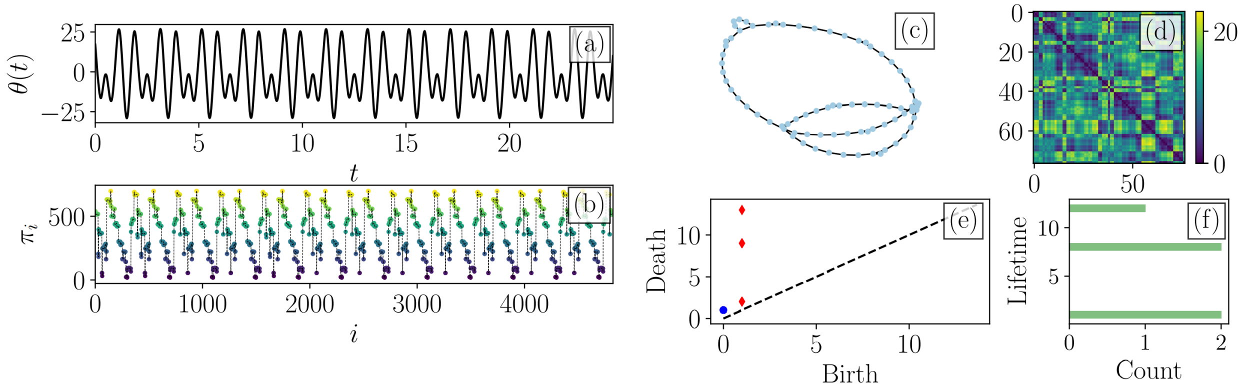


Figure 3.12: Example of method applied to experimental data with a periodic response Fig. (a). In Fig. (b) the sequence of permutations are shown for $n = 6$ with the associated ordinal partition network in Fig. (c). In Fig. (d) the distance matrix (using an unweighted network and short path distance) is shown, which was used to compute a persistence diagram with multiplicity shown in Fig. (e) and (f), respectively.

respectively. The network exhibits a rather simple structure with one large loop, two smaller loops, and two insignificantly small loops. The distance between nodes is shown through a shortest-path distance matrix (see Fig. 3.12-(d)). With the distance matrix known, the persistence diagram is obtained, which summarizes the loops as 1-D features with lifetimes of $[12, 8, 8, 1, 1]$. Additionally, a histogram is used to show the lifetime multiplicity, i.e., how many points are overlaid in each location of the persistence diagram. The periodicity score was calculated as $P(D) \approx 0.61$ and the persistent entropy was calculated as $E'(D) \approx 0.45$ using the lifetimes in Fig. 3.12-(f).

To make a fair comparison, the same process as shown in Fig. 3.12 is applied to a time series generated from a base excitation with $A = 0.085$ and frequency $\omega = 1.5Hz$, which results in a chaotic response. The resulting network from the permutation sequence is shown in Fig. 3.13-(a). It is clear that the network from the chaotic time series shows significantly more loops with, in general, smaller loop sizes. The size and quantity of these loops are shown in the persistence diagram of the network with the lifetimes (with multiplicity) shown in Fig. 3.13-(b) and (c), respectively. The periodicity score was calculated as $P(D) \approx 0.95$ and the persistent entropy was calculated as $E'(D) \approx 0.90$. This examples show how persistent homology of complex networks can be used to detect a change in complexity of the time series from experimental data.

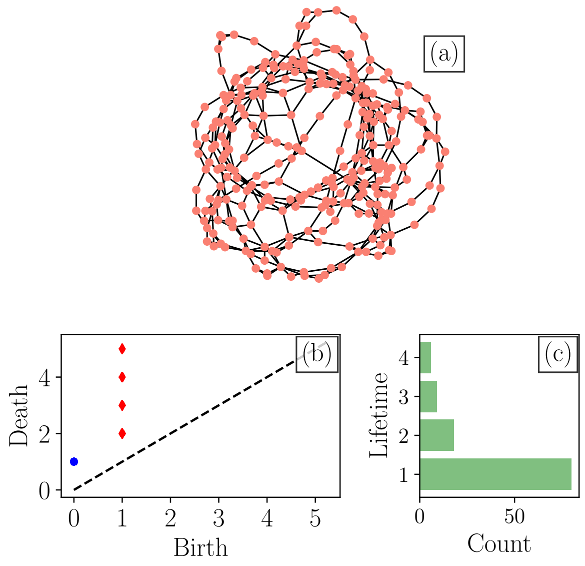


Figure 3.13: Example of method applied to experimental data with a chaotic response Fig. (a). In Fig. (b) the sequence of permutations are shown for $n = 6$ with the associated ordinal partition network in Fig. (c). In Fig. (d) the distance matrix (using an unweighted network and short path distance) is shown, which was used to compute a persistence diagram with multiplicity shown in Fig. (e) and (f), respectively.

3.4 Results

This section compares the persistence-based point summaries and the standard network scores, and illustrates the ability of these scores to detect dynamic state changes. Specifically, I compare the point summaries $M(D_1)$, $P(D_1)$, and $E'(D_1)$ to some commonly used network quantitative characteristics such as the mean out degree $\langle k \rangle$, the out degree variance σ^2, and the number of vertices N. These comparisons are shown in Section 3.4.1 for a family of trajectories from the Rössler system, while Section 3.4.1 tabulates the different scores for a variety of dynamical systems. In Section 3.4.2 I contrast the noise robustness of our approach to the standard network scores for ordinal partition networks.

3.4.1 Dynamic State Change Detection on the Rössler System

Letting the parameter a in the Rossler system vary in the range $0.37 < a < 0.43$ in steps of $\Delta a = 0.001$ and setting $\beta = 2$ and $\gamma = 4$, I obtain 1201 time series of length 1000 seconds for the state x. I only retain the last 400 seconds of the simulation to allow the trajectory to settle on an attractor. For the construction of the corresponding k-NN networks, I sample the time series at 2 Hz in order to capture a sufficient number of oscillations while avoiding overly large point clouds for computing persistence. For the Takens' embedding I use the mutual information function approach and the nearest neighbor method, respectively, to choose the parameters $\tau = 4$ and $d = 7$.

For constructing the ordinal partition networks use the higher sampling frequency of 20 Hz, and I use MPE to select $\tau = 40$ and $d = 6$. I found that a higher sampling rate for ordinal partition networks and the resulting longer time series is not an issue due to the maximum number of vertices not being dependent on the length of the time series, but rather on the motif dimension d and time series complexity. Furthermore, a higher sampling rate tends to improve the detection of periodic and chaotic time series for ordinal partition networks.

The resulting point summaries were found for both ordinal partition networks (left column plots of Fig. 3.14) and k-NN of Takens' embedding networks (right column plots of Fig. 3.14). The top two graphs in 3.14 show the bifurcation diagram depicting the local extrema of x and the Lyapunov exponent [19], respectively. The periodic regions (shown as the regions between vertical,dashed, green lines with a solid green line below) were identified by investigating the bifurcation diagram and the Lyapunov exponent plots.

For the ordinal networks, the left columns plots of Figure 3.14 show a significant drop in all six scores for the large periodic window corresponding to approximately $0.409 \leq a \leq 0.412$. There are also less pronounced drops in these scores for the other shorter periodic windows. These drops are especially evident for $\langle k \rangle$, $E'(D_1)$, and $P(D_1)$ where the scores significantly decrease in comparison to their surrounding values. However, some scores such as $\langle k \rangle$ are not normalized, e.g., so that $0 \leq \langle k \rangle \leq 1$. Given one time series, and not a parameterized set of series, this makes it difficult or even impossible to distinguish between chaotic and periodic regions. On the other hand, the normalized scores that I introduce in this paper, $E'(D_1)$ and $P(D_1)$, suggest periodic regions when $E'(D_1) < 0.5$ and $P(D_1) < 0.75$. It should be noted that the difference between chaotic and periodic regions, as shown in Section 3.4.2, starts degrading as noise levels are increased.

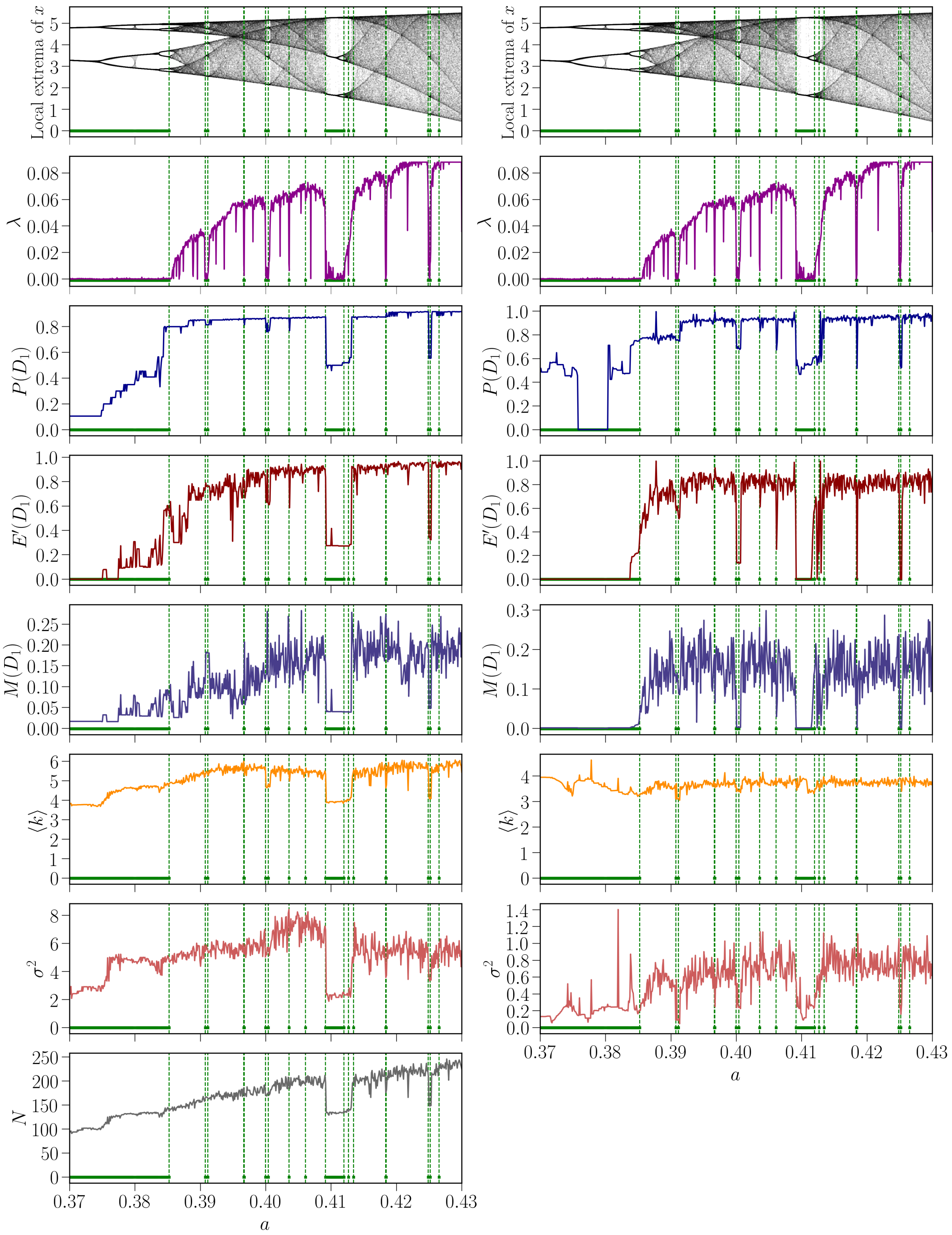


Figure 3.14: Rössler system bifurcation for $0.37 < a < 0.43$ with steps of 0.001. Left column plots include point summaries calculated from ordinal partition networks with parameters $\tau = 40$ and $d = 6$; Right column plots show the same results for the k-NN networks generated from Takens' embedding with parameters $\tau = 4$ and $d = 7$. The figure compares point summaries $P(D_1)$, $M(D_1)$, and $E'(D_1)$ with the Lyapunov exponent λ [19] and some common network parameters including the number of vertices N, mean out degree $\langle k \rangle$, and out degree variance σ^2.

Table 3.1: A comparison between persistence diagram point summaries $M(D_1)$, $P(D_1)$, and $E'(D_1)$ for detecting differences in the networks generated from for periodic (Per.) and chaotic (Ch.) time series using both k-NN graphs and ordinal partition graphs.

System/ Data	**Ref.**	**k-NN Graph from Takens' Embedding**						**Ordinal Partition Graph**					
		$E'(D_1)$		$M(D_1)$		$P(D_1)$		$E'(D_1)$		$M(D_1)$		$P(D_1)$	
		Per.	**Ch.**	**Per.**	**Ch.**	**Per.**	**Ch.**	**Per.**	**Ch.**	**Per.**	**Ch.**	**Per.**	**Ch.**
Chua Circuit	C.1	0.00	0.80	0.001	0.19	0.54	0.89	0.21	0.72	0.051	0.19	0.42	0.88
Lorenz	C.1	0.04	0.84	0.005	0.16	0.64	0.93	0.18	0.95	0.026	0.36	0.28	0.96
Rossler	C.1	0.00	0.85	0.001	0.18	0.50	0.94	0.00	0.89	0.036	0.28	0.33	0.85
Coupled Lorenz-Rossler	C.1	0.00	0.82	0.003	0.16	0.46	0.94	0.00	0.87	0.033	0.35	0.56	0.92
Bi-directional Rossler	C.1	0.00	0.76	0.004	0.13	0.55	0.87	0.25	0.91	0.064	0.29	0.40	0.92
Mackey-Glass	C.1	0.00	0.67	0.001	0.07	0.56	0.93	0.30	0.96	0.077	0.37	0.25	0.93
Logistic Map	C.1	NA						0.00	0.93	0.125	0.70	0.00	0.91
Henon Map	C.1							0.00	0.88	0.111	0.48	0.00	0.96
ECG	C.1	0.95	0.86	0.282	0.14	0.97	0.97	0.82	0.89	0.268	0.45	0.92	0.97
EEG	C.1	0.96	0.94	0.627	0.33	0.99	0.98	0.89	0.84	0.513	0.31	0.97	0.93

For the k-NN Takens' embedding networks, the right column plots of Figure 3.14 show a significant drop in $P(D_1)$, $M(D_1)$, and $E'(D_1)$ during periodic windows. However, for the traditional graph scores $\langle k \rangle$ and σ^2 this drop does not clearly correspond to the beginning and end of the periodic window. Further, for the smaller periodic windows interspersed with the chaotic regions I found that $\langle k \rangle$, σ^2, and $M'(D_1)$ are too noisy to identity the dynamic state changes in these areas. In contrast, our scores $P(D_1)$ and $E'(D_1)$ retain the ability to distinguish between dynamics regimes, and for k-NN networks of Takens' embedding I suggest tagging the time series as periodic when $E'(D_1) < 0.5$ and $P(D_1) < 0.7$.

Tabulated Results This section uses a variety of dynamical systems to validate the observations I made for the Rössler system in 3.4.1 related to the point summaries $E'(D_1)$, $M(D_1)$, and $P(D_1)$ that I introduced in 3.2.3. The results for each system when using ordinal partition networks and the k-NN network from Takens' embedding are provided side by side in Table 3.1. The model and time series information for all of these systems are provided in C.1. The table can be categorized into three types of dynamical systems: (1) systems of differential equations (Chua circuit, Lorenz, Rössler, coupled Lorenz-Rössler, bi-directional Rössler, and Mackey-Glass equations), (2) discrete-time dynamical systems (Logistic map, and Hénon map), and (3) ECG and EEG signals. The paragraphs below discuss the results for each one of these systems.

Systems of differential Equations: As shown in Table 3.1, our point summaries from both networks yield distinguishable differences between periodic and chaotic time series. The k-NN graph results in Table 3.1 show that periodic time series have $E'(D_1) < 0.5$, $M(D_1) < 0.15$, and $P(D_1) < 0.7$. Similarly, the ordinal partition graph scores in Table 3.1 show that periodic time series have $E'(D_1) < 0.5$, $M(D_1) < 0.07$, and $P(D_1) < 0.75$.

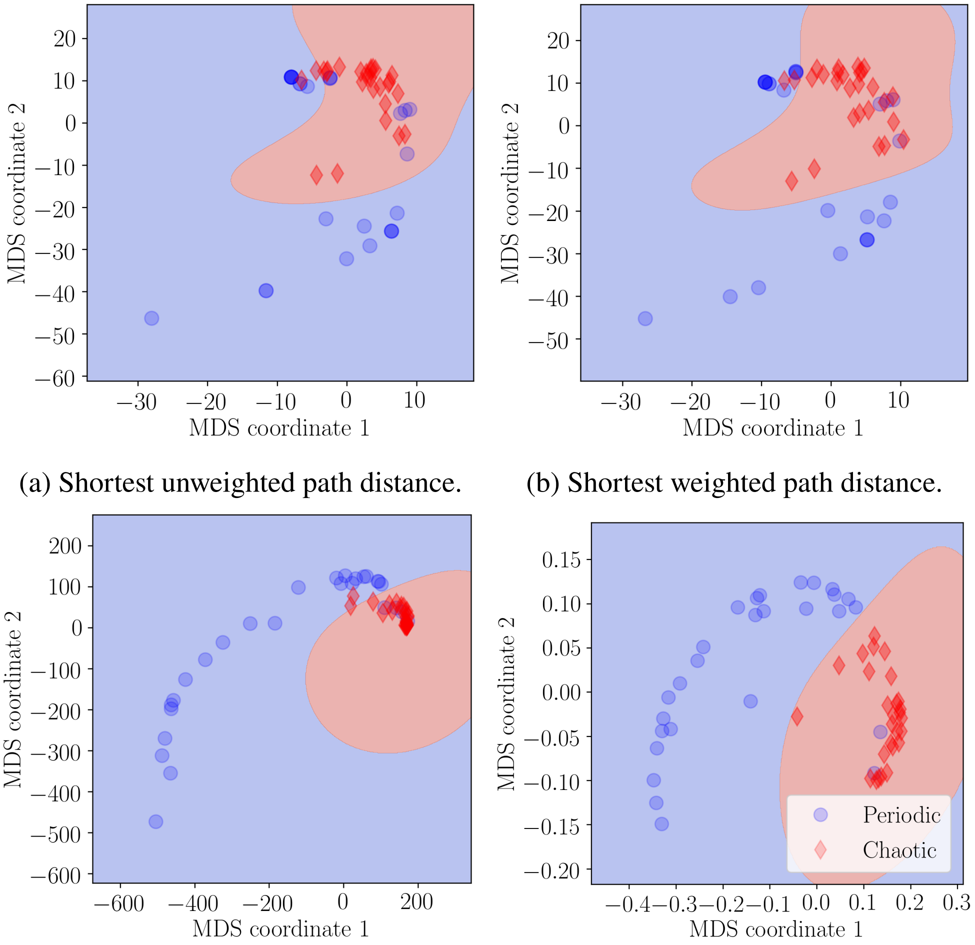
Discrete dynamical systems: The results for the discrete dynamical equations in Table 3.1 show distinguishable differences between periodic maps in comparison to chaotic maps when using ordinal partition networks. Takens' embedding was not applied to the discrete dynamical systems, and only the ordinal partition network results are reported here because working with these networks is more natural for maps.

EEG and ECG Results: The point summary results from real world data sets (ECG and EEG) shown in Table 3.1 have inherent noise, which causes the differences between the compared states to be less significant as shown in Fig. 3.18. The k-NN graph results in Table 3.1 do not show a significant difference between the two groups for either ECG and EEG data. This is most likely due to the sensitivity of Takens' embedding to noise and perturbations. However, I did find a difference between epileptic and healthy patients through the networks formed by ordinal partitions for ECG [153] and EEG [7] data. 3.4.2 discusses the effect of additive noise on the point summaries in more detail. As a note, there have been other methods for characterizing EEG data using TDA and persistent entropy [184], but our method is different from prior works because I apply persistent homology to the generated networks.

In this section we discuss the empirical results on the dynamic state detection capabilities and stability of the persistent homology of ordinal partition networks using the distance methods for incorporating weight information.

3.4.2 Dynamic State Detection Using Machine Learning on Persistence Diagrams

To determine the viability of the persistence diagram for categorizing the dynamic state of a signal using the persistent homology of the shortest weighted path, weighted shortest path, and diffusion distances compared to the shortest unweighted path distance we use the lower dimensional projection of the persistence diagrams. Specifically, we implemented the Multi-Dimensional Scaling (MDS) projection to two dimensions using the bottleneck distance matrix for our 23 systems (see Table C.1 for a list). These systems were simulated from the dynamical systems module in the Python package `Teaspoon` with details on the simulations provided in Appendix C. We then use a Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel to delineate periodic and chaotic dynamics based on the two dimensional MDS projection. The SVM fit was done using default parameters for the `SKLearn` SVM package in Python.



(a) Shortest unweighted path distance.

(b) Shortest weighted path distance.

(c) Weighted shortest path distance

(d) Diffusion distance

Figure 3.15: Comparison between the (a) shortest unweighted path, (b) shortest weighted path, (c) weighted shortest path, and (d) lazy diffusion distances using a two dimensional MDS projection (random seed 42) of the bottleneck distances between persistence diagrams of the OPN for chaotic and periodic dynamics with an SVM radial bias function kernel separation.

I generate results separating persistence diagrams from periodic and chaotic dynamic using the following graph distances: shortest unweighted path, shortest weighted path, weighted shortest path, and diffusion distance. These distance are used when defining the distance matrix that is used to calculated the persistent homology of the complex network. In the following paragraphs I apply this machine learning analysis to both the OPNs and CGSSNs.

Machine Learning on the Ordinal Partition Network's Persistent Homology The results for the OPN using the shortest unweighted path (Fig. 3.15 a), shortest weighted path (Fig. 3.15 b), weighted shortest path (Fig. 3.15 c), and diffusion distance (Fig. 3.15 d) are shown in Fig. 3.15.

The average and standard deviation of the accuracy for each SVM kernel are provided as the percent accuracy in Table 3.2. These accuracy statistics were generated using random seed 1 to 100.

Based on this initial analysis it is clear that the diffusion distance significantly outperforms the other distance methods with an accuracy of 95.0% ± 0.9% in comparison to the second best accuracy of 89.5% using the weighted shortest path. The worst performance was from the shortest unweighted path distance, which has an accuracy of 80.3% for this random seed (42).

We theorize that one reason for the increased performance when using the diffusion distance is in how it tends to normalize the scale of the persistence diagram. Specifically, when comparing the 23 dynamical systems, the maximum lifetimes for $t = 2d$ walk steps ranges from 0.08 to 0.21 with a mean of 0.147 and standard deviation of 0.042 or 28.6% of the average. In comparison, the maximum lifetimes for the shortest unweighted path distance range from 2 to 24 with an average of 9.38 and standard deviation of 6.36 or 67.8% of the average. This demonstrates that the persistence diagrams from the diffusion distance calculation tends to be more consistent in magnitude. We can further show this relationship using the cycle graph $G_{\text{cycle}}(n)$, where n as the number of nodes is increased from 2 to 500 with the maximum persistence calculated for each graph (see Appendix Section D.1). In comparison to the shortest path distances, this result shows that the persistence of the cycle graph does not continue to grow with a larger cycle graph when using the diffusion distance and trends to a plateau.

Overall, none of the distances in combination with the ordinal partition networks were able to accurately separate 100% of the periodic from chaotic persistence diagrams.

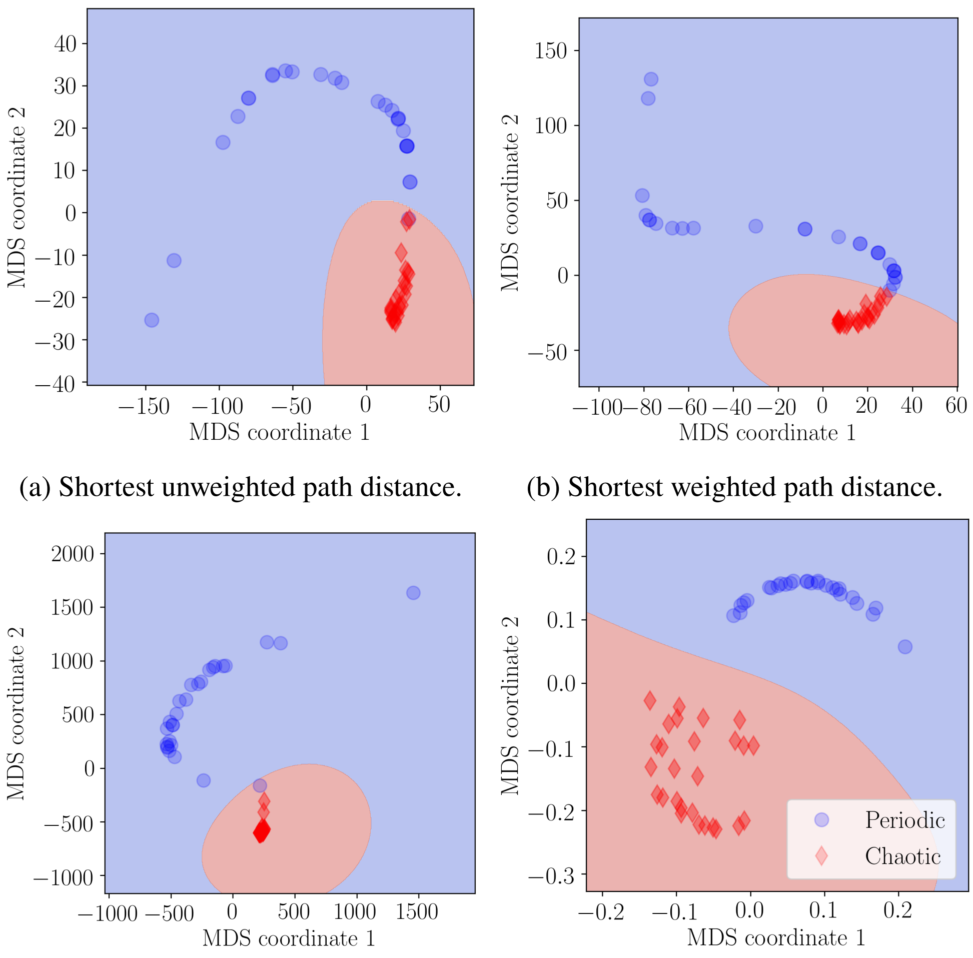
Table 3.2: Accuracies of the distance methods for both ordinal partition and coarse grained state space networks.

Network	**Distance Method**	**Percent Accuracy (%)**
OPN	Shortest unweighted path	80.7 ± 1.5
OPN	Shortest weighted path	88.9 ± 0.0
OPN	Weighted shortest path	88.9 ± 0.0
OPN	Lazy diffusion distance	95.0 ± 0.9
CGSSN	Shortest unweighted path	98.1 ± 0.0
CGSSN	Shortest weighted path	100.0 ± 0.0
CGSSN	Weighted shortest path	98.1 ± 0.0
CGSSN	Lazy diffusion distance	100.0 ± 0.0

Machine Learning on the Coarse Grained State Space Network's Persistent Homology I next repeat the previous SVM analysis on the coarse grained state space network. As mentioned previously, the CGSSN has better stability qualities than the OPN and thus may be able to better distinguish between dynamic states. Further, the CGSSN takes into account the amplitude of the state space vectors, which is discarded information when creating OPNs. For this analysis we used $b = 12$ bins and $n = 4$ for generating CGSSNs for all of the systems. An appropriate delay was selected using the multi-scale permutation entropy method. The resulting SVM separations are shown in Fig. 3.16 for random seed 42.

The average and standard deviation of the accuracy for each SVM kernel are provided as the percent accuracy in Table 3.2. These accuracy statistics were generated using random seed 1 to 100.

These results show that all of the distances applied to the CGSSN outperformed the OPN alternative. Specifically, both the shortest weighted path and diffusion distances were able to have 100% accuracy for seperating dynamics based on the persistence diagrams, while the shortest unweighted path and weighted shortest path both had 98.1% accuracy. I again theorize this is due



(a) Shortest unweighted path distance.

(b) Shortest weighted path distance.

(c) Weighted shortest path distance

(d) Diffusion distance

Figure 3.16: Comparison between the (a) shortest unweighted path, (b) shortest weighted path, (c) weighted shortest path, and (d) lazy diffusion distances using a two dimensional MDS projection (random seed 42) of the bottleneck distances between persistence diagrams of the CGSSN for chaotic and periodic dynamics with an SVM radial bias function kernel separation.

to the CGSSN taking into account the state space vector amplitude information that is discarded by the OPN.

Stability Analysis One drawback to using MDS in our setting is that it cannot be used for true supervised learning as data points not in the original training set cannot be assigned a projection after the fact. We can at least analyze how sensitive the bottleneck distance between persistence diagrams is to differences in the input time series, showing that the results are resilient to noise. While we would like to be able to provide a stability proof in the spirit of [49], such an investigation is outside the scope of this work.

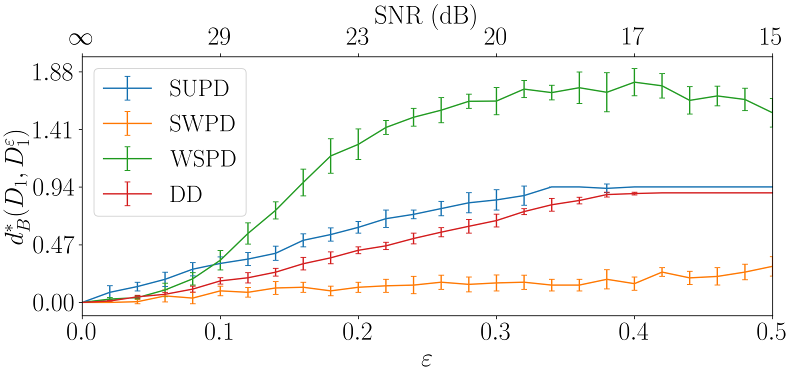


Figure 3.17: Bottleneck distance stability analysis of the periodic Lorenz system (see Eq. (4.3)) with standard deviation normalized signal and bounded ($\varepsilon = 6\sigma$) Gaussian additive noise. Analysis shows stability results using Shortest Unweighted Path Distance (SUPD), Shortest Weighted Path Distance (SWPD), Weighted Shortest Path Distance (WSPD), and Diffusion Distance (DD).

Instead we use an empirical study of the stability of the bottleneck distance using the same systems with the periodic signals (both dissipative autonomous and driven). Specifically, we tested the stability by adding bounded Gaussian noise to the signal. The noise had Signal to Noise Ratios (SNR) from ∞ (no noise) to 15 dB (extremely noisy). The additive noise followed a zero-mean Gaussian distribution that was truncated at three standard deviations from the mean and set $\epsilon = 6\sigma$. To make a fair comparison between each of the distance methods in terms of stability and sensitivity to noise we normalize the bottleneck distance as

$$d_B^*(D_1, D_1^\epsilon) = \frac{d_B(D_1, D_1^\epsilon)}{\frac{1}{2}\sum_{x \in D_1} \text{pers}(x)}, \tag{3.10}$$

where d_B is the bottleneck distance function and D_1 and D_1^ϵ are the noise free and noise contaminated one-dimensional persistence diagrams, respectively.

Figure 3.17 provides a demonstrative example of the effects of noise and the stability of the persistence diagram for the Lorenz system. The persistence diagrams as ϵ is increased are drawn overlaid in Fig.3.17 (b) In Fig.3.17, we see the bottleneck distance from the the noise free diagram to the noise contaminated diagram as the noise amplitude ϵ is increased. In the case of Lorenz, all four distance methods are stable with an approximately linear change in the bottleneck distance with respect to the noise level ϵ for small levels of noise (less than 25 dB). Additionally, d_B^* tends

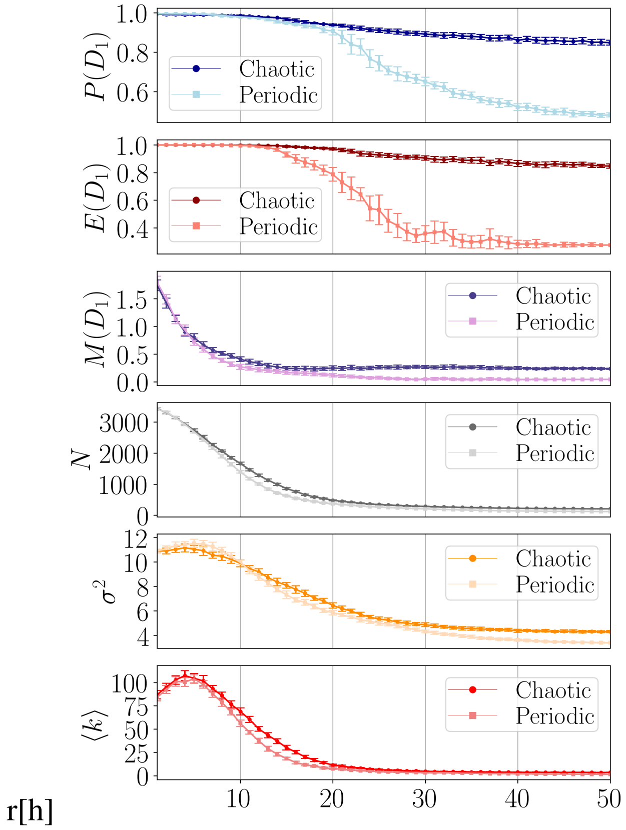


Figure 3.18: Average point summaries and network parameters for varying SNRs from Gaussian noise added to time series generated from periodic and chaotic Rössler systems. For each SNR, 25 separate samples are taken to provide mean values and standard deviations, which are shown as the error bars.

to plateau at noise levels greater than approximately 18 dB. This is due to the minimum pairing between diagrams matching to the diagonal. It is also clear the shortest weighted path distance is significantly less sensitive to additive noise with only slight changes in its normalized bottleneck distance as ϵ is increased.

Some of these characteristics seen in the Lorenz systems seem to be consistent across all of the other 22 systems; see Appendix for similar figures for the remaining systems. The shortest weighted path distance tends to be the least sensitive to additive noise. Additionally, the bottleneck distance tends to plateau at approximately 20 dB for most systems. Most importantly, all of the distance methods tend to have an approximately linear relationship between d_B^* and ϵ for low levels of noise (SNR $\leq$ 25 dB). These results empirically demonstrate that the persistence diagram is stable in this setting for limited levels of additive noise.

Some characteristics that tend to be highly dependent on the system is the sensitivity of the shortest unweighted path, weighted shortest path, and diffusion distances to additive noise. For some systems (e.g. the Rabinocih Frabrikant attractor), the weighted shortest path distance is the least sensitive to high levels of additive noise, while in other systems (e.g. the Thomas cyclically symmetric attractor) the weighted shortest path distance is the most sensitive to additive noise. In most systems the diffusion distance and shortest unweighted path are comparably sensitive to additive noise.

Effects of Additive Noise I investigate the noise robustness of the point summaries in comparison to some common network parameters—mean out degree $\langle k \rangle$, out degree variance σ^2, and the number of vertices N. The ordinal partition networks are based on time series from the Rössler system with parameters $b = 2.0$, $c = 4.0$, and either $a = 0.41$ or $a = 0.43$ for a periodic or chaotic response, respectively.

To make comparisons on the noise robustness I add Gaussian noise to the signal and calculate the point summaries and network parameters at various Signal-to-Noise Ratios (SNR) for both periodic and chaotic Rössler systems. The chosen SNR values were all the integers from 1 to 50, and at each SNR value I obtain 25 realizations of noisy signals.

To determine the 68% confidence interval at each SNR, I repeat the calculation of the point summaries and network parameters for all noise realizations at each SNR level, and I set our confidence interval to $\bar{x}(SNR) \pm s(SNR)$ where $\bar{x}(SNR)$ and $s(SNR)$ are the sample average and sample standard deviation, respectively, at a specific SNR value. Figure 3.18 shows the mean values and confidence intervals for each SNR. To assess the ability of point summaries to assign a distinguishing score to a periodic versus a chaotic system in the presence of noise, I check for an overlap in the confidence intervals for the periodic and chaotic results at each SNR. If for a particular point summary there is an overlap between the scores for periodic and the chaotic time series, then that point summary is not effective in distinguishing the dynamics at that specific SNR. Table 3.3 summarizes the noise robustness by providing the lowest SNR at which each point summary and network parameter no longer has an overlap between the periodic and chaotic confidence intervals. This result shows a lower capable SNR for the persistence based point summaries than the mean out degree $\langle k \rangle$ and variance σ^2. Another trend that should be noted is the reduction in difference between periodic and chaotic time series for high levels of noise. This should be taken into account when applying the point summaries to real world data with intrinsic noise.

Table 3.3: Noise robustness comparison for persistence diagram point summaries and network parameters using ordinal partition network.

Point Summary Network Parameter	Lowest Distinguishing SNR
$E'(D_1)$	14
$M(D_1)$	19
$P(D_1)$	20
$\langle k \rangle$	29
σ^2	29
N	8

CHAPTER 4

PERSISTENT HOMOLOGY OF DYNAMICAL NETWORKS

A dynamical system is any system whose future state is dependent on the current state. Many real-world dynamical systems are simulated using approximate models. Standard dynamical system models occupy a wide range of applications from population models [144] to aeronautical dynamics [204]. A common characteristic of a dynamical system is that its behavior can change with a system parameter, known as a bifurcation. For example, the airflow over an aircraft's wing can change from laminar to turbulent with a change in the angle of attack resulting in stall [208,251], the load on a power-grid system can push a line to fail causing a cascade failure and black-out [206,207], or a change in atmospheric chemistry can cause for severe weather [93]. Capturing the characteristic changes of a dynamical system through a measurement signal is critical in detecting, predicting, and possibly preventing some of these catastrophic failures. Outside of detecting imminent events, many other important characteristics of a system are studied through the lens of dynamics. These include population models transitioning from stable values to chaotic oscillations based on environmental factors [144], economic bubbles showing dynamics with bimodal distribution bifurcations [67], or chaotic fluctuations in power-grid dynamics through period-doubling [103].

A common avenue to study these systems is through time series or signals, which are widely utilized to analyze real-world dynamical system bifurcations. For example, a change in measured biophysical signals can indicate upcoming health problems [87,166,195] or a change in the vibratory signals of machines or structures can be the harbinger of imminent failure [12,218]. Time series typically originate from real-life systems measurements, and they provide only finitely sampled information from which the underlying dynamics must be gleaned. Time series analysis methods have many useful foundational tools for bifurcation and dynamic state analysis, such as frequency spectrum analysis [24,64] and autocorrelation [194].

While time series analysis tools can be leveraged for bifurcation detection and dynamic state analysis, many complex and high-dimensional dynamical systems and their corresponding measurements can more naturally natural be represented as complex networks. For example, there are dynamical systems models for social networks [215], disease spread dynamics [98], manufacturer-supplier networks [244], power grid network [206], transportation networks [56]. These dynamical system models demonstrate how dynamical networks can be representative of highly complex real-world systems. Many important characteristics of a dynamical network can be extracted from the data. These include source and rate of disease spread as well as predictions on future infections [72], weak branches in supply chains and possible failures [169,244], changes in infrastructure to avoid cascade failures in power grids [206, 219], transportation network optimal routing (finding an optimal minimum time route between) [17], fault analysis (detecting transportation disruptions) in transportation networks [224], and flow pattern analysis (visualization) [90].

If the studied system only has a single one-dimensional signal output, we can still represent the dynamical system as a temporal network. This is done using complex networks representations of windowed sections of time series data to visualize how the graph structure of the windowed time series data changes. Examples of graph formation techniques from time series include k-nearest-neighbors networks [118], epsilon-recurrence networks [101], coarse-grained state-space network [216, 217], or ordinal partition networks [146].

We are only using the data to construct the evolving networks in this work. As such, we will aptly refer to them as temporal graphs [94]. While a complex dynamical system typically drives the temporal graphs, the underlying equations of motion are unknown. Temporal graph data is commonly represented using attributed information on the edges for the time intervals or instances in which the edges are active [41, 97]. Using this attributed information, we can represent the graph in several ways [238]. In this work we will first represent the data in the standard attributed temporal graph structure and then use the graph snapshots approach. The graph snapshots represent the temporal graph as a sequence of static graphs $G_0, G_1, \dots, G_n$.

The standard network analysis tools for studying temporal networks often include measures such as centrality or flow measures [23], temporal clustering for event detection [53, 156, 247], and connectedness [108]. However, these tools do not account for higher-dimensional structures (e.g., loops as a one-dimensional structure). It may be important to account for evolving higher-dimensional structures in temporal networks to understand the changing structure better. For example, a highly connected network may only have one connected component with no clear clusters, but the number of loops within the network may detect the change.

To study the evolving higher dimensional structures within a temporal network, we will leverage zigzag persistence [35] from the field of Topological Data Analysis (TDA) [34]. TDA is typically used to study point cloud data through the flagship tool persistent homology.

Persistent homology, colloquially referred to as persistence, encodes structure by analyzing the changing shape of a simplicial complex (a higher dimensional generalization of a network) over a filtration (a nested sequence of subcomplexes). It should be noted that the majority of these applications utilize a relatively standard pipeline to construct this filtration. Namely, given point cloud data embedded in $\mathbb{R}^n$ as input, construct the Vietoris Rips (VR) at multiple distance filtration values. The VR complex is generated for incremented filtration values such that the result is a nested sequence of simplicial complexes. The homology of the point cloud data can then be measured for each simplicial complex. The homologies that persist over a broader range of filtration values are significant. We provide a more detailed introduction in Section 4.1. It is also possible to apply this framework to graph data using geodesic distance measures such as the shortest path as done in [162].

Unfortunately, the standard persistent homology pipeline does not account for temporal information. To account for temporal changes, we use zigzag persistence. Instead of measuring the shape of static point cloud data through a distance filtration, zigzag persistence measures how long a structure persists through a sequence of changing simplicial complexes. For example, in [233] the Hopf bifurcation is detected through zigzag persistence (i.e., a loop is detected through the one-dimensional zigzag persistence diagram). The zigzag persistence algorithm incorporates the two essential characteristics of temporal graphs we are looking to study—namely, the temporal and structural information stored within a temporal network.

In this work, we will use zigzag persistence to visualize these changes. Zigzag persistence compactly represents both temporal and structural changes using a persistence diagram. The persistence diagram is a two-dimensional summary of persistent homology. The resulting persistence diagram is commonly analyzed through either a qualitative analysis, standard one-dimensional statistical summaries, or machine learning via vectorizing the persistence diagram.

Organization We will start in section 4.1 with an introductory background on persistent homology and zigzag persistence. Following this, we introduce the two systems we will study. The first is a dataset collected over a week of the Great Britain transportation system. The second is an intermittent Lorenz system simulation, where we generate a temporal network through complex networks of sliding windows. Next, in Section 4.2, we overview the general pipeline for applying zigzag persistence to temporal graph data. We couple this explanation with a demonstrative toy example. In Section 4.3 we apply zigzag persistence to our two examples and show how the resulting persistence diagrams help visualize the underlying dynamics in comparison to standard temporal network analysis techniques.

4.1 Background

4.1.1 Zigzag Persistence

A problem with the standard application of persistent homology is that it requires each subsequent simplicial complex to be a subset of the previous simplicial complex. This directionality problem results in limited applications where new simplexes can not be included in the simplicial complex filtration, which occurs in many real-world datasets. This issue was alleviated through zigzag persistence [35, 36], which allows for a zigzagging of the subset directions as

$$K_0 \leftrightarrow K_1 \leftrightarrow K_2 \leftrightarrow \ldots \leftrightarrow K_n, \tag{4.1}$$

where there isn't necessarily a filtration parameter for the ordered simplicial complexes. The subset direction is determined based on which is the subset. However, it is possible to force the direction to zigzag if we can create a simplicial complexes $K_{i,i+1}$ with both K_i and K_{i+1} as subset as shown in Eq. (4.2).

$$K_0 \hookrightarrow K_{0,1} \hookleftarrow K_1 \hookrightarrow K_{1,2} \hookleftarrow K_2 \hookrightarrow \ldots \hookleftarrow K_{n-1} \hookrightarrow K_{n-1,n} \hookleftarrow K_n. \tag{4.2}$$

We can now determine when homology features are born and die based on the zigzag persistence. We again track this with a persistence diagram consisting of persistence pairs (b_i, d_i). However, b_i and d_i are the times or indices when the homology was born and died instead of the filtration value. If there are times associated with the indices, then the time value can be used in substitution of the indices. Additionally, the complexes $K_{i,i+1}$ have half step indices (e.g., $i + 0.5$), or the average time between the two can be used. This work will have times associated with the simplicial complexes instead of indices. For more details, an example demonstrating zigzag persistence on a temporal graph is provided in Section 4.2.1.

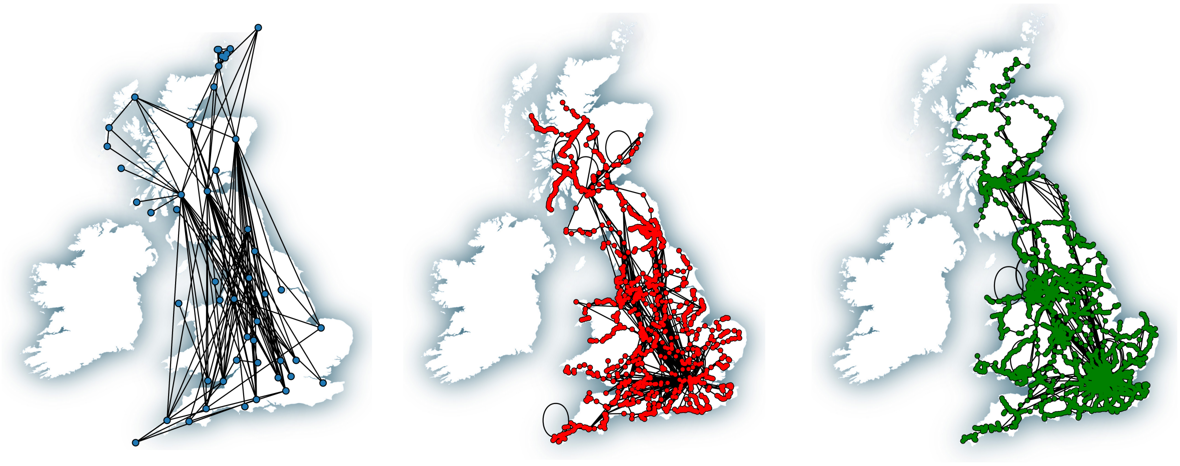
4.1.2 Temporal Graphs

A temporal graph is a graph structure that incorporates information on when edges and/or nodes are present in the graph. We will only be using the case on temporal information attributed to the edges in this work.

We apply zigzag persistence to two main temporal networks described in the subsequent subsections. The first is the Great Britain transportation network, and the second is the temporal ordinal partition network.

Great Britain Multi-layered Temporal Transportation Network We use temporal networks created from the Great Britain (GB) temporal transportation dataset [79] for the air, rail, and coach transportation methods. This data provides the destinations (nodes) and connections (edges) for public transportation in GB. Additionally, the departure and arrival times are provided to allow for a temporal analysis. This temporal data was collected for one week.

The graphs constructed without the use of temporal information are shown in Fig. 4.1 where the destinations are overlaid with a GB map. As shown, the network's destination encompasses both cities and remote towns as well as the connections between them. As such, the network's structure



(a) Air Travel Network (b) Coach Network (c) Rail Network

Figure 4.1: Transportation networks of Great Britain for air, coach, and rail travel.

encodes the transportation connectivity. In section 4.2 we introduce our method for generating snapshots for different time intervals over the entire week period that the transportation data was collected.

Temporal Ordinal Partition Network Ordinal partition networks [146] are a graph representation of time series data based on permutation transitions. As such, they encapsulate the state space structure of the underlying system. While we only use the ordinal partition network in this work, there are several other transitional complex networks from time-series data that a similar analysis could be done. These include k-nearest-neighbors [118], epsilon-recurrence [101], coarse-grained state-space networks [216, 217].

The ordinal partition network is formed by first generating a sequence of permutations from the time series $x = [x_0, x_1, x_2, \dots, x_n]$ using a permutation dimension m and delay τ. These are the same permutations in the information statistic permutation entropy [14]. In this work we choose $m = 6$ and τ using the multi-scale permutation entropy method as suggested in [160]. We generate a sequence of permutation by assigning each vector embedding

$$v_i = [x_i, x_{i+\tau}, x_{i+2\tau}, \dots, x_{i+(m-1)\tau}] = [v_i(0), v_i(1) \dots, v_i(m-1)]$$

to one of the $m!$ possible permutations. We assign the permutation $\pi_i = [\pi_i(0), \ldots \pi_i(n-1)] \in \mathbb{Z}^m$ based on the ordinal pattern of v_i such that $v_i(\pi(0)) \leq v_i(\pi(1)) \leq v_i(\pi(2)) \leq \ldots \leq v_i(\pi(n-1))$.

Using the sequence of permutations $\Pi = [\pi_0, \pi_1, \ldots, \pi_{n-m-2\tau}]$ we can form a graph $G(E, V)$ by sett the vertices V as all permutations used and edges for transitions from π_i to $\pi_i + 1$. We will not add weight or directionality to the graph for this formation. However, we will include the index i and the corresponding time at which each edge is activated as temporal data for the graph. For more details on the ordinal partition network, we direct the reader to [146, 162].

4.2 Method

To apply zigzag persistence to study temporal graphs, we need a process as outlined in the pipeline shown in Fig. 4.2. This process needs to take a temporal graph to a sequence of snapshot graphs, which can then be represented as zigzagging subset simplicial complexes. This procedure then allows for the application of zigzag persistence.

We begin with a dataset as a temporal graph where each edge has intervals or instances in time representing when the edge is active.

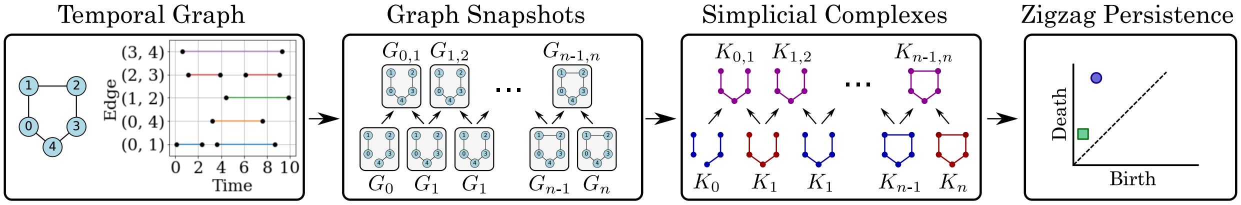


Figure 4.2: Pipeline for applying zigzag persistence to temporal networks. Begin with an unweighted and undirected **temporal graph** where each edge is on at a point or interval of time. Create **graph snapshots** using a sliding window interval over the time domain. Create a sequence of **simplicial complexes** from the graphs and apply **zigzag persistence** to the union zigzag simplicial complexes.

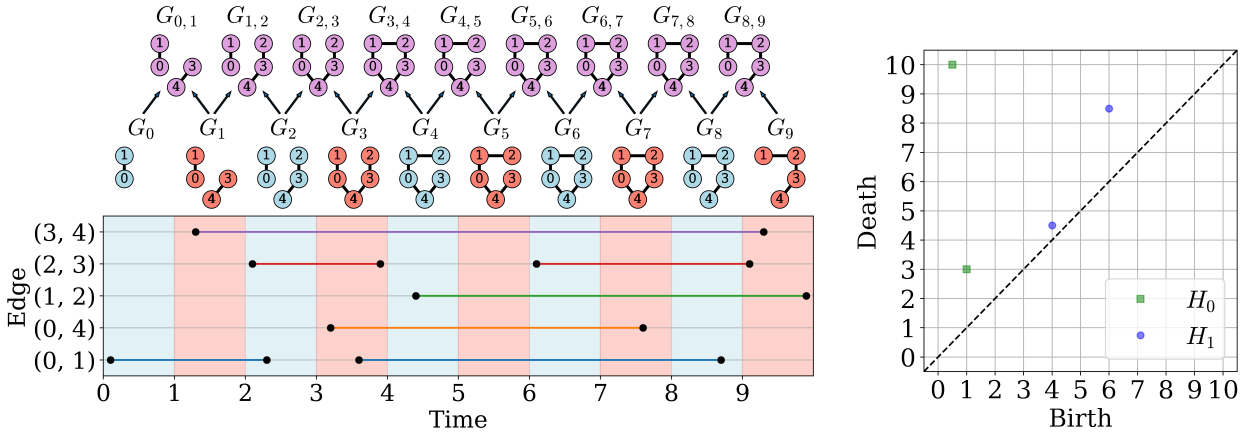
Graph snapshots G_i are generated using a sliding window technique using the temporal information. The sliding window for graph snapshot G_i is defined as $SW_i(w, t_i^{SW})$ with width δ and centered at time t_i^{SW}. The sliding windows can also be set to overlap by choosing window times such that $t_{i+1}^{SW} - t_i^{SW} \leq w$. We further need to include union windows for the use of zigzag persistence, which are defined as $G_{i,i+1}$ and re-generated from the union of two adjacent sliding windows $SW_i \cup SW_{i+1}$.

From the graph snapshots using the sliding windows and their unions, we create a sequence of simplicial complexes using a Vietoris-Rips (VR) complex with distance filtration value r. The choice of an appropriate r is dependent on the application, but in general, we suggest $1 \leq r \leq 3$. The VR complex K_i for each G_i is generated using the unweighted and undirected shortest path distance between nodes and filtration value r. If $r = 1$, the original graph is returned by filling in only the edges as 1-dimensional simplices. Similarly, higher r values fill in r-dimensional simplices. Choosing higher r values for generating simplicial complexes results in small higher-dimensional features not being represented in the persistence. For example, if $r = 2$ and there is a 3-node cycle subgraph in the graph, the cycle would be filled with the 2-simplex. This would result in the cycle not being present in the one-dimensional homology.

We use the resulting sequence of simplicial complexes to calculate zigzag persistence to study the changing structure of the temporal graph. In the following simple example shown in Fig. 4.3, we describe the method in more detail and show how to interpret the resulting zigzag persistence diagram.

4.2.1 Example

In this example, we demonstrate how to use zigzag persistence to measure the changing structure of a simple 5-node cycle graph as edges are added and removed based on the temporal information. Figure 4.3a shows the temporal information of the simple cycle graph as the intervals on each edge. The sliding windows for this example are created with width $w = 1$ and $t_i^{SW} = 0.5$ such that the windows are the non-overlapping intervals $SW_i = [i, i + 1]$. For each window a graph snapshot G_i is created, where G_i is the edge induced subgraph with edges added if the window SW_i overlaps with the edge interval. The union graphs $G_{i,i+1}$ are also created using the union of adjacent sliding windows as $SW_i \cup SW_{i+1} = [i, i + 2]$. By using the union subgraphs we have $G_i \subset G_{i,i+1}$ and $G_{i+1} \subset G_{i,i+1}$.



(a) Edge intervals with sliding windows highlighted (alternating blue-red) with corresponding graphs and union graphs above.

(b) Zigag persistence diagram for both H_0 and H_1.

Figure 4.3: Example zigzag persistence applied to a simple temporal cycle graph.

To calculate the zigzag persistence for this example we created VR complexes K_i and $K_{i,i+1}$ for each graph G_i and union graph $G_{i,i+1}$, respectively, using the unweighted and undirected shortest path distance with distance filtration value $r = 1$. Setting $r = 1$ creates the graph equivalent simplicial complex. At the end of the sliding windows, we consider the graph empty and set the death of any remaining homology features as the end time of the last window (i.e., $t = 10$ for this example). The resulting zigzag persistence diagram is shown in Fig. 4.3b.

This persistence diagram shows the zero-dimensional and one-dimensional features as H_0 and H_1, respectively. There are two one-dimensional features at persistence pairs $(1, 3)$ and $(0.5, 10)$. The persistence pair $(0.5, 10)$ was born first at G_0 which occurred at $t = 0.5$ as the first connected component. The second component and persistence pair appears in $G_{0,1}$ at time $t = 1$. Both of these components persist until $G_{2,3}$ at $t = 3$, where, based on the elder's rule, the first-born feature persists with the later-born feature or component dying. This explains the persistence pair $(1, 3)$ with the component born at $G_{0,1}$ and dying at the merging of components in $G_{2,3}$. The first-born component continues to persist until the last window. Based on our definition, we set the death of this feature as the end interval of the last window, with the second persistence pair at $(0.5, 10)$. The one-dimensional feature (the cycle represented in H_1) is present twice in the persistence diagram. This is due to it first appearing in $G_{3,4}$ and then disappearing at G_4 with the first corresponding persistence pair at $(4, 4.5)$. The cycle then reappears at $G_{5,6}$ and again disappears at G_7 corresponding to the second persistence pair at $(6, 8.5)$.

This example demonstrates how zigzag persistence captures the changing structure of temporal graphs at multiple dimensions. We can also capture higher-dimensional structures using the persistence diagram, but we do not investigate them in this work.

4.3 Results

To demonstrate the functionality of zigzag persistence for analyzing temporal graphs, we will use two examples. The first is an analysis of transportation data from Great Britain discussed in Section 4.1. The second is a simulated dataset from the Lorenz system that exhibits intermittency, a dynamical system phenomenon where the dynamic state transition from periodic to chaotic in irregular intervals.

We compare our results for both examples to some standard networks tools to analyze temporal networks. Namely, we will compare two connectivity statistics and three centrality statistics.

The two connectivity statistics analyze the Connected Components (CCs). The first CC statistic is the number of connected components N_{cc}, which provides a simple shape summary of the graph snapshots by understanding the number of disconnected subgraphs. The second statistic is the average size of the connected components $\bar{S}_{cc}$. This statistic provides insight into how significant the components are for each graph snapshot.

The second statistic type is on centrality measures. The three centrality measures we use are the average and standardized degree centrality $\bar{C}_d$, betweenness centrality $\bar{C}_b$, and closeness centrality $\bar{C}_c$. The degree centrality measures the number of edges connected to a node, the betweenness centrality measures how often a node is used all possible shortest paths, and the closeness centrality measures how close the node is to all other nodes through the shortest path. For details on the implementation of each centrality measure, we direct the reader to [125].

4.3.1 Great Britain Temporal Transportation Network

From the Great Britain transportation data discussed in Section 4.1, we created temporal graphs from the air, rail, and coach transportation methods. We created these temporal graphs using the sliding window technique for graph snapshots introduced in Section 4.2. For the sake of brevity, in this section, we will only show the results of applying zigzag persistence to the temporal rail network. Results for the other two networks (air and coach) are provided in the appendix and show similar behavior.

We set the sliding windows with width $w = 20$ minutes. We chose this window size based on the average weight time being 7 minutes and 7 seconds with a standard deviation of 7 minutes and 24 seconds from a collected sample [235]. Additionally, we used an overlap of 50% between adjacent windows. To create simplicial complexes from the graph snapshots, we used a distance filtration of $r = 1$.

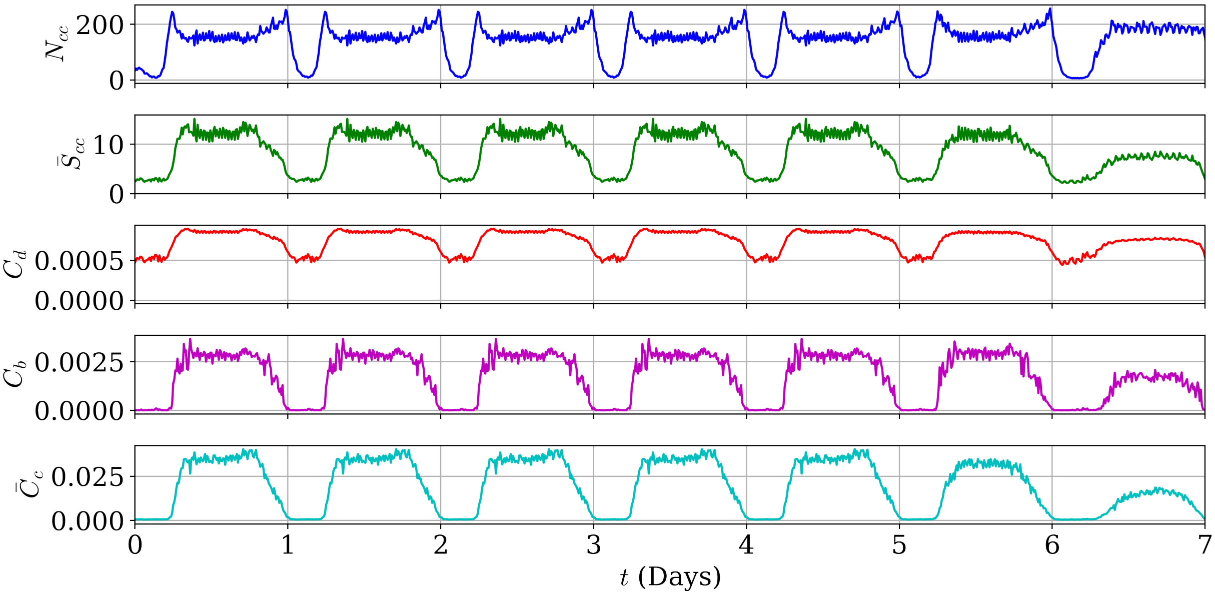
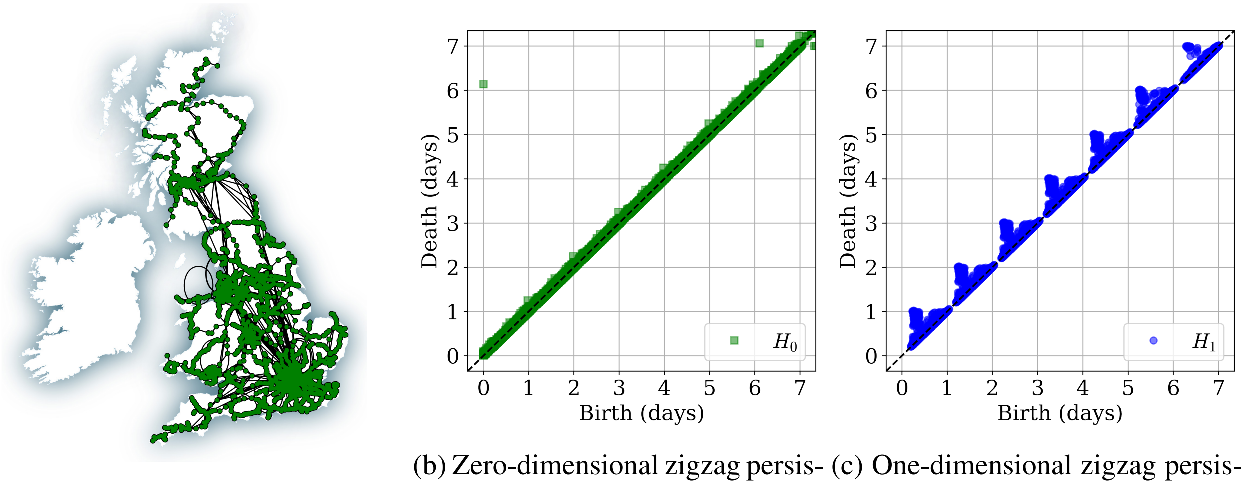


Figure 4.4: Connectivity and centrality analysis on temporal Great Britain rail network.

As a first approach to understand the dynamics of this graph, we implement the standard centrality and connectivity statistics as shown in Fig. 4.4. The standard tools show us the general daily trends. Specifically, all the connectivity and centrality measures increase during peak travel hours. However, further information is difficult to glean from these statistics. On the other hand, in



(a) Full Rail Travel Network.

(b) Zero-dimensional zigzag persistence.

(c) One-dimensional zigzag persistence.

Figure 4.5: Zigzag persistence diagrams of the rail transportation network of Great Britain.

Fig. 4.5 the zigzag persistence provides us with much more information. It also shows daily trends, but it also conveys through H_0 that a main connected component persists for the first six days and a second component for the last day. This provides an understanding of the long-term connectivity of this component that was not present in the standard statistics. Further, the H_1 encapsulates that travel loops form during peak travel times and persist daily.

4.3.2 Temporal Ordinal Partition Network for Intermittency Detection

Using a sliding window technique, we can represent ordinal partition networks as temporal graphs. However, instead of each edge having a set of intervals associated with it as in the example in Section 4.2, they instead have time instances each edge is active. The instances are based on when a transition between unique permutations occurs. For example, the transition from π_i to π_{i+1} occurring at time t_i would be active for that moment in time t_i. If the sliding window overlaps with an edge's activation instance, we add that edge to the sliding windows graph.

We will show how this procedure can be used to detect chaotic and periodic windows in a signal exhibiting intermittency (i.e., the irregular transitions from periodic to chaotic dynamics). The signal is x solution to the simulated Lorenz system defined as

$$\frac{dx}{dt} = \sigma(y - x), \ \frac{dy}{dt} = x(\rho - z) - y, \ \frac{dz}{dt} = xy - \beta z \tag{4.3}$$

with system parameters $\sigma = 10.0$, $\beta = 8.0/3.0$, and $\rho = 166.15$ for a response with type 1 intermittency [188]. We simulated the system with a sampling rate of 100 Hz for 500 seconds with only the last 70 seconds used. We set the sliding windows for generating graph snapshots to have a width of $w = 10\tau$ and 80% overlap between adjacent windows. For each window, we generated ordinal partition networks using $\tau = 30$ and $n = 6$, where τ was selected using the multi-scale permutation entropy method [160].

The resulting signal $x(t)$ from simulating the Lorenz system in Eq. (4.3) is shown in Fig. 4.6 with example ordinal partition networks generated at a chaotic window highlighted in red and a periodic window highlighted in blue. These sample graph snapshots show that the structure of the ordinal partition network significantly changes depending on the dynamic state of the window's time-series segment. Further, we expect to see little change in the graph structure while the window slides along a periodic region of $x(t)$ compared to significant changes when overlapping with a chaotic region.

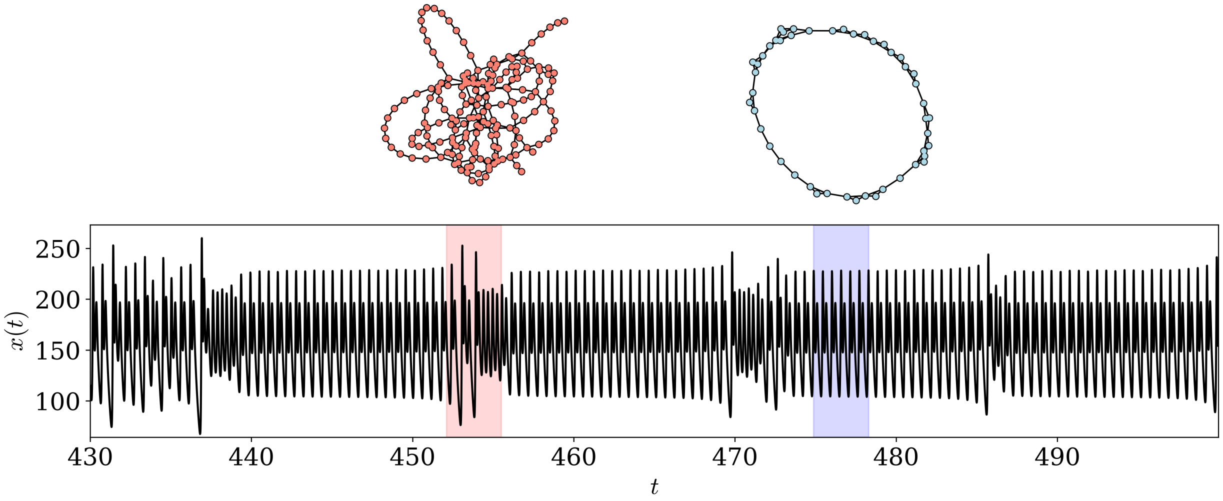


Figure 4.6: The $x(t)$ solution to simulation of Lorenz system from Eq. (4.3) exhibiting intermittency with example sliding windows for both periodic (blue) and chaotic (red) dynamics with their respective ordinal partition networks.

We show the standard tools for connectivity and centrality measures of the graph snapshots in Fig. 4.7. The number of components N_{cc} is constant due to the nature of the ordinal partition network, where the sequence of permutation transitions creates a chain of connected edges. As such, there is no structural information in the number of components. However, the size of the components does increase during the chaotic windows. This increase is due to, in general, more unique permutations and thus nodes used in a chaotic signal compared to periodic. Of the centrality statistics, only the average closeness centrality shows an apparent increase during chaotic regions. The increase in centrality is most likely due to the chaotic regions causing a more highly connected graph as demonstrated in the chaotic window and corresponding network of Fig. 4.6. While these statistics do provide some insight into the changing dynamics, they do not show how the higher-dimensional structure of the graph evolves through the sliding windows and graph snapshots.

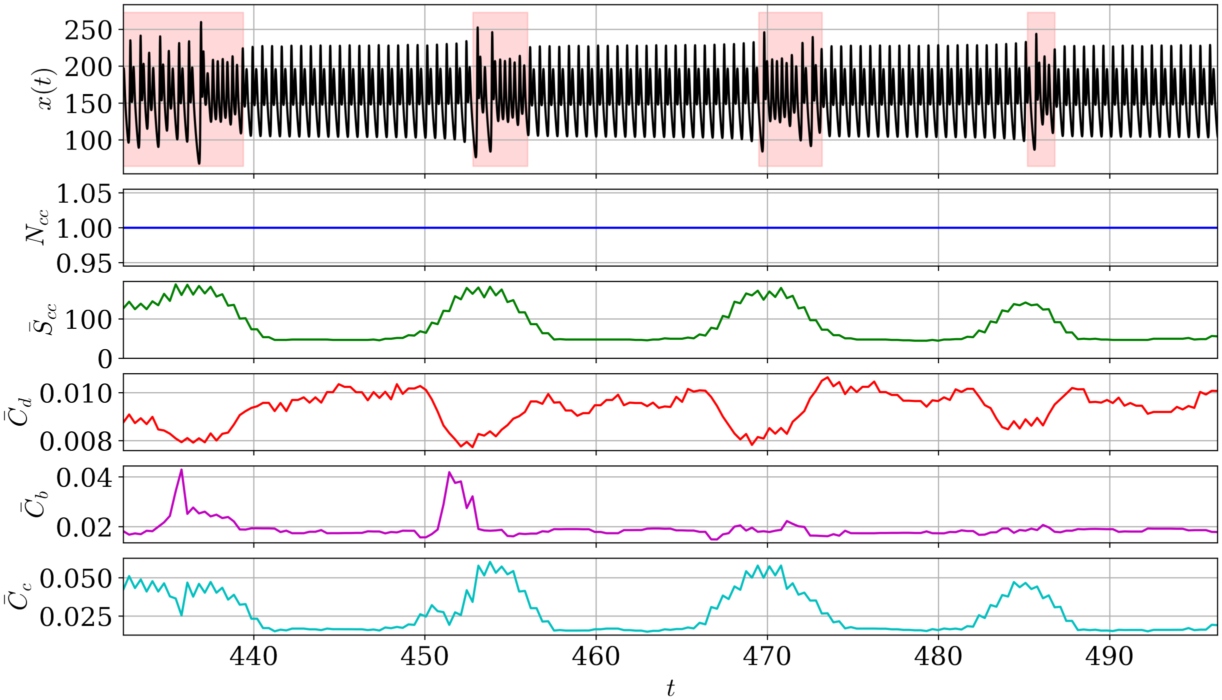


Figure 4.7: Connectivity and centrality analysis on temporal ordinal partition network with chaotic regions of $x(t)$ highlighted in red.

In comparison to the standard statistics, the H_1 in Fig. 4.8 shows us a persistent loop structure that persists between the chaotic windows, which is representative of the periodic nature. Further, the

H_1 shows that the chaotic windows characteristically have many low-lifetime persistence pairs. This

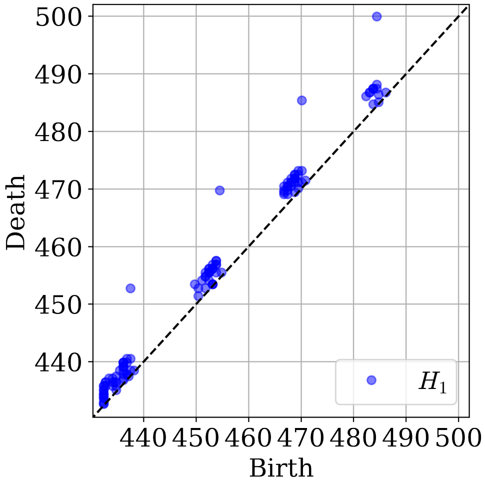


Figure 4.8: One-dimensional zigzag persistence of the temporal ordinal partition network from the x solution of the intermittent Lorenz system described in Eq. (4.3).

is in line with the results in [162] that showed ordinal partition networks from chaotic signals tend to have persistence diagrams with many features in H_1 in comparison to their periodic counterpart. These additional insights through the zigzag persistence provide a helpful insight into analyzing temporal graphs that is not possible with standard statistics.

4.4 Conclusion

In this work we studied how to effectively apply zigzag persistence to temporal graphs. Zigzag persistence provides a unique perspective when studying the evolving structure of a temporal graph by tracking the standard lower-dimensional features (e.g., connected components), but also higher-dimensional features (e.g., loops and voids). We showed the benefits of using zigzag persistence on two examples: the Great Britain transportation network and ordinal partition networks. Our results showed that the informative zero and one-dimensional zigzag persistence provided insights into the structure of the temporal graph that were not easily gleaned from standard centrality and connectivity statistics.

We believe zigzag persistence could also be leveraged to study other temporal graphs including flock behavior models (e.g., viscsek model) and the emergence of coordinated motion, power grid dynamics with the topological characteristics of a cascade failures, and supplier-manufacture networks through the effects of trade failures on production and consumption.

Future work to improve this method would involve an analysis on deciding an optimal window size and overlap, a method to incorporate edge weight and directionality, and temporal information on both the nodes and edges. It would also be worth investigating higher-dimensional features (e.g., voids through H_2).

CHAPTER 5

PERSISTENT HOMOLOGY OF DYNAMICAL NETWORKS

This auxillary chapter of my research introduces some of the data sets used throughout my research and the software packages developed. Namely, the two main experimental data sets are from a magnetic single pendulum (see Section 5.1) and a tracked double pendulum [165]. I did not include the extensive double pendulum documentation in this document. However, the open-source publication is available [165]. Throughout my research project I have also been contributing and developing the website documentation for teaspoon, which is an open source topological signal processing package available through Python.

5.1 Experiment: Magnetic Pendulum

Note: a Computer Aided Design (CAD) model and design document for the pendulum used for the experimental section of this manuscipt is available through *GitHub* at *https://github.com/Khasawneh-Lab/simple_pendulum*.

The driven magnetic pendulum is a well known system to exhibit chaos [117,214,231]. Therefore, I designed and built a magnetic pendulum apparatus, and utilized the ordinal partition embedding and TDA to characterize the dynamics of the resulting signals.

In this section I derive a simplified equation of motion using Lagrange's approach. The design, manufacturing, and equipment used for the experiment are also explained. Additionally, I describe our methods for estimating and measuring the constants that appear in the equation of motion.

5.1.1 Mathematical Model

I begin by deriving the equations of motion for the physical system shown in Fig. 5.1. Let the total mass of the rotating components be M, the distance from the rotation center O to the mass center of the rotating assembly r_{cm}, and the mass moment of inertia of the rotating components about their mass center be I_{cm}. Further, assume that the magnetic interactions are well approximated by a dipole model with $m_1 = m_2 = m$ representing the magnitudes of the dipole moment. To develop

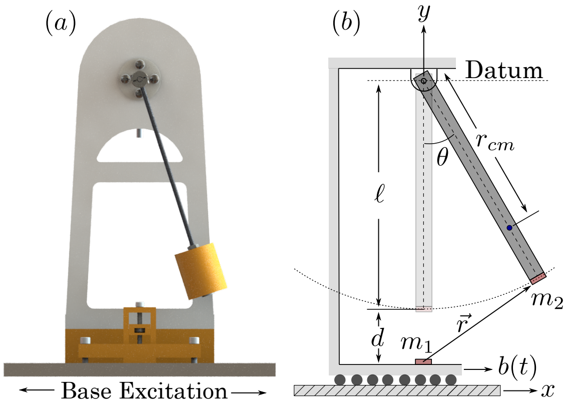


Figure 5.1: Rendering of experimental setup in comparison to reduced model, where $b(t) = A\sin(\omega t)$ is the base excitation with frequency ω and amplitude A, r_{cm} is the effective center of mass of the pendulum, d is the minimum distance between magnets $m_1 = m_2 = m$ (modeled as dipoles), and ℓ is the length of the pendulum.

the equation of motion, I use Lagrange's equation (Eq. (5.9)), so the potential energy V, kinetic energy T, and non-conservative moments R are needed. In this analysis the damping moments and the moments generated from the magnetic interaction are treated as non-conservative. The potential and kinetic energy are defined as

$$\begin{aligned} T &= \frac{1}{2}M|\vec{v}_{cm}|^2 + \frac{1}{2}I_{cm}\dot{\theta}^2, \\ V &= -Mgr_{cm}\cos(\theta), \end{aligned} \tag{5.1}$$

where $\vec{v}_{cm}$ is the velocity of the mass center given by

$$\vec{v}_{cm} = r_{cm}\dot{\theta}\left[\cos(\theta)\hat{\epsilon}_x + \sin(\theta)\hat{\epsilon}_y\right] + A\cos(\omega t)\hat{\epsilon}_x. \tag{5.2}$$

In Eq. (5.2), $A\cos(\omega t)$ is introduced from the base excitation $b(t) = A\cos(\omega t)$ in the x direction with A as the amplitude and ω as the frequency and $\hat{\epsilon}_x$ and $\hat{\epsilon}_y$ are the unit vectors in the x and y directions, respectively.

The non-conservative moments are caused by the energy lost to damping. For our analysis, I consider three possible mechanisms of energy dissipation: Coulomb damping τ_c, viscous damping τ_v, and quadratic damping τ_q. I chose to use all three mechanisms of damping due to previous work on damping estimation for a pendulum similar to the one I used [183]. These three moments are defined as

$$\begin{aligned}\tau_c &= \mu_c \mathrm{sgn}(\dot{\theta}),\\ \tau_v &= \mu_v \dot{\theta},\\ \tau_q &= \mu_q \dot{\theta}^2 \mathrm{sgn}(\dot{\theta}),\end{aligned} \tag{5.3}$$

where μ_c, μ_v, and μ_q are the coefficient for Coulomb, viscous, and quadratic damping, respectively.

To begin the derivation of the torque induced from the magnetic interaction τ_m, consider two, in-plane magnets as shown on the left side of Fig. 5.2. The red side of the magnet in the figure represents its north-pole. From this representation, the magnetic force acting on each magnet is calculated as

$$\begin{aligned}F_r &= \frac{3\mu_o m^2}{4\pi r^4}\left[2c(\phi-\alpha)c(\phi-\beta) - s(\phi-\alpha)s(\phi-\beta)\right],\\ F_\phi &= \frac{3\mu_o m^2}{4\pi r^4}\left[s(2\phi-\alpha-\beta)\right],\end{aligned} \tag{5.4}$$

where m_1 and m_2 are the magnetic moments, μ_o is the magnetic permeability of free space, and $c(*) = \sin(*)$ and $s(*) = \sin(*)$. Equation (5.4) assumes that the cylindrical magnets used in the experiment can be approximated as a dipole. I later show that this assumption is satisfactory in Fig. 5.4 of Section 5.1.3. These magnetic forces are then adapted to the physical pendulum as shown on the right side of Fig. 5.2, with $\alpha = \pi/2$ and $\beta = \pi/2 - \theta$. Additionally, ϕ and r are

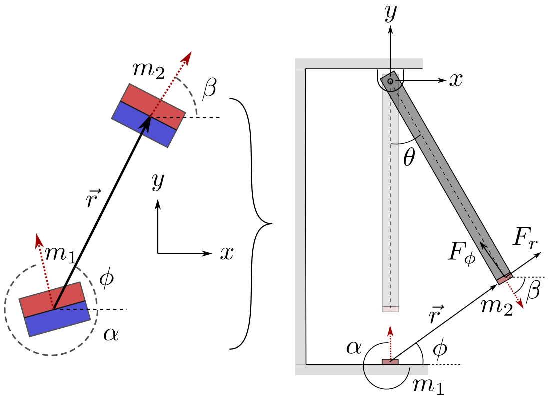


Figure 5.2: A comparison between a generic, in-plane magnetic model in global coordinates and the equivalent magnetic forces in the pendulum model F_r and F_ϕ (see Eq. (5.4)).

calculated from θ, d, and ℓ from Fig. 5.1 as

$$\phi = \frac{\pi}{2} - \arcsin\left(\frac{\ell}{r}\sin(\theta)\right), \quad \text{and} \tag{5.5}$$

$$r = \sqrt{[\ell\sin(\theta)]^2 + [d + \ell(1 - \cos(\theta))]^2}. \tag{5.6}$$

The moment induced by the magnetic interaction is then

$$\tau_m = \ell F_r \cos(\phi - \theta) - \ell F_\phi \sin(\phi - \theta). \tag{5.7}$$

Using τ_m from Eq. (5.7) and the non-conservative torques from Eq. (5.3), R is defined as

$$R = \tau_c + \tau_v + \tau_q + \tau_m. \tag{5.8}$$

Finally, the equation of motion for the base-excited magnetic single pendulum is found by substituting the above expressions into Lagrange's equation and noting that $L = T - V$

$$\frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} + R = 0. \tag{5.9}$$

Equation (5.9) was symbolically manipulated to express it in state space format using Python's *Sympy* package. Then, the system was simulated at a frequency of $f_s = 60$ Hz using Python's *odeint* function from the *Scipy* library.

5.1.2 Equipment and Experimental Design

The setup of the experiment was manufactured by extending the capabilities of a previously manufactured simple pendulum [183]. To increase the non-linearity, in-plane magnets on the base as well as at the end of the pendulum were added. To assume a permeability of free space μ_0, any ferromagnetic material within the vicinity was removed, which made the use of 3D printed components critical. In Fig. 5.3 an overview of the utilized, 3D-printed components are shown. Specifically, Figs. 5.3 (a) and (b) show exploded views of the end mass of the pendulum, and the linear stage for controlling the distance d, respectively. The magnets used are two, approximately identical, rare-earth (neodymium) N52 permanent magnets with a radius and length of 6.35 mm (1/4").

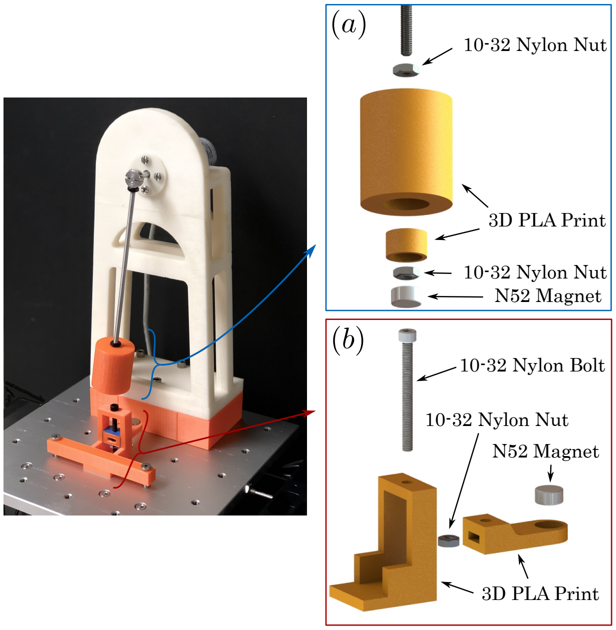


Figure 5.3: Manufacturing overview with experimental setup. In Fig. (a), an exploded view of the end mass (100% infill 3D printed PLA components) is shown with the magnet press fit into end of pendulum. In Fig. (b), an exploded view of the linear stage controlling the vertical position of the lower magnet.

Table 5.1 provides a list of the item, description, and manufacturer for all of the experimental equipment used to collect the rotational data from the magnetic single pendulum under base excitation.

Table 5.1: Equipment used for experimental data collection.

Item	Description	Manufacturer
Shaker	113 Electro-Seis	APS
DC Power Supply	Model 1761	BK Precision
Accelerometer	Model 352C22	Piezotronics
Rotary Encoder	UCD-AC005-0413	Posital
Data Acquisition	USB-6356	Nat. Inst.
PC	OptiPlex 7050	Dell

5.1.3 Physical Parameters and Constants

To estimate the magnetic dipole moment m of the cylindrical magnets used (see Fig. 5.3), I performed an experiment similar to the one described in [85]. When the distance between the magnets is less than a critical value r_c, modeling the magnets as dipoles can lead to large errors since the dipole model does not accurately approximate the repulsive force between the magnets. This distance was estimated as $r_c = 0.035$ m (see Fig. 5.4). Additionally, in the region where $r > r_c$, the force curve, a function of scale r^{-4}, was fit to the curve to estimate the magnetic dipole moment as $m = 0.85$ Cm.

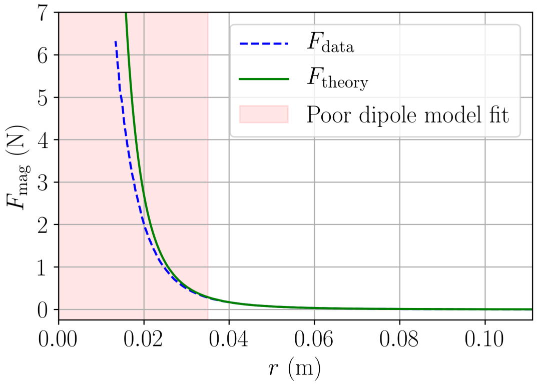


Figure 5.4: Measured repulsion force as a function of distance compared to theoretical force in Eq. (5.4) with $\theta = 0$. The theoretical force F_{theory} is based on dipole model with a dipole moment $m = 0.85$ cm, which was estimated using a curve fit to the region where the magnetic thickness $T \ll r$. Region of poor fit is marked for $r < 0.035$ m.

The other parameter values as well as their uncertainties (when applicable) are provided in Table 5.2, which are in reference to Fig. 5.1. Most of these parameters were either estimated using *SolidWorks* or by multiple direct measurements.

Table 5.2: Equation of motion parameters to simulated pendulum with associated uncertainty.

Parameter (units)	Value	Uncertainty ($\pm\sigma$)
d (m)	0.36	0.005
ℓ (m)	0.208	0.005
g (m/s^2)	9.81	-
M (kg)	0.1038	0.005
r_{cm} (m)	0.188	-
ω (rad/s)	3π	-
μ_0 (Cm)	1.257×10^{-6}	-
m (Cm)	0.85	-
μ_c (-)	0.002540	0.000020
μ_v (-)	0.000015	0.000003
μ_q (-)	0.000151	0.000020

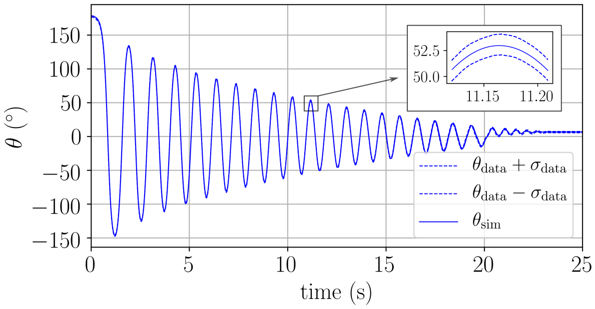


Figure 5.5: Free drop test between collect angular position data θ_{data} with encoder uncertainty σ_{data} and the simulated response θ_{sim}. As shown in the zoomed-in region, the simulated response is within the bounds of uncertainty of the actual response.

To validate the parameters, an experiment and simulation of a free drop of the pendulum are compared. The resulting angle $\theta(t)$ is shown in Fig. 5.5, which shows a very similar response between simulation and experiment. Additionally, the simulation is within the bounds of uncertainty of the encoder $\sigma_{\text{data}} = 1^\circ$ as shown in the zoomed in region of Fig. 5.5.

5.2 Teaspoon: A comprehensive python package for topological signal processing

Topological signal processing is a newly emerging field with an ever growing collection of tools. Using Topological Data Analysis (TDA) for signal processing allows for an analysis of the underlying shape of a time series. These methods are well backed by theory [177,203] and have shown success in numerous application areas including machining dynamics [111–114,116], finance [82,83], and gene expression [21,179].

Here I present the python package, teaspoon, that provides state-of-the-art topological signal processing tools as well as wrappers for available persistent homology software. While some TDA based packages exist for python (e.g. Scikit-TDA and Giotto-TDA), the teaspoon package specifically provides modules design to tackle questions related to signal processing and time series analysis from the viewpoint of topology. In comparison, other existing packages are designed for more general applications for TDA.

In the teaspoon package there are currently five main modules: dynamical systems, machine learning, complex networks, information, and parameter selection with several sub-modules for each as shown in Fig. 5.6. The dynamical systems library is currently hosting 60 dynamical systems including maps, flows, and collected data sets. The machine learning library contains code for numerous persistence diagram featuriztion and kernel methods. Specifically, this module includes the template function featurization methods described in [181,232] as well as persistence landscapes [28], persistence images [1], Carlsson coordinates [2], persistence paths and signature [43,44] and the multi-scale kernel method [196]. The complex networks module contains code to represent a time series as a network using ordinal partitions [146] or k nearest neighbors [118]. This module also provides several methods for calculating distances between nodes based on the adjacency matrix, which allows for the calculation of the persistent homology of the resulting networks. The information theory module implements entropy based functions for signal processing persistence diagram analysis. Lastly, the parameter selection module currently provides multiple algorithms for the automatic selection of the delay τ and dimension n parameters for state space reconstruction and permutation entropy.

In this work, I outline the features available in each module as well as features that will be added in the future. The goal of this package is to provide a range of topological signal processing tools in one unified framework. Additionally, for most of these modules, further documentation and examples of the functions are provided in the teaspoon documentation webpage[1].

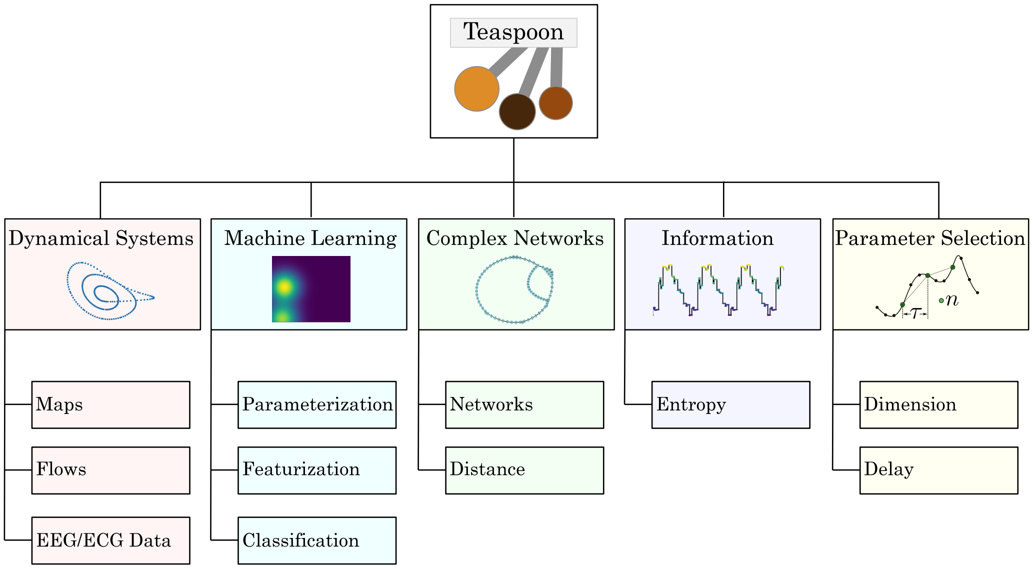


Figure 5.6: Tree structure of teaspoon.

5.2.1 Dynamical Systems Library (DynSysLib)

The dynamical systems library (DynSysLib) is a teaspoon module that provides a wide selection of dynamical system simulation models with many from [220]. Most of the available dynamical systems are able to exhibit both periodic and chaotic responses. In general, these systems can be separated into three categories: (1) flows, (2) Maps, and (3) Collected data. A full list of the available dynamical systems are provided in tables C.2 and C.3 of the appendix. The module has a single function *DynamicSystems*, which allows a wide range of the simulation control with the user being able to control as little as the system of interest and the desired dynamical state (chaotic or periodic) or provide detailed simulation parameters such as initial conditions, system parameters, and solution time. The function output is the resulting time series response. For details on the default parameters used, equations of motion, and examples, please see the teaspoon documentation webpage[1].

5.2.2 Machine Learning Module

In this section, I describe the machine learning module in teaspoon. Machine learning module provides automated feature matrix generation and classification, and it is suitable for the applications where persistence diagrams can be computed. There are three main files inside the module and these are $Base.py$, $feature\_functions.py$, and $PD\_Classification.py$. Here, I will explain the necessary functions in each of these files and show how to use these functions to perform machine learning using Topological Data Analysis (TDA).

Parameter Buckets The parameter bucket is a tool to hold all necessary parameters for the featurization functions as well as the classification algorithms. This includes parameters such as the classification algorithm, the size of the test set, as well as the desired persistence diagram featurization method. The parameter buckets are implemented as classes in the $Base.py$ file. The basic structure is implemented as a class $ParameterBucket$, however there are two more specialized

[1] http://elizabethmunch.com/code/teaspoon

classes, $InterPolyParameters$ and $TentParameters$ that are dedicated for parameters to the template functions introduced in Ref. [181]. These parameter buckets also have the functionality to use the template function featurization on localized regions of the persistence diagrams, using an adaptive partitioning method described in Ref. [232].

The rest of the parameter buckets are used for other featurization methods. The $LandscapesParameterBucket$ is for persistence landscapes [28], which requires an input for the landscape number that will be used to generate feature matrix. The $CL\_ParameterBucket$ is used to set parameters for classification using Persistence Images [1], Carlsoon Coordinates [2], persistence paths and signature [43, 44] and kernel method [196].

Featurization The file $feature\_functions.py$ contains functions that compute the topological features mentioned above. $F\_$ and $CL\_$ suffixes indicate that corresponding functions are designed for featurization and classification, respectively. First, for the template featurizations, there are two main functions, $tents$ and $interp\_polynomial$. These functions compute the collection of template functions based on a grid formed using parameters from the corresponding parameter buckets.

In addition to these, there is $PLandscape$ class that uses $PLandscapes$ function to compute the persistence landscapes for a given persistence diagram [28]. This class has an option to define L_number which returns specific landscapes in an array. Output of $PLandscapes$ is a dictionary that includes all landscapes, total number of landscapes and the desired landscapes if user defines L_number. $PLandscape$ class can also plot persistence landscapes. If user does not define the desired landscapes to plot, all landscapes will be plotted. $F\_Landscape$ uses persistence landscapes to compute feature matrix as explained in Ref. [147]. The inputs of the function are persistence landscapes, parameter bucket object that is explained in Sec. 5.2.2.

The second featurization method is persistence images. I utilized https://gitlab.com/csu-tda/PersistenceImagesP Images package to compute persistence images. F_Image takes persistence diagrams, pixel size, variance of the Gaussian distribution, the numbers of persistence diagrams whose image will be plotted, and transfer learning option. If transfer learning option is set to true, second set of persistence diagrams should be provided. Then, it will compute feature matrices for both sets of diagrams. Carlsson Coordinates is the third featurization method [2]. It has five coordinates that depend on birth and death times of persistence diagrams. $F\_CCoordinates$ takes persistence diagrams and computes these five features. It has second input, FN that defines how many feature will be computed. Feature vectors are generated using $\sum\limits_{i=1}^{FN} \binom{FN}{i}$ combinations of five coordinates. $F\_CCoordinates$ will return these feature vectors, number of combinations and combinations in a list.

Another featurization method is persistence path and signatures [43, 44]. $F\_PSignature$ function computes signatures on persistence landscapes. The first two levels of signatures are currently coded in the function. The inputs are persistence landscapes and the number of the landscape which will be used to compute the signatures. Then it returns the feature matrix to be used in the classification. Final featurization method is kernel method for persistence diagrams. $KernelMethod$ computes the kernel between given two persistence diagrams. It also has $sigma$ input which is a variable in the formula of the kernel given in Ref. [196]. After computing pairwise kernels between the diagrams, it can be used as pre-defined kernel in Support Vector Machine (SVM) algorithm for classification.

Classification Classification functions are embedded in $PD\_Classification$. Most of the functions take feature functions and parameter bucket object as input. They divide the given feature matrix into training set and test set with respect to test size defined in the parameter bucket. Classification can be performed using four classification algorithms: Support Vector Machine (SVM), Logistic Regression (LR), Random Forest (RF) and Gradient Boosting (GB). For the kernel method, LibSVM package [39] is utilized to insert pre-computed kernel matrix for classification. Additionally, the featurization methods can be used to create feature vectors compatible with any scikit-learn classification algorithm.

I also include the option of Transfer learning in classification for most of the featurization methods except kernel method. In this type of classification, a classifier is trained on a data set and tested on another one. One can refer to Ref. [171] for more details about transfer learning. When user defines the transfer learning as true in parameter bucket, feature functions will be computed for training and test persistence diagrams separately. In both classification type, training and test set will be generated 10 times randomly. Mean classification score, standard deviation for training and test set and total runtime for the classification are given as output.

5.2.3 Complex Networks Module

The teaspoon module provides the Python implementation of the algorithms used in [162], which provides methods for analyzing the dynamic state of a time series based on the persistent homology of the network representations of time series. The general pipeline, as shown in Fig. 5.7, is as follows: (1) represent a time series as a network as described in Section 5.2.3, (2) Generate a distance matrix from the undirected and weighted adjacency matrix as described in section 5.2.3, and (3) apply 1-D persistent homology to the distance matrix. The persistence diagram point summaries can be generated to analyze the dynamic state of the underlying time series.

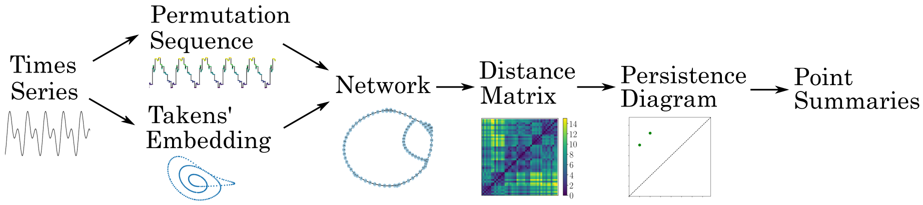


Figure 5.7: The persistent homology of complex networks pipeline.

Network Representations of Time Series There are currently two available algorithms in the complex networks module to represent a time series as a complex network. Specifically, these are k Nearest Neighbor (k-NN) networks [118] and ordinal partition networks [146]. For the implementation of these algorithms I use the adjacency matrix as the graph data structure.

For the ordinal partition network a permutation sequence needs to be generated by using the function *permutation_sequence*, which requires a time series and the permutation dimension

n and delay τ. For selecting the dimension and delay I suggest using the parameter selection module. Using the permutation sequence, the resulting adjacency matrix is formed using the *AdjacenyMatrix_OP* function, which creates edges in the graph based on permutation transitions.

Two steps are required to generate k-NN networks. First, the time series needs to have its state space reconstructed through Takens' embedding, which is done through the function *Takens_Embedding*. This function requires the time series and the embedding dimension and delay. The dimension and delay can be selected using the parameter selection module. Next, the k-NN are found using the *k_NN* function and specifying k which has a default of $k = 4$. Using the list of neighbors, an adjacency matrix is formed using the *Adjacency_KNN* function by treating each embedded vector as a node and adding edges when two nodes are k-NN.

The next step in the pipeline is to define algorithms to represent distances between nodes in the network based on the adjacency matrix, which is discussed in the subsequent section.

Distance Matrix Two steps are required to assign distances between nodes in a network: (1) apply an edge weight algorithm to represent distances for adjacent nodes and (2) implement a distance algorithm for non-adjacent nodes.

For the first step I provide the following edge weight functions: unweighted, inverse, and difference. Specifically, the unweighted option changes all the edge weights to 1, the inverse sets the weight to the element-wise reciprocal, and the difference finds the maximum edge weight and sets the new edge weight as the difference between the max edge weight and that edge's weight.

The second step requires a method for defining distances between non-adjacent nodes. To do this I offer two options: the shortest-path distance and effective network resistance [71]. Both of these steps are implemented through the *DistanceMatrix* function.

5.2.4 Information Module

The information theory module currently provides three functions for information entropy calculations. The first two are the calculation of the permutation entropy [14] and multi-scale permutation entropy as *PE* and *MsPE*, respectively. Permutation entropy has been shown to be a useful tool for analyzing signal complexity and has very few requirments for its application. The third function is the persistent entropy [9] through the function *PersistentEntropy*, which calculates the entropy of a persistence diagram given the lifetimes from the persistence diagram.

5.2.5 Parameter Selection Module

The parameter selection module provides code for the functions used in [161] and [11] for automatically calculating the dimension n and delay τ parameters for both permutation entropy and Takens' embedding (state space reconstruction). For details on each of the methods please reference their respective publications as some are more suitable for non-linear time series or have time series requirements. A comprehensive list of the available methods are provided in Table C.4.

APPENDICES

APPENDIX A

PERMUTATION ENTROPY PARAMETER SELECTION

A.1 MPE Effects of Noise

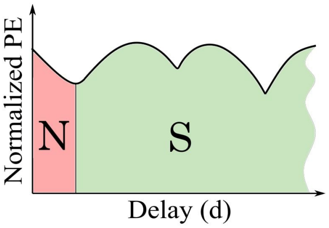


Figure A.1: Region N is affected by noise in the MPE plot, and region S is unaffected.

Effects of Noise We found that the main advantage of using MPE for determining the embedding delay is its robustness to noise. Noise on an MPE plot has minimal effects on regions B and C from Fig. 2.11, while only significantly affecting region A as shown in Fig A.1. Furthermore, depending on the signal to noise ratio, there will only be an effect at the beginning of region A. Figure A.1 shows the first region N where noise is affecting the permutation entropy. The effect of noise causes the MPE plot to start at a maxima and decrease to a local minima. When the time delay becomes large enough, the permutations are no longer influenced by the noise causing this minima. We found that the location of the minima is based on the condition

$$m_{\text{avg}}\tau_N \approx A_{\text{noise}} f_s, \tag{A.1}$$

where m_{avg} is the average of the absolute value of the slope and A_{noise} is approximately the maximum amplitude of the noise, τ_N is the value of τ great enough to surpass the noise amplitude. We derived this condition from the need for, on average, $|f(t) - f(t+\tau)| > A_{\text{noise}}$. This shows that MPE is robust to noise as long as the noise amplitude does not exceed the amplitude of the signal.

A.2 Autocorrelation Methods and Example

Pearson Correlation The Pearson correlation coefficient $\rho_{xy} \in [-1, 1]$ measures the linear correlation of two time series x and y. Using these two data sets the correlation coefficient is calculated as

$$\rho_{xy} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^{n} (y_i - \bar{y})^2}}. \tag{A.2}$$

The possible values of ρ_{xy} represent the relationship between the two data sets, where $\rho_{xy} = 1$ represents a perfect positive linear correlation, $\rho_{xy} = 0$ represents no linear correlation, while $\rho_{xy} = -1$ represents a perfect negative linear correlation. However, Pearson correlation is limited because it only detects linear correlations. This limitation is somewhat alleviated by using Spearman's Correlation which operates on the ordinal ranking of the two time series instead of their numeric values.

Spearman's Correlation Spearman's correlation is also calculated using Eq. (A.2) with the substitution of x and y for their ordinal ranking. This substitution allows for detecting nonlinear correlation trends to be represented as long as the correlation is monotonic. To demonstrate the difference, Fig. A.2 shows two sequences x and y calculated from $y = x^4$ with $x \in [0, 10]$. Using this example, the Pearson correlation is calculated as $\rho \approx 0.86$, while Spearman's ranked correlation yields $\rho = 1.0$. This result demonstrates how Spearman's correlation coefficient accurately detects the non-linear, monotonic correlation between x and y whereas Pearson correlation may miss it.

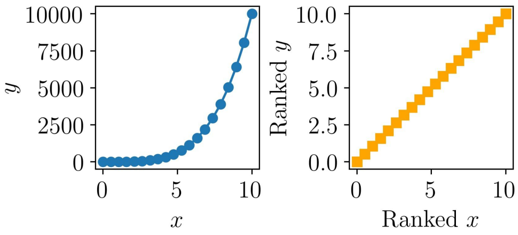


Figure A.2: A comparison between (left) unranked values and (right) ranked values for calculating correlation coefficients. Using the ranked x and y, Spearman's correlation coefficient can be used to accurately reveal existing nonlinear monotonic correlations.

Autocorrelation Example We can use the concept of correlation to select a delay τ by calculating the correlation coefficient using Eq. (A.2) between a time series and its τ-lagged version. As an example, take the time series $x(t) = \sin(2\pi t)$, with $t \in [0, 5]$ having a sampling frequency of 100 Hz. This results in a suggested delay $\tau = 20$ at the first folding time using both Spearman's and Pearson correlation.

A.3 MI methods

MI using Equal-sized Partitions For the calculation of MI, the joint and independent probabilities of the original $x(t)$ and time lagged $x(t+\tau)$ time series are needed. However, since x is a discrete time series, we approximate these probabilities using bins, which segment the range of the series into discrete groups. The simplest method for approximating the probabilities using this discretization method is to use equal sized bins. However, the size of these bins is dependent on the number of bins k. We investigated various methods for estimating an appropriate number of bins using the length of the time series N. These methods include the common square-root choice $k = \lceil\sqrt{N}\rceil$, Sturge's formula [223] $k = \lceil\log_2(N)\rceil + 1$, and Rice Rule [126] $k = \lceil 2N^{1/3}\rceil$. After comparing each method using a variety of examples, we found that the use of Sturge's formula provided the best results for selecting τ for PE using MI.

MI using Adaptive Partitions Darbellay and Vajda [55] introduced a multistep, adaptive partitioning scheme to select appropriate binning sizes in the observation space formed by the plane $x(t)$ and $x(t+\tau)$. Their method is often considered state-of-the-art for estimating the mutual information function [119]. In this approach, the bins are recursively created where in the first function call, the space of the signal and its τ-lagged version is divided into an equal number of 2D bins. Then a A chi-squared test is used to test the null hypothesis that the data within the newly created bins are independent. Any segment that fails the test is further divided until the resulting sub-segments contain independent data (or a certain number of divisions is satisfied). Using this partitioning method, the MI is calculated using Eq. (2.14).

Kraskov MI Kraskov et al. [119] developed a method for approximating the MI using entropy estimates using partition sizes based on k-nearest neighbors. Specifically, the method begins by first calculating the MI using entropy [52] as

$$I(X;Y) = H(X) + H(Y) - H(X,Y), \tag{A.3}$$

where H is the Shannon entropy. Next, an approximation of $H(X)$ with digamma functions is done, but the probability density of X and Y still needs to be estimated. To do this, adaptive partitions using the k-nearest neighbor are formed. Specifically Kraskov et al. develop two different partitioning methods with similar results. The first method uses the maximum Chebyshev distance to the $k = 1$ nearest neighbor j to form square bins as shown in Fig. A.3-a, and the second method in Fig. A.3-b uses rectangular partitions using the horizontal and vertical distances to the $k = 1$ nearest neighbor j. To continue with the example shown in Fig. A.3, the density probability is estimated using the strips formed from these bins. To highlight the difference, Fig. A.3-a shows a horizontal strip of width $\epsilon(i)$ encapsulating $n_x(i) = 2$ points (strip does not include the point i), while in Fig. A.3-b only $n_x(i) = 1$ point is enclosed. Using these probability density approximations and the digamma function ψ, MI between X and Y can be estimated. Using the partitioning method shown in Fig. A.3-a the MI is estimated as

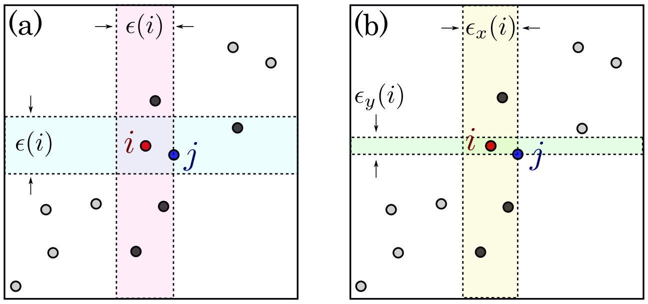


Figure A.3: Example showing two different partition methods for Mutual Information estimation using $k = 1$ nearest neighbor adaptive partitioning.

$$I^{(1)}(X;Y) = \psi(k) - (\psi(n_x + 1) + \psi(n_y + 1) + \psi(N). \tag{A.4}$$

Using the partitioning method shown in Fig. A.3-b the MI is estimated as

$$I^{(2)}(X;Y) = \psi(k) - 1/k - [\psi(n_x) + \psi(n_y)] + \psi(N). \tag{A.5}$$

A.4 Tabulated PE parameters

Table A.1: A comparison between the calculated and suggested values for the delay parameter τ for multiple MI approximation methods. The cells in bold highlight the methods that yielded the closest match to the suggested delay. The equal-sized partition method is described in Section A.3, Kraskov et al. methods 1 and 2 in Section A.3, and the adaptive partitioning approach in Section A.3.

System	**Mutual Information**				**Suggested Delay tau**	**Ref.**
	Equal-sized Partitions	**Kraskov et al. Method 1**	**Kraskov et al. Method 2**	**Adaptive Partitions**		
White Noise	**1**	3	3	**1**	1	[201]
Lorenz	13	**9**	**9**	**9**	10	[201]
Rossler	14	13	**11**	**9**	9	[227]
Bi-directional Rossler	**16**	**14**	**14**	**15**	15	[201]
Mackey-Glass	**7**	**8**	**7**	**7**	1 to 700	[201]
Sine Wave	4	**17**	**13**	1	15	[227]
Logistic Map	**5**	8	11	**5**	1 to 5	[201]
Henon Map	12	15	13	8	1 to 5	[201]
ECG	22	16	9	8	1 to 4	[201]
EEG	6	5	5	5	1 to 3	[201]

Table A.2: A comparison between the calculated and suggested values for the delay parameter τ. The cells in bold show the methods that yielded the closest match to the suggested delay. The following conditions or abbreviations were used in the table: the range under PAMI results is from using the range $(4 < n < 6)$, AP under MI is an abbreviation for adaptive partitioning, and AC is an abbreviation for autocorrelation.

Catagory	**System**	**Traditional Methods**		**Modified/Proposed Methods**			**Suggested Delay (τ)**	**Ref.**
		MI using AP	**Spearman's AC**	**Freq. App.**	**MPE**	**PAMI ($4 \leq n \leq 6$)**		
Noise	White Noise	1	**1**	**1**	**1**	**1**	1	[201]
Chaotic Differential Equation	Lorenz	**9**	15	6	17	**5 to 9**	10	[201]
	Rossler	**9**	12	**7**	19	**6 to 10**	9	[227]
	Bi-directional Rossler	**15**	**12**	7	20	6 to 10	15	[201]
	Mackey-Glass	**7**	**5**	**3**	**8**	**2 to 4**	1 to 700	[201]
Periodic	Sine Wave	1	**10**	**21**	**16**	5 to 8	15	[227]
Nonlinear Difference Eq.	Logistic Map	**5**	**1**	**1**	**1**	**1**	1 to 5	[201]
	Henon Map	8	**1**	**1**	**1**	**1**	1 to 5	[201]
Medical Data	ECG	8	21	**2**	13	**1 to 2**	1 to 4	[201]
	EEG	**5**	**4**	**1**	**4**	**2 to 4**	1 to 3	[201]

Table A.3: A comparison between the calculated and suggested values for the embedding dimension n. The cells in bold show the methods that yielded the closest match to the suggested dimension.

Catagory	**System**	**Traditional Methods**		**Modified Method**	**Suggested Dim. (n)**	**Ref.**
		FNN	**SSA**	**MPE**		
Noise	White Noise	**4**	23	**5**	3 to 7	[201]
Chaotic Differential Equation	Lorenz	3	4	**5**	5 to 7	[201]
	Rossler	4	4	4	6	[227]
	Bi-directional Rossler	4	4	4	6 to 7	[201]
	Mackey-Glass	**4**	**6**	**4**	4 to 8	[201]
Periodic	Sine Wave	**4**	2	3	4	[227]
Nonlinear Difference Equation	Logistic Map	**4**	**3**	**5**	2 to 16	[201]
	Henon Map	**4**	2	**5**	3 to 10	[201]
Medical Data	ECG	**7**	8	**5**	3 to 7	[201]
	EEG	**5**	11	**6**	3 to 7	[201]

APPENDIX B

SUBLEVEL SET PERSISTENCE AND DAMPING PARAMETER ESTIMATION

In appendix B, we provide an omitted proof and algorithm. Specifically, we have included the theorem showing the relationship between the mean lifetime and mean birth and death times of a persistence diagram and the algorithm for calculating the sublevel set persistence diagram.

B.1 Proof of Expected Lifetime Equation

The following proof supports a claim made in 1.2.1. In what follows, we will use the notation μ_S to denote the expected value of the distribution over the multi-set S.

Theorem B.1.1 (Expected Lifetime)**.** *Let $\mathcal{D} = \{(b_i, d_i)\}_{i=1}^n$ be a persistence diagram. Let B, D, and L be the multi-sets of birth times, death times, and lifetimes, respectively. Then, the average lifetime is:*

$$\mu_L = \mu_D - \mu_B.$$

Proof. By definition, $B = \{b_i\}_{i=1}^n$, $D = \{d_i\}_{i=1}^n$, and $L = \{d_i - b_i\}_{i=1}^n$. By definition of mean and of L, the mean lifetime is

$$\mu_L = \frac{1}{n}\sum_{i=1}^n (d_i - b_i). \tag{B.1}$$

Expanding the sum to two separate sums and using the commutative property of addition, we get:

$$\begin{aligned} \mu_L &= \frac{1}{n}\sum_{i=1}^n (d_i - b_i) \\ &= \frac{1}{n}\sum_{i=1}^n d_i - \frac{1}{n}\sum_{i=1}^n b_i \\ &= \mu_D - \mu_B, \end{aligned} \tag{B.2}$$

where the last equality is by definition of μ_D and μ_B. Thus, we conclude that $\mu_L = \mu_D - \mu_B$. □

APPENDIX C

DYNAMICAL SYSTEMS

C.1 Dynamic State Analysis System Models

The following 18 continuous and 12 discrete dynamical systems were used throughout this work. For details on their equations of motion and system parameters we direct the reader to the MakeData module in the python package `teaspoon` [161].

Table C.1: Continuous and discrete dynamical Systems used throughout manuscript.

Autonomous Continuous Dynamical Systems	**Driven Continuous Dynamical Systems**	**Discrete Dynamical Systems**
Lorenz	Driven Van der Pol Oscillator	Logistic Map
Rossler	Shaw Van der Pol Oscillator	Henon Map
Double Pendulum	Forced Brusselator	Sine Map
Diffusionless Lorenz Attractor	Ueda Oscillator	Tent Map
Complex Butterfly	Duffing Van der Pol Oscillator	Ricker's Population Map
Chen's System	Base Excited Magnetic Pendulum	Gauss Map
ACT Attractor		Sine Circle Map
Rabinovich Frabrikant Attractor		Lozi Map
Halvorsen's Cyclically Symmetric Attractor		Tinkerbell Map
Burke Shaw Attractor		Holmes Cubic Map
Rucklidge Attractor		Kaplan-Yorke Map
WINDMI		Gingerbread Man Map

C.2 All Available Dynamic System Models

Table C.2: Available flows and maps in dynamic systems library module.

Dissipative Flows	Conservative Flows	Driven Dissipative Flows	Maps
Lorenz Att.	Simple Driven	Driven Pendulum	Logistic
Rossler Att.	Nose-Hoover Osc.	Driven Van der Pol Osc.	Henon
Chua Circuit	Labyrinth Chaos	Shaw Van der Pol Osc.	Sine
Coupled Lorenz-Rossler	Henon-Heiles Osc.	Forced Brusselator	Tent
Coupled Rossler-Rossler		Ueda Osc.	Linear Congruent
Double Pendulum		Duffing's Two-well Osc.	Ricker's Pop.
Diffusionless Lorenz Att.		Duffing Van der Pol Osc.	Gauss
Complex Butterfly		Rayleigh-Duffing Osc.	Cusp
Chen's Att.			Pincher's
Hadley Att.			Sine-circle
ACT Att.			Lozi
Rabinovich-Fabrikant Att.			Delayed Logistic
Rigid Body Feedback			Tinkerbell
Moore-Spiegel Osc.			Burgers
Thomas Att.			Holmes
Halvorsen's Att.			Kaplan-Yorke
Burke-Shaw Att.			
Rucklidge Att.			
WINDMI			
Simple Quadratic Flow			
Simple Cubic Flow			
Simple Piecewise Flow			
Double Scroll			

Table C.3: Available functions, noise models, and medical data in dynamical systems library module.

Functions	Noise Models	Medical Data
Sine	Gaussian	Electrocardiogram
Incommensurate Sine	Uniform	Electroencephalogram
	Rayleigh	

Table C.4: Parameter selection methods available in parameter selection module for both the delay and dimension parameters.

Algorithm	Reference(s)	Dimension or Delay
Mutual Information	[78, 161]	Delay
Autocorrelation	[25, 161]	Delay
Frequency Analysis	[11, 149, 161]	Delay
Multi-scale Permutation Entropy	[161, 200]	Delay
Permutation Auto-mutual Information	[135, 161]	Delay
SW1PerS	[11, 177]	Delay
False Nearest Neighbors	[109, 161]	Dimension
Multi-scale Permutation Entropy	[161, 200]	Dimension
Singular Spectrum Analysis	[27, 161]	Dimension

APPENDIX D

ADDITIONAL DIFFUSION DISTANCE ANALYSIS

D.1 Persistence of Cycle Graph

The cycle graph on n vertices is the graph $G = (V, E)$ with $V = \{v_1, \cdots, v_n\}$, and $E = \{v_i v_{i+1} \mid 1 \leq i < n\} \cup \{v_n v_1\}$; i.e. it forms a closed path (cycle) where no repetitions occur except for the starting and ending vertices. If we increase the number of nodes from 2 to 500 and calculate the maximum persistence or maximum lifetime, we find that it quickly reaches a maximum of $L_1 = 0.216$ at $n = 32$, and then steadily declines seeming to approach a plateau as shown in Fig. D.1. This is in comparison to the unweighted shortest path distance of the cycle graph which has a maximum persistence of $\lceil n/3 \rceil - 1$ as shown in [162].

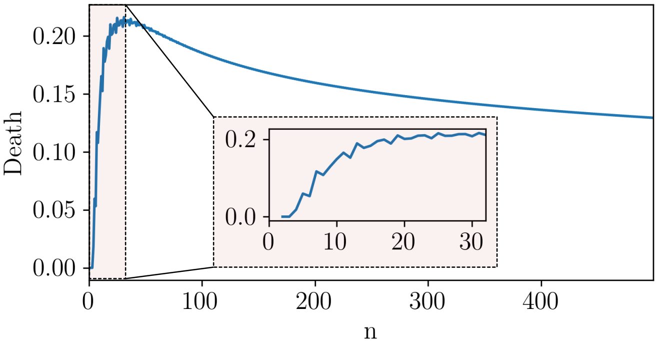


Figure D.1: Numerical analysis of the maximum persistence of the cycle graph $G_{\text{cycle}}(n)$ with size n when using diffusion distance with $t = 2d$.

D.2 Analysis on Random Walk Steps

In this section we vary the number of random walk steps t with respect to the graph diameter d to determine how many steps is suitable for calculating the persistent homology based on the diffusion distance. We vary t/d from 1 to 5 as shown in Fig. D.2. To decide on the optimal t we calculate the maximum lifetime and number of persistence pairs in each resulting persistence diagram for each of the 23 dynamical systems investigated in this work. Additionally, the average for both the maximum lifetime and number of lifetimes is plotted as shown in Fig. D.2.

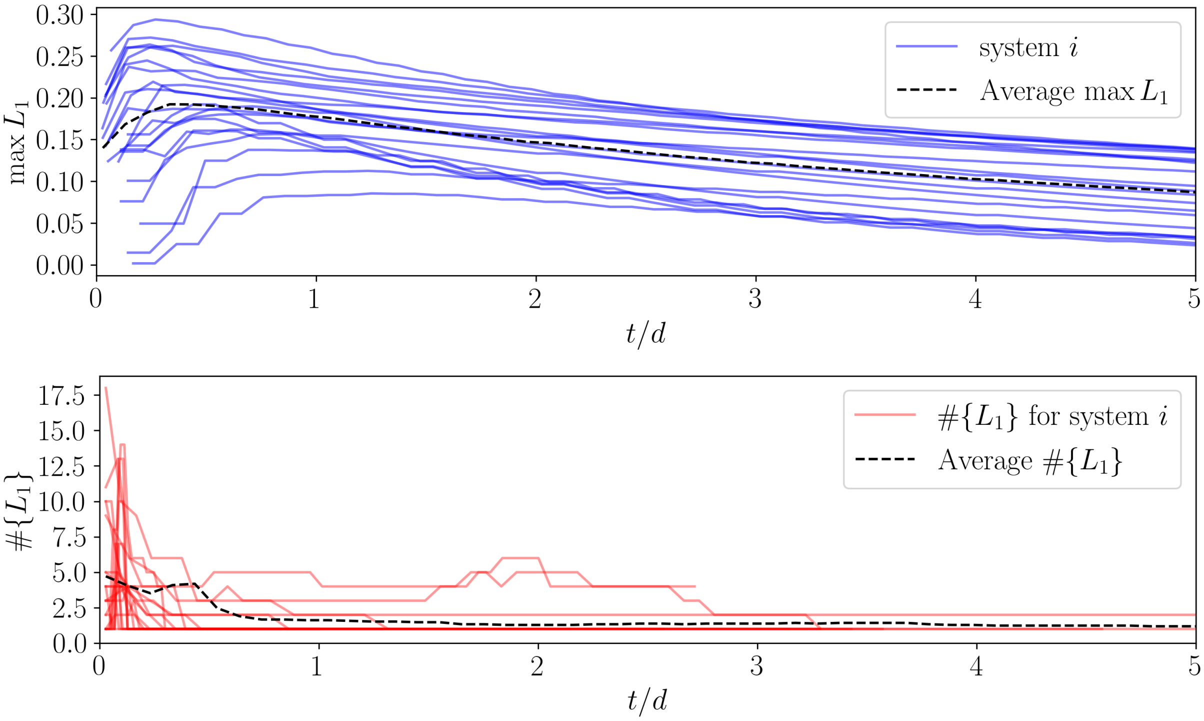


Figure D.2: Comparison of max L_1 and $\#\{L_1\}$ for each system and mean when varying t in P^t with respect to the diameter ($t \in [d, 5d]$).

Based on the each systems maximum lifetimes, a suitable value for t should be greater than d based on having a t large enough that each system reaches a maximum of the $\max(L_1)$. We can also note that the number of persistence pairs or lifetimes in the persistence diagram does not stabalize for the majority of systems until approximately $t = 2d/3$. This again supports a minimum suggest $t > d$. The only downfall of larger values of t is that the maximum lifetime tends to diminish as shown in the $\max(L_1)$ figure. Therefor, we conclude that a suitable t should be within the range $d < t < 3d$. In this work we chose $t = 2d$.

BIBLIOGRAPHY

REFERENCES

[1] Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18(8):1–35, 2017.

[2] Aaron Adcock, Erik Carlsson, and Gunnar Carlsson. The ring of algebraic functions on persistence bar codes. *Homology, Homotopy and Applications*, 18(1):381–402, 2016.

[3] Robert J. Adler, Omer Bobrowski, Matthew S. Borman, Eliran Subag, and Shmuel Weinberger. Persistent homology for random fields and complexes. In *Institute of Mathematical Statistics Collections*, pages 124–143. Institute of Mathematical Statistics, 2010.

[4] Robert J. Adler, Omer Bobrowski, and Shmuel Weinberger. Crackle: The homology of noise. *Discrete & Computational Geometry*, 52(4):680–704, aug 2014.

[5] Mehran Ahmadlou and Hojjat Adeli. Visibility graph similarity: A new measure of generalized synchronization in coupled dynamic systems. *Physica D: Nonlinear Phenomena*, 241(4):326–332, feb 2012.

[6] José M. Amigó, Roberto Monetti, Thomas Aschenbrenner, and Wolfram Bunk. Transcripts: An algebraic approach to coupled time series. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(1):013105, mar 2012.

[7] Ralph G. Andrzejak, Klaus Lehnertz, Florian Mormann, Christoph Rieke, Peter David, and Christian E. Elger. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. *Physical Review E*, 64(6), nov 2001.

[8] Nakhlé H Asmar. *Partial differential equations with Fourier series and boundary value problems*. Courier Dover Publications, 2016.

[9] N. Atienza, L. M. Escudero, M. J. Jimenez, and M. Soriano-Trigueros. Persistent entropy: a scale-invariant topological statistic for analyzing cell arrangements.

[10] Nieves Atienza, Rocio Gonzalez-Diaz, and Matteo Rucco. Persistent entropy for separating topological features from noise in vietoris-rips complexes. *Journal of Intelligent Information Systems*, 52(3):637–655, jul 2017.

[11] Brittany T. Fasy Audun D. Myers, Firas A. Khasawneh. Separating persistent homology of noise from time series data using topological signal processing. *arXiv:2012.04039 [math.AT]*, 2020.

[12] Onur Avci, Osama Abdeljaber, Serkan Kiranyaz, Mohammed Hussein, Moncef Gabbouj, and Daniel J. Inman. A review of vibration-based damage detection in civil structures: From traditional methods to machine learning and deep learning applications. *Mechanical Systems and Signal Processing*, 147:107077, jan 2021.

[13] Massoud Babaie-Zadeh and Christian Jutten. A general approach for mutual information minimization and its application to blind source separation. *Signal Processing*, 85(5):975–995, may 2005.

[14] Christoph Bandt and Bernd Pompe. Permutation entropy: A natural complexity measure for time series. *Physical Review Letters*, 88(17), apr 2002.

[15] Christoph Bandt and Bernd Pompe. Permutation entropy: a natural complexity measure for time series. *Physical review letters*, 88(17):174102, 2002.

[16] Aurelio F. Bariviera, Luciano Zunino, and Osvaldo A. Rosso. An analysis of high-frequency cryptocurrencies prices dynamics using permutation-information-theory quantifiers. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(7):075511, jul 2018.

[17] Hannah Bast, Daniel Delling, Andrew Goldberg, Matthias Müller-Hannemann, Thomas Pajor, Peter Sanders, Dorothea Wagner, and Renato F. Werneck. Route planning in transportation networks.

[18] Pierre Baudot and Daniel Bennequin. Topological forms of information. AIP Publishing LLC, 2015.

[19] Giancarlo Benettin, Luigi Galgani, Antonio Giorgilli, and Jean-Marie Strelcyn. Lyapunov characteristic exponents for smooth dynamical systems and for hamiltonian systems: A method for computing all of them. part 2: Numerical application. *Meccanica*, 15(1):21–30, mar 1980.

[20] T. Berry, J. R. Cressman, Z. Greguric-Ferenček, and T. Sauer. Time-scale separation from diffusion-mapped delay coordinates. *SIAM Journal on Applied Dynamical Systems*, 12(2):618–649, jan 2013.

[21] Jesse Berwald and Marian Gidea. Critical transitions in a model of a genetic regulatory system. *Mathematical Biosciences & Engineering*, 11(4):723–740, 2014.

[22] A Block, W Von Bloh, and HJ Schellnhuber. Efficient box-counting determination of generalized fractal dimensions. *Physical Review A*, 42(4):1869, 1990.

[23] Stephen P. Borgatti. Centrality and network flow. *Social Networks*, 27(1):55–71, jan 2005.

[24] L. Borkowski and A. Stefanski. FFT bifurcation analysis of routes to chaos via quasiperiodic solutions. *Mathematical Problems in Engineering*, 2015:1–9, 2015.

[25] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[26] David S Broomhead and Gregory P King. Extracting qualitative dynamics from experimental data. *Physica D: Nonlinear Phenomena*, 20(2-3):217–236, 1986.

[27] David S Broomhead and Gregory P King. Extracting qualitative dynamics from experimental data. *Physica D: Nonlinear Phenomena*, 20(2-3):217–236, 1986.

[28] Peter Bubenik. Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research*, 16:77–102, 2015.

[29] John Butterworth, Jin Hee Lee, and Barry Davidson. Experimental determination of modal damping from full scale testing. In *13th world conference on earthquake engineering*, volume 310, pages 1–15, 2004.

[30] Th Buzug and G Pfister. Optimal delay time and embedding dimension for delay-time coordinates by analysis of the global static and local dynamical behavior of strange attractors. *Physical review A*, 45(10):7073, 1992.

[31] Andriana S. L. O. Campanharo, M. Irmak Sirer, R. Dean Malmgren, Fernando M. Ramos, and Luís A. Nunes. Amaral. Duality between time series and networks. *PLoS ONE*, 6(8):e23378, aug 2011.

[32] M S Cao, G G Sha, Y F Gao, and W Ostachowicz. Structural damage identification using damping: a compendium of uses and features. *Smart Materials and Structures*, 26(4):043001, mar 2017.

[33] Yinhe Cao, Wen-wen Tung, JB Gao, Vladimir A Protopopescu, and Lee M Hively. Detecting dynamical changes in time series using the permutation entropy. *Physical review E*, 70(4):046217, 2004.

[34] Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, January 2009. Survey.

[35] Gunnar Carlsson and Vin de Silva. Zigzag persistence. *Foundations of Computational Mathematics*, 10(4):367–405, apr 2010.

[36] Gunnar Carlsson, Vin de Silva, and Dmitriy Morozov. Zigzag persistent homology and real-valued functions. In *Proceedings of the 25th annual symposium on Computational geometry*. ACM Press, 2009.

[37] Gunnar Carlsson, Jackson Gorham, Matthew Kahle, and Jeremy Mason. Computational topology for configuration spaces of hard disks. *Physical Review E*, 85(1), jan 2012.

[38] M. J. Casiano. Extracting damping ratio from dynamic data and numerical solutions. *Nasa Technical Reports*, 2016.

[39] Chih-Chung Chang and Chih-Jen Lin. LIBSVM. *ACM Transactions on Intelligent Systems and Technology*, 2(3):1–27, apr 2011.

[40] Frédéric Chazal, Brittany Terese Fasy, Fabrizio Lecci, Alessandro Rinaldo, Aarti Singh, and Larry Wasserman. On the bootstrap for persistence diagrams and landscapes. *arXiv preprint arXiv:1311.0376*, 2013.

[41] Xiaoying Chen, Chong Zhang, Bin Ge, and Weidong Xiao. Temporal query processing in social network. *Journal of Intelligent Information Systems*, 49(2):147–166, dec 2016.

[42] Yang Chen, Harish Chintakunta, Le Xie, Yuliy M. Baryshnikov, and P. R. Kumar. Persistent-homology-based detection of power system low-frequency oscillations using PMUs. In *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, dec 2016.

[43] Ilya Chevyrev and Andrey Kormilitzin. A Primer on the Signature Method in Machine Learning. 2016.

[44] Ilya Chevyrev, Vidit Nanda, and Harald Oberhauser. Persistence paths and signature features in topological data analysis.

[45] Harish Chintakunta, Thanos Gentimis, Rocio Gonzalez-Diaz, Maria-Jose Jimenez, and Hamid Krim. An entropy-based persistence barcode. *Pattern Recognition*, 48(2):391–401, feb 2015.

[46] S. Chowdhury and F. Mémoli. Convergence of hierarchical clustering and persistent homology methods on directed networks. *ArXiv*, abs/1711.04211, 2017.

[47] Yu-Min Chung, Chuan-Shen Hu, Yu-Lun Lo, and Hau-Tieng Wu. A persistent homology approach to heart rate variability analysis with an application to sleep-wake classification. *arXiv preprint arXiv:1908.06856*, 2019.

[48] Septima Poinsette Clark. Estimating the fractal dimension of chaotic time series. *Lincoln Laboratory Journal*, 3(1), 1990.

[49] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, dec 2006.

[50] Ronald R. Coifman and Stéphane Lafon. Diffusion maps. 21(1):5–30, jul 2006.

[51] Madalena Costa, Ary L Goldberger, and C-K Peng. Multiscale entropy analysis of complex physiologic time series. *Physical review letters*, 89(6):068102, 2002.

[52] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[53] Joseph Crawford and Tijana Milenković. ClueNet: Clustering a temporal network based on topological similarity rather than denseness. *PLOS ONE*, 13(5):e0195993, may 2018.

[54] S. Czesla, T. Molle, and J. H. M. M. Schmitt. A posteriori noise estimation in variable data sets. *Astronomy & Astrophysics*, 609:A39, jan 2018.

[55] G.A. Darbellay and I. Vajda. Estimation of the information by an adaptive partitioning of the observation space. *IEEE Transactions on Information Theory*, 45(4):1315–1321, may 1999.

[56] Bin Ran David Boyce. *Modeling Dynamic Transportation Networks*. Springer Berlin Heidelberg, 2012.

[57] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 2008.

[58] Luciana De Micco, Juana Graciela Fernández, Hilda A Larrondo, Angelo Plastino, and Osvaldo A Rosso. Sampling period, statistical complexity, and chaotic attractors. *Physica A: Statistical Mechanics and its Applications*, 391(8):2564–2575, 2012.

[59] Luciana De Micco, Juana Graciela Fernández, Hilda A Larrondo, Angelo Plastino, and Osvaldo A Rosso. Sampling period, statistical complexity, and chaotic attractors. *Physica A: Statistical Mechanics and its Applications*, 391(8):2564–2575, 2012.

[60] Cecil Jose A. Delfinado and Herbert Edelsbrunner. An incremental algorithm for Betti numbers of simplicial complexes on the 3-sphere. *Computer Aided Geometric Design*, 12(7):771–784, 1995.

[61] Alfonso Delgado-Bonal and Alexander Marshak. Approximate entropy and sample entropy: A comprehensive tutorial. *Entropy*, 21(6):541, may 2019.

[62] Bin Deng, Li Liang, Shunan Li, Ruofan Wang, Haitao Yu, Jiang Wang, and Xile Wei. Complexity extraction of electroencephalograms in alzheimers disease with weighted-permutation entropy. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(4):043105, apr 2015.

[63] Varad Deshmukh, Elizabeth Bradley, Joshua Garland, and James D. Meiss. Using curvature to select the time lag for delay reconstruction. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(6):063143, jun 2020.

[64] T. Detroux, L. Renson, L. Masset, and G. Kerschen. The harmonic balance method for bifurcation analysis of large-scale nonlinear mechanical systems. *Computer Methods in Applied Mechanics and Engineering*, 296:18–38, nov 2015.

[65] Tamal K Dey and Yusu Wang. *Computational Topology for Data Analysis*. Cambridge University Press, 2021.

[66] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[67] Andrey Dmitriev, Victor Dmitriev, Oleg Sagaydak, and Olga Tsukanova. The application of stochastic bifurcation theory to the early detection of economic bubbles. *Procedia Computer Science*, 122:354–361, 2017.

[68] Reik V Donner, Yong Zou, Jonathan F Donges, Norbert Marwan, and Jürgen Kurths. Recurrence networks—a novel paradigm for nonlinear time series analysis. *New Journal of Physics*, 12(3):033025, mar 2010.

[69] Herbert Edelsbrunner and John Harer. Persistent homology-a survey. *Contemporary mathematics*, 453:257–282, 2008.

[70] Herbert Edelsbrunner and John Harer. *Computational Topology - an Introduction.* American Mathematical Society, 2010.

[71] W. Ellens, F.M. Spieksma, P. Van Mieghem, A. Jamakovic, and R.E. Kooij. Effective graph resistance. *Linear Algebra and its Applications*, 435(10):2491–2506, nov 2011.

[72] Jessica Enright and Rowland Raymond Kao. Epidemics on dynamic networks. *Epidemics*, 24:88–97, sep 2018.

[73] Brittany Terese Fasy, Fabrizio Lecci, Alessandro Rinaldo, Larry Wasserman, Sivaraman Balakrishnan, Aarti Singh, et al. Confidence sets for persistence diagrams. *The Annals of Statistics*, 42(6):2301–2339, 2014.

[74] Temple H. Fay. Coulomb damping. *International Journal of Mathematical Education in Science and Technology*, 43(7):923–936, oct 2012.

[75] Temple H. Fay. Quadratic damping. *International Journal of Mathematical Education in Science and Technology*, 43(6):789–803, sep 2012.

[76] Birgit Frank, Bernd Pompe, Uwe Schneider, and Dirk Hoyer. Permutation entropy improves fetal behavioural state classification based on heart rate analysis from biomagnetic recordings in near term fetuses. *Medical and Biological Engineering and Computing*, 44(3):179, 2006.

[77] Andrew M Fraser and Harry L Swinney. Independent coordinates for strange attractors from mutual information. *Physical review A*, 33(2):1134, 1986.

[78] Andrew M. Fraser and Harry L. Swinney. Independent coordinates for strange attractors from mutual information. *Physical Review A*, 33(2):1134–1140, feb 1986.

[79] Riccardo Gallotti and Marc Barthelemy. The multilayer temporal network of public transport in great britain. *Scientific Data*, 2(1), jan 2015.

[80] Joshua Garland, Tyler Jones, Michael Neuder, Valerie Morris, James White, and Elizabeth Bradley. Anomaly detection in paleoclimate records using permutation entropy. *Entropy*, 20(12):931, 2018.

[81] Joshua Garland, Tyler R Jones, Elizabeth Bradley, Michael Neuder, and James WC White. Climate entropy production recorded in a deep antarctic ice core. *arXiv preprint arXiv:1806.10936*, 2018.

[82] Marian Gidea. Topological data analysis of critical transitions in financial networks. In Puzis R. Shmueli E., Barzel B., editor, *3rd International Winter School and Conference on Network Science NetSci-X 2017*, Springer Proceedings in Complexity. Springer, Cham, 2017.

[83] Marian Gidea and Yuri Katz. Topological data analysis of financial time series: Landscapes of crashes. *Physica A: Statistical Mechanics and its Applications*, 491:820–834, 2018.

[84] C. Gontier, M. Smail, and P.E. Gautier. A time domain method for the identification of dynamic parameters of structures. *Mechanical Systems and Signal Processing*, 7(1):45–56, jan 1993.

[85] Manuel I González. Forces between permanent magnets: experiments and model. *European Journal of Physics*, 38(2):025202, dec 2016.

[86] Peter Grassberger and Itamar Procaccia. Measuring the strangeness of strange attractors. *Physica D: Nonlinear Phenomena*, 9(1-2):189–208, 1983.

[87] Aixia Guo, Bettina F. Drake, Yosef M. Khan, James R. Langabeer II, and Randi E. Foraker. Time-series cardiovascular risk factors and receipt of screening for breast, cervical, and colon cancer: The guideline advantage. *PLOS ONE*, 15(8):e0236836, aug 2020.

[88] T. C. Gupta. Identification and experimental validation of damping ratios of different human body segments through anthropometric vibratory model in standing posture. *Journal of Biomechanical Engineering*, 129(4):566–574, dec 2006.

[89] Gregory Gutin, Toufik Mansour, and Simone Severini. A characterization of horizontal visibility graphs and combinatorics on words. *Physica A: Statistical Mechanics and its Applications*, 390(12):2421–2428, jun 2011.

[90] Jürgen Hackl and Bryan T. Adey. Estimation of traffic flow changes using networks in networks approaches. *Applied Network Science*, 4(1), may 2019.

[91] Frank R Hampel. The influence curve and its role in robust estimation. *Journal of the american statistical association*, 69(346):383–393, 1974.

[92] Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.

[93] Assaf Hochman, Pinhas Alpert, Tzvi Harpaz, Hadas Saaroni, and Gabriele Messori. A new dynamical systems perspective on atmospheric predictability: Eastern mediterranean weather regimes as a case study. *Science Advances*, 5(6), jun 2019.

[94] Petter Holme and Jari Saramäki. Temporal networks. *Physics Reports*, 519(3):97–125, oct 2012.

[95] J. Hu, J.B. Gao, and K.D. White. Estimating measurement noise in a time series by exploiting nonstationarity. *Chaos, Solitons & Fractals*, 22(4):807–819, nov 2004.

[96] Fang-Lin Huang, Xue-Min Wang, Zheng-Qing Chen, Xu-Hui He, and Yi-Qing Ni. A new approach to identification of structural damping ratios. *Journal of Sound and Vibration*, 303(1-2):144–153, jun 2007.

[97] Silu Huang, Ada Wai-Chee Fu, and Ruifeng Liu. Minimum spanning trees in temporal graphs. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, may 2015.

[98] Ismail Husein, Herman Mawengkang, Saib Suwilo, and Mardiningsih. Modeling the transmission of infectious disease in a dynamic network. *Journal of Physics: Conference Series*, 1255(1):012052, aug 2019.

[99] Boris Iglewicz and David Hoaglin. *Volume 16: how to detect and handle outliers, The ASQC basic references in quality control: statistical techniques, Edward F. Mykytka*. PhD thesis, Ph. D., Editor, 1993.

[100] Daniel J. Inman. *Engineering Vibration*. Pearson, 2014.

[101] Rinku Jacob, K. P. Harikrishnan, R. Misra, and G. Ambika. Weighted recurrence networks for the analysis of time-series data. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 475(2221):20180256, jan 2019.

[102] N JAKSIC and M BOLTEZAR. An approach to parameter identification for a single-degree-of-freedom dynamical system based on short free acceleration response. *Journal of Sound and Vibration*, 250(3):465 – 483, 2002.

[103] W. Ji and V. Venkatasubramanian. Hard-limit induced chaos in a fundamental power system model. *International Journal of Electrical Power & Energy Systems*, 18(5):279–295, jun 1996.

[104] Matthew Kahle and Elizabeth Meckes. Limit theorems for betti numbers of random simplicial complexes. *Homology, Homotopy and Applications*, 15(1):343–374, 2013.

[105] Holger Kantz and Thomas Schreiber. *Nonlinear Time Series Analysis*. Cambridge University Press, nov 2003.

[106] Holger Kantz and Thomas Schreiber. *Nonlinear Time Series Analysis*. Cambridge University Press, 2004.

[107] Karsten Keller, Teresa Mangold, Inga Stolz, and Jenna Werner. Permutation entropy: New ideas and challenges. *Entropy*, 19(3):134, mar 2017.

[108] David Kempe, Jon Kleinberg, and Amit Kumar. Connectivity and inference problems for temporal networks. *Journal of Computer and System Sciences*, 64(4):820–842, jun 2002.

[109] Matthew B. Kennel, Reggie Brown, and Henry D. I. Abarbanel. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physical Review A*, 45(6):3403–3411, mar 1992.

[110] Matthew B Kennel, Reggie Brown, and Henry DI Abarbanel. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physical review A*, 45(6):3403, 1992.

[111] Firas A. Khasawneh and Elizabeth Munch. Stability determination in turning using persistent homology and time series analysis. In *Proceedings of the ASME 2014 International Mechanical Engineering Congress & Exposition, November 14-20, 2014, Montreal, Canada*, 2014. Paper no. IMECE2014-40221.

[112] Firas A. Khasawneh and Elizabeth Munch. Chatter detection in turning using persistent homology. *Mechanical Systems and Signal Processing*, 70-71:527–541, 2016.

[113] Firas A. Khasawneh and Elizabeth Munch. *Utilizing Topological Data Analysis for Studying Signals of Time-Delay Systems*, pages 93–106. Springer International Publishing, Cham, 2017.

[114] Firas A. Khasawneh and Elizabeth Munch. Topological data analysis for true step detection in periodic piecewise constant signals. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 474(2218):20180027, oct 2018.

[115] Firas A. Khasawneh, Elizabeth Munch, and Jose A. Perea. Chatter classification in turning using machine learning and topological data analysis. In Tamas Insperger, editor, *14th IFAC Workshop on Time Delay Systems TDS 2018: Budapest, Hungary, 28–30 June 2018*, volume 51, pages 195–200, 2018. Accepted for publication at IFAC Workshop on Time Delay Systems; Budapest, Hungary; June 2018.

[116] Firas A. Khasawneh, Elizabeth Munch, and Jose A. Perea. Chatter classification in turning using machine learning and topological data analysis. *IFAC-PapersOnLine*, 51(14):195–200, 2018.

[117] Giorgi Khomeriki. Parametric resonance induced chaos in magnetic damped driven pendulum. *Physics Letters A*, 380(31-32):2382–2385, 2016.

[118] Alexander Khor and Michael Small. Examining k-nearest neighbour networks: Superfamily phenomena and inversion. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 26(4):043101, apr 2016.

[119] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical Review E*, 69(6), jun 2004.

[120] L. Lacasa, B. Luque, J. Luque, and J. C. Nuño. The visibility graph: A new method for estimating the hurst exponent of fractional brownian motion. *EPL (Europhysics Letters)*, 86(3):30001, may 2009.

[121] L. Lacasa, A. Nuñez, É. Roldán, J. M. R. Parrondo, and B. Luque. Time series irreversibility: a visibility graph approach. *The European Physical Journal B*, 85(6), jun 2012.

[122] Lucas Lacasa, Bartolo Luque, Fernando Ballesteros, Jordi Luque, and Juan Carlos Nuño. From time series to complex networks: The visibility graph. *Proceedings of the National Academy of Sciences*, 105(13):4972–4975, mar 2008.

[123] Lucas Lacasa and Raul Toral. Description of stochastic and chaotic series using visibility graphs. *Physical Review E*, 82(3), sep 2010.

[124] HJ Landau. Sampling, data transmission, and the nyquist rate. *Proceedings of the IEEE*, 55(10):1701–1706, 1967.

[125] Andrea Landherr, Bettina Friedl, and Julia Heidemann. A critical review of centrality measures in social networks. *Business & Information Systems Engineering*, 2(6):371–385, oct 2010.

[126] David Lane, Joan Lu, Camille Peres, Emily Zitek, et al. Online statistics: An interactive multimedia course of study. *Retrieved January*, 29:2009, 2008.

[127] Peter Lawson, Andrew B. Sholl, J. Quincy Brown, Brittany Terese Fasy, and Carola Wenk. Persistent homology for the quantitative evaluation of architectural features in prostate cancer histology. *Scientific Reports*, 9(1), feb 2019.

[128] Hyekyoung Lee, Hyejin Kang, M. K. Chung, Bung-Nyun Kim, and Dong Soo Lee. Persistent brain network homology from the perspective of dendrogram. *IEEE Transactions on Medical Imaging*, 31(12):2267–2277, dec 2012.

[129] Michel CR Leles, João Pedro H Sansão, Leonardo A Mozelli, and Homero N Guimarães. Improving reconstruction of time-series based in singular spectrum analysis: A segmentation approach. *Digital Signal Processing*, 77:63–76, 2018.

[130] Christophe Leys, Christophe Ley, Olivier Klein, Philippe Bernard, and Laurent Licata. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology*, 49(4):764–766, 2013.

[131] Duan Li, Zhenhu Liang, Yinghua Wang, Satoshi Hagihira, Jamie W. Sleigh, and Xiaoli Li. Parameter selection in permutation entropy for an electroencephalographic measure of isoflurane anesthetic drug effect. *Journal of Clinical Monitoring and Computing*, 27(2):113–123, dec 2012.

[132] J. W. Liang and B. F. Feeny. Identifying coulomb and viscous friction from free-vibration decrements. *Nonlinear Dynamics*, 16(4):337–347, 1998.

[133] Jin-Wei Liang and Brian F. Feeny. Balancing energy to estimate damping parameters in forced oscillators. *Journal of Sound and Vibration*, 295(3-5):988–998, aug 2006.

[134] Jin-Wei Liang and Brian F. Feeny. Balancing energy to estimate damping in a forced oscillator with compliant contact. *Journal of Sound and Vibration*, 330(9):2049–2061, apr 2011.

[135] Zhenhu Liang, Yinghua Wang, Gaoxiang Ouyang, Logan J Voss, Jamie W Sleigh, and Xiaoli Li. Permutation auto-mutual information of electroencephalogram in anesthesia. *Journal of Neural Engineering*, 10(2):026004, feb 2013.

[136] R.M. Lin and J. Zhu. Model updating of damped structures using FRF data. *Mechanical Systems and Signal Processing*, 20(8):2200–2218, nov 2006.

[137] Jared A. Little and Brian P. Mann. Optimizing logarithmic decrement damping estimation via uncertainty analysis. In *Special Topics in Structural Dynamics & Experimental Techniques, Volume 5*, pages 19–22. Springer International Publishing, jun 2019.

[138] Chein-Shan Liu. Identifying time-dependent damping and stiffness functions by a simple and yet accurate method. *Journal of Sound and Vibration*, 318(1-2):148–165, nov 2008.

[139] Tiebing Liu, Wenpo Yao, Min Wu, Zhaorong Shi, Jun Wang, and Xinbao Ning. Multiscale permutation entropy analysis of electrocardiogram. *Physica A: Statistical Mechanics and its Applications*, 471:492–498, 2017.

[140] Bartolo Luque, Lucas Lacasa, Fernando J. Ballesteros, and Alberto Robledo. Feigenbaum graphs: A complex network perspective of chaos. *PLOS ONE*, 6(9):1–8, 09 2011.

[141] Bartolo Luque, Lucas Lacasa, Fernando J. Ballesteros, and Alberto Robledo. Analytical properties of horizontal visibility graphs in the feigenbaum scenario. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(1):013109, mar 2012.

[142] B.P. Mann and F.A. Khasawneh. An energy-balance approach for oscillator parameter identification. *Journal of Sound and Vibration*, 321(1-2):65–78, mar 2009.

[143] Desire L Massart, Leonard Kaufman, Peter J Rousseeuw, and Annick Leroy. Least median of squares: a robust method for outlier and model error detection in regression and calibration. *Analytica Chimica Acta*, 187:171–179, 1986.

[144] Robert M. May. Chaos and the dynamics of biological populations. *Nuclear Physics B - Proceedings Supplements*, 2:225–245, nov 1987.

[145] Michael McCullough, Michael Small, Thomas Stemler, and Herbert Ho-Ching Iu. Time lagged ordinal partition networks for capturing dynamics of continuous dynamical systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(5):053101, 2015.

[146] Michael McCullough, Michael Small, Thomas Stemler, and Herbert Ho-Ching Iu. Time lagged ordinal partition networks for capturing dynamics of continuous dynamical systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(5):053101, may 2015.

[147] Andreas Otto Melih C. Yesilli, Firas A. Khasawneh. Topological feature vectors for chatter detection in turning processes. *arXiv:1905.08671*, 2019.

[148] Michał Melosik and W Marszalek. On the 0/1 test for chaos in continuous systems. *Bulletin of the Polish Academy of Sciences Technical Sciences*, 64(3):521–528, 2016.

[149] Michał Melosik and W Marszalek. On the 0/1 test for chaos in continuous systems. *Bulletin of the Polish Academy of Sciences Technical Sciences*, 64(3):521–528, 2016.

[150] Craig Meskell. A decrement method for quantifying nonlinear and linear damping parameters. *Journal of Sound and Vibration*, 296(3):643–649, sep 2006.

[151] T. Mimura and A. Mita. Automatic estimation of natural frequencies and damping ratios of building structures. *Procedia Engineering*, 188:163–169, 2017.

[152] Luis Montesinos, Rossana Castaldo, and Leandro Pecchia. On the use of approximate entropy and sample entropy with centre of pressure time-series. *Journal of NeuroEngineering and Rehabilitation*, 15(1), dec 2018.

[153] George B Moody and Roger G Mark. Mit-bih arrhythmia database, 1992.

[154] George B Moody and Roger G Mark. The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, 20(3):45–50, 2001.

[155] John R. Moore and Douglas A. Maguire. Natural sway frequencies and damping ratios of trees: concepts, review and synthesis of previous studies. *Trees - Structure and Function*, 18(2):195–203, mar 2004.

[156] Pablo Moriano, Jorge Finke, and Yong-Yeol Ahn. Community-based event detection in temporal networks. *Scientific Reports*, 9(1), mar 2019.

[157] Elizabeth Munch. A user's guide to topological data analysis. *Journal of Learning Analytics*, 4(2), 2017.

[158] James R. Munkres. *Elements of Algebraic Topology*. Addison Wesley, 1993.

[159] Audun Myers and Firas A. Khasawneh. Delay parameter selection in permutation entropy using topological data analysis. *arXiv:1905.04329 [physics.data-an]*, 2019.

[160] Audun Myers and Firas A. Khasawneh. Dynamic state analysis of a driven magnetic pendulum using ordinal partition networks and topological data analysis. In *Volume 7: 32nd Conference on Mechanical Vibration and Noise (VIB)*. American Society of Mechanical Engineers, aug 2020.

[161] Audun Myers and Firas A. Khasawneh. On the automatic parameter selection for permutation entropy. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(3):033130, mar 2020.

[162] Audun Myers, Elizabeth Munch, and Firas A Khasawneh. Persistent homology of complex networks for dynamic state detection. *arXiv preprint arXiv:1904.07403*, 2019.

[163] Audun D. Myers, Firas A. Khasawneh, and Brittany T. Fasy. Separating persistent homology of noise from time series data using topological signal processing. December 2020.

[164] Audun D. Myers, Firas A. Khasawneh, and Brittany T. Fasy. ANAPT: Additive noise analysis for persistence thresholding. *Foundations of Data Science*, 0(0):0, 2022.

[165] Audun D. Myers, Joshua R. Tempelman, David Petrushenko, and Firas A. Khasawneh. Low-cost double pendulum for high-quality data collection with open-source video tracking and analysis. *HardwareX*, 8:e00138, oct 2020.

[166] Suraj K. Nayak, Arindam Bit, Anilesh Dey, Biswajit Mohapatra, and Kunal Pal. A review on the nonlinear dynamical system analysis of electrocardiogram signal. *Journal of Healthcare Engineering*, 2018:1–19, 2018.

[167] Angel Nuñez, Lucas Lacasa, Eusebio Valero, Jose Patricio Gómez, and Bartolo Luque. Detecting series periodicity with horizontal visibility graphs. *International Journal of Bifurcation and Chaos*, 22(07):1250160, jul 2012.

[168] Angel M Nuñez, Lucas Lacasa, Jose Patricio Gomez, and Bartolo Luque. Visibility algorithms: A short review. *New Frontiers in Graph Theory*, pages 119–152, 2012.

[169] Philip Nuss, T.E. Graedel, Elisa Alonso, and Adam Carroll. Mapping supply chain risk by network analysis of product platforms. *Sustainable Materials and Technologies*, 10:14–22, dec 2016.

[170] S. Y. Oudot. *Persistence theory: from quiver representations to data analysis*, volume 209 of *AMS Mathematical Surveys and Monographs*. American Mathematical Society, 2015.

[171] Sinno Jialin Pan and Qiang Yang. A Survey On Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, oct 2010.

[172] George A. Papagiannopoulos and George D. Hatzigeorgiou. On the use of the half-power bandwidth method to estimate damping in building structures. *Soil Dynamics and Earthquake Engineering*, 31(7):1075–1079, jul 2011.

[173] Athanasios Papoulis and S Unnikrishna Pillai. *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Education, 2002.

[174] Jose A. Perea. A brief history of persistence.

[175] Jose A. Perea. Persistent homology of toroidal sliding window embeddings. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, mar 2016.

[176] Jose A. Perea. Topological time series analysis. *Notices of the American Mathematical Society*, ??(05):1, may 2019.

[177] Jose A. Perea, Anastasia Deckard, Steve B. Haase, and John Harer. SW1PerS: Sliding windows and 1-persistence scoring; discovering periodicity in gene expression time series data. *BMC Bioinformatics*, 16(1), Aug 2015.

[178] Jose A Perea, Anastasia Deckard, Steve B Haase, and John Harer. Sw1pers: Sliding windows and 1-persistence scoring; discovering periodicity in gene expression time series data. *BMC bioinformatics*, 16(1):257, 2015.

[179] Jose A. Perea, Anastasia Deckard, Steve B. Haase, and John Harer. SW1pers: Sliding windows and 1-persistence scoring; discovering periodicity in gene expression time series data. *BMC Bioinformatics*, 16(1), 2015.

[180] Jose A Perea and John Harer. Sliding windows and persistence: An application of topological methods to signal analysis. *Foundations of Computational Mathematics*, 15(3):799–838, 2015.

[181] Jose A. Perea, Elizabeth Munch, and Firas A. Khasawneh. Approximating continuous functions on persistence diagrams using template functions.

[182] Yakov Borisovich Pesin. Characteristic lyapunov exponents and smooth ergodic theory. *Uspekhi Matematicheskikh Nauk*, 32(4):55–112, 1977.

[183] David Petrushenko and Firas A. Khasawneh. Uncertainty propagation of system parameters to the dynamic response: An application to a benchtop pendulum. In *Volume 4B: Dynamics, Vibration, and Control*. American Society of Mechanical Engineers, nov 2017.

[184] Marco Piangerelli, Matteo Rucco, and Emanuela Merelli. Topological classifier for detecting the emergence of epileptic seizures. 2016.

[185] Steven M Pincus. Approximate entropy as a measure of system complexity. *Proceedings of the National Academy of Sciences*, 88(6):2297–2301, 1991.

[186] Steven M Pincus. Approximate entropy as a measure of system complexity. *Proceedings of the National Academy of Sciences*, 88(6):2297–2301, 1991.

[187] Pavel M. Polunin, Yushi Yang, Mark I. Dykman, Thomas W. Kenny, and Steven W. Shaw. Characterization of MEMS resonator nonlinearities using the ringdown response. *Journal of Microelectromechanical Systems*, 25(2):297–303, apr 2016.

[188] Yves Pomeau and Paul Manneville. Intermittent transition to turbulence in dissipative dynamical systems. *Communications in Mathematical Physics*, 74(2):189–197, jun 1980.

[189] Anton Popov, Oleksii Avilov, and Oleksii Kanaykin. Permutation entropy of eeg signals for different sampling rate and time lag combinations. In *Signal Processing Symposium (SPS), 2013*, pages 1–4. IEEE, 2013.

[190] Alberto Porta, Vlasta Bari, Andrea Marchi, Beatrice De Maria, Paolo Castiglioni, Marco di Rienzo, Stefano Guzzetti, Andrei Cividjian, and Luc Quintin. Limits of permutation-based entropies in assessing complexity of short heart period variability. *Physiological Measurement*, 36(4):755–765, mar 2015.

[191] M Prandina, J E Mottershead, and E Bonisoli. Damping identification in multiple degree-of-freedom systems using an energy balance approach. *Journal of Physics: Conference Series*, 181:012006, aug 2009.

[192] Marco Prandina, John E. Mottershead, and Elvio Bonisoli. An assessment of damping identification methods. *Journal of Sound and Vibration*, 323(3-5):662–676, jun 2009.

[193] Fengyong Qian, Shuhung Leung, Yuesheng Zhu, Waiki Wong, Derek Pao, and Winghong Lau. Damped sinusoidal signals parameter estimation in frequency domain. *Signal Processing*, 92(2):381–391, feb 2012.

[194] Thomas Quail, Alvin Shrier, and Leon Glass. Predicting the onset of period-doubling bifurcations in noisy cardiac systems. *Proceedings of the National Academy of Sciences*, 112(30):9358–9363, jul 2015.

[195] Rangayyan. *Biomed Signal Analysis 2E*. John Wiley & Sons, 2015.

[196] Jan Reininghaus, Stefan Huber, Ulrich Bauer, and Roland Kwitt. A stable multi-scale kernel for topological machine learning. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2015.

[197] Mark A Richards. The discrete-time fourier transform and discrete fourier transform of windowed stationary white noise. *Georgia Institute of Technology, Tech. Rep*, 2013.

[198] Joshua S Richman and J Randall Moorman. Physiological time-series analysis using approximate entropy and sample entropy. *American Journal of Physiology-Heart and Circulatory Physiology*, 278(6):H2039–H2049, 2000.

[199] Joshua S Richman and J Randall Moorman. Physiological time-series analysis using approximate entropy and sample entropy. *American Journal of Physiology-Heart and Circulatory Physiology*, 278(6):H2039–H2049, 2000.

[200] M. Riedl, A. Müller, and N. Wessel. Practical considerations of permutation entropy. *The European Physical Journal Special Topics*, 222(2):249–262, jun 2013.

[201] Müller Riedl, A Müller, and N Wessel. Practical considerations of permutation entropy. *The European Physical Journal Special Topics*, 222(2):249–262, 2013.

[202] Michael Robinson. *Topological Signal Processing*. Springer, 2014.

[203] Michael Robinson. *Topological signal processing*. Springer, 2014.

[204] G. Rohith and Nandan K. Sinha. Routes to chaos in the post-stall dynamics of higher-dimensional aircraft model. *Nonlinear Dynamics*, 100(2):1705–1724, apr 2020.

[205] Konstantinos Sakellariou, Thomas Stemler, and Michael Small. Estimating topological entropy using ordinal partition networks. *Physical Review E*, 103(2):022214, feb 2021.

[206] Benjamin Schäfer, Dirk Witthaut, Marc Timme, and Vito Latora. Dynamically induced cascading failures in power grids. *Nature Communications*, 9(1), may 2018.

[207] Benjamin Schäfer and G. Cigdem Yalcin. Dynamical modeling of cascading failures in the turkish power grid. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(9):093134, sep 2019.

[208] N. V. Semionov, Yu. G. Yermolaev, A. D. Kosinov, A. N. Semenov, B. V. Smorodsky, and A. A. Yatskikh. The effect of small angle of attack on the laminar-turbulent transition in boundary layer on swept wing at mach number m=2. In *AIP Conference Proceedings*. Author(s), 2017.

[209] Songwon Seo. *A review and comparison of methods for detecting outliers in univariate data sets*. PhD thesis, University of Pittsburgh, 2006.

[210] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, jul 1948.

[211] Claude E Shannon, Warren Weaver, and Arthur W Burks. The mathematical theory of communication. 1951.

[212] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55, 2001.

[213] He Shaobo, Sun Kehui, and Wang Huihai. Modified multiscale permutation entropy algorithm and its application for multiscroll chaotic systems. *Complexity*, 21(5):52–58, nov 2014.

[214] Azad Siahmakoun, Valentina A French, and Jeffrey Patterson. Nonlinear dynamics of a sinusoidally driven pendulum in a repulsive magnetic field. *American Journal of Physics*, 65(5):393–400, 1997.

[215] B. Skyrms and R. Pemantle. A dynamic model of social network formation. *Proceedings of the National Academy of Sciences*, 97(16):9340–9346, aug 2000.

[216] Michael Small. Complex networks from time series: Capturing dynamics. In *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*. IEEE, may 2013.

[217] Michael Small, Jie Zhang, and Xiaoke Xu. Transforming time series into complex networks. In *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 2078–2089. Springer Berlin Heidelberg, 2009.

[218] Hoon Sohn and Charles Reed Farrar. Damage diagnosis using time series analysis of vibration signals. *Smart Materials and Structures*, 10:446–451, 2001.

[219] Saleh Soltan, Dorian Mazauric, and Gil Zussman. Cascading failures in power grids. In *Proceedings of the 5th international conference on Future energy systems*. ACM, jun 2014.

[220] J. Sprott. Chaos and time-series analysis. *Choice Reviews Online*, 41(06):41–3492–41–3492, feb 2004.

[221] MATTHÄUS STANIEK and KLAUS LEHNERTZ. PARAMETER SELECTION FOR PERMUTATION ENTROPY MEASUREMENTS. *International Journal of Bifurcation and Chaos*, 17(10):3729–3733, oct 2007.

[222] MATTHÄUS STANIEK and KLAUS LEHNERTZ. PARAMETER SELECTION FOR PERMUTATION ENTROPY MEASUREMENTS. *International Journal of Bifurcation and Chaos*, 17(10):3729–3733, oct 2007.

[223] Herbert A Sturges. The choice of a class interval. *Journal of the american statistical association*, 21(153):65–66, 1926.

[224] Kashin Sugishita and Yasuo Asakura. Vulnerability studies in the fields of transportation and complex networks: a citation network analysis. *Public Transport*, 13(1):1–34, sep 2020.

[225] Floris Takens. Detecting strange attractors in turbulence. In *Dynamical systems and turbulence, Warwick 1980*, pages 366–381. Springer, 1981.

[226] Floris Takens. Detecting strange attractors in turbulence. In David Rand and Lai-Sang Young, editors, *Dynamical Systems and Turbulence, Warwick 1980*, volume 898 of *Lecture Notes in Mathematics*, pages 366–381. Springer Berlin Heidelberg, 1981.

[227] Mei Tao, Kristina Poskuviene, Nizar Alkayem, Maosen Cao, and Minvydas Ragulskis. Permutation entropy based on non-uniform embedding. *Entropy*, 20(8):612, 2018.

[228] Mei Tao, Kristina Poskuviene, Nizar Alkayem, Maosen Cao, and Minvydas Ragulskis. Permutation entropy based on non-uniform embedding. *Entropy*, 20(8):612, 2018.

[229] Joshua Tempelman. Chaos detection with persistent homology, 2020.

[230] Joshua R. Tempelman, Audun Myers, Jeffrey T. Scruggs, and Firas A. Khasawneh. Effects of correlated noise on the performance of persistence based dynamic state detection methods. In *Volume 7: 32nd Conference on Mechanical Vibration and Noise (VIB)*. American Society of Mechanical Engineers, aug 2020.

[231] Vy Tran, Eric Brost, Marty Johnston, and Jeff Jalkio. Predicting the behavior of a chaotic pendulum with a variable interaction potential. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 23(3):033103, sep 2013.

[232] Sarah Tymochko, Elizabeth Munch, Jason Dunion, Kristen Corbosiero, and Ryan Torn. Using persistent homology to quantify a diurnal cycle in hurricane felix.

[233] Sarah Tymochko, Elizabeth Munch, and Firas A. Khasawneh. Using zigzag persistent homology to detect hopf bifurcations in dynamical systems. *Algorithms*, 13(11):278, oct 2020.

[234] Krzysztof Urbanowicz and Janusz A. Hołyst. Noise-level estimation of time series using coarse-grained entropy. *Physical Review E*, 67(4), apr 2003.

[235] M. van Hagen. *Waiting experience at train stations*. PhD thesis, University of Twente, April 2011.

[236] Xiang Wan, Wenqian Wang, Jiming Liu, and Tiejun Tong. Estimating the sample mean and standard deviation from the sample size, median, range and/or interquartile range. *BMC Medical Research Methodology*, 14(1), dec 2014.

[237] Minggang Wang and Lixin Tian. From time series to complex networks: The phase space coarse graining. *Physica A: Statistical Mechanics and its Applications*, 461:456–468, nov 2016.

[238] Yishu Wang, Ye Yuan, Yuliang Ma, and Guoren Wang. Time-dependent graphs: Definitions, applications, and algorithms. *Data Science and Engineering*, 4(4):352–366, sep 2019.

[239] Tongfeng Weng, Jie Zhang, Michael Small, Rui Zheng, and Pan Hui. Memory and betweenness preference in temporal networks induced from time series. *Scientific Reports*, 7(1), feb 2017.

[240] Alan Wolf, Jack B Swift, Harry L Swinney, and John A Vastano. Determining lyapunov exponents from a time series. *Physica D: Nonlinear Phenomena*, 16(3):285–317, 1985.

[241] G.R. Wood and B.P. Zhang. Estimation of the lipschitz constant of a function. *Journal of Global Optimization*, 8(1), jan 1996.

[242] Hui Xiong, Pengjian Shang, and Jiayi He. Nonuniversality of the horizontal visibility graph in inferring series periodicity. *Physica A: Statistical Mechanics and its Applications*, 534:122234, nov 2019.

[243] Boyan Xu, Christopher J. Tralie, Alice Antia, Michael Lin, and Jose A. Perea. Twisty takens: A geometric characterization of good observations on dense trajectories.

[244] Mengkai Xu, Srinivasan Radhakrishnan, Sagar Kamarthi, and Xiaoning Jin. Resiliency of mutualistic supplier-manufacturer networks. *Scientific Reports*, 9(1), sep 2019.

[245] Jiawei Xue and Ruipeng Diao. A frequency domain interpolation method for damping ratio estimation. In *2014 IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2014)*. IEEE, nov 2014.

[246] Melih C. Yesilli, Sarah Tymochko, Firas A. Khasawneh, and Elizabeth Munch. Chatter diagnosis in milling using supervised learning and topological features vector. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. IEEE, dec 2019.

[247] Jingyi You, Chenlong Hu, Hidetaka Kamigaito, Kotaro Funakoshi, and Manabu Okumura. Robust dynamic clustering for temporal networks. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. ACM, oct 2021.

[248] Hong Zhang and Xuncheng Liu. Analysis of parameter selection for permutation entropy in logistic chaotic series. In *Intelligent Transportation, Big Data & Smart City (ICITBS), 2018 International Conference on*, pages 398–402. IEEE, 2018.

[249] J. Fang Zhang and Z. Gang Shao. Complex networks from lévy noise. *Indian Journal of Physics*, 85(9):1425–1432, sep 2011.

[250] Jianye Zhang and Peng Zhang. *Time Series Analysis Methods and Applications for Flight Data*. Springer Berlin Heidelberg, 2017.

[251] Yang Zhang, Zhou Zhou, Kelei Wang, and Xu Li. Aerodynamic characteristics of different airfoils under varied turbulence intensities at low reynolds numbers. *Applied Sciences*, 10(5):1706, mar 2020.

[252] Luciano Zunino, Miguel C Soriano, Ingo Fischer, Osvaldo A Rosso, and Claudio R Mirasso. Permutation-information-theory approach to unveil delay dynamics from time-series analysis. *Physical Review E*, 82(4):046212, 2010.

[253] Luciano Zunino, Miguel C Soriano, Ingo Fischer, Osvaldo A Rosso, and Claudio R Mirasso. Permutation-information-theory approach to unveil delay dynamics from time-series analysis. *Physical Review E*, 82(4):046212, 2010.