From Abstract Spaces to Physical Structures: A Topological and Geometric Perspective of Structural Dynamics

A thesis submitted to the University of Sheffield for the degree of Doctor of Philosophy in the Faculty of Engineering

by

Tristan Gowdridge

Department of Mechanical Engineering
University of Sheffield

September 2023

Abstract

Many academic areas of the sciences, which might seem contrived at their outset, often find relevance in practical fields like engineering. Prime examples include quantum mechanics in semiconductor production, number theory's importance in cybersecurity, and adjustments for general relativity in global positioning systems. This thesis looks up the abstraction hierarchy, focussing on the concepts of algebraic topology and, to a lesser degree, differential geometry, to address challenges in structural health monitoring (SHM) and nonlinear dynamics.

The first component of this thesis explores obtaining series solutions to nonlinear differential equations by considering *generating series* expansions, a framework rooted in differential geometry. Here, novel optimisations are presented for computationally determined series solutions, which output impulse responses of nonlinear systems. The process is applied to a benchmark study, yielding unseen solution depths, providing greater insights into the behaviour of nonlinear systems.

Much of the data-based SHM literature is dedicated to machine learning methods; overlooking informative shape features within data. The cornerstone of this thesis introduces and applies Topological Data Analysis (TDA), harnessing the data's shape, such as holes and voids, for SHM decision-making. Topological arguments are shown to enhance SHM insights regarding damage detection and environmental and operational variation removal, as well as augmenting established machine-learning approaches. A multi-faceted understanding of data is crucial for SHM since information is limited, and decisions carry dire safety and economic consequences.

A key challenge in SHM is discerning damage effects from benign environmental fluctuations. This thesis addresses this problem by imbuing 1D time series with a topology, where different trends give unique shapes, and inferences are made via topological reasoning. A significant portion of this thesis evaluates the Z24 Bridge, showcasing TDA's capability to identify damage amidst dominant external factors.

This thesis shows that abstract mathematical concepts yield beneficial outcomes by providing unique insights; underscoring the potential of TDA and generating series in SHM and nonlinear dynamics. By leveraging such novel analyses, one might identify data subtleties, indicative of structural issues, which might otherwise be overlooked with machine learning and traditional SHM means.

Acknowledgements

First and foremost, the sincerest thanks go to Prof. Keith Worden, whose brilliant supervision and vast expertise were instrumental throughout this project. Thank you for all your help, support, countless funny stories in the pub, and simply believing in me. I would also like to thank my secondary supervisor, Prof. Nikolaos Dervilis, for his valuable insights during discussions. Without his reassurances about the efficiency of my code, I would likely still be optimising! Together, you both never cease to make me smile and motivate me enormously. An extended thank you to Prof. Lizzy Cross for providing access to and discussions about the Tamar Bridge data, as well as pointers on cointegration.

The Dynamics Research Group has been a blessing to be a part of. Here, I have met like-minded individuals, be it through research, sports, or leisure and I could not imagine more supportive friends. The camaraderie from friends like Aidan, Brandon, Chris, Dan B, Jack P, Marcus, Matt T, Matty, Max, and Tina has been invaluable in making life more enjoyable. Having such an amazing set of people to pursue out-of-hours activities be it through climbing, squash, or even enjoying a pint in the Red Deer (or maybe a few more in Woody's) has helped during some of the more testing times. Thank you to everyone in the DRG old, new, past and present.

Thank you to Chris and Jack, with whom I have been lucky enough to share an office. Their insightful, humorous, and engaging conversations have enriched the past few years, making the PhD journey much more enjoyable. Special appreciation goes to Chris for his great company during the write-up period.

Outside of the office, a special thanks to Harry, Kiernan, Jordan, Falcone, Ethan, and Alex F for being constant sources of laughter, entertainment, and joy. Thanks to the guys back home Wardle, Nathan, Tom, Kieran, and Jack D for always being there.

Last but by no means least, a heartfelt thanks to Gemma. Not only for being incredibly caring and loving over the past six years but also unimaginably understanding during the hectic last three months of the thesis write-up. Without your help and support, completing this PhD would not have been possible.

Table of Contents

Chapter 1

INTRODUCTION

Mechanical Engineering tends to favour pragmatic and inherently geometric techniques, be it through modelling or data analysis. This trend is evident in the subdisciplines of nonlinear dynamics and structural health monitoring (SHM), which often overlook abstract approaches like topology and differential geometry. However, in the light of recent computational advancements and the emergence of new theories, there is value in integrating these abstract notions; providing modern insights to age-old engineering problems. This thesis aims to incorporate topology and differential geometry concepts, thereby diversifying an engineer's analytical toolbox.

1.1 Structural Health Monitoring

SHM is a multi-disciplinary engineering field concerning the continuous, real-time, and online monitoring of engineering structures. To achieve SHM, sensors are placed over a structure to observe its damage-sensitive features, inferring information regarding the structure's past, present, and ideally future health and operational states. Common applications of SHM are in assessing civil, aerospace, and mechanical infrastructure [1].

The information extracted from SHM systems is tiered via *Rytter's hierarchy*. Each successive level offers greater insight into the structure's health state, but determining them becomes increasingly complex. Rytter's hierarchy, first outlined in [2] and later expanded [3] to a form in popular usage today, is characterised as follows:

I Detection: is damage present within a system?

II Localisation: where is the damage?

III Type: how could the damage be classified?

IV Quantification: how severe is the damage?

V Prognosis: how long can the structure operate as intended?

Solving for each tier in Rytter's hierarchy requires sophisticated data analysis or modelling strategies, in conjunction with sensing technologies. While advanced stages of damage might be visibly evident, e.g. a noticeable crack, at this point, the structure is likely beyond cost-effective repair. Additionally, damage may occur at a microscopic or subsurface level. For these reasons, in SHM, a reliance is placed on measuring features that display indirect signs of damage. These effects indicative of damage are often subtle, particularly in the early stages. Consequently, extracting these damage subtleties requires mathematically-involved methodologies, where the two main philosophies of SHM are *model-based* and *data-based*.

Model-based SHM centres around developing a high-fidelity physics-based model of a structure, factoring in material properties, geometry, loading conditions, among other factors. This model is validated against data gathered from the structure. If there are statistically significant discrepancies between the model and observed data, the structure is behaving abnormally, potentially indicating the presence of damage.

Data-based SHM does not rely on predefined models. Instead, such approaches aim to identify patterns within the data, commonly via statistical or machine-learning techniques, which then guide monitoring decisions.

The work in this thesis leans towards a data-based approach, but it is important to acknowledge the unique advantages and disadvantages of each method. Model-based approaches are grounded in the mechanics of the structure, leading to interpretable results and the ability to model in hypothesised contexts, such as earthquakes or extreme storms. However, physics-based models are complex, bespoke, costly to develop, and limited to current physical knowledge; hence, they will never truly capture all real-world phenomena. On the other hand, data-based approaches are relatively cheap, adaptable, generalisable and scalable. However, a data-based method's efficacy is tied to the quality of the input data, and their inner workings are often opaque, aptly giving them the name *black-box* models. Lately, there has been an interest in *grey-box* modelling [4], which aims to imbue data-based approaches with some physical understanding of the structure.

If SHM is so challenging to implement well, and structures have withstood the test of time long before the conception of the field, why is SHM now crucial for infrastructure? Several reasons underpin SHM's usage, with some of the most important benefits being [5]:

Safety: SHM systems, with their continuous and online monitoring, can detect damage in its initial stages. This information informs asset managers about the damage, enabling timely intervention. This action reduces the likelihood of catastrophic failure, which often carries severe safety implications. Operating a structure in the presence of damage, unbeknownst of otherwise, only exacerbates the damage. Hence, early detection increases safety.

Economic: As a corollary to the safety benefits, similar logic applies to economic advantages. A catastrophic breakage could cause undue damage to adjacent parts of the machinery. For example, addressing a faulty bearing early on might prevent the need to repair a costlier gearbox later. Furthermore, catastrophic failures in large-scale structures can lead to prolonged downtimes. Given that these structures have correspondingly large outputs, any period of inactivity is an economic inefficiency.

Automation: Automated monitoring is preferential to manual monitoring in situations where accessing parts is problematic e.g. a centrally-located part within a large machine with many surrounding parts, or hard-to-reach structures like offshore wind turbines. SHM also can increase the period or even remove the need for manual-routine maintenance, thereby reducing downtime and expensive engineer assessment.

Operational Optimisation: SHM systems gather data regarding environmental and operational variations (EOVs), which can be utilised to finetune a structure's operation. For instance, adjusting a wind turbine's blade pitch and nacelle yaw according to wind speed and direction for maximal energy production [6].

1.1.1 SHM in Practice

Implementing SHM systems in real-world settings presents various deployment challenges. In controlled laboratory settings, structural excitation and boundary conditions are precise and controllable. In contrast, real-world conditions present multiple unpredictable excitations and constantly shifting circumstances which complicate SHM. Structures face daily and seasonal temperature variations. Traffic loads intensify during busy periods. Foundations degrade over time. These complexities accumulate, giving an ever-increasingly complicated scenario that an SHM system must consider to provide reliable assessments.

A major problem is that the effects from EOVs often provide more variation in the damage-sensitive features than the early signs of damage. Distinguishing effects, such as temperature, from genuine damage effects, is a classic SHM problem. The primary EOVs of interest in this research are temperature and traffic loading.

SHM systems are frequently used for large-scale, high-value structures. Acquiring data from a damaged state presents the ill-opposed financial burden of willingly damaging such costly and often bespoke structures. Consequently, uncertainty often arises in the influence of damage on damage-sensitive features. Even if some damage information is known, damage can manifest in different forms and locations, each possibly having unique effects on the structure. It becomes unfeasible, if not impossible, to account for every possible damage type, location, and their combinations. Therefore, SHM necessitates clever data analysis to maximise information extraction.

1.2 Topological Data Analysis for SHM

Topological data analysis (TDA) is an advanced data analysis method not yet in common usage in SHM. Since its inception in the early noughties [7], TDA has experienced substantial growth, finding application across many mathematically-centred academic disciplines.

TDA uses the shape of data as an analytical feature via theories adapted from the more abstract side of mathematics, specifically, a subfield called *algebraic topology.* This research aims to merge SHM and TDA, thereby assisting SHM decision-making via novel topological reasoning. The objective is to expand the SHM data-driven toolkit by incorporating topological methods both as a standalone process and in conjunction with well-established machine-learning methods.

The primary tool in the TDA arsenal is *persistent homology*, which offers multi-scale descriptions of the shape of high-dimensional data by extending the traditional mathematical concept of *homology.* In essence, homology determines the number of k-dimensional holes in a continuous space. In TDA, homology is generalised to consider discrete point clouds, like those from data sets. The term ‘persistent’ arises because the homology is evaluated across various length scales, tracking when the k-dimensional holes appear and subsequently disappear; gauging the persistence of topological features in data. The persistent homology distinctively represents a data set and can be compared to persistent homologies from other data sets, quantifying the degree of similarity in shape between the two.

This thesis proposes that damage induces some unique and observable topological change in SHM data, which can be identified and leveraged via TDA.

1.3 Nonlinear Dynamics

Besides SHM, another complementary theme of this thesis concerns *nonlinear dynamics.* These two studies often intersect, as SHM often requires modelling nonlinear phenomena pervasive in nature, areas where linear models otherwise fall short. Nonlinear dynamics, at its core, studies systems that do not follow a proportional input-output relationship, giving rise to rich and descriptive theories.

Modelling nonlinearities in structures is key, as all real-world structures exhibit nonlinearity to some degree; be it via friction in joints, loading beyond a material’s linear-elastic region, or even a structure’s geometry [8]. Nonlinear systems also exhibit such phenomena as chaos and bifurcations, which are not represented by linear analysis. The effects of such complex behaviours could profoundly affect the longevity of structures and machines. Hence, nonlinear dynamics is not just an academic pursuit, but necessary to understand and predict real-world structural happenings.

In this thesis, nonlinear dynamics is explored in two different contexts. One treatment is under the lens of topological methods, delving into some chaotic properties of systems of differential equations and the intriguing phase-space characteristics of their solutions. The other context employs a method rooted in differential geometry, using the *generating series* to provide solutions to nonlinear differential equations.

1.4 Generating Series

Although the generating series have strong foundations in differential geometry via Lie groups and Lie algebras [9], such depths are not covered in this thesis directly. In lieu of deep mathematical abstraction, the novelty lies within application and optimisation, given the engineering focus of this thesis.

The generating series serves as a tool in this work, specifically a determination mechanism for a series solution to nonlinear differential equations. The generating series approach parallels the Laplace transform used for ordinary differential equations but tailored for nonlinear differential equations. The general procedure transforms a nonlinear differential equation into an alternative domain, enabling algebraic manipulation of what was a calculus problem – exactly like Laplace transforms in the s-domain. In this generating series domain, polynomial nonlinearities are expanded via the *shuffle product*[1]. Following this algebraic expansion, the result is inverse transformed back to the time domain, yielding an analytical series solution to nonlinear differential equations.

The appeal of the generating series approach stems from its compatibility with computer programs. The shuffle product is intuitively represented via a recursive breakdown. While manually calculating this expansion is impractical, computers manage it favourably; opening the door for computational optimisation to uncover unforeseen series-solution depths.

This thesis applies the generating series method to a modified version of Duffing's equation [11], a common nonlinear differential equation in structural dynamics; specifically, when excited by an impulse response.

1.5 Thesis Novelty

The overarching theme of this thesis is the application of topological data analysis to structural health monitoring. In doing so, TDA – which is previously unseen to the SHM community – is introduced and applied to problems on the softer end of Rytter's hierarchy. There are a few detours along the way where some pure

---

[1] Shuffle algebra, like topology, traces its origins from the more abstract side of mathematics as first discovered by Eilenberg and Mac Lane [10]. Coincidentally, or perhaps a testament to their genius, Samuel Eilenberg and Saunders Mac Lane were also pioneers of algebraic topology via *category theory* and *homological algebra*.

nonlinear dynamics problems are considered, both by topological data analysis and the generating series. The specific contributions to the existing literature are as follows:

- Two topology-based novelty detection mechanisms are presented. The SHM decisions are formed solely on the relative shape of data partitions. The first damage-detection procedure is native to persistent homology, using a common metric over the space of persistent homologies. The second manipulates persistent homology into features suitable for machine-learning classification. In both, the damage influence on the data's shape is uniquely recognisable, even in the presence of a magnitude-dominant EOV.
- Furthering the understanding of the impact of *cointegration* – a data normalisation procedure – on the removal of EOVs from damage-sensitive features. Embedded topologies are formed from time series, and assessed before and after cointegration to measure the effectiveness of trend removal, especially when multiple EOVs with distinct effects are present.
- The creation of a computer program that efficiently determines the shuffle product, incorporating several novel computational and mathematical optimisations. The interest in determining an impulse response prompted its derivation in the generating-series domain. Together, these findings provide a highly-optimised, generalisable framework for the impulse response of nonlinear differential equations. This framework is applied to a modified Duffing's equation, revealing unseen series-solutions depths.

1.6 Thesis Outline

A summary of each chapter is as follows:

Chapter 2: A standalone chapter that introduces the generating series method, detailing its application for determining series solutions to nonlinear differential equations. This chapter provides a derivation of the impulse response in the generating series domain and outlines novel optimisation strategies in the shuffle product's implementation. The chapter concludes with an application to a variant of Duffing's equation and showcases the results.

Chapter 3: Provides an overview of the background theory needed for topological data analysis. Assuming no prior knowledge of abstract mathematical concepts, it begins with basic set theory and progressively builds up through topics such as groups, metric spaces, topological space, simplicial complexes, homology, persistent homology, and the Wasserstein distance. The chapter concludes with some common TDA pitfalls and shortcomings.

Chapter 4: A topological approach to SHM is demonstrated using the classic Z24 Bridge case study. The bridge's natural frequencies are partitioned into data sets according to the temperature at the instance, or the presence of damage. A topology-based metric is considered over the partitions to identify a significant topological change in the presence of damage.

Chapter 5: Here, the previous chapter is extended to consider the persistent homologies as features in machine learning. Sliding windows are taken over the Z24 bridge natural frequencies, and the persistent homologies are used to decide a binary classification signalling the presence of damage.

Chapter 6: A nonlinear dynamics case study is presented, demonstrating the capability of topological data analysis to calculate the fractal dimension of strange attractors. This detour from SHM shows information exists in short-lived topological features, which is not always considered to be the case. Concluding this chapter is a novel method for determining the optimal delay for reconstructing attractors.

Chapter 7: The beginning of analysing EOV normalisation via both linear and nonlinear cointegration when applied to the Z24 natural frequencies. Each natural frequency time series is embedded, giving them a topology quantifiable via persistent homology. The embedded time series are compared before and after cointegration, evaluating the effectiveness of trend removal over cointegration based on the extent of shape removal in the embeddings.

Chapter 8: This chapter topologically quantifies the relative removal of multiple nonstationary trends present from distinct EOVs when passed through a cointegration procedure. The Tamar Bridge is introduced as a benchmark because of its pronounced effects from multiple EOVs. This exploration introduces a novel method for determining the optimal delay, coupled with a pre-existing method for the optimal embedding dimension. Embeddings are created for each EOV and compared to embedded structural parameters before and after cointegration to evaluate the most significant topological changes.

Chapter 9: This thesis concludes with a discussion of the proposed techniques, highlighting the advantages and disadvantages of applying more abstract analyses to engineering problems, then noting some possible areas for future study.

Chapter 2

A GENERATING SERIES APPROACH FOR NONLINEAR OSCILLATORS

This chapter explores the *generating series* approach for determining the *Volterra series* of nonlinear systems. An analytical determination of the response for a nonlinear single-degree-of-freedom (SDOF) oscillator subject to an impulse excitation will be provided. The theoretical aspects of this approach may, at times, appear verbose and complex. However, this approach allows for an automatic computational expansion of higher-order terms. A walkthrough of the required background theory is provided, this is then supplemented with a classic nonlinear dynamics problem. This chapter will also shed light on several optimisation strategies for implementing this generating-series approach.

One of the attractive features of the Volterra series is that the generalised coefficients of the series have physical interpretations; the coefficients are actually linear and nonlinear impulse responses, and their Fourier transforms can be regarded as Higher-dimensional Frequency Response Functions (HFRFs) [8]. Unfortunately, calculations with the Volterra series are very demanding in algebraic terms and rapidly become intractable (at least by hand), as the order of the expansion increases. Fortunately, such a reformulation exists in the form of the *generating series.* The ground-breaking idea involved the representation of the Volterra series – a sequence of high-dimensional integrals – as a purely algebraic expansion. The overriding benefit of the generating-series approach was that the operations in the series expansion are implementable in computer algebra.

Analytical approximations, including the Volterra series, offer a greater level of insight and comprehension for nonlinear systems when compared to numerical counterparts. Analytic expressions are succinctly embodied in formulae, enabling analysts to appreciate the impact of individual parameters on the solution. Furthermore, analytical expressions allow for further examination, including the application of calculus to gain additional system insights. In contrast, numerically-determined solutions often obscure this intuitive understanding. However, it is important to note that when compared to numerical solutions, such as an initial-value problem using a *fourth-order Runge-Kutta* [12], analytical solutions require a richer theory, are harder to implement, and require more time-consuming calculations.

The generating-series approach to nonlinear differential equations bears a significant resemblance to the Laplace transform approach to ordinary differential equations. The general procedure for both is as follows: the differential equation undergoes a transformation into a different domain, which is then algebraically manipulated, and finally transformed back into the time domain, giving the solution to the differential equation. However, this generating-series theory extends the standard Laplace transform to incorporate nonlinear terms, which massively complicate matters. For the generating-series approach, one must convert the nonlinear differential equation into integral form. Following this, previous work [9, 13] has outlined a method of transforming the integral form into a different domain where the nonlinearities may be expanded algebraically. The transformed equivalent of multiplication – known as the *shuffle product* – plays a crucial role in computing the Volterra kernels. An iterative process in the transformed domain centres around the shuffle-product expansion of the generating series. Once the necessary accuracy is obtained, the terms are decomposed into a form applicable for inverse transformation back into the time domain, thereby providing an analytical (albeit truncated), solution to the nonlinear differential equation.

The generating-series approach is readily extensible to solve more complex systems than the example given later in this chapter. Increased complexity could come from higher-order derivatives or more nonlinearities. The excitation of the system is also not limited to an impulse. Solutions are well-defined for other excitations, such as step, sinusoidal, exponential, polynomial [14, 15, 16], and Gaussian White Noise (GWN) [17, 18]. In the current literature, a definition is given for the generating series response to a Dirac-delta excitation [19]. However, in this paper, no derivation or reference is given for this result[1]. Hence, this chapter will provide an original derivation of the impulse response for generating series.

An example is provided, where an SDOF oscillator with quadratic and cubic stiffness nonlinearities is subject to an impulse excitation with amplitude A. A walkthrough of the procedure and calculations is presented, highlighting some key points that might deter someone from attempting to implement this challenging analysis. The results will then be presented. To cast light on the credibility of this approach, a comparison to previously-determined Volterra kernels is given, and an error analysis compared to a fourth-order Runge-Kutta solution. Finally, the example will conclude with a discussion regarding the performance and stability of the Volterra series.

2.1 Background Theory

Rather than exploring the intricate details of algebraic systems, or offering proof as to why this procedure is a valid method for determining the Volterra series, this work will focus on practical applications of previous work by Fleiss et al. [9, 13, 19]. However, the reader should be made aware of the rich interplay between *generating series*, *differential geometry* and *Lie algebras* [9, 10, 20, 21, 22]. In this section, only the required nomenclature and concepts are introduced, to ensure that readers can fully appreciate the following material.

2.1.1 Volterra Series

The Volterra series serves as a mapping for nonlinear input-output processes [23, 24]. For structural dynamicists, the Volterra series is a generalisation of Duhamel's integral (equation (2.1)) to include nonlinear behaviour. For the more mathematically inclined, the Volterra series can be thought of as a generalisation of the Taylor series from functions to functionals, incorporating a reliance on previous values in the series – memory.

---

[1] The author has completed a comprehensive literature survey and suspects a derivation for the impulse excitation may exist within Moustanir Lamnabhi's 1986 PhD Thesis titled 'Analyse des systèmes non linéaires par les méthodes de développements fonctionnels'. However, the only records of this work are at the University of Lille and Paris-Saclay University in paper and microfiche formats. The author's requests for Inter-Library Loans and scans of the documents were denied, and attempts to contact Moustanir Lamnabhi proved fruitless.

The input-output behaviour for a linear system is,

$$y(t) = \int_{-\infty}^{+\infty} h(\tau)u(t-\tau)\,\mathrm{d}\tau \tag{2.1}$$

where $h(\tau)$ represents the system's linear impulse response [8]. For nonlinear systems, Duhamel's integral denotes the first term, $y_1(t)$, in the Volterra series

$$y(t) = y_1(t) + y_2(t) + y_3(t) + \cdots + y_i(t) + \cdots \tag{2.2}$$

where in general

$$y_i(t) = \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} h_i\left(\tau_1, \ldots, \tau_i\right) u\left(t-\tau_1\right) \cdots u\left(t-\tau_i\right)\, \mathrm{d}\tau_1 \cdots \mathrm{d}\tau_i \tag{2.3}$$

The terms denoted by h_i are referred to as the *Volterra kernels*, and are essentially generalisations of the linear impulse-response function (IRF). Determining the Volterra kernels often requires complex and manual (therefore error-prone) calculations; be it through contour integration [25], harmonic-probing [26], or exponential-inputs [27]. Such methods of determining the Volterra kernels, the generating-series approach included, become cumbersome even for some lower-order terms. However, the generating-series method discussed in this chapter allows for a relatively simplistic implementation on computers. Allowing for the calculation of much higher-order terms when compared to hand calculations.

2.1.2 Noncommutative Algebra

Let $X = \{x_0, x_1\}$, this set is called the *alphabet* and the elements are called *letters*. X generates the *free monoid* set X^*; whose elements are sequences of the form $x_{j_v} \ldots x_{j_0}$, called *words* [13].

The length of a word, $w \in X^*$, denoted $|w|$, refers to its number of letters. For instance, $|x_0x_1x_1x_1x_0| = 5$, while the length of the empty word 1 is zero as it contains no letters.

A product of words signifies their concatenation, expressed as:

$$\left(x_{j_\nu} \cdots x_{j_0}\right)\left(x_{k_\mu} \cdots x_{k_0}\right) = x_{j_\nu} \cdots x_{j_0} x_{k_\mu} \cdots x_{k_0} \tag{2.4}$$

The operation of concatenating words is noncommutative, the reasons for this will be elaborated in Section 2.1.4.

In this context, it is necessary to define two letters and their corresponding time-domain counterparts

$$x_0 \leftrightarrow \int_0^t \mathrm{d}\tau \quad \text{and} \quad x_1 \leftrightarrow \int_0^t u(\tau)\,\mathrm{d}\tau \tag{2.5}$$

where x_0 represents an integration in the time domain and x_1 is the integration of the system excitation $u(t)$ in the time domain i.e. $y(t) = S\left[u(t)\right]$, where S represents the system equation. The usage and understanding of these letters will be expanded in Section 2.1.4, where the noncommutativity constraint will also be further discussed. The definitions given in equation (2.5) will then form the basis for the transformation into the generating-series domain.

A noncommutative series g, is called the generating series and represents a system with input-output behaviour. It is possible to transform a differential equation in y into the generating-series domain, where g serves as the analogue of y. This transformation is performed by following the letter definitions given in equation (2.5), and substituting y^k with $g^{⧢k}$. More detail will be given on this in Sections 2.1.3, 2.1.4 and 2.1.5.

2.1.3 Basic Shuffle Product

A prominent operator inside the algebraic framework is the *shuffle product*, denoted by ⧢; this is an essential operator in the expansion of generating series. As briefly touched upon, the shuffle product is the transformed analogue of multiplication, specific to the dependent variable in the differential equation. Once all the necessary background theory has been introduced, the shuffle product will be revisited in Section 2.1.6, specifically concerning its application to generating series. At this stage, focusing on the shuffle product's simpler application over letters and words will suffice; this will help build intuition over the fundamental workings of the shuffle product.

The shuffle product is best described as the sum of all the permutations formed from a riffle shuffle of the letters of the two arguments. The following identities, given $x_1, x_2, y_1, y_2 \in X$, may help support this definition:

1. $1 ⧢ 1 \equiv 1$, gives the terminating criterion for the shuffle expansion

2. $1 ⧢ x_1 \equiv x_1 ⧢ 1 = x_1$, shuffling with the identity element.
3. $x_1 ⧢ y_1y_2 \equiv x_1y_1y_2 + y_1x_1y_2 + y_1y_2x_1$, demonstration of the result for a non-trivial case.
4. $x_1x_2 ⧢ y_1y_2 \equiv x_1(x_2 ⧢ y_1y_2) + y_1(x_1x_2 ⧢ y_2)$, the shuffle product can be recursively simplified into many reduced problems – a pivotal requirement for this work.

To further clarify matters, the full expansion of identity 4 is given as:

$$\begin{aligned} x_1x_2 ⧢ y_1y_2 &= x_1(x_2 ⧢ y_1y_2) + y_1(x_1x_2 ⧢ y_2) \\ &= x_1x_2y_1y_2 + x_1y_1x_2y_2 + x_1y_1y_2x_2 + y_1x_1x_2y_2 + y_1x_1y_2x_2 + y_1y_2x_1x_2 \end{aligned} \tag{2.6}$$

As highlighted in equation (2.6), the letters constructing the words passed into the shuffle product do not permute in any of the output terms. Specifically, x_1 always precedes x_2, the same is true for y_1 and y_2. This statement holds true across all shuffle-product expansions.

Most handily, the shuffle product can be recursively expressed along the length of the two arguments, by splitting the arguments into a concatenation of a letter and word. This fact will be used extensively when the shuffle product is described in more detail over generating series of a certain form in Section 2.1.6. Given $w_1, w_2 \in X^*$ and $x_1, x_2 \in X$

$$x_1w_1 ⧢ x_2w_2 \equiv x_1(w_1 ⧢ x_2w_2) + x_2(x_1w_1 ⧢ w_2) \tag{2.7}$$

2.1.4 Iterated Integrals

Iterated integrals serve as a key transformation mechanism when converting nonlinear differential equations into the generating-series domain. The iterated integrals arise when converting the nonlinear differential equations into integral form. For a system with the highest-order derivative n, the k^{th} derivative where $k < n$, requires $n - k$ iterations of integrals to eliminate that differential (this process is exemplified in equation (2.14)).

Consider the product of two iterated integrals. For notational convenience, the number of iterated integrals is determined by the number of bounds, and only a single integral is displayed:

$$\begin{gathered}\left(\int_0^t d\xi_{j_v}\dots d\xi_{j_0}\right)\left(\int_0^t d\xi_{k_\mu}\dots d\xi_{k_0}\right)= \\ \int_0^t \mathrm{d}\xi_{j_v}\left[\left(\int_0^\tau \mathrm{d}\xi_{j_{v-1}}\dots \mathrm{d}\xi_{j_0}\right)\left(\int_0^\tau \mathrm{d}\xi_{k_\mu}\dots \mathrm{d}\xi_{k_0}\right)\right]+ \\ \int_0^t \mathrm{d}\xi_{k_\mu}\left[\left(\int_0^\tau d\xi_{j_v}\dots \mathrm{d}\xi_{j_0}\right)\left(\int_0^\tau \mathrm{d}\xi_{k_{\mu-1}}\dots \mathrm{d}\xi_{k_v}\right)\right]\end{gathered} \tag{2.8}$$

Interestingly, the expansion of the product using integration-by-parts shares a striking resemblance to equation (2.7), wherein terms are removed from the product, and the problem is redefined on a smaller subset of the initial problem. This resemblance forms the foundation for transforming between the two domains and is, in fact, a theorem.

Theorem 2.1.1. *The product of two analytic causal functionals is again an analytic causal functional of the same kind, the generating power series of which is the shuffle product of the two generating power series [9, 13, 20]*

$$y \cdot y \leftrightarrow g \sqcup\!\sqcup\, g \tag{2.9}$$

This is further extended to higher-order terms too; for instance,

$$y^k \leftrightarrow \underbrace{g \sqcup\!\sqcup \cdots \sqcup\!\sqcup\, g}_{\text{k-times}} = g^{\sqcup\!\sqcup k} \tag{2.10}$$

The term ‘analytic causal functional’ may seem complex at first glance, but a step-by-step breakdown can help uncover its meaning. Firstly, ‘analytic’ means a function that can be locally expressed as a convergent power series. Such functions are notable for their infinite differentiability. Secondly, ‘causal’ simply pertains to the reliance or causality on another variable’s past and present values, such as x_1 or the system excitation (depending on the domain) in this context. Finally, a ‘functional’ is a special kind of function that takes another function as an input and produces a scalar as its output. Thence, combining these terms, an ‘analytic causal functional’ is a scalar-producing function that is infinitely differentiable, has a dependency on another variable, and takes another function as its input.

By letting g denote the generating series associated with y, the letter x_0 be the integral $\int_0^t y(\tau)\,\mathrm{d}\tau$ and x_1 the integral with respect to the excitation $\int_0^t u(\tau)\,\mathrm{d}\tau$ [13] this outlines all the requirements to transform the nonlinear differential equations into the generating-series domain. For clarification, k iterated integrals in the time domain will be transformed to x_0^k. For k iterated integrals applied to the excitation, this results in $x_0^{k-1}x_1$. Since only the first application of the integral is applied to the excitation, this results in the single x_1 term. The noncommutativity constraint is now transparent, as changing the sequence of letters within a word analogously permutes the integration bounds.

Example 2.1.2. Consider the excitation $u(t) = t^k$, and then consider the two products x_0x_1 and x_1x_0. Expanding their time domain equivalent terms gives:

$$x_0x_1 \leftrightarrow \int_0^t \int_0^{\tau_2} \tau_1^k \;\mathrm{d}\tau_1\,\mathrm{d}\tau_2 = \frac{t^{k+2}}{(k+1)(k+2)} \tag{2.11}$$

and

$$x_1x_0 \leftrightarrow \int_0^t \tau_2^k \int_0^{\tau_2} \;\mathrm{d}\tau_1\,\mathrm{d}\tau_2 = \frac{t^{k+2}}{k+2} \tag{2.12}$$

Hence, it is immediately evident that $x_0x_1 \neq x_1x_0$ in this case; highlighting the requirement to respect noncommutativity.

2.1.5 Generating Series

Generating series serve as a powerful tool for solving nonlinear differential equations. The generating series, g, is the transformed equivalent of the dependent variable in a differential equation (much like the relation between $y(t)$ and $Y(s)$ in standard Laplace transforms). The generating series can be applied to nonlinear differential equations, those encapsulated by the general form,

$$\sum_{i=0}^{n} l_i \frac{\mathrm{d}^i y}{\mathrm{d}t^i} + \sum_{j=2}^{m} b_j y^j = u(t),\; l_n = 1 \tag{2.13}$$

Where y represents the response variable which is dependent on t, u is the excitation, l_i are the coefficients of the differential terms, and b_j are the coefficients of the polynomial nonlinearities. This formula describes a family of nonlinear differential equations with polynomial nonlinearities, including Duffing's equation [11].

Upon converting equation (2.13) into integral form, the following integral equation is obtained:

$$\begin{aligned} y + l_{n-1}\int_0^t y \,\mathrm{d}\tau_1 + \cdots + l_0 \int_0^t \mathrm{d}\tau_n \cdots \int_0^{\tau_2} y \,\mathrm{d}\tau_1 + \sum_{j=2}^{m} b_j \int_0^t \mathrm{d}\tau_n \cdots \int_0^{\tau_2} y^j \,\mathrm{d}\tau_1 \\ = \int_0^t \mathrm{d}\tau_n \cdots \int_0^{\tau_2} u \,\mathrm{d}\tau_1 \end{aligned} \tag{2.14}$$

Given the transform to the generating-series domain described in Section 2.1.4, each repetition of an integral of y can be substituted with the x_0, and the integral of the excitation replaced with x_1. This substitution creates a direct analogue of the nonlinear differential equation in the generating-series domain,

$$\left(1 + \sum_{i=0}^{n-1} l_i x_0^{n-i}\right) g + x_0^n \sum_{j=2}^{m} b_j g^{\sqcup\!\sqcup j} = x_0^{n-1} x_1 \tag{2.15}$$

Naturally, there are very strong parallels between equations (2.13) and (2.15). However, a notable alteration involves a bound shift on the first summation to factorise g out as $l_n = 1$. The most intriguing part about the generating-series domain form is that the differential equation has been reduced to a purely algebraic problem. The coefficient of g represents a polynomial in x_0, which can subsequently be factorised into a product of its roots,

$$1 + \sum_{i=0}^{n-1} l_i x_0^{n-i} = \prod_{i=0}^{p} (1 - a_i x_0)^{\alpha_i} \tag{2.16}$$

where $\sum_{i=0}^{p} \alpha_i = n$. Substituting the result from equation (2.16) into equation (2.15) and then rearranging for g gives,

$$g = \frac{x_0^{n-1} x_1}{\prod_{i=0}^{p} (1 - a_i x_0)^{\alpha_i}} - \frac{x_0^n \sum_{j=2}^{m} b_j g^{\sqcup\!\sqcup j}}{\prod_{i=0}^{p} (1 - a_i x_0)^{\alpha_i}} \tag{2.17}$$

The division of this equation into two terms is purposeful. The first term represents the linear approximation to the Volterra series, as it does not contain g. Whereas, the second term contains repeated shuffles of g. Equation (2.17) provides the basis for an inductive iterative scheme, where the first term is the initial term, and the second term provides the method of deducing higher-order terms. This iterative method closely aligns with a perturbation expansion, a technique commonly used in nonlinear dynamics [8]. A solution for g is simply the sum of all the generating-series terms created over this iteration,

$$g = g_0 + g_1 + \cdots + g_k \tag{2.18}$$

Here, k signifies the number of iterations computed of the iterative scheme. Naturally, the greater the number of terms, the greater the accuracy for convergent series approximations.

In order to begin the iterative scheme, the first term, g_0 is taken to be the linear term from equation (2.17) and the iterative expansion is defined by expanding the shuffles of the second term in equation (2.17). This results in the first term of the iterative scheme of the form,

$$g_0 = \frac{x_0^{n-1} x_1}{\prod_{i=0}^{p} (1 - a_i x_0)^{\alpha_i}} \tag{2.19}$$

and higher-order terms are determined by the following,

$$g_{i+1} = -\frac{x_0^n \sum_{j=2}^{m} b_j \sum_{\nu_1+\cdots+\nu_j=i} g_{\nu_1} ⧢ \ldots ⧢ g_{\nu_j}}{\prod_{i=0}^{p} (1 - a_i x_0)^{\alpha_i}} \tag{2.20}$$

The term $\sum_{\nu_1+\cdots+\nu_j=i} g_{\nu_1} ⧢ \ldots ⧢ g_{\nu_j}$ is the sum over all the permutations of the previous generating-series expansions that sum to i, the current iteration depth. Following the expansion of the shuffle products and reverse transforming the generating series to the time domain, this procedure allows for series expansions of nonlinear differential equations being computed entirely algebraically.

Before proceeding, some subtleties need to be addressed that will aid further steps. Equation (2.19) states that the first term g_0 is of a particular form $R_0 \cdot x_{i_1} \cdot R_1 \cdot x_{i_2} \cdot \ldots \cdot x_{i_p} \cdot R_p$, where R_j is a rational fraction in x_0 and $\{i_1, i_2, \ldots, i_p\} \in \{0, 1\}$. Additionally, the iterative scheme is also of a similar form. Meaning that if the first term is of this form, and the formula to create new terms preserves this form; all the generating-series terms derived from this iterative scheme maintain the same structure. Consequently, all terms in the expansion will adopt the form:

$$\frac{x_{i_1}}{(1-a_1x_0)^{\alpha_1}}\cdot\ldots\cdot\frac{x_{i_q}}{(1-a_qx_0)^{\alpha_q}} \tag{2.21}$$

For simplicity further down the line, reducing the integer exponents α to unity will allow for a more concise definition of the shuffle product of two generating series. Fortunately, this issue is not too contentious, as repeated applications of the identity shown in equation (2.22) will reduce any exponent,

$$\frac{1}{(1-ax_0)^{\alpha}}\equiv\frac{1}{(1-ax_0)^{\alpha-1}}+\frac{ax_0}{(1-ax_0)(1-ax_0)^{\alpha-1}} \tag{2.22}$$

Repeated applications allow any integer exponent to be reduced to unity, thus paving the way for an easier shuffle-product expansion of generating series. As observed throughout this clarification, the application of the equation (2.22) preserves the form within the terms. After a full decomposition, such that all exponents are reduced to unity, each generating-series term is of the form,

$$\frac{\lambda}{1-a_0x_0}\cdot\frac{x_{i_1}}{1-a_1x_0}\cdot\ldots\cdot\frac{x_{i_n}}{1-a_nx_0} \tag{2.23}$$

where λ is the coefficient of the term. After this abstract work in the generating-series domain, it is nice to relate back to the Volterra series and time domain. The generating series form in equation (2.23) is equivalent to the following in the time domain [13, 17].

$$\lambda\int_0^t\int_0^{\tau_n}\cdots\int_0^{\tau_2}e^{a_0(t-\tau_n)}u_{i_1}(\tau_n)e^{a_1(\tau_n-\tau_{n-1})}\cdots e^{a_{n-1}(\tau_2-\tau_1)}u_{i_n}(\tau_1)e^{a_n\tau_1}\mathrm{d}\tau_1\ldots\mathrm{d}\tau_n \tag{2.24}$$

2.1.6 Shuffle Product of Generating Series

To define the shuffle product of two generating series, it is best to assume the generating series form of rational fractions and the letters x_0 and x_1, as described at the end of Section 2.1.5. Consequently, the shuffle-product expansion of generating-series terms is more involved than the ones shown in Section 2.1.3.

Given two generating series g_1^p and g_2^q, with lengths p and q respectively. The forms of these generating series are given by:

$$g_1^p = \frac{\lambda_1}{1-b_0x_0} \cdot \frac{x_{i_1}}{1-b_1x_0} \cdot \ldots \cdot \frac{x_{i_p}}{1-b_px_0} = g_1^{p-1} \cdot \frac{x_{i_p}}{1-b_px_0} \tag{2.25}$$

$$g_2^q = \frac{\lambda_2}{1-d_0x_0} \cdot \frac{x_{j_1}}{1-d_1x_0} \cdot \ldots \cdot \frac{x_{j_q}}{1-d_qx_0} = g_2^{q-1} \cdot \frac{x_{j_q}}{1-d_qx_0} \tag{2.26}$$

Where λ_i denote the coefficients of the generating series. The earlier work ensuring that all terms in the iterative expansion share this form allows for the shuffle product to be well-defined for all iteration depths. The shuffle product of two generating series is recursively defined along its length and terminates when both generating have length 0, much like equation (2.7). For the shuffle product of the two generating series in equations (2.25) and (2.26) would require $p+q$ reductions.

To examine the shuffle product, take the two generating series with the end letter and fraction taken out from each [13].

$$\begin{aligned}
g_1^p \shuffle g_2^q &= g_1^{p-1} \cdot \frac{x_{i_p}}{1-b_px_0} \shuffle g_2^{q-1} \cdot \frac{x_{j_q}}{1-d_qx_0} \\
&= \left[g_1^{p-1} x_{i_p} \left(1 + \frac{b_px_0}{1-b_px_0} \right) \right] \shuffle \left[g_2^{q-1} x_{j_q} \left(1 + \frac{d_qx_0}{1-d_qx_0} \right) \right] \\
&= \left[\frac{x_{i_p}}{1-b_px_0} g_1^{p-1} \shuffle g_2^{q-1} \right] x_{j_q} + \left[g_1^{p-1} \shuffle g_2^{q-1} \frac{x_{j_q}}{1-d_qx_0} \right] x_{i_p} \\
&\qquad + \left[\frac{x_{i_p}}{1-b_px_0} g_1^{p-1} \shuffle g_2^{q-1} \frac{x_{j_q}}{1-d_qx_0} \right] b_px_0 \\
&\qquad + \left[\frac{x_{i_p}}{1-b_px_0} g_1^{p-1} \shuffle g_2^{q-1} \frac{x_{j_q}}{1-d_qx_0} \right] d_qx_0 \\
&= (g_1^{p-1} \shuffle g_2^q) \underbrace{\frac{x_{i_p}}{1-(b_p+d_q)x_0}}_{g_1 \text{ reduction term}} + (g_1^p \shuffle g_2^{q-1}) \underbrace{\frac{x_{j_q}}{1-(b_p+d_q)x_0}}_{g_2 \text{ reduction term}}
\end{aligned} \tag{2.27}$$

The denominators of the two reduction terms are the same and are formed by applying the identity [13]:

$$\frac{1}{1-bx_0} \shuffle \frac{1}{1-dx_0} \equiv \frac{1}{1-(b+d)x_0} \tag{2.28}$$

The numerator of the reduction term is simply the last letter of the term being reduced.

The shuffle product is now defined in a succinct recursive manner over the lengths of the two generating series (much like equation (2.7)). From equation (2.27), the shuffle product is shown to be commutative over generating series, as reversing the order only impacts the order of addition. Commutativity will allow for optimisations at a later stage.

Adopting an Array Form

To enable a more compact representation of the generating series, an array form is adopted as a notational aid. This form is particularly useful for expanding the shuffle products inside a computer program, where the shuffle product can then be defined as a sequence of array manipulations. Given the generating series from equations (2.25) and (2.26) their corresponding array forms are as follows:

$$g_1^p = \begin{bmatrix} \lambda_1 & x_{i_1} & x_{i_2} & \cdots & x_{i_{p-1}} & x_{i_p} \\ b_0 & b_1 & b_2 & \cdots & b_{p-1} & b_p \end{bmatrix}, \quad g_2^q = \begin{bmatrix} \lambda_2 & x_{j_1} & x_{j_2} & \cdots & x_{j_{q-1}} & x_{j_q} \\ d_0 & d_1 & d_2 & \cdots & d_{q-1} & d_q \end{bmatrix} \tag{2.29}$$

Example 2.1.3. It is useful to imagine the expansion of the shuffle product as a binary tree, a very natural form arising from recursing in this way. In this representation, the root node denotes the start point of the shuffle expansion, whilst the left and right children imply reductions in g_1 and g_2, respectively. At each node, a reduction term is placed, created from equation (2.27). In array form, the g_1 reduction term is constructed by removing the final column of g_1 and adding the last denominator coefficient in g_2; g_2's reduction term is calculated similarly. The result of the shuffle expansion is the cumulative sum of these concatenated reduction terms along all branches.

The bottom layer of the tree is slightly different from the rest, as no letters remain in either generating series, therefore a reduction is not possible. In this case, equation (2.28) is applied to the fraction terms, and the coefficients are multiplied. Given the two generating series,

$$g_1^2 = \frac{\lambda_1}{1 - b_0 x_0} \cdot \frac{x_{i_1}}{1 - b_1 x_0} \cdot \frac{x_{i_2}}{1 - 0 \cdot x_0}, \quad g_2^1 = \frac{\lambda_2}{1 - d_0 x_0} \cdot \frac{x_{j_1}}{1 - 0 \cdot x_0} \tag{2.30}$$

These are represented in their array forms as,

$$g_1^2 = \begin{bmatrix} \lambda_1 & x_{i_1} & x_{i_2} \\ b_0 & b_1 & 0 \end{bmatrix}, \quad g_2^1 = \begin{bmatrix} \lambda_2 & x_{j_1} \\ d_0 & 0 \end{bmatrix} \tag{2.31}$$

The shuffle product of g_1^2 and g_2^1, using the array and tree representations, is as follows,

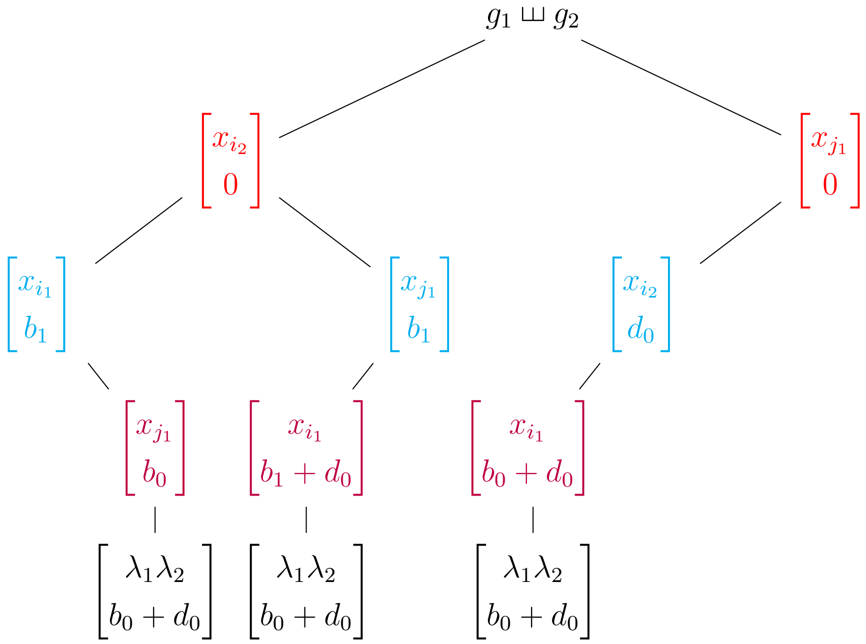


Figure 2.1: A binary tree used to help demonstrate the shuffle product of two generating series. Red, cyan, purple, and black indicate their depths, where they reflect the first, second, third, and final-constant reductions, respectively.

Figure 2.1 extends equation (2.27) to fully compute the shuffle product of two generating series. Performing the concatenation along the branches and summing the terms gives,

$$\begin{aligned} g_1 ⧢ g_2 = &\begin{bmatrix} \lambda_1\lambda_2 & x_{j_1} & x_{i_1} & x_{i_2} \\ b_0 + d_0 & b_0 & b_1 & 0 \end{bmatrix} + \begin{bmatrix} \lambda_1\lambda_2 & x_{i_1} & x_{j_1} & x_{i_2} \\ b_0 + d_0 & b_1 + d_0 & b_1 & 0 \end{bmatrix} \\ &+ \begin{bmatrix} \lambda_1\lambda_2 & x_{i_1} & x_{i_2} & x_{j_1} \\ b_0 + d_0 & b_1 + d_0 & d_0 & 0 \end{bmatrix} \end{aligned} \tag{2.32}$$

Here, the three terms are formed by concatenating all the reduction terms along the three branches of the tree.

Further collecting of generating-series terms may be possible, depending on the values i_n, j_n, b_n, d_n. The generating-series terms can only be collected when all the letters and denominator coefficients are identical; in this situation, the λ coefficients of the like generating series are added.

It is interesting to note the strong overlaps between the shuffle-product expansion and combinatorics, more specifically lattice paths. From this theory, the number of branches in the tree formed from the shuffle-product expansion of the two generating series can be succinctly expressed as the number of unique paths formed over a lattice. This being, with generating series lengths p and q the number of terms formed from the shuffle-product expansion is $\frac{(p+q)!}{p!\,q!}$. The parallel between the shuffle-product expansion and lattices is further explored in Section 2.2.1 and allows for huge computational optimisations in the implementation of the shuffle product.

n **Shuffles**

All prior instances have been strictly concerned with the application of one shuffle product. However, it is imperative to acknowledge the expansions for repeated applications of the shuffle product. This step will be required when computing the cubic nonlinearity shown in Section 2.3. The extension from a single to multiple applications is trivial as the shuffle product is operating on the words inside of a monoid algebraic structure, in which associativity is axiomatic [28]. Therefore, to determine the higher-order shuffles, the result of one shuffle expansion is passed in as an argument into the subsequent shuffle, and so on.

2.1.7 Laplace-Borel Transforms

The *Laplace-Borel transform* and its associated inverse transform are methods of converting closed-form expressions from the time domain to the generating-series domain, and vice versa. While the transformation into the generating-series domain was largely been discussed in Section 2.1.4, the specifics of transforming the excitation were not discussed. Additionally, the inverse transform out of the generating-series domain, resulting in the Volterra series, has also not been discussed.

At this point, it should be noted that there is not always a one-to-one mapping between g_i and y_i. To determine specific values for each Volterra term y_i, the generating series must be grouped by their number of x_1 terms. The generating-series terms that contain p x_1 letters relate to the Volterra term $y_p(t)$. This discrepancy arises from the fact the iterative scheme (equation (2.20)) may contain multiple shuffle expansions. Such is the case in Section 2.3, where a cubic nonlinearity is considered.

The generating series allows for the calculation concerning any excitation where a Laplace-Borel transform can be determined. During the process of iterating the generating series, the excitation is represented by the letter x_1; corresponding to a general excitation $u(t)$. Once the generating series' have been inductively expanded using the shuffle product, the term $g_{u(t)} \sqcup\!\sqcup \cdot$ is substituted in place of x_1 [9, 13], where $g_{u(t)}$ denotes the Laplace-Borel transform of $u(t)$. Since the computationally-expensive iterative expansion is done generally, different excitations can be substituted giving multiple solutions to different excitations at a relatively-low computational cost.

$q(t)$	$g_{q(t)}$
Unit step	1
$\dfrac{t^n}{n!}$	x_0^n
$\left(\sum_{i=0}^{n-1}\binom{i}{n-1}\dfrac{a^i t^i}{i!}\right)e^{at}$	$(1-ax_0)^{-n}$
$\cos(\omega t)$	$(1+\omega^2 x_0^2)^{-1}$

Table 2.1: Standard Laplace-Borel Transforms.

For the case of the impulse response, this task is not as trivial, since a Laplace-Transform of the Dirac-delta function cannot be defined. However, the process of determining an impulse response using generating series is still possible. In this case, a certain form of the generating series results directly when setting $u(t) = A\delta(t)$ in the time domain. Using a Dirac-delta excitation results in a definition over the length of the generating series; this result is derived from first principles and is discussed in much greater detail in Section 2.1.8.

A selection of standard Laplace-Borel transforms is shown in Table 2.1. These are used for transforming the excitation, $u(t)$, and for the inverse transform of generating series into the time domain. More intricate Laplace-Borel transforms may be constructed as a composition of these [13], or derived from first principles [16, 17, 29].

After processing the excitation letter x_1, the generating series consist solely of x_0

terms, yet remain in the form of a product of rational fractions in x_0; which is not a form displayed in Table 2.1. The generating-series terms must be decomposed into partial fractions until all the generating-series terms conform to standard inverse Laplace-Borel transforms. Following the partial fractions decomposition, the generating series can be readily inverse transformed into the time domain, yielding the response of the system.

2.1.8 Impulse Response

In circumstances where the Laplace-Borel transform does not exist, such as the impulse response, the usual approach of substituting the transformed excitation and expanding the shuffle products no longer applies. This scenario is similar to determining the statistics of the response for Gaussian White Noise [17, 21]. While determining the response of nonlinear differential equations subject to an impulse response is still possible, it requires a little more care and attention.

The general approach is to substitute the excitation $u(t) = A\delta(t)$ into the time domain form of a general generating-series term (as shown in equation (2.34)). This form is then manipulated primarily by making use of the sifting property of Dirac-delta functions; whilst still considering the parallel generating-series terms. Consequently establishing the structure of a typical term related to an impulse response in the generating-series domain, without explicitly determining the Laplace-Borel transform. This process enables the filtering of generating-series terms following this dual time-generating-series-domain framework. To deduce the generating series representation of the impulse response, the Dirac-delta function's sifting property [30] is extensively used,

$$\int_{-\infty}^{+\infty} \delta(x) f(x)\,\mathrm{d}x = f(0) \tag{2.33}$$

For this derivation, firstly consider the time domain analogue of the generating-series term as depicted in equation (2.23); represented by the repeated integral [13],

$$\int_0^t \cdots \int_0^{\tau_2} e^{a_0(t-\tau_n)} u_{i_1}(\tau_n) \cdots u_{i_{n-1}}(\tau_2) e^{a_{n-1}(\tau_2-\tau_1)} u_{i_n}(\tau_1) e^{a_n\tau_1}\,\mathrm{d}\tau_1 \ldots \mathrm{d}\tau_n \tag{2.34}$$

Where for the case of an impulse excitation, $u_0(t) = 1$ and $u_1(t) = A\delta(t)$. The first case to consider is when $i_1 = 0$. In this case, the result is the same as when determining the GWN excitation [17, 18, 21], since x_0 does not contribute towards the excitation. The integral may be separated from the others, giving

$$\int_0^t e^{a_0(t-\tau_n)}\,\mathrm{d}\tau_n \int_0^{\tau_n} \cdots\,\mathrm{d}\tau_1 \ldots \mathrm{d}\tau_{n-1} \leftrightarrow \frac{x_0}{1+a_0x_0}\cdot\ldots \tag{2.35}$$

The equivalent generating-series domain term is also shown. The process of removing the first integral may be iteratively performed along the length of every instance of the letter x_0.

When $i_1 = 1$, two potential scenarios may arise. The first involves the generating series forming a sequence of x_1 letters to the end; for instance $i_1 = i_2 = \cdots = i_n = 1$. In the time domain, $A\delta(t)$ replaces all $u_j(t)$ as all letters correspond to the excitation letter x_1. The sifting property then simplifies the integration, with the corresponding generating-series equivalent shown as,

$$\frac{A^n}{n!}e^{a_0t} \leftrightarrow \frac{A^n}{n!}\frac{1}{(1+a_0x_0)} \tag{2.36}$$

The second case is when $i_1 = 1$ and there exists a p where $p > 1$ and $i_p = 0$. This case is perhaps best summarised by considering the last integral where $i_n = 0$

$$\int_0^t e^{a_n\tau}\,\mathrm{d}\tau = \frac{1}{a_n}\left[e^{a_nt} - 1\right] \tag{2.37}$$

Now, there is a difference between the two terms, formed from evaluating the integral corresponding to x_0. Therefore, if an x_1 precedes this in the generating-series domain, which relates to integration with a Dirac-delta function in the time domain, they cancel to zero upon evaluating the sifting property,

$$\int_0^t (e^{a_n\tau} - 1)\delta(\tau)\,\mathrm{d}\tau = 0 \tag{2.38}$$

Therefore reducing this case and all subsequent integrals to 0. In this context, an x_1 term immediately precedes the x_0. However, there is potential for any number of x_0 letters before the x_1. This condition doesn't pose a problem to the logic as the $f(\tau_i)$ in the sifting property definition will be some combination of $e^{a}_{n-i}\tau_i$ and constant terms and when evaluating at $f(0)$, the terms will likewise cancel, resulting in 0 in every case.

Taking into account the previous scenarios, these are sufficient to define the structure of a generating-series term that contributes to the impulse response. By ensuring that all generating series are of this form, the impulse response of the system is determined. Thus, performing this analysis bypasses the need for a direct Laplace-Borel transform. This analysis then allows for the definition of the impulse response of generating-series expansions:

Definition 2.1.4. Given a generating series of the form shown in equation (2.23) the function, $\mathrm{IRF}_A()$, that determines the response of a generating series to a Dirac input function $u(t) = A\delta(t)$ with amplitude A is defined implicitly over its length [19]:

$$\mathrm{IRF}_A(g_1^p) = \begin{cases} \dfrac{x_0}{(1+a_0x_0)}\mathrm{IRF}_A(g_1^{p-1}), & i_1 = 0 \\ \dfrac{A^n}{n!}\dfrac{1}{(1+a_0x_0)}, & i_1 = i_2 = \cdots = i_n = 1 \\ 0, & i_1 = 1 \text{ and } \exists p,\ p > 1,\ i_p \neq 1 \end{cases} \tag{2.39}$$

Following this, sufficient material has now been covered to determine the impulse response of nonlinear differential equations.

2.2 Optimising the Shuffle Product

To extend this analysis into the higher-order Volterra terms, optimising the shuffle product became a key requirement. Although the techniques discussed are not exhaustive, only the most significant optimisations are discussed, which can be split into two main categories: a more efficient restructuring of the original problem, and eliminating terms during iteration. The bottleneck in the calculations currently lies with the partial fraction decomposition of the generating series.

2.2.1 Tabulation

Given two generating series g_1^p and g_2^q, with lengths p and q, programmatically recursing over the shuffle-product expansion tree (Figure 2.1) becomes unwieldy, even for small p and q. Furthermore, recursion involves extra computational overheads associated with large function stacks. In addition, Python also implements a recursion depth limit, which is fast-approached in this analysis. To bypass these issues a *tabulation* approach was implemented.

By tabulation, the tree formed from expanding the shuffle product is remodelled as a $2D$ grid. The coordinate (i, j) then relates to i reductions of g_1^p and j reductions

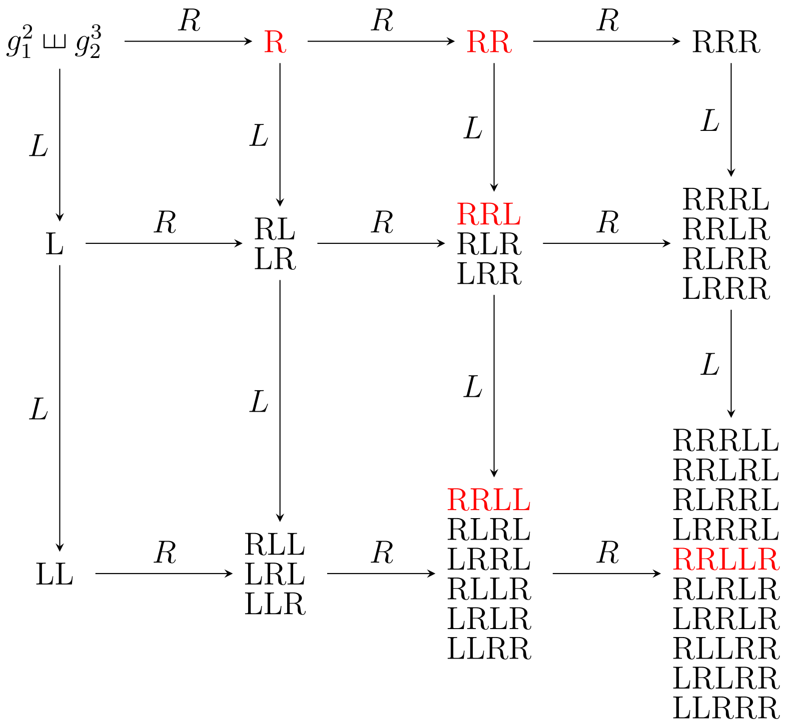


Figure 2.2: For the case when $p = 2$ and $q = 3$, a diagram for tabulation of shuffle product. L and R denote reductions in the left and right hand generating series, respectively.

in g_2^q. Consequently, the grid used to expand the shuffle product contains $p+1$ rows and $q+1$ columns. This reformatting then changes the recursive problem into a nested 'for' loop over the lengths of the two generating series. At each grid node, the left and right reduction terms, as outlined in equation (2.27), are concatenated with the terms currently at the node. The terms formed from the left-hand term concatenation are pushed to the adjacent row beneath, and the result formed from the right-hand term concatenation are pushed to the adjacent column, as demonstrated in Figure 2.2. Therefore, repeating this process by iterating over the grid, all possible unique paths are traversed. The notion of covering all the unique paths is analogous to determining all possible shuffles of two generating series. As a demonstrative aid, the grid nodes shown in Figure 2.2 display the paths to get to that node; in actuality, the concatenations of terms are stored there. The final grid node at coordinates (p, q), contains the concatenation of terms formed over all possible reduction paths; identical to the tree shown in Figure 2.1.

The substantial efficiency gains of the tabulation approach stem from grouping the concatenations and the reduction terms, in contrast to the tree approach where reduction terms are only concatenated with one term at each node. Therefore, the computational complexity of the tree and grid cases is proportional to the number of nodes in the two representations, as each node requires the calculation of a reduction term. With the grid approach the number of nodes is $n_{\text{grid}}(p, q) = (p + 1) \cdot (q + 1)$. Whereas the number of nodes in the tree form is represented by

$$n_{\text{tree}}(p, q) = \binom{p+q+2}{p+1} - 1 = \frac{(p+q+2)!}{(p+1)!\,(q+1)!} - 1 \qquad (2.40)$$

Unsurprisingly, the tree approach contains a considerably greater number of nodes, leading to many repetitions when calculating the reduction terms.

Reformulating the shuffle product using tabulation changes the shuffle product from recursive to non-recursive, thereby skirting the recursion depth limit and additional function stack overheads. Most importantly, tabulation reduces the computational complexity from combinatorial to quadratic, significantly reducing the number of required calculations.

2.2.2 Memoisation

When working with higher-order expansions of the generating series, identical arguments are frequently passed into the shuffle product. For this reason, a lookup table over the shuffle product provides a significant reduction in computation time. A hash function defined over the generating series acts as a key for the lookup table. Before determining the shuffle product, the key representing the shuffle-product expansion is checked against a lookup table. If no value is present at that key, the shuffle product is expanded and its value is stored at this position. If a value is present in the lookup table, this is returned and the calculation is bypassed. This becomes especially important for larger generating series and/or a high number of repeated shuffles.

As highlighted in Section 2.1.6, the shuffle product being commutative and associative allows for the arguments of repeated shuffles to be sorted without changing the results. Therefore, before expanding the shuffle product, it is useful to sort the generating series passed in as arguments based on their hash values. The sorted hash values are then used as a key in a lookup table. First sorting the generating series before checking the lookup table will increase the number of lookups, consequently decreasing the number of expensive shuffle-product expansions. For instance, when expanding a shuffle product relating to a cubic nonlinearity for g_2, the summation criteria will be: $\nu_1 + \nu_2 + \nu_3 = 1$. There are three permutations of this calculation of (ν_1, ν_2, ν_3) that satisfy this criterion, which are $(0, 0, 1)$, $(0, 1, 0)$, $(1, 0, 0)$. By sorting and caching the results, it means that only one of these permutations is expanded, with the other two being retrieved via memoisation.

2.2.3 Remove Non-Impulse Form

Defintion 2.1.4 allows for a significant computational optimisation by leveraging the preservation of letter order within the arguments of the shuffle product. As stated in Defintion 2.1.4, if x_0 ever succeeds x_1, this term is reduced to zero. Consequently, the words of the generating series corresponding with an impulse response *must* be of the form,

$$\underbrace{x_0 \ldots x_0}_{r\text{-times}} \underbrace{x_1 \ldots x_1}_{s\text{-times}} \tag{2.41}$$

where the term represented by this form will contribute to the Volterra term y_s. Since the shuffle product represents the riffle shuffle between two generating series, as highlighted in Sections 2.1.3 and 2.1.5, the order of the letters within one shuffle product argument does not permute. Therefore, it can be determined that any generating-series terms where all the x_0 letters do not precede all the x_1 terms will be reduced to zero, along with all its corresponding terms created in the shuffle-product expansion, according to Definition 2.1.4.

By discarding terms that contradict the form in equation (2.41) during the iterative process, a substantial proportion of terms are removed from the shuffle-product expansion, significantly reducing the number of shuffle-product expansions, as described in the case study in Section 2.3. However, a limitation of this optimised approach restricts the calculations strictly to the impulse response. Despite this limitation, this optimised iterative expansion is instrumental for determining higher-order terms in the Volterra series.

2.2.4 Filter Large Generating Series

As mentioned in Section 2.1.7, when generating series are transformed back into the time domain, the highest fully-complete Volterra term from i iterations is y_{i+1}. Since the iteration depth is known at the point of iteration, any shuffle-product calculations where the x_1 count of the two shuffle-product arguments is greater than $i + 1$ are deemed unnecessary, as the Volterra term they contribute towards will be incomplete. Not considering such expansions is especially important, as two generating series with a x_1 count greater than the iteration depth will be among the largest generating series in the expansion, and consequently, the most expensive to compute.

2.3 Nonlinear Oscillator Impulse Response

To supplement the theoretical walkthrough presented in the previous section, a nonlinear SDOF oscillator with both quadratic and cubic stiffness nonlinearities is considered. This model is a more advanced extension of Duffing's equation [11] to also include an asymmetric quadratic term,

$$m\frac{\mathrm{d}^2y}{\mathrm{d}t^2} + c\frac{\mathrm{d}y}{\mathrm{d}t} + k_1y + k_2y^2 + k_3y^3 = u(t) \tag{2.42}$$

where m, c, k_1, k_2, and k_3 are the mass, damping, linear stiffness, quadratic stiffness, and cubic stiffness coefficients. The excitation $u(t) = A\delta(t)$ is an impulse excitation with amplitude AN. To simplify this problem, zero-initial conditions are assumed. Firstly, equation (2.42) is converted into integral form,

$$\begin{aligned} my(t) + c\int_0^t y(\tau)\,\mathrm{d}\tau + k_1\int_0^t\int_0^\tau y(\sigma)\,\mathrm{d}\sigma + k_3\int_0^t\int_0^\tau y^3(\sigma)\mathrm{d}\sigma + k_2\int_0^t\int_0^\tau y^2(\sigma)\,\mathrm{d}\sigma \\ = \int_0^t \mathrm{d}\tau\int_0^\tau u(\sigma)\,\mathrm{d}\sigma \end{aligned} \tag{2.43}$$

This expression is then transformed into the generating-series domain by substituting the letters x_0 and x_1 in place of their respective integrals. Furthermore, the variable g is also factored out, leading to a polynomial in the x_0 coefficient,

$$g(m + cx_0 + k_1x_0^2) = x_0x_1 - k_2x_0^2\, g \sqcup\!\sqcup\, g - k_3x_0^2\, g \sqcup\!\sqcup\, g \sqcup\!\sqcup\, g \tag{2.44}$$

Readers acquainted with the structural dynamics literature might find the quadratic in x_0 peculiar because of the k_1 leading coefficient; typically, m is expected. However, this discrepancy is not problematic when compared to traditional approaches. Usually when factorising the polynomial, the desired form would be $(x - r_1)(x - r_2)$, whereas in this application the form $(1 - a_1x)(1 - a_2x)$ is required. To illustrate the equivalence between the two forms consider their expansions,

$$(x - r_1)(x - r_2) \tag{2.45}$$

$$x^2 - (r_1 + r_2)x + r_1 r_2 \tag{2.46}$$

$$(1 - a_1 x)(1 - a_2 x) \tag{2.47}$$

$$a_1 a_2 x^2 - (a_1 + a_2) + 1 \tag{2.48}$$

The coefficients in equations (2.46) and (2.48) are reversed. Therefore, to calculate the values of a_1 and a_2 in equation (2.44), the quadratic formula is applied to the coefficients in reverse order, giving the familiar form

$$a_1, a_2 = \frac{-c \pm \sqrt{c^2 - 4mk_1}}{2m} = \left(-\zeta \pm \sqrt{\zeta^2 - 1}\right) \omega_n \tag{2.49}$$

For a more in-depth and generalised analysis of polynomials greater than degree two, the theory regarding reciprocal polynomials should be consulted [31]. With the polynomial factorised and the values a_1 and a_2 well-defined, equation (2.44) is rearranged into its iterative-inductive scheme, adopting the form $g_{i+1} = f(g_i)$:

$$g_{i+1} = -\underbrace{\frac{x_0}{(1 - a_1 x_0)} \frac{x_0}{(1 - a_2 x_0)}}_{\text{Iterative Multiplier}} \left[\underbrace{k_2 \cdot \sum_{\mu_1 + \mu_2 = i} g_{\mu_1} \shuffle g_{\mu_2}}_{\text{Quadratic Nonlinearity}} + \underbrace{k_3 \cdot \sum_{\nu_1 + \nu_2 + \nu_3 = i} g_{\nu_1} \shuffle g_{\nu_2} \shuffle g_{\nu_3}}_{\text{Cubic Nonlinearity}} \right] \tag{2.50}$$

with the linear base term defined as:

$$g_0 = \frac{x_0}{(1 - a_1 x_0)} \frac{x_1}{(1 - a_2 x_0)} \tag{2.51}$$

For the first iteration, two shuffle expansions need to be addressed – one for each nonlinearity. The expansion $g_0 \shuffle g_0$ relates to the quadratic nonlinearity, while $g_0 \shuffle g_0 \shuffle g_0$ applies to the cubic nonlinearity. After the expansions, all terms are concatenated with the iterative multiplier and multiplied by the respective stiffness coefficient; these terms then constitute g_1.

Calculating g_2 introduces additional complexity, arising from the multiple unique combinations that satisfy the summation criteria $\mu_1 + \mu_2 = 1$ and $\nu_1 + \nu_2 + \nu_3 = 1$. For the quadratic nonlinearity expansion the values $(\mu_1, \mu_2) = (1, 0), (0, 1)$ satisfy the condition. Similarly, $(\nu_1, \nu_2, \nu_3) = (1, 0, 0), (0, 1, 0), (0, 0, 1)$ satisfy the cubic nonlinearity criterion. For instance, with the quadratic nonlinearity expansion, both $g_0 \shuffle g_1$ and $g_1 \shuffle g_0$ must be evaluated.

The previous paragraph begins to highlight the additional computational expense involved when determining higher-order nonlinear terms. This increased expense stems from three main factors. Firstly, there are simply a greater number of shuffle products to compute. Secondly, the expansions associated with higher-order nonlinearities generate more terms, an effect that is compounded over repeated iterations. Finally, there are more possibilities to satisfy the summation criteria. Nonetheless, the mathematical and computational optimisations discussed in Section 2.2 help alleviate the computational expense for higher-order terms.

The base term g_0 and the resultant term g_1 formed from the inductive scheme are shown in their array form,

$$g_0 = \begin{bmatrix} 1 & x_0 & x_1 \\ a_1 & a_2 & 0 \end{bmatrix} \tag{2.52}$$

$$\begin{aligned}
g_1 = &\begin{bmatrix} -2k_2 & x_0 & x_0 & x_0 & x_1 & x_0 & x_1 \\ a_2 & a_1 & 2a_1 & a_1+a_2 & a_1 & a_2 & 0 \end{bmatrix} \\
&+ \begin{bmatrix} -4k_2 & x_0 & x_0 & x_0 & x_0 & x_1 & x_1 \\ a_2 & a_1 & 2a_1 & a_1+a_2 & 2a_2 & a_2 & 0 \end{bmatrix} \\
&+ \begin{bmatrix} -6k_3 & x_0 & x_0 & x_0 & x_1 & x_0 & x_1 & x_0 & x_1 \\ a_2 & a_1 & 3a_1 & 2a_1+a_2 & 2a_1 & a_1+a_2 & a_1 & a_2 & 0 \end{bmatrix} \\
&+ \begin{bmatrix} -12k_3 & x_0 & x_0 & x_0 & x_0 & x_1 & x_1 & x_0 & x_1 \\ a_2 & a_1 & 3a_1 & 2a_1+a_2 & a_1+2a_2 & a_1+a_2 & a_1 & a_2 & 0 \end{bmatrix} \\
&+ \begin{bmatrix} -24k_3 & x_0 & x_0 & x_0 & x_0 & x_1 & x_0 & x_1 & x_1 \\ a_2 & a_1 & 3a_1 & 2a_1+a_2 & a_1+2a_2 & a_1+a_2 & 2a_2 & a_2 & 0 \end{bmatrix} \\
&+ \begin{bmatrix} -12k_3 & x_0 & x_0 & x_0 & x_1 & x_0 & x_0 & x_1 & x_1 \\ a_2 & a_1 & 3a_1 & 2a_1+a_2 & 2a_1 & a_1+a_2 & 2a_2 & a_2 & 0 \end{bmatrix} \\
&+ \begin{bmatrix} -36k_3 & x_0 & x_0 & x_0 & x_0 & x_0 & x_1 & x_1 & x_1 \\ a_2 & a_1 & 3a_1 & 2a_1+a_2 & a_1+2a_2 & 3a_2 & 2a_2 & a_2 & 0 \end{bmatrix}
\end{aligned} \tag{2.53}$$

In the shuffle-product expansion, the quantity and growth rate of the terms is striking, with term counts 1, 7, 636, 219954 inside g_0, g_1, g_2, and g_3 respectively. Despite the significant number of terms, most are reduced to 0 after the application of the impulse-response criterion per Definition 2.1.4. Only a small number are significant and contribute to the impulse response, their counts being: 1, 2, 18, 456 for g_0, g_1, g_2, and g_3 respectively. Highlighting the importance of the optimisation addressing this problem, given in Section 2.2.3. The relevant terms inside g_1 that contribute to the impulse response are,

$$\begin{gathered} g_1 = \begin{bmatrix} -4k_2 & x_0 & x_0 & x_0 & x_0 & x_1 & x_1 \\ a_2 & a_1 & 2a_1 & a_1+a_2 & 2a_2 & a_2 & 0 \end{bmatrix} \\ \begin{bmatrix} -36k_3 & x_0 & x_0 & x_0 & x_0 & x_0 & x_1 & x_1 & x_1 \\ a_2 & a_1 & 3a_1 & 2a_1+a_2 & a_1+2a_2 & 3a_2 & 2a_2 & a_2 & 0 \end{bmatrix} \end{gathered} \tag{2.54}$$

In g_1 there is a single contribution from each nonlinearity. Higher-order terms, such as g_2 will begin to include a mixing term where the generating-series terms will include the coefficients k_2^2, k_2k_3, and k_3^2. However, these terms are already too long to be displayed compactly. As a result, only lower-order terms will be displayed for demonstrative purposes, but the full scope of the analysis when performed computationally includes all terms up to g_5.

The count of x_1 letters present in the generating-series term signifies its corresponding Volterra term. Once the desired expansion is determined, it is then required to sort the generating-series terms into their corresponding Volterra terms. To avoid confusing notation, these sorted terms in the generating-series domain are represented by the Laplace-Borel transform of the Volterra term, where $\mathcal{LB}(y_i)$ denotes the transformed variant of the i^{th} Volterra term. Only the Volterra terms up to one more than the iteration depth may be considered (to account for the zero-offset used throughout to represent the generating-series expansion). There will be terms in the generating-series domain with more x_1 letters than the iteration depth, as a consequence of the cubic nonlinearity. However, the Volterra terms after the iteration depth will lack all contributions and therefore, are unsuitable for consideration. For instance, in equation (2.54), the first term contains two x_1 letters and the second term contains three x_1 letters. Thus, the first term will contribute towards y_2 and the second to y_3. However, the second term does not completely constitute y_3 as the other contributions will be formed in the next iteration, g_2. Hence, highlighting the importance of the optimisation given in Section 2.2.4.

Once organised according to the appropriate Volterra terms, one can then apply Definition 2.1.4, giving the generating-series impulse response. Therein, converting the generating series into a form only consisting of x_0 letters. These terms are now compact enough to be represented in their fractional form:

$$\begin{aligned}\mathcal{LB}(y_1) &= \frac{Ax_0}{(1-a_1x_0)(1-a_2x_0)} && (2.55)\\ \mathcal{LB}(y_2) &= \frac{-2k_2Ax_0^4}{(1-a_2x_0)(1-a_1x_0)(1-2a_1x_0)(1-(a_1+a_2)x_0)(1-2a_2x_0)} && (2.56)\end{aligned}$$

Subsequently, the terms are decomposed using partial fractions. The coefficients are shown in the numerator to clarify the Laplace-Borel transform form:

$$\mathcal{LB}(y_1) = \frac{A/(a_1-a_2)}{1-a_2x_0} - \frac{A/(a_1-a_2)}{1-a_1x_0} \tag{2.57}$$

$$\begin{aligned}\mathcal{LB}(y_2) = {} & \frac{2A^2k_2/(a_1^3a_2-2a_1^2a_2^2+a_1a_2^3)}{1-(a_1+a_2)x_0} - \frac{2A^2k_2/(2a_1^3a_2-3a_1^2a_2^2+a_1a_2^3)}{1-a_2x_0}\\ & + \frac{A^2k_2/(a_1^3a_2-4a_1^2a_2^2+5a_1a_2^3-2a_2^4)}{1-2a_2x_0} - \frac{2A^2k_2/(a_1^3a_2-3a_1^2a_2^2+2a_1a_2^3)}{1-a_1x_0}\\ & + \frac{A^2k_2/(-2a_1^4+5a_1^3a_2-4a_1^2a_2^2+a_1a_2^3)}{1-2a_1x_0}\end{aligned} \tag{2.58}$$

Even for y_2, which was constructed from a single generating-series term with length four, these expansions become overwhelming, very quickly. With all terms now decomposed using partial fractions, they are in a form suitable for transformation back into the time domain. The terms shown in equations (2.57) and (2.58) are in the exponential form shown in Table 2.1. Applying the inverse Laplace-Borel transform over all these terms gives the Volterra terms:

$$y_1 = \frac{Ae^{-a_2t}}{a_1-a_2} - \frac{Ae^{-a_1t}}{a_1-a_2} \tag{2.59}$$

$$\begin{aligned}y_2 = {} & \frac{A^2k_2e^{-2a_1t}}{-2a_1^4+5a_1^3a_2-4a_1^2a_2^2+a_1a_2^3} + \frac{A^2k_2e^{-2a_2t}}{a_1^3a_2-4a_1^2a_2^2+5a_1a_2^3-2a_2^4}\\ & - \frac{2A^2k_2e^{-a_1t}}{a_1^3a_2-3a_1^2a_2^2+2a_1a_2^3} - \frac{2A^2k_2e^{-a_2t}}{2a_1^3a_2-3a_1^2a_2^2+a_1a_2^3} + \frac{2A^2k_2e^{t(-a_1-a_2)}}{a_1^3a_2-2a_1^2a_2^2+a_1a_2^3}\end{aligned} \tag{2.60}$$

The term y_3, derived using the generating series, is provided in Appendix A. The first three terms of the Volterra series were previously calculated by hand, using a contour integration approach [25], the result of which is shown in Appendix B. A comparison between these two approaches is given in Section 2.3.1 and shows they give identical results. However, the terms y_4, y_5, and y_6 have never been determined before. These terms are of special interest, as from y_4 onwards the terms exhibit a mixing between the k_2 and k_3 nonlinearities.

2.3.1 Results

The results section is divided into two parts: initially, a comparison between the Volterra series obtained using contour integration and the generating series is given. Following this, the implications of including successive Volterra terms are discussed, by considering their effect on the modelling precision at different impulse excitations. An error analysis is provided when compared to the numerical simulation of a 4th-order Runge-Kutta [12]. Here, the focus is on modelling accuracy and the counterproductive effect of including higher-order terms for this example.

However, the treatment of a Runge-Kutta solution as an absolute truth should be approached with caution. Previous work [32] comparing the generating series and Runge-Kutta solutions for different systems demonstrates that the Runge-Kutta method can yield suboptimal results because of excessive smoothing between data points. To further explore the mathematical relationship between the generating series and the Runge-Kutta solution, one might refer to [33].

The generating series method was chosen for this problem due to its flexible framework, particularly its connection to bilinear approximations, as discussed in previous works [34, 35]. This approach was preferred over techniques like Carleman linearisation, which approximates a nonlinear system by representing state variables as power series and then truncating the series to the desired length [36]. While Carleman linearisation involves approximating the system with an infinite-dimensional linear model, the generating series method provides a more effective solution for this context, by not introducing extra state variables.

The parameters defining the system behaviour are $m = 1$kg, $c = 20$N/(m/s), $k_1 = 10000$N/m, $k_2 = 1 \cdot 10^7$N/m^2, and $k_3 = 5 \cdot 10^9$N/m^3 with initial conditions $\dot{y}(0) = 0$ and $y(0) = 0$. These are the same values as used in [25] and are specifically chosen to result in an undamped natural frequency $\omega_n = 100$rad/s and a damping ratio $\zeta = 0.1$. These parameters, result in an underdamped system response, giving complex a_1 and a_2. However, this isn't an issue, as the theory of generating series holds in the complex domain [13, 19]. In the time domain, the complex components will accordingly cancel, resulting in real-valued responses; as expected for physical systems.

Comparison with Contour Integration

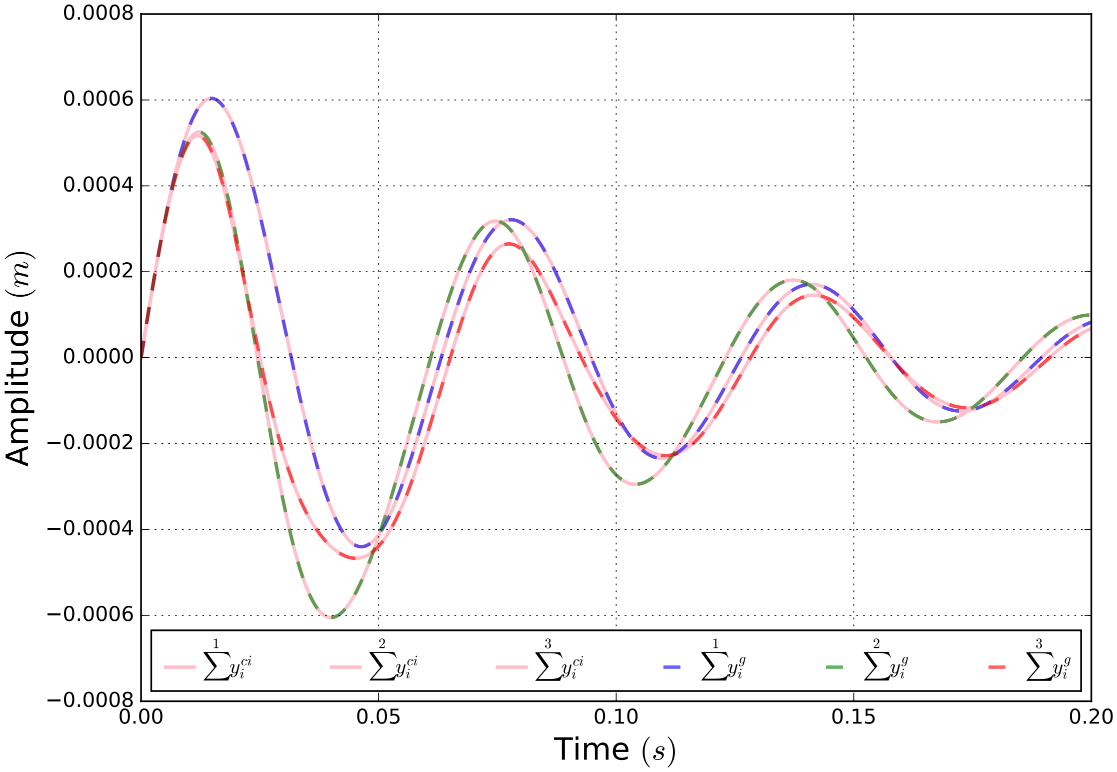


Figure 2.3: Generating series and contour integration comparison for a quadratic cubic nonlinear oscillator with $A = 0.07N$. Where ci relates to the terms from contour integration and g refers to the terms formed via the generating series.

Previous work [25] determined the Volterra series, including the terms y_1, y_2, and y_3 for the same system outlined here; these findings can be found in Appendix B. To validate the generating series and contour integration approaches for determining the Volterra series, it is useful to compare their results. Given the contrasting representations of the two forms, it might not be immediately clear that these two forms are equivalent. Nevertheless, their equivalence is shown in Figure 2.3, where there is an exact alignment over the Volterra series.

Validating the two result sets against one another highlights that two completely different approaches to determining the Volterra series give the same results. Whereas the contour integration approach was determined using pen and paper and is impractical to be taken further, the generating series method was performed entirely computationally, facilitating the determination of even higher-order terms.

Inclusion of Successive Terms

The Runge-Kutta numerical method is used as a benchmark for the error analysis. To determine the numerical solution with Runge-Kutta, equation (2.42) is required to be broken down into an initial-value problem,

$$\begin{aligned} y_1 &= y \\ \frac{\mathrm{d}y_1}{\mathrm{d}t} &= y_2 \\ \frac{\mathrm{d}y_2}{\mathrm{d}t} &= \frac{1}{m}\left(A\delta(t) - cy_2 - k_1 y_1 - k_2 y_2^2 - k_3 y^3\right) \end{aligned} \tag{2.61}$$

A time step $\mathrm{d}t = 10^{-4}$s was used. To ensure that the impulse excitation has area A, a condition outlined by the Dirac-delta function, an excitation amplitude $A/\mathrm{d}t$ is used. All calculations were executed over the time interval $[0, 1]$s. However, all plots only display $[0, 0.2]$s, to highlight the more significant sections.

To better understand the effect of varying impulse amplitude over the Volterra series, systems were calculated at logarithmically-spaced intervals from 10^{-4} to 10^{0}. Wherein the error from the first six truncations of the Volterra series is compared against the respective Runge-Kutta solution. To make this comparison more objective, the *root relative squared-error* (RRSE) between the numerical approximation and the Volterra series was computed.

$$RRSE = 100\sqrt{\frac{\sum_{i=1}^{i=N}\left(y_{v_i}(t) - y_{rk_i}(t)\right)^2}{\sum_{i=1}^{i=N}\left(\overline{y_{rk}(t)} - y_{rk_i}(t)\right)^2}} \tag{2.62}$$

Where y_{rk_i} and y_{v_i} are the i^{th} sample points of the Runge-Kutta and Volterra series solutions respectively and $\overline{y_{rk}}$ is the average of the Runge-Kutta solution. The errors associated with the first six expansions of the Volterra series are given in Figure 2.4.

Figure 2.4 shows that all the Volterra series give close to the same error until $A = 5 \cdot 10^{-3}$N. Until this point, the Volterra series are dominated by the y_1 term. The individual Volterra terms at this amplitude are shown in Figure 2.5, and it can be seen that the terms y_2 to y_6 are negligible compared to y_1; hence, y_1 dominates. However, Figure 2.5 indicates y_2 is becoming significant compared to y_1, hence the RRSE values diverging from the linear case. A similar situation arises at

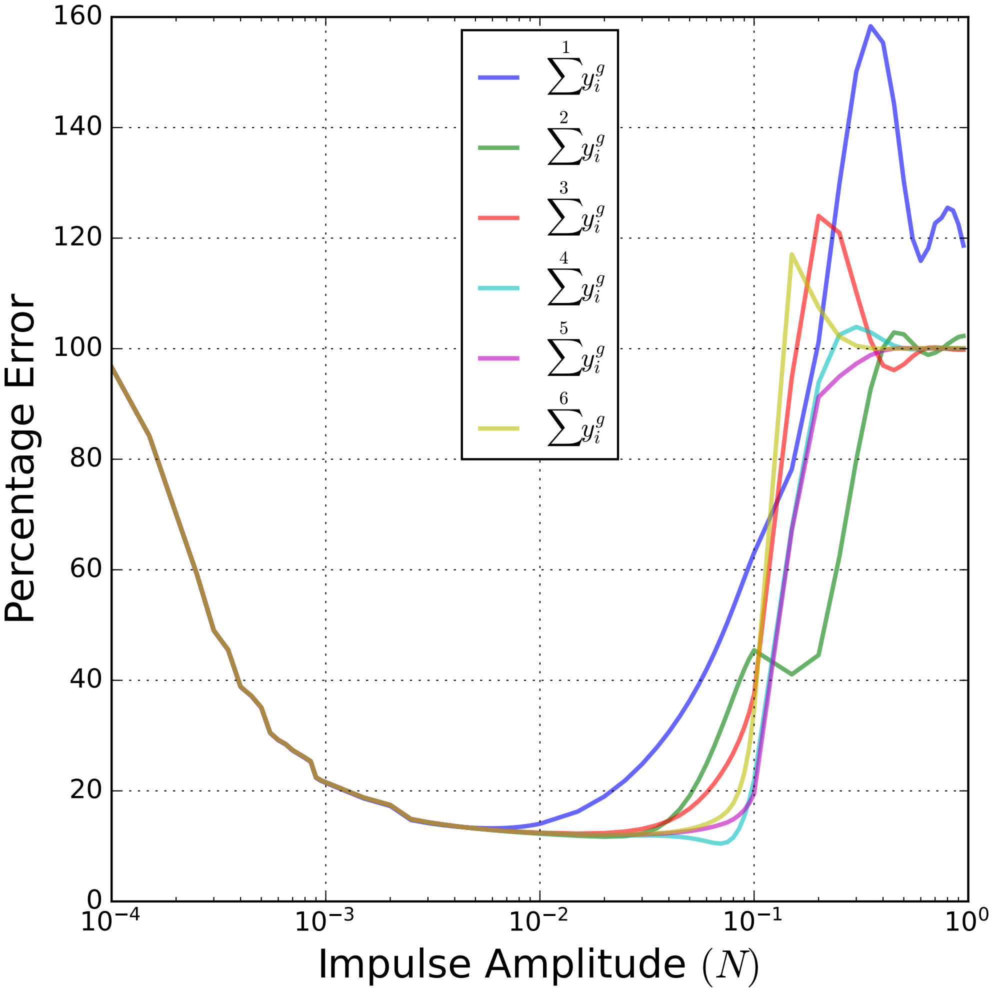


Figure 2.4: RRSE of successive Volterra series compared against a 4th order Runge-Kutta approximation of the nonlinear system.

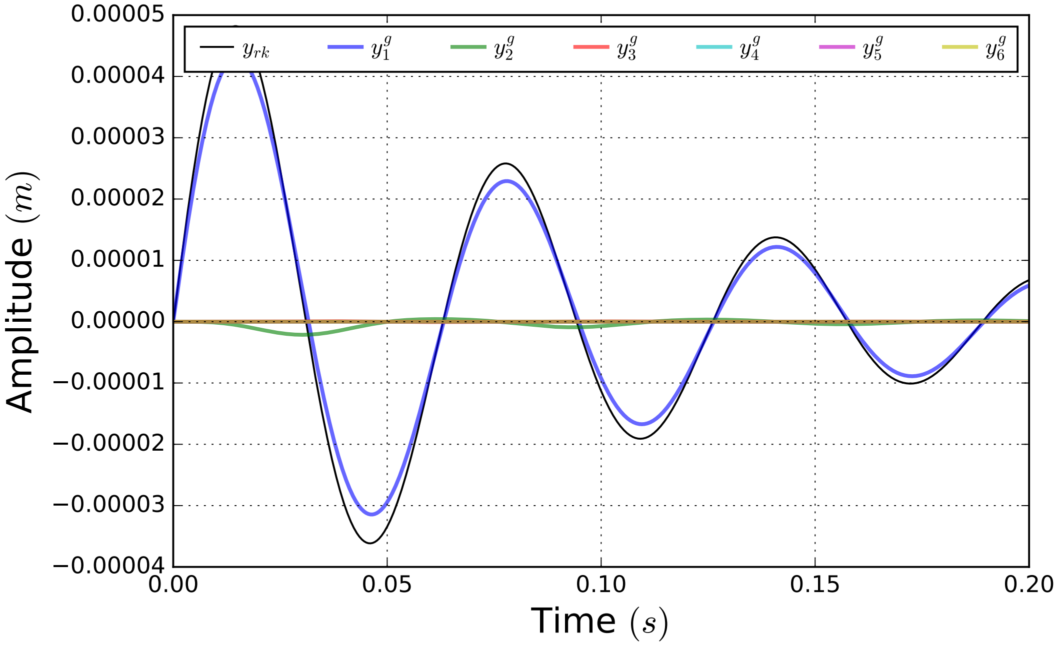


Figure 2.5: Volterra terms with impulse amplitude $A = 0.005$N; showing the Volterra series are dominated by y_1, with y_2 just becoming significant.

$A = 3 \cdot 10^{-2}$N (Figure 2.6), where until then the Volterra series have been dominated by $y_1 + y_2$. Approximately around this amplitude, all other terms become significant, therefore giving varying accuracies between the Volterra series. When the higher-order terms become significant, the higher-order series become more accurate. Owing to the A^i part of the coefficient in the Volterra series, the higher-order terms become significant at higher and higher impulse amplitudes, before subsequently becoming unstable. When increasing the impulse amplitude, the general trend of the nonlinear Volterra terms is to increase at an ever-increasing rate when compared to y_1.

At the other end of the spectrum, whilst the inclusion of additional Volterra terms provides greater accuracy at lower amplitudes, it is observed that higher-order terms become unstable at lower amplitudes. The instability arises as the higher-order Volterra terms are increasing at a significantly greater rate as a function of the impulse amplitude, and therefore reducing the interval to which all terms are stable. Specifically, at $A = 0.08$N on Figure 2.4, the inclusion of the y_5 and y_6 terms makes the Volterra series less accurate than when compared to the Volterra series capped at y_4. Additionally, the rapid increase in the error, or divergence, of the Volterra series including y_6 happens at a lower impulse amplitude than the Volterra series

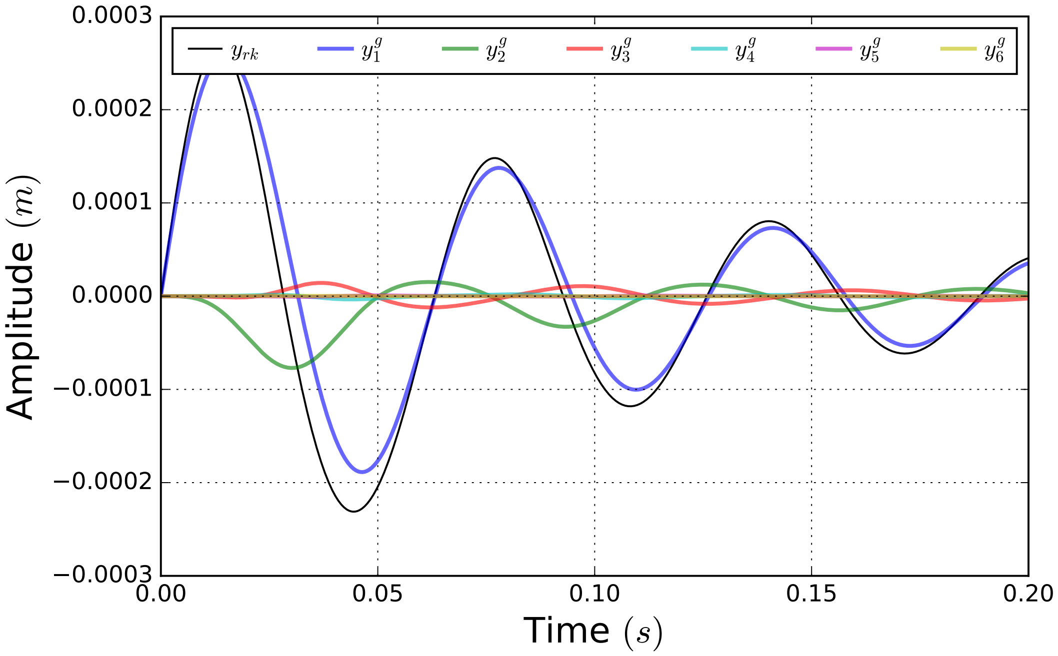


Figure 2.6: Volterra terms with impulse amplitude $A = 0.03$N; showing the Volterra series are dominated by $y_1 + y_2$, with y_3 just becoming significant.

including y_5.

Inevitably, the two propositions discussed in the preceding paragraphs must converge, at a certain impulse amplitude. Therefore, the inclusion of extra Volterra terms in the series beyond this point will no longer be as accurate when compared to the lower-order Volterra series, as the higher-order terms will be unstable for such a small region.

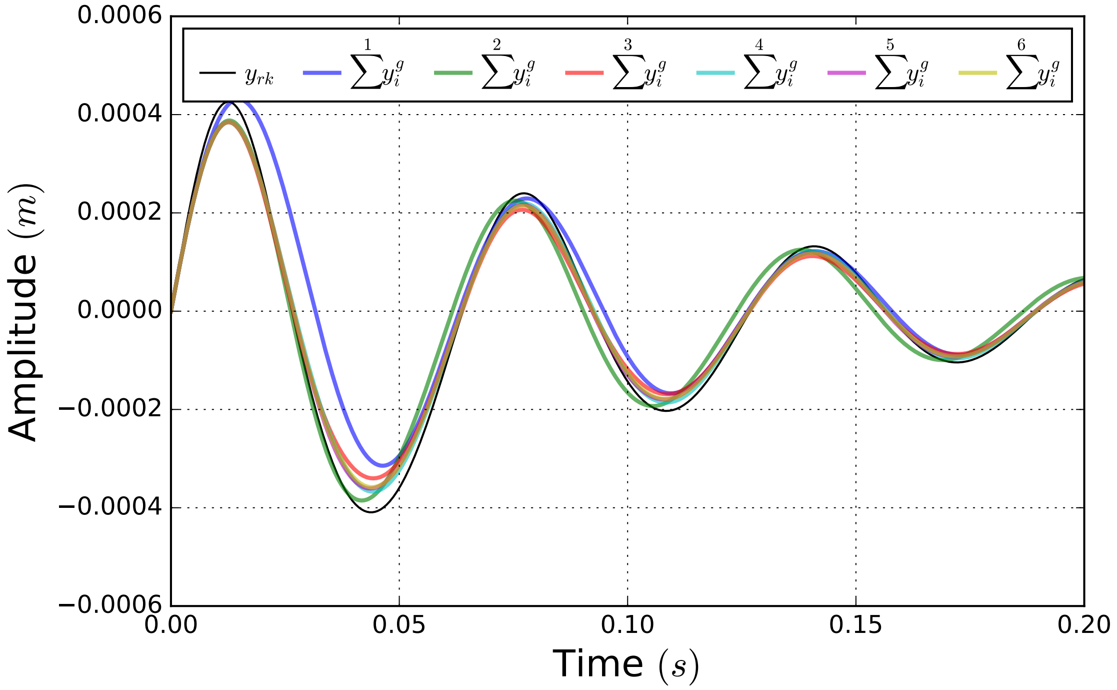


Figure 2.7: Volterra series with impulse amplitude $A = 0.05\text{N}$.

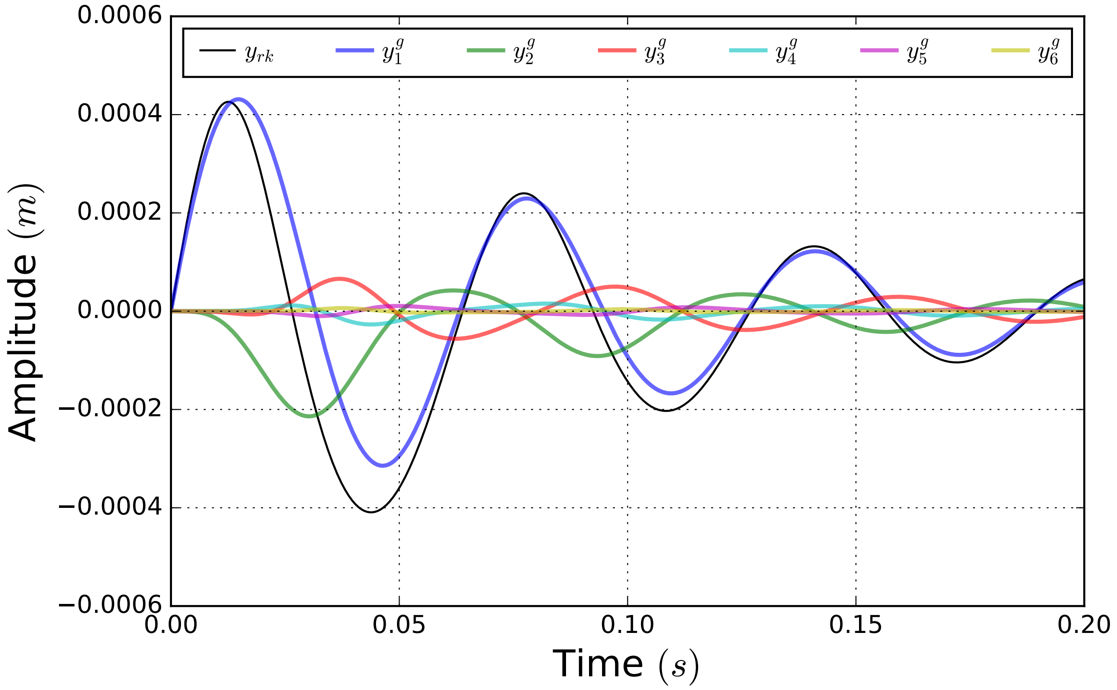


Figure 2.8: Volterra terms with impulse excitation $A = 0.05\text{N}$.

To exemplify this point, consider where $A = 0.05\text{N}$, as shown in Figures 2.7 and 2.8. At this impulse amplitude, the Volterra terms are well-behaved, where the maximum amplitudes of the terms are in decreasing order. The order of the Volterra series, being denoted by their maximum term, from most accurate to least accurate is y_4, y_5, y_6, y_3, y_2, y_1. Therefore, at this amplitude, the terms y_5 and y_6 only degrade the approximation. Figure 2.4 shows that the Volterra series including up to y_4 is always better than the higher-order Volterra series. The interest then becomes in identifying the optimal number of Volterra terms to include, as the addition of extra terms is computationally expensive and narrows the valid range of the Volterra series.

Increasing the impulse amplitude amplifies the nonlinear stiffness effects in the system, meaning the linear approximation, y_1 decreases in accuracy at greater impulse amplitudes. This effect is objectively illustrated in Figure 2.4, where the linear approximation gradually increases in error. This effect is also visually present over Figures 2.5, 2.6, 2.8, and 2.10, where the difference between the y_1 and the Runge-Kutta approximation increases as impulse amplitude increases. However, the inclusion of higher-order Volterra terms, with their ability to model nonlinear effects decreases the error arising from nonlinearities in the system at low impulse amplitudes. Yet, as these higher-order terms grow at an accelerating rate with respect to impulse amplitude, the rate of divergence of successive series also increases; as shown in Figure 2.4. This interplay between impulse amplitude, nonlinearity, stability, and series order is key to understanding an optimal Volterra series approximation.

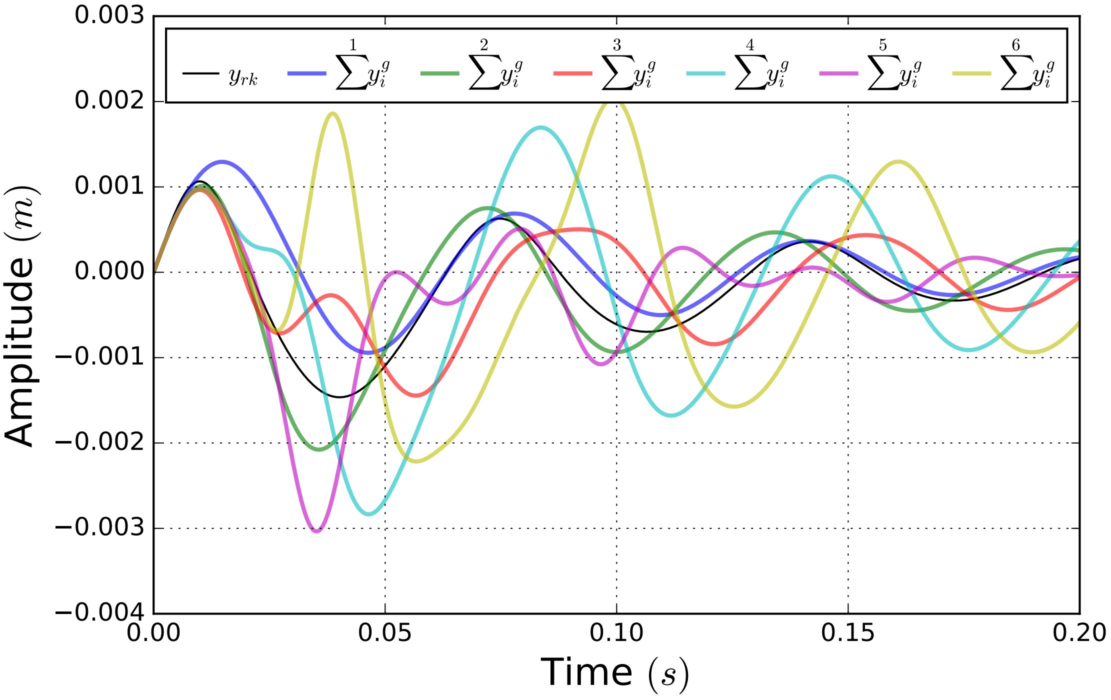


Figure 2.9: Volterra series with impulse amplitude $A = 0.15$N.

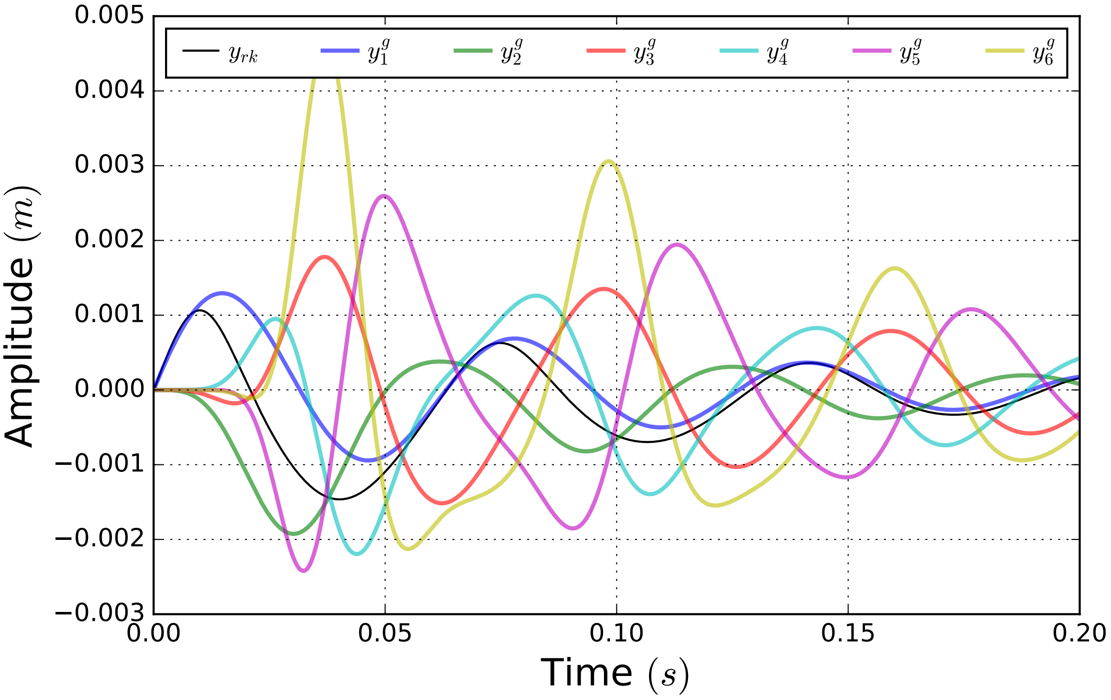


Figure 2.10: Volterra terms with impulse excitation $A = 0.15$N.

To highlight the fact that higher-order terms are unstable at lower amplitudes, consider Figure 2.10, where the order of the maximum amplitudes in descending order is y_6, y_5, y_3, y_4, y_2, y_1. Since y_6 is the last term to provide significant contributions to the Volterra series, and at this amplitude, it now has the largest amplitude, it has to be increasing at a greater rate than lower-order terms. This observation holds true for lower-order terms when compared to their lower-order counterparts.

A more general understanding of this idea is shown in Figures 2.11 and 2.12, which depict the relationship between the maximum amplitudes of the Volterra terms and their excitation amplitudes. These figures demonstrate that each successive term increases at a faster rate than the previous terms. By plotting data on a log-log scale, the lines follow the form of a straight line, $y = mx + c$. The gradient, m, approximates the exponent of A, as successive terms are strongly dependent on the repeated multiplication of A. The y-intercept, c, is determined by the rest of the term, which is a function of the system parameters.

If the Volterra terms are naively considered to be divergent at the impulse amplitude, A, when the maximum value in that Volterra term is greater than that of the previous term. Using this definition, regions of stability can be established for the generating series. Figure 2.12 shows that the first intersection between two lines, therefore the first instability arises from y_5 surpassing y_4. This finding aligns with earlier observations highlighting why the Volterra series up to y_4 was the most accurate in Figure 2.4.

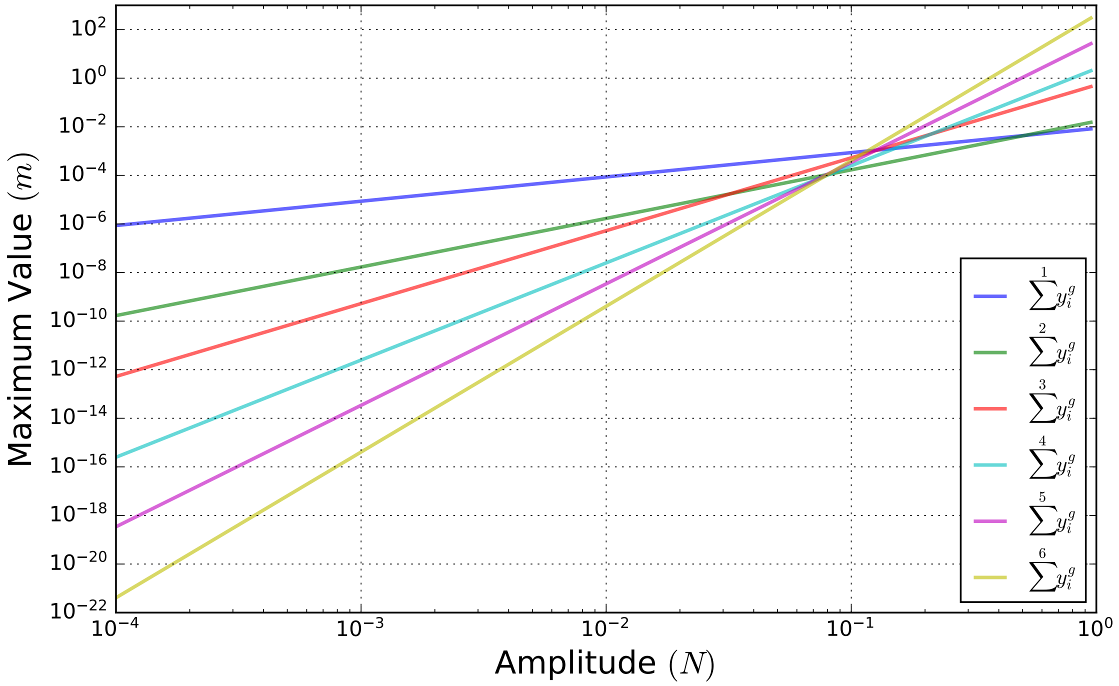


Figure 2.11: Log-log plot showing the maximum values of the Volterra terms over a range of impulse amplitudes.

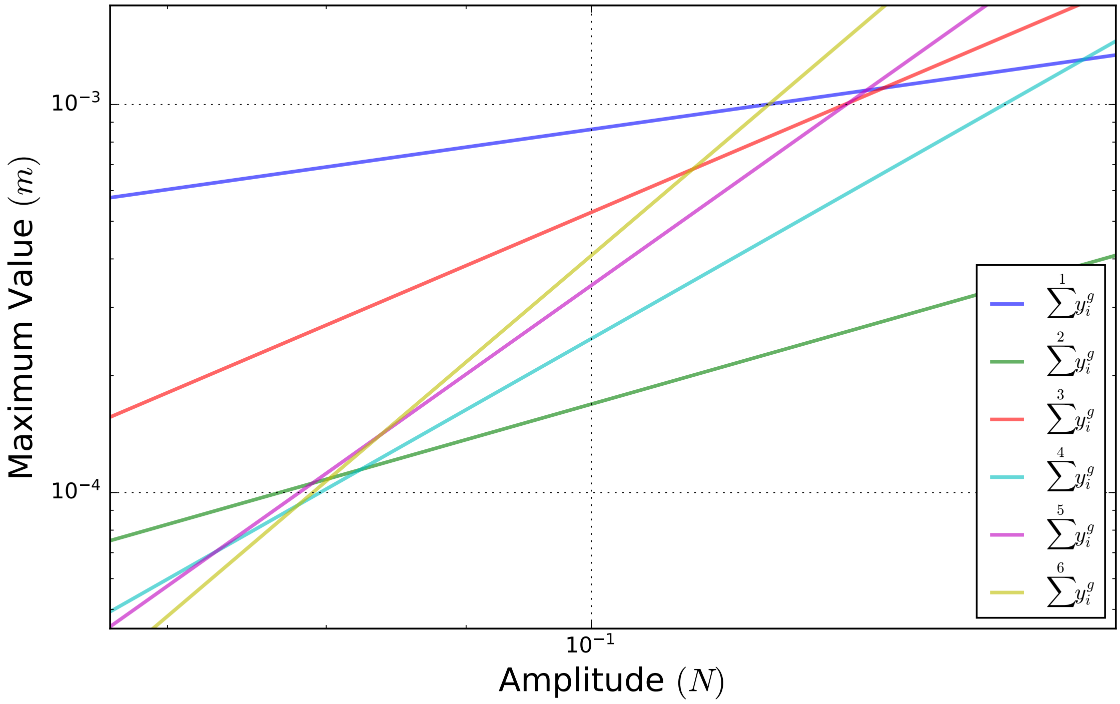


Figure 2.12: Log-log plot showing the maximum values of the Volterra terms over a range of impulse amplitudes, zoomed in on the critical region.

2.4 Conclusion

This chapter has introduced and explored the use of generating series for nonlinear differential equations, specifically with an interest in a nonlinear SDOF oscillator with quadratic and cubic nonlinearities. This generating-series method of determining the Volterra kernels requires a more abstract take than processes usually considered in the Mechanical Engineering literature; requiring the use of iterated integrals, noncommutative algebra, and monoids. However, this approach provides an elegant and computational method of reducing the nonlinear differential equation into an algebraic/ combinatorial problem.

The novel derivation of the impulse response in the generating-series domain casts light on its form. This derivation also extends on work done by Lamnabhi [19] to include a varying impulse amplitude, A.

This work introduced a very natural recursive tree representation for the shuffle-product expansion to help understand the concatenation of terms; which was then significantly optimised using a tabulation method, from drawing on parallels with combinatorics. All other optimisations discussed in this work were also novel, and without these, the calculation of y_5 and y_6 would be infeasible.

The results section explored the effectiveness of including higher-order Volterra series, taking into account complications arising from stability and nonlinearity. An error analysis against a 4th-order Runge-Kutta showed that counterintuitively including extra Volterra terms doesn't necessarily give the best accuracy in every case. The generating series approach was also validated against previously determined results.

In conclusion, this chapter has shown that the generating-series approach is a valid and effective method for handling complex nonlinear differential equations, specifically when perturbed by an impulse. The optimisation techniques have significantly improved the computational efficiency of the shuffle product (where the code is freely available on GitHub), making this approach more attainable for real-world applications.

Moving forward, several avenues for future research and development in this area include:

1. Determining a scheme that automatically outputs the Volterra stability region.

2. Determining optimisations for expanding deterministic excitations and Gaussian White Noise.
3. Apply this approach to other difficult nonlinear differential equations of the required form. Such as expanding this analysis to include a quartic stiffness term.
4. Derive the Laplace-Borel transform of other common excitations, such as coloured noises.

Chapter 3

TOPOLOGICAL DATA ANALYSIS–BACKGROUND THEORY

The main theme of this thesis makes use of data sets' shape, wherein quantitative descriptors are used to form decisions based solely on the form of the data. The central ideas required to perform this analysis are firmly rooted in *algebraic topology*, an abstract mathematical field concerning the shape of mathematical objects. In this work, algebraic-topological ideas are considered over discrete data sets, giving rise to a computational field of study called *Topological Data Analysis* (TDA). In order to motivate one of the main themes in TDA, called *persistent homology*, the required background theory to understand this concept is presented in this chapter.

This topological-based data analysis technique introduces a fresh perspective for SHM datasets, focussing on their inherent shapes. This research will unveil novelty detection schemes centred on the voids within data, distinguishing damage from other EOVs. These schemes will be executed both within the native TDA framework and in tandem with machine-learning techniques. Beyond the evident SHM applications aligned with Rytter's hierarchy, there is also an exploratory topological analysis of cointegrated residuals for data normalisation. In essence, this rich theory reveals novel aspects of data to support SHM decision-making.

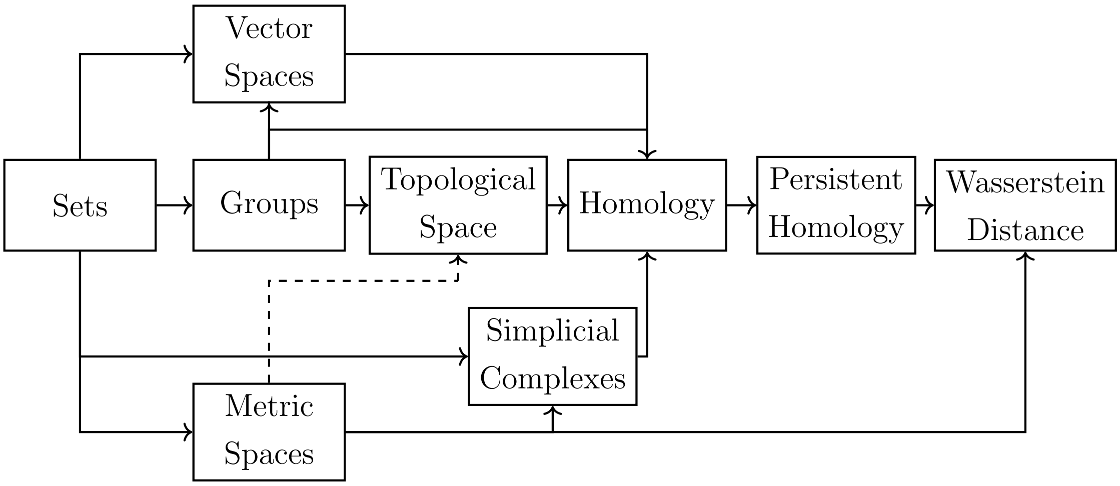


Figure 3.1: Flowchart of required background knowledge for TDA, with dependencies shown as arrows.

This background section is relatively expansive, given that the requirements for TDA are fairly mathematically involved, and the foundational ideas are seldom encountered in Engineering. Figure 3.1 lists the main topics required for persistent homology, and attempts to outline *rough* connections and dependences between them; helping navigate the seemingly complex web of theory.

The use of persistent homology in TDA has seen many sophisticated applications [37, 38, 39, 40]; despite this, many TDA analyses rely on the same general procedure:

1. A finite sample of n points from a d-dimensional data set are assumed to lay on a continuous *manifold*, with some notion of distance between the points.

2. An ordered family of objects is built atop the data, where each object is a snapshot of the shape at a given scale. These objects, referred to individually as *simplicial complexes*, and collectively as a *filtration*, encapsulate information about the connections formed between the data points over a range of distances.

3. Consider a single simplicial complex, which is a geometric object defining connections between data points. The same information can be represented combinatorially, removing the need for a geometric representation and reformatting the problem more abstractly. Extending this concept to a filtration, an abstract representation of the data is formed over a range of length scales. As the data were assumed to be sampled from a manifold, the information

now enclosed combinatorially, represents this underlying manifold.

4. After some analysis, the *persistent homology* can be derived from the *abstract simplicial complex.* The persistent homology is indicative of the shape, or *topology*, of the manifold; providing a quantitative description of the shape.

5. Metrics exist over the space of persistent homologies, which allow for the comparison of different data sets, entirely based on their shape; thereby providing a topological inference of data.

This chapter will discuss this process in greater detail, whilst also thoroughly covering the mathematical concepts involved. By the end of this chapter, sufficient material is covered to establish the fundamental analytical ideas discussed in the succeeding chapters. In the following chapters, application-specific ideas are introduced, which are then analysed alongside TDA. In this theory walkthrough, where concepts require more attention and are pivotal in their definition of homology, examples are provided to complement their description.

3.1 Sets

A set is a foundational concept, underpinning nearly all the ideas discussed within TDA, and therefore demands a rigorous introduction. The definitions listed for sets also have analogues for more complex algebraic structures. Discussing these topics at the most fundamental level will assist in understanding later, more abstract, ideas.

A set can be thought of as an unordered collection of unique elements. The elements of a set need not share the same type; they might be algebraic, numeric, or even other sets. Sets provide few restrictions, and hence, exhibit less structure compared to more restrictive concepts like *groups*, *topological spaces*, and *vector spaces* – topics covered at a later stage. Sets are depicted here with curly braces, e.g, $S = \{a, c, b, 3, \{\}, d, 1, 2\}$ represents the set S containing eight unordered unique elements of varying types.

The *empty set* $\emptyset$, is the set containing no elements, represented as $\emptyset = \{\}$. When notating $\emptyset$, it is already considered a set and should not be wrapped in curly braces. There is a subtle but important distinction between the two, as $\emptyset \neq \{\emptyset\}$. Consider the analogy that $\emptyset$ represents an empty bag, whereas $\{\emptyset\}$ represents an empty bag within a bag. Thus, the outer bag indeed contains something – the inner empty bag – and therefore, it contains an element and is not empty.

There are two common ways of merging sets: the *union* and *intersection.* The union of two sets, X and Y, denoted by $X \cup Y$, returns the set where the elements are found in X, Y, or both. It can be formally expressed as $X \cup Y = \{x \colon x \in X \text{ or } x \in Y\}$. Conversely, the intersection of two sets X and Y, $X \cap Y$, gives the set where the elements are common to both X and Y, being formally written as $X \cap Y = \{x \colon x \in X \text{ and } x \in Y\}$ [41]. Hence, the intersection of two disjoint sets results in $\emptyset$. Given two sets, $X = \{a, b, c\}$ and $Y = \{c, d, e\}$. The union of these two sets, $X \cup Y = \{a, b, c, d, e\}$, is the set of unique elements in either X or Y. Meanwhile, their intersection is $X \cap Y = \{c\}$, as only this element is common to both X and Y. For definitions later in this chapter, it is useful to note that the union may be iterated over multiple sets using the notation, $\bigcup_i^n S_i$, akin to sigma notation.

For topology, it will prove useful to construct new sets from preexisting ones. One such method is by creating a *subset* Y of a set X, wherein every element $y \in Y$ is also contained in X. This relationship is denoted $Y \subseteq X$. Extending on this idea, by taking the sets of all subsets, will naturally lead to defining a *topology.* Another useful method to create new sets is by taking the *Cartesian product.* The Cartesian product gives every combination of the two sets as an ordered pair, $X \times Y = \{(x, y) \mid x \in X \text{ and } y \in Y\}$. It is worth noting that sets can be infinite, a classic example of this is $\mathbb{R}$, the set of all real numbers. Using only a few definitions that have been provided, it is already possible to construct the set specifying Euclidean space. By taking the Cartesian product of $\mathbb{R}$ with itself, it is possible to construct the xy-plane with, $\mathbb{R} \times \mathbb{R} = \mathbb{R}^2$, where an element is denoted by its ordered pair $(x_i,\, y_i)$, more commonly known as a coordinate. This idea can be extended to repeated applications to construct k-dimensional Euclidean space by computing k Cartesian products, $\mathbb{R}^k \cong \prod_{i=1}^k \mathbb{R}$.

Within the context of algebraic topology, and by extension topological data analysis, maps between algebraic structures underpin a considerable number of definitions and theories. One such set mapping is defined as a function, $f \colon X \to Y$, where X and Y are commonly termed the *domain* and *image*, respectively. It is then useful to classify f as either *surjective*, *injective* or *bijective* [42].

Surjection: Each element in Y is mapped to, or has a pre-image in X. Surjection

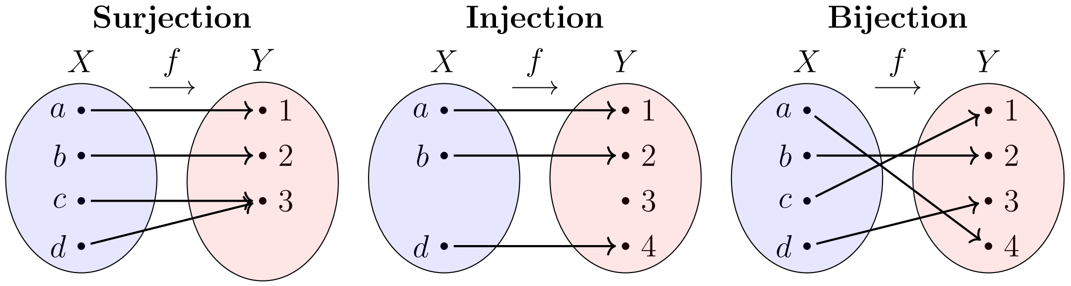


Figure 3.2: Surjection, Injection, and Bijection.

is mathematically stated as $\forall y \in Y,\ \exists\, x \in X$ such that $f(x) = y$.

Injection: Every element in X maps to a unique element in Y. Mathematically, injection is expressed as $\forall a, b \in X$ if $f(a) = f(b)$ then $a = b$.

Bijection: A map that is both injective and surjective. Therefore, every element in the domain is uniquely mapped into the image, where every element in the image is covered.

For a visual representation, Figure 3.2 helps support these mapping classifications. Following on, some advanced concepts can be built atop injection, surjection and bijection. Specifically, a mapping $f\colon X \to Y$ is called *invertible* if it has an *inverse* $g\colon Y \to X$, where the composite map $g \circ f : X \to Y \to X$ is id_X, the identity map on X, and $f \circ g\colon Y \to X \to Y$ is the identity map, id_Y, on Y. From this premise, only bijective functions can be invertible [38]. Furthermore, sets X and Y are said to be *isomorphic* when a bijective map exists between them, written as $X \cong Y$, and the bijective map is referred to as an *isomorphism*. Another integral concept is the countability of sets. Specifically, a set X is *countable* when it can be bijectively mapped to the set of natural numbers, represented as $X \to \mathbb{N}$. All of these concepts are extended, forming elements of definitions at a later stage, where further concepts find application in the definition of homology.

Finally, the abstract idea of an *equivalence relation* will be invaluable when classifying spaces by specific attributes. In essence, an equivalence relation establishes a relation among elements within a set. These equivalent elements form a subset, where each element shares a similar property, that the equivalence relation is assessing. The symbol $\sim$ represents the binary equivalence operator between two sets, and for an equivalence relation to be valid, the arguments must satisfy the following conditions

Reflexivity: $a \sim a$, a is the same as a.

Symmetry: $a \sim b$ if and only if $b \sim a$. If a is the same as b, then b is the same as a.

Transitivity: if $a \sim b$ and $b \sim c$ then $a \sim c$.

Arising from the equivalence relation is the notion of an *equivalence class.* Where a set is divided into distinct equivalence classes, with the elements of each equivalence class sharing a common characteristic [43].

3.2 Groups

Beyond sets is the idea of a group, which has a richer structure resulting from an accompanying binary operator that acts over the set elements. For the set-operator pair to qualify as a group they must satisfy some important, albeit stringent axioms. Via these restrictions, a more complete and robust algebraic structure emerges, where information can be studied in one case and transferred via isomorphism to another case. Many of the succeeding definitions for groups are built on the ones found in Section 3.1. The intricacies of algebraic topology are fundamentally tied to the machinery of group theory. Consequently, this section will discuss concepts fundamental to group theory, setting the stage for an understanding of TDA.

Definition 3.2.1. *Groups* are an extension of sets, to include a *binary operator.* The group, G, formed from this set-operator pair is written as $G = (S, \circ)$. There are also four necessary axioms associated with groups, these are [44]:

G1: If $g_1, g_2 \in G$ then, $g_1 \circ g_2 \in G$. In words, the result of applying the operator over two elements of the group is also contained within the group. This property, known as group *closure* is denoted $G \circ G \to G$.

G2: Given any three distinct elements g_1, g_2, $g_3 \in G$ then $(g_1 \circ g_2) \circ g_3$ is equivalent to $g_1 \circ (g_2 \circ g_3)$. This axiom highlights that the evaluation order does not affect the outcome, and is called the *associative* property.

G3: A group contains a unique *identity element*, e, such that when e is passed as an argument into the group operation, the other element will remain unchanged in the result. Specifically, $g \circ e = g = e \circ g$ for all $g \in G$.

G4: Every element of the group must have an inverse, which is also an element of the group. The result of combining an element with its inverse is the identity element. Formally, $\forall g \in G$, $\exists\, g^{-1} \in G$ such that $g \circ g^{-1} = e$, where e is the identity element of the group. Where $\forall$ and $\exists$ mean 'for all' and 'there exists'.

Since the concept of a group is vital in this work, three short examples are provided to demonstrate how to verify the axioms. Firstly, consider the set $\mathbb{R}$ forming a group under addition, $G = \{\mathbb{R}, +\}$. The axioms listed in Definition 3.2.1 are satisfied by G. **G1:** For any $a, b \in \mathbb{R}$, then $a + b \in \mathbb{R}$, this statement is true for all $a, b \in \mathbb{R}$. **G2:** Addition is associative, meaning $a + (b + c) = (a + b) + c$, $\forall a, b, c \in \mathbb{R}$. **G3:** The addition identity element, $0 \in \mathbb{R}$: preserves other elements over addition: $a + 0 = 0 + a = a$, $\forall a \in \mathbb{R}$. **G4:** For any $a \in \mathbb{R}$, there exists an element $-a$ which is also in $\mathbb{R}$ such that their sum, $a + (-a) = 0$, results in the group identity. On the other hand, $\mathbb{R}$ does not form a group under multiplication as **G4** is not satisfied, since $0 \in \mathbb{R}$ and $\frac{1}{0} \notin \mathbb{R}$, thus 0 does not have a multiplicative inverse.

Example 3.2.2. Finite groups are often represented as Cayley tables. The table is populated with the elements as if the elements in the rows and columns were passed as arguments into the group operator. Table 3.1 shows the Cayley table for the group formed under multiplication modulo 9, where only considering coprime values, yielding $U(9) = \{1, 2, 4, 5, 7, 8\}$, as 3 and 6 share a factor with 9 [45].

$\times_9$	1	2	4	5	7	8
1	1	2	4	5	7	8
2	2	4	8	1	5	7
4	4	8	7	2	1	5
5	5	1	2	7	8	4
7	7	5	1	8	4	2
8	8	7	5	4	2	1

Table 3.1: The Cayley table for the group formed under multiplication modulo 9.

The Cayley table allows for fast verification of the group axioms.

G1: The elements populating the table are all elements of $U(9)$, therefore the group is closed.

G2: Multiplication is an associative operator.

G3: 1 is the multiplicative identity.

G4: This is visually proven as each row and column in the table contains a single 1, the identity element, therefore proving the existence of a unique inverse.

This example will be later revisited in Example 3.2.5 where the group's structure is preserved over a mapping.

3.2.1 Subgroups and Cosets

Just as sets can create subsets, groups can similarly create subgroups. However, with subgroups, information is inherited from the parent group, such as the identity in the most basic case. Subgroups then motivate the concepts of *cosets*, *sets of cosets*, and *normal subgroups*. Normal groups hold significant importance since they can be used to construct *quotient groups*, which provide the notion of group division. All of these group types are directly involved with the definitions building up to, and directly associated with the concept of homology.

Definition 3.2.3. A *subgroup* H of a group G is a subset such that $H \subseteq G$, which adheres to the group axioms in Definition 3.2.1. Both the operation and the identity element of the subgroup are directly inherited from the parent group [38].

Building on the notion of subgroups, a natural avenue for extension is the idea of cosets. Given a group G and one of its subgroups H, then:

A *left coset* of H in G is defined as a subset of G expressed as $gH = \{gh \ : \ h \in H\}$, for some $g \in G$. The set of left cosets of H in G is written G/H.

A *right coset* of H in G is a subset of G of the form $Hg = \{hg \ : \ h \in H\}$, for some $g \in G$, and the set of right cosets is written $H \backslash G$ [44].

A *normal* subgroup H of G occurs when the left and right cosets are equal, $gH = Hg$, $\forall g \in G$, this is written $H \lhd G$.

Example 3.2.4 shows an example of calculating the cosets, and it is shown that elements of cosets share some equivalence relation.

Furthermore, a common and useful type of group naturally arises when the result is unaffected by the ordering of operands. Such a group is termed *Abelian* and is characterised by its *commutative* operator: $x \circ y = y \circ x$, $\forall x, y \in G$. In the case of Abelian groups, the left and right cosets are always identical, and inherently all subgroups are normal. However, the converse is not true, as not all normal subgroups are formed from Abelian groups. If $H \triangleleft G$, then the set of cosets G/H becomes a group itself, known as the *quotient group* [44]. The notation for the quotient group closely aligns with forming numerical quotients, which is also reflected in its meaning. However, it is perhaps more informative to motivate and explain this material with an example,

Example 3.2.4. Consider the group, $G = (\mathbb{Z}, +)$, there then exists a subgroup H of G, such that $H = (2\mathbb{Z}, +)$. Choosing an element from G, for instance, $g = 1$, and adding it to every element in the subgroup to find a left coset, the result is $1 + H = 1 + 2\mathbb{Z}$; corresponding to the odd set of all odd numbers.

Continuing this process with the element $g = 2$, the result is $2 + 2\mathbb{Z} = 2\mathbb{Z}$, representing the set of all even numbers. Since the subgroup H is infinite, adding $g = 3$ gives the same results as adding $g = 1$. Applying this logic for subsequent odd and even values of g concludes that the set of all left cosets is given by $G/H = \{2\mathbb{Z}, 1 + 2\mathbb{Z}\}$. Where the first element G/H is formed from even values of g, and the second from odd values of g, which represent the equivalence classes with the cosets. Because addition is commutative, both H and G are Abelian. Consequently, the left and right cosets are the same, thus H is a normal subgroup of G. Thus, the set of right cosets is also given by $H \backslash G = \{2\mathbb{Z},\ 2\mathbb{Z} + 1\}$.

Following this, since H is a normal subgroup, $G/H = \{2\mathbb{Z},\ 2\mathbb{Z} + 1\}$ represents the quotient group. Quotient groups are conceptually similar to quotients of integers, hence the naming. When performing an integer quotient, say a/b, one is essentially asking how many times b fits into a. This is much the same in group theory, where G/H can be thought of as measuring how many copies of H fit into G. Each coset in G/H is like a copy of H that has been translated. Then G/H describes all the ways H can be translated with respect to the elements of G, without changing their internal structure. An understanding of quotient groups is especially important in this work, as homology groups are quotient groups.

3.2.2 Maps

Group maps are a key focus of group theory since they facilitate the transfer of information between groups. By mapping between groups with the same underlying structure, a deep and abstract symmetry can be transferred across seemingly disconnected problems. For example, if a group G_1 is well studied, with a large list of properties catalogued, and a *homomorphism*[1] can be formed to another group G_2, then there can be a transfer of knowledge via the homomorphism from $G_1 \to G_2$ as the group structure is preserved over the homomorphism.

Consider two groups $(G_1, *)$ and $(G_2, \circ)$. A group *homomorphism*, $h\colon G_1 \to G_2$, maps from G_1 to G_2 and is defined if the following conditions are satisfied [38]:

1. The identity element in one group maps to the identity element of the other group, $h(e_{G_1}) = e_{G_2}$.
2. The group operation distributes over the homomorphism, $h(u * v) = h(u) \circ h(v)$, $\forall u, v \in G_1$.

Given a group homomorphism $h : G_1 \to G_2$, the *kernel* of h, $\ker(h) \subset G_1$, is the set of elements x such that $h(x) = e$, where e is the identity element. The *image* of h, $\mathrm{Im}(h) \subset G_2$, is the set of elements y such that $y = h(x)$ for some x [46]. Having a solid understanding of the kernel and the image of a homomorphism is invaluable in later sections. These ideas will form the backbone when deriving the homology groups.

Example 3.2.5. Consider the set created by the sixth complex roots of unity, $z = \{x \ : \ x^6 = 1\} = \{1, \omega, \omega^2, \omega^3, \omega^4, \omega^5\}$ where $\omega = e^{2\pi i/6}$. This set satisfies the group axioms under multiplication, and its Cayley table is shown below.

$\times$	1	ω	ω^2	ω^3	ω^4	ω^5
1	1	ω	ω^2	ω^3	ω^4	ω^5
ω	ω	ω^2	ω^3	ω^4	ω^5	1
ω^2	ω^2	ω^3	ω^4	ω^5	1	ω
ω^3	ω^3	ω^4	ω^5	1	ω	ω^2
ω^4	ω^4	ω^5	1	ω	ω^2	ω^3
ω^5	ω^5	1	ω	ω^2	ω^3	ω^4

Table 3.2: The Cayley table formed when multiplying the sixth roots of unity.

Interestingly, there is a homomorphism between the set elements of the sixth roots

---

[1] The term homomorphism is used when mapping between algebraic structures but homeomorphism is used when mapping between topological spaces.

of unity and $U(9)$, given in Example 3.2.2, and is given by,

$$\begin{pmatrix} 1 \\ \omega \\ \omega^2 \\ \omega^3 \\ \omega^4 \\ \omega^5 \end{pmatrix} \mapsto \begin{pmatrix} 1 \\ 2 \\ 4 \\ 8 \\ 7 \\ 5 \end{pmatrix} \equiv \begin{pmatrix} 2^0 \\ 2^1 \\ 2^2 \\ 2^3 \\ 2^4 \\ 2^5 \end{pmatrix} \mod 9 \tag{3.1}$$

This mapping is a homomorphism, but it is actually bijective as well, so is an *isomorphism.* Notice that the order of the elements over the mapping has been permuted between the representations given in Tables 3.1 and 3.2. It is not a requirement of an isomorphism that the ordering presented in the Cayley table is preserved. Since sets are unordered collections, permuting the rows and columns is a legitimate process. In essence, there is a structural symmetry between the sixth roots of unity and multiplication of coprime factors mod 9.

3.2.3 Generators and Free Groups

The groups presented in Tables 3.1 and 3.2 can both be *generated* by a single element, called the *generator.* A group generator is an element such that every element of the group may be expressed as a combination of the generator and its inverse. For these two groups, repeated multiplication by ω and 2 respectively, results in all the elements of their groups. In general, a group may have multiple generators which are a subset of the group, such that they can create the entire group.

Intriguingly, the generator structure is preserved under an isomorphism. From the geometric perspective of complex roots of unity, the role of ω as the group generator is intuitive. However, the repeated multiplication by 2 for multiplication mod 9 is not so straightforward. This reasoning is where group theory becomes a very powerful tool, by transferring intuitive structure via isomorphism from one domain where reasoning is intuitive to another where things aren't so obvious.

Extending on the idea of group generators is the idea of a *free group.* A free group F_S on a set S is a group wherein every element can be written as a product of elements of S and their inverses [45]. Less rigorously, this says that a group can be created by applying some operator to any repeated number of elements of a set. Thus any free group generated by a nonempty subset is infinite. The only requirement for a free group is that each element and its inverse cancel each other out. For example consider the set $\{a,\, b\}$. Some elements of the free group generated by this set would be $a,\, a^2,\, a^3,\, \dots,\, b,\, ab,\, ba,\, a^{-1},\, b^{-1},\, a^2b^3a^{-5}b^2, \dots$. The concepts of group generators and free groups are *extremely* important in defining the homology groups.

3.3 Spaces

Three types of spaces are required for TDA, these being *vector spaces*, *metric spaces*, and *topological spaces*; all will be used in unique ways. Vector spaces represent the form of homology, whilst metric spaces serve as a familiar prerequisite to topological space, as well as being used as a similarity measure of the persistent homologies. Finally, topological spaces are the most relaxed and abstract form considered, which allows for the definition of homology.

3.3.1 Vector Spaces

Definition 3.3.1. Let $\mathbb{F}$ be a set, in which addition, subtraction, multiplication and division are defined, this is called the *scalar-field.* Let V be a non-empty set where the elements, $\mathbf{v} \in V$, are *vectors.* The vectors have two closed operations assigned, vector addition $+\colon V \times V \to V$ and scalar multiplication $\cdot\colon F \times V \to V$. V is called an $\mathbb{F}$-*vector space* if the following criteria are satisfied [47]:

V1: $(V, +)$ is an Abelian group.

V2: Scalar multiplication is distributive over vector addition; mathematically stated as $a \cdot (\mathbf{v}_1 + \mathbf{v}_2) = a \cdot \mathbf{v}_1 + a \cdot \mathbf{v}_2,\ \forall a \in \mathbb{F},\ \forall \mathbf{v}_1, \mathbf{v}_2 \in V$.

V3: Vector addition is distributive over scalar multiplication; mathematically stated as $(a + b) \cdot \mathbf{v} = a \cdot \mathbf{v} + b \cdot \mathbf{v},\ \forall a, b \in \mathbb{F},\ \forall \mathbf{v} \in V$.

V4: Scalar multiplication is associative; mathematically stated as $(ab) \cdot \mathbf{v} = a(b \cdot \mathbf{v}),\ \forall a, b \in \mathbb{F},\ \forall \mathbf{v} \in V$.

V5: $\forall \mathbf{v} \in V$ then $1 \cdot \mathbf{v} = \mathbf{v}$.

For an $\mathbb{F}$-vector space, V. A finite subset of $\{\mathbf{e}_1, \dots, \mathbf{e}_n\} \subseteq V$ is the set of *basis vectors* if every $\mathbf{v} \in V$ can be expressed as a unique linear combination of elements of the basis vectors and V is said to be n-*dimensional.* The dimension of the homology groups gives arguably the most important descriptor in the persistent homology, called the *Betti numbers.*

3.3.2 Metric Spaces

Before moving onto the more general theory of topological spaces, it is instructive to first motivate with *metric spaces.* Approaching the topics in this order aids comprehension of the more abstract topological definitions. Whilst the topological definitions are more nuanced, they do still draw parallels to the metric equivalents. Additionally, this detour in theory is not in vain, or indeed a detour, since metric spaces are also useful when comparing persistent homologies. A specific metric, known as the *Wasserstein distance*, is used extensively as a similarity measure, allowing for inference between data sets based solely on their shape. Arguably the most familiar metric is the Euclidean distance, $\partial_2(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$, which allows a generalisation of Pythagoras's theorem into n dimensions.

Definition 3.3.2. A *metric space* is defined by a set-metric pair $(X,\, \partial_X)$, that fulfils the requirements outlined by some axioms. Here, X represents the set containing all the elements of the metric space and ∂_X is the associated metric, or distance function, such that,

$$\partial_X \colon X \times X \to \mathbb{R} \tag{3.2}$$

The metric takes two elements from the set and maps these elements to the real numbers, giving the notion of distance between the two elements. To qualify as a metric space, the distance function must satisfy three fundamental properties [38]:

$\mathbf{d}(\boldsymbol{x}, \boldsymbol{y}) \geq \mathbf{0}$**:** This simply states that the distance between any two points is always positive. The sole exception occurs when $x = y$, wherein $\partial_X(x,\, y) = 0$.

$\mathbf{d}(\boldsymbol{x}, \boldsymbol{y}) = \mathbf{d}(\boldsymbol{y}, \boldsymbol{x})$**:** This states that the distance is scalar, i.e. nondirectional. The distance from $x \to y$ is the same as the distance from $y \to x$.

$\mathbf{d}(\boldsymbol{x}, \boldsymbol{z}) \leq \mathbf{d}(\boldsymbol{x}, \boldsymbol{y}) + \mathbf{d}(\boldsymbol{y}, \boldsymbol{z})$**:** Known as the triangle inequality, which states that the direct route between two points is never more than the distance between the two points via an intermediary point. The triangle inequality is only ever an equality when the three points are colinear.

Once a metric is established within a space, it paves the way for many familiar and useful ideas, such as continuity and convergence. The procedure to define these is done through *open balls* and *sequences*, which will be adapted and generalised for topological spaces.

An *open ball*, defined as $B_\varepsilon(x) = \{y \in X \;:\; \partial_X(y, x) < \varepsilon\}$, is the region around a point x, encompassing all the space that is less than the distance ε from x. The term 'open' in this context refers to the exclusion of the ball's boundary from the space, whereas a *closed ball* would include the boundary. This space is often referred to as the *ε-neighbourhood* of x [43]; and this idea is instrumental in the formation of topological objects from data, presented in Section 3.4.

3.3.3 Topological Spaces

Topological spaces are a relaxation of metric spaces, removing the need for a distance function, and allowing for a more general theory. As a consequence, topological spaces are a superset of metric spaces. Whilst metric spaces have a familiar geometric interpretation of distance, topological spaces more abstractly approach an analogous result through a notion of proximity, via a set of nested open sets. Topological spaces offer a generalised idea of closeness, leading to a broader perspective on continuity and convergence. The importance of studying topological spaces stems from the need to understand the continuous properties of spaces without a precise notion of distance. These ideas then give rise to ideas such as homology, allowing for the quantification of the data's shape, forming the main analytical tool used in this thesis for SHM data analysis.

Definition 3.3.3. A *topological space* is represented by a pair, $(X, \mathcal{T})$, where X is a set and $\mathcal{T}$ is a collection of subsets, called the *topology.* $\mathcal{T}$ contains *open sets* which must satisfy the following axioms [43]:

- The space itself and the empty set are elements of the topology, $X \in \mathcal{T}$ and $\emptyset \in \mathcal{T}$.
- The union of any collection of sets in $\mathcal{T}$ is also an element of $\mathcal{T}$.
- The intersection of a finite collection of elements of $\mathcal{T}$ is an element of $\mathcal{T}$.

Similarly to metric spaces, topological spaces are a pair, where X captures the span of the space. Whereas, $\mathcal{T}$, the topology, is a set of open continuous subregions of X, giving a vague notion of distance. Given that the subsets in $\mathcal{T}$ are continuous, they inherently convey a sense of closeness. Larger subsets in $\mathcal{T}$ do not necessarily imply the points contained within the set are proximate, whereas smaller subsets are more restrictive. For an intuitive, discrete, and probabilistic (albeit not mathematically rigorous) way of thinking, the subsets in $\mathcal{T}$ encapsulate closeness as their elements will be arbitrarily close in the topological space. Then, the frequency that any two

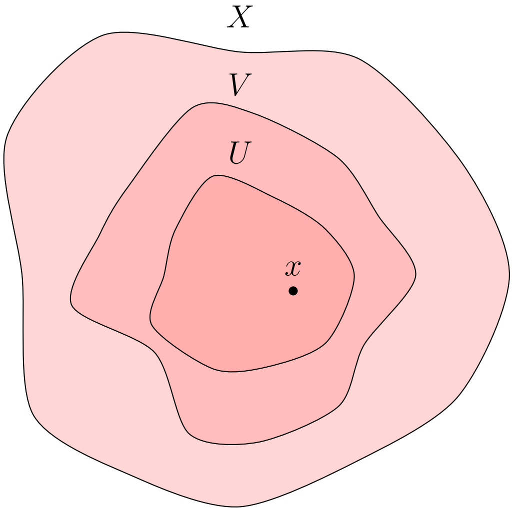


Figure 3.3: The neighbourhood, V, of $x \in X$.

unique elements appear within these subsets offers an insight into their proximity: a higher frequency suggests more proximate points, as smaller subsets naturally house more proximate points resulting from the absence of discontinuities in the space. However, this logic breaks down when considering continuous spaces, as all distinct pairs of points will occur infinitely frequently.

While topological spaces provide a broad theory for understanding closeness, for the case of real-world physical and engineering analysis, it is useful to apply additional constraints to ensure the mathematics in the space are 'well behaved' by being consistent and predictable. Most real-world systems enforce the requirement for a space to be *second-countable* and *Hausdorff*.

Firstly, to motivate space countability, a *base* of a topological space $(X, \mathcal{T})$ is a collection of sets $\{\mathcal{T}_\alpha\}$, such that any set in the topology, $\mathcal{T}$, can be expressed as a union of the base sets [38]. As a result, topological bases provide a foundation for building all open sets within the space. Extending on this idea, the space is then termed *second countable* if there are a countable number of base sets [48]. Countability, as defined in Section 3.1, refers to a mapping to the natural numbers. Consequently, second countability provides insight into the magnitude of a topological space, even if infinite.

To then describe a Hausdorff space, it is best to first describe a topological *neighbourhood*. In essence, a neighbourhood indicates which points are deemed close to a point, which is more abstract in the absence of a traditional geometric metric. A

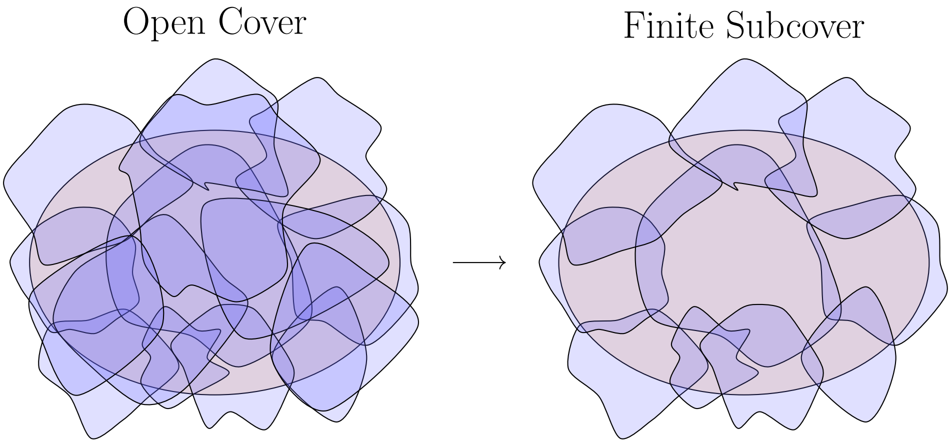


Figure 3.4: Open cover and finite subcover of a topological space, X.

neighbourhood of a point $x \in X$ is a subset $V \subseteq X$, such that there is an open set $U \subseteq V$ where $x \in U$. Simply put, if there's an open set containing x which is entirely contained within V, then V is a neighbourhood. Essentially, a neighbourhood is an immediate area around a point. This fundamental idea underpins many other ideas in topology, such as continuity and convergence, effectively acting as the topological equivalent of a metric open ball. Now, leading to the Hausdorff property, a topological space is *Hausdorff* if $\forall x, y \in X$ with $x \neq y$, there exist two non-overlapping neighbourhoods V_x and V_y, such that $x \in V_x$ and $y \in V_y$; mathematically, $V_x \cap V_y = \emptyset$. That is, each point is distinct and can be uniquely distinguished and isolated from all other points [48].

One final essential property of topological spaces to consider is *compactness*. Compactness seeks to generalise the concept of boundedness of a region within a topological space. Given a topological space X, then an *open cover* of X is a collection of open sets $\{U_\alpha\}$ such that the union of all these sets contains the entire space X, which is formally expressed as $X \subseteq \bigcup_\alpha U_\alpha$. The topological space is then called *compact* if any open cover of X has a finite *subcover*, where a subcover is some subset of the cover, $\{U_\alpha\}$, which still covers X [46].

Transitioning from pure topological concepts, it is important to acknowledge how these principles are pertinent to engineering and physical sciences. By imposing that a space is second countable and Hausdorff, one asserts that limits are uniquely defined, an essential requirement for calculus. Furthermore, the Hausdorff property ensures continuous functions are well-defined, as the image of a compact set under a continuous function remains compact. The preservation of compactness across

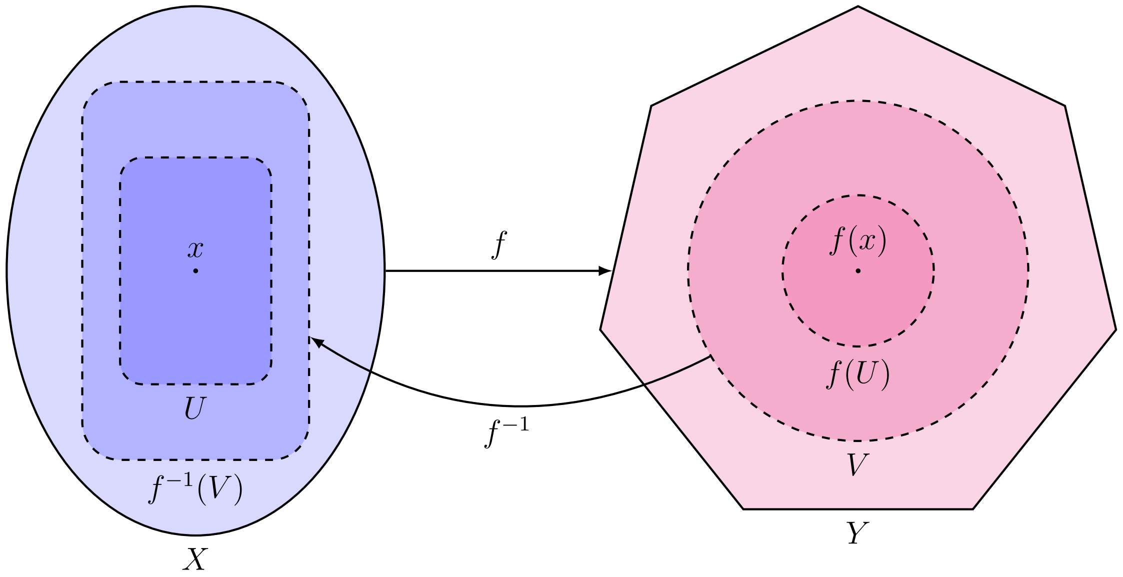


Figure 3.5: Continuous map between two topological spaces, X and Y. The dashed boundaries suggest that the region is open.

mappings is crucial for physical applications when the conservation of an attribute is expected, such as energy and momentum. Without compactness, there is no notion of finiteness, potentially giving mathematically-incoherent results with real-world observations. Having such restrictions in place ensures that spaces behave analogously to well-behaved counterparts, such as $\mathbb{R}^n$. Further generalisation into calculus on manifolds is discussed in Section 3.3.4. However, before discussing this, continuity is still to be addressed.

The metric and topological ideas for continuous maps are analogous, where metric spaces consider a more restrictive ε-neighbourhood (or open ball), and topological spaces consider a topological neighbourhood. Consider two topological spaces: $(X, \mathcal{T}_X)$ and $(Y, \mathcal{T}_Y)$. Suppose a map $f\colon X \to Y$, and that there is a neighbourhood U of x where $f(U) \subset V$. The map $f : X \to Y$ is *continuous* if, for *every* open set $V \subseteq Y$, its pre-image $f^{-1}(V)$ is open in X [48]. As the definition for topological continuity is fairly verbose, Figure 3.5 has been provided to assist the definition. Much like sets, continuous maps are extended to a *homeomorphism* when they are bijective. In this scenario, the spaces X and Y are said to be *homeomorphic*.

Two topological spaces being homeomorphic essentially states that two spaces are topologically congruent, meaning one can seamlessly morph into the other without discontinuities, and vice versa. Two homeomorphic spaces are denoted $X \cong Y$. The attributes of countability, compactness, and the Hausdorff property are all preserved under a homeomorphism. These properties serve as the first taste of *topological invariants*, so termed as they are invariant under homeomorphism. The useful converse argument can be considered: if the two spaces exhibit different topological invariants, then these space *cannot* be homeomorphic, as per the definition. The homology of a space, and subsequently, the persistent homology are also topological invariants, which precisely quantify differences between spaces.

3.3.4 Manifolds

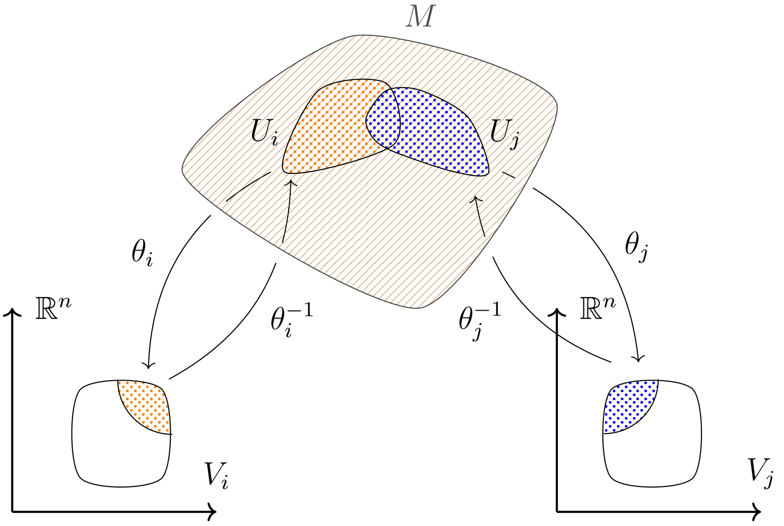


Figure 3.6: A manifold with charts U_i and U_j, with their homeomorphisms to $\mathbb{R}^n$.

Having introduced the concept of topological spaces, second countability, Hausdorff spaces, and homeomorphisms, these all motivate the application to *topological manifolds*. Within these manifolds, are well-defined regions of space where data analysis may take place. At the heart of TDA, some point cloud data are assumed to lie on a continuous manifold, allowing for its quantifiable analysis. The most fundamental, and often quoted, definition of a manifold is as a topological space that is locally homeomorphic to $\mathbb{R}^n$. However, this obscures details regarding countability and the Hausdorff property, which are assumed for $\mathbb{R}^n$

Given a topological space, M, then an open subset $U \subseteq M$ is a *chart* when U is homeomorphic to some open subset, V, of $\mathbb{R}^n$. The homeomorphism $\theta\colon U \to V$ then induces a local coordinate system onto U by the inverse map $\theta^{-1} : V \to U$ [48]. By extension, an *atlas* of a topological space, M, is a union of charts, such that the charts form a cover of M. That is, all of M is locally homeomorphic to some subset of $\mathbb{R}^n$ [48]. It is worth noting that atlases are not typically unique, as manifolds can be covered in various ways. For illustrative purposes, a partial cover of a manifold is displayed in Figure 3.6. To qualify as a manifold, M should be fully covered, though, this has been omitted for clarity. The interest in this work is with topological manifolds, which are when a manifold M is second-countable and satisfies the Hausdorff property. As a consequence, topological manifolds also have an atlas where each chart is a subset of $\mathbb{R}^n$.

When applying TDA in later chapters, there is an implicit assumption that the point-cloud data originate from a manifold. Then, by understanding the shape of the point cloud, it is assumed that this is a fair approximation to the underlying manifold. A deeper understanding of manifold shape and structure can then begin to offer insight into system mechanics, forming the analysis section of this thesis.

3.4 Simplicial Complexes

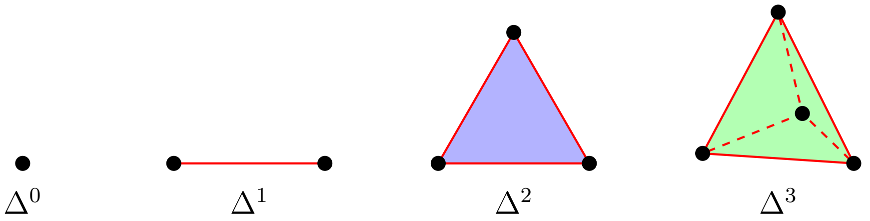


Figure 3.7: The first four simplices.

After a deep dive into the abstraction of pure mathematics, it is refreshing to shift towards more tangible engineering-esque, geometric-in-nature concepts. One could call this the calm before the storm before coming head-on with homology. However, before proceeding, a pressing question still looms: How is a notion of continuous shape extracted from discrete data? Since topological spaces were discussed in Section 3.3.3, sufficient material has been discussed to create mathematical objects from data. The solution to this problem lies with *simplicial complexes*.

Simplicial complexes are geometric objects used to attribute a quantifiable shape to data. Often described as higher-dimensional analogues of graphs, simplicial complexes play a significant role in this thesis when analysing data sets. Specifically, simplicial complexes will allow for the conversion of data to a discrete topological object. By doing this, simplicial complexes permit topological analysis of data, allowing for the calculation of invariants, and comparison of data sets.

A simplicial complex is a structure made up of fundamental units called *simplices*, denoted Δ^k where k refers to the dimension that the simplex occupies. The first four simplices are shown in Figure 3.7, where each vertex in a simplex connects to all other vertices, and the enclosed space also forms part of the simplex. Throughout this work, when displaying simplices and simplicial complexes, red is used to denote edges, blue for areas, and green for volumes[2]. This colour choice closely aligns when plotting persistent homology at a later stage. The first four simplices can be intuitively visualised and described:

Δ^0 is a point.

Δ^1 is a line segment spanning two points.

Δ^2 consists of three vertices, the edges spanning the vertices and the area enclosed by the edges.

Δ^3 includes the four faces and the volume bound by the faces, as well as the six edges and four vertices.

A pattern may be spotted in forming successive simplices. For a given Δ^k, it consists of $(k+1)$ connected Δ^{k-1} simplices, which form the *faces* of the k-simplex. The connection over the Δ^{k-1} simplices encloses a k dimensional region, and thus a k-simplex is k-dimensional. When considering simplices where $k > 3$, they occupy a space greater than three dimensions, requiring a grasp of visualising in higher dimensional space. Alas, simplices and simplicial complexes of higher dimensions are used during this thesis, and therefore a reliance on the theory (over intuition) is unavoidable.

Definition 3.4.1. A *k-simplex*, Δ^k, is the space spanned by the set of points $\{x_0, x_1, \ldots, x_k\}$ where each $x_i \in \mathbb{R}^{k+1}$, given by:

$$\Delta^k = \{(\lambda_0 x_0, \ldots, \lambda_k x_k) \in \mathbb{R}^{k+1} \mid \lambda_i \geq 0 \ \forall i, \sum_{i=1}^{k+1} \lambda_i = 1\} \tag{3.3}$$

---

[2] It is impossible to display both the green volume and the blue areas on a 2D plot. Where green is present in simplicial complex plots, blue faces should also be there. For clarification, Δ^3 *does* contain the 2D areas spanning the points but are not shown as they would mask the green volume. In this case, the blue faces are assumed, which leads to a later idea of a maximal simplex.

where $x_1, \ldots, x_k$ are all linearly independent, and these are termed the *vertices* of the simplex [49].

The λ_i coefficients are called the *barycentric coordinates* of the simplex. For a point $x_i \in \Delta^k$ corresponding to a given set of barycentric coordinates $\lambda_1, \ldots, \lambda_{k+1}$, this can be thought of as the system with $\lambda_1, \ldots, \lambda_{k+1}$ attributed masses to the corresponding vertices $x_1, \ldots, x_{k+1}$. Subsequently, if all values of λ_i are nonzero, the corresponding set of points x represents the interior of Δ^k, while if any values of $\lambda_i = 0$, then the set of points x represents a face of the simplex [42].

The set $\{\sum_{i=1}^{k+1} \lambda_i x_i \mid \lambda_j = 0\}$ is the j^{th}- *face* of the simplex, Δ^k. This face is opposite the vertex x_j. This, in words, says that the faces of a line segment are the vertices at the end, the faces of a triangle are the lines connecting the points, and the faces of a tetrahedron are the triangles. Following this logic, there exist $(k+1)$ Δ^{k-1} faces of Δ^k, as arrived at previously in a more qualitative manner.

Definition 3.4.2. The *boundary* refers to the parts of the simplex, Δ^k, that lie outside its interior.

Where the *interior* of a simplex, Δ^k, denoted $\text{int}(\Delta^k)$, is the space spanned by the subset of points $\{x_0, \ldots, x_k\}$ given that $\forall \lambda_i > 0$. As a less formal follow-on definition, the boundary is also the union of all the faces of a simplex. Definition 3.4.2 bridges understanding until a more mathematically rigorous one, which holds for simplicial complexes, is given in Section 3.5. Understanding the boundary operator is key in defining homology groups.

However, a simplex can only capture limited information about space, arising from the restriction that every vertex must connect to all others. Simplices alone cannot represent shapes like the perimeter of a square. To extend the use of simplices and allow the mathematical description of more complex spaces, the idea of a simplicial complex must be introduced.

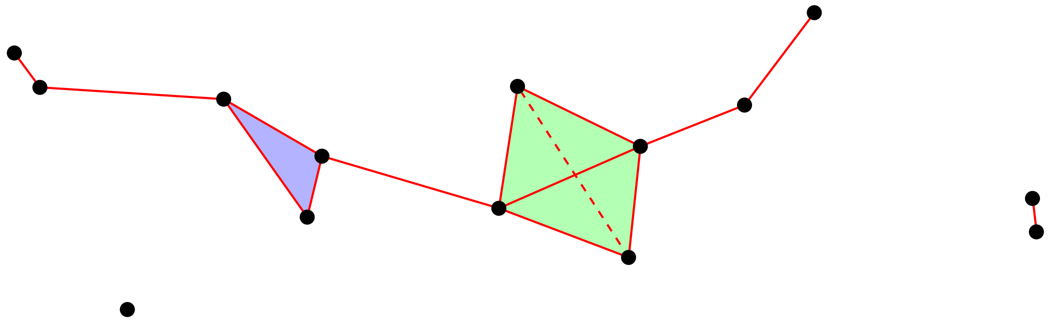


Figure 3.8: A randomly generated simplicial complex containing three disjoint regions, vertices, edges, areas, and a volume.

Definition 3.4.3. A *simplicial complex*, K, is a finite collection of simplices in some space $\mathbb{R}^n$, that satisfies the following rules [50]:

1. Every face of a simplex $\Delta^k \in K$ is also a face of K.
2. For any two simplices Δ^j, $\Delta^k \in K$ then either $\Delta^j \cap \Delta^k = \emptyset$ (suggesting the simplices are disjoint) or $\Delta^j \cap \Delta^k$ is a common face of the two simplices.

These rules then impose that the maximal dimension of a simplicial complex can never be greater than the largest dimension of the constituent simplices, i.e. forming simplex connections does not create new dimensions, as the enclosed space is not included [51].

A simplicial complex is a collection, or complex, of simplices, which are sometimes, but not necessarily, connected along simplex faces. Consider a simplex of four vertices $\{x_0, x_1, x_2, x_3\}$, by definition these encode a tetrahedron. Yet, if these four points were part of a simplicial complex, this is not necessarily true. One example of the combinatorially many is the connection of four Δ^1- simplices, giving the perimeter of a square. Therefore, the more general approach of simplicial complexes allows for a richer spatial description.

Figure 3.9: A simplex, Δ^3, containing four vertices, compared to a simplicial complex, a square perimeter, also containing four vertices.

There still remains a reliance on a geometric definition of a simplicial complex. However, when determining the homology groups, simplicial complexes need to be described abstractly. The conversion from a geometric object to an abstract setting is done via *abstract simplicial complexes*, wherein the same information is represented combinatorially as a nested set structure.

Definition 3.4.4. An *abstract simplicial complex*, $\tilde{K}$, consists of a set of vertices, $\text{vert}(K)$ and a set of abstract simplices, $\text{simp}(K)$, such that [52]:

1. Every simplex $\Delta^k \in \text{simp}(K)$ is a non-empty subset of $\text{vert}(K)$, or the simplices are a union of the vertices.
2. For all vertices, $v \in \text{vert}(K)$, there is also an abstract simplex $\{v\} \in \text{simp}(K)$.
3. For every non-empty abstract simplex, Δ^k, a non-empty subset of Δ^k is also an abstract simplex. This is referred to as a face of Δ^k.
4. For an abstract simplex $\Delta^k \in \text{vert}(K)$, the dimension of Δ^k is $\dim(\Delta^k) = |\Delta^k| - 1$, where $|.|$ denotes the number of elements in the set.

This notion of an abstract simplicial complex, $\tilde{K}$ allows for a combinatorial representation of a geometric simplicial complex K. The converse is also true; a topological space $\tilde{K}$ can always be associated with a geometric realisation $|\tilde{K}|$. Whilst potentially geometrically different, a homeomorphism will exist between K and $|\tilde{K}|$ [53].

Abstract simplicial complexes allow for the mapping between *categories*, specifically transitioning a geometric realisation of a simplicial complex embedded in Euclidean space to its abstract form. The abstract simplicial complex form can be analysed using group theory and maps to output information regarding the shape. By extension, abstract simplicial complexes can be formed from data and algorithmically analysed with computer packages [54, 55], to output topological information.

Drawing a parallel with daily transactions, the idea of mapping between categories can be likened to depositing money into an ATM. A paper £20 note and £20 represented digitally on a debit card fundamentally represent the same value but in vastly different forms. Physical cash can be held, whilst the value on a debit card is encoded as a sequence of ones-and-zeros inside a computer program. However, the same monetary value is attributed to the two forms. Each form also offers unique advantages: cash can be transferred hand-to-hand, without the need for online banking, whereas money on a debit card can be transferred instantaneously

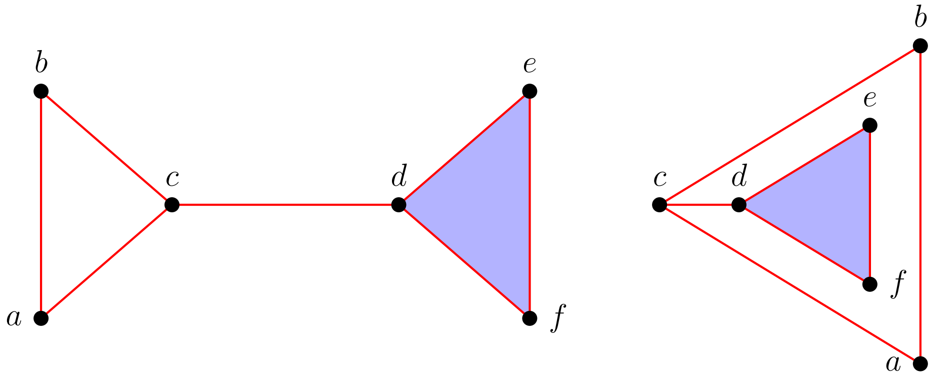


Figure 3.10: Two distinct geometric simplicial complexes with the same abstract simplicial complex.

to the other side of the planet. In this analogy, the paper cash and debit card represent different categories, and a mapping between categories – depositing at an ATM – is called a *functor*.

In the context of this thesis, geometric simplicial complexes are needed for creating simplicial complexes from data. Meanwhile, the tools in an algebraic framework called a *chain group* are required to define the homology groups, which is simply not possible over geometric simplicial complexes. Chain groups represent yet another mapping, transforming the combinatorial information of a simplicial complex into a group theory representation. The key takeaway is that the same information can be represented in various ways, each providing unique functionality and advantages.

Example 3.4.5. To help show the form of an abstract simplicial complex and its derivation from a geometric simplicial complex, consider the following example. Figure 3.10 displays a $2D$ shape, where the blue shaded region represents the interior of that simplex. Qualitatively analysing the depiction, there is a simplicial complex consisting of six vertices $\{a, b, c, d, e, f\}$, an edge set that connects these vertices $\{ab, ac, bc, cd, de, df, ef\}$, and an area spanned by the vertices $\{def\}$. This information is referred to as the *combinatorial information*, which only describes the simplicial complex's connections and not how the connections are arranged. Notably, these connections represent the topology and not necessarily the geometry, leading to countless possibilities of geometric realisations of this combinatorial information, a second realisation is shown in Figure 3.10.

From a given geometric simplicial complex, its combinatorial information forms the basis of an abstract simplicial complex, removing the need for a pictorial representation. By applying the rules outlined in Definition 3.4.4, its abstract simplicial complex is formed as a set of sets.

$$\mathrm{vert}(K) = \{a,\ b,\ c,\ d,\ e,\ f\} \tag{3.4}$$

$$\begin{aligned} \mathrm{simp}(K) = \{&\{a\},\ \{b\},\ \{c\},\ \{d\},\ \{e\}, \{f\},\ \{a,\ b\},\ \{a,\ c\}, \\ &\{b,\ c\},\ \{c,\ d\},\ \{d,\ e\},\ \{d,\ f\},\ \{e,\ f\},\ \{d,\ e,\ f\}\} \end{aligned} \tag{3.5}$$

However, much of this information can be inferred from the biggest simplices. This insight gives rise to a more efficient encoding method, called the *maximal simplex.* For example, the face $\{def\}$ implies the existence of three vertices $\{d,\ e,\ f\}$ and the edges connecting them $\{de,\ df,\ ef\}$. A maximal simplex contains all this information and it need not be repeated. As a result, a maximal simplex is not the face of another simplex in a simplicial complex. Consequently, the set of maximal simplices, $\max(K)$, provides sufficient information to represent a simplicial complex. This example can be adequately described with the abbreviated notation:

$$\max(K) = \{ab,\ ac,\ bc,\ cd,\ def\} \tag{3.6}$$

For simplicity, the braces are omitted. Throughout this thesis, this simplified notation using maximal simplices is used to represent simplicial complexes.

In the context of TDA, there is a requirement to analyse a spectrum of simplicial complexes parameterised by a metric; motivating the need for a *filtration.* A filtration of a simplicial complex, K, is an ordered list of subsets beginning with, $\emptyset$, and ending with K, such that:

$$\emptyset = K^0 \subset K^1 \subset \ldots \subset K^m = K \tag{3.7}$$

For any $i = \{0,\ \ldots,\ m-1\}$ such that $K^{i+1} = K^i \cup \Delta_{i+1}^{k_i}$ where $\Delta_{i+1}^{k_i}$ is the $(i+1)$-th simplex in the filtration [7]. Essentially, a filtration describes how the simplicial complex grows, concerning some notion of distance, by sequentially adding simplices.

Given that much of the data analysed in this thesis is not embedded in easily-visualised dimensions, it is especially important to have a dependable method of constructing simplicial complexes. Doing this ensures that the topological inferences drawn from higher-dimensional data are trustworthy, whilst also removing the need to visualise in higher dimensions. While there are various methods of con-

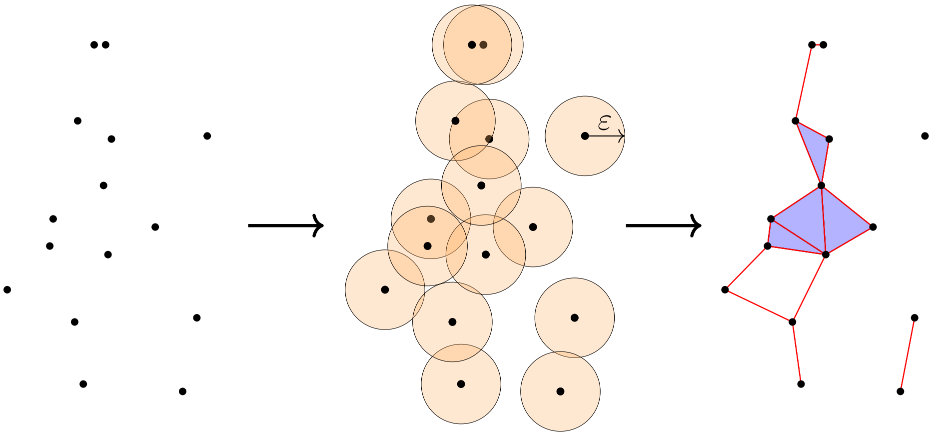


Figure 3.11: The process of constructing a VR complex from 2D point data.

structing simplicial complexes from data, this work focuses on the *Vietoris-Rips complex* (or VR complex) [56]. This technique allows for topological information to be extracted from data. Other mechanisms of constructing simplicial complexes from data, such as Čech [57] and Alpha [58] complexes, will yield minimally-different results given their distinct definitions. Nevertheless, with correct usage, all methods of constructing simplicial complexes should yield nearly identical results.

The foundational idea behind constructing complexes from data centres around the notion of placing k-dimensional open balls at each point; where each open ball has a radius, ε. How these balls intersect defines how the simplices connect the data points, representing the nodes in the complex. The rules determining the way these balls connect give rise to the different types of construction methods. For an example of forming a VR complex please refer to Figure 3.11, where a VR complex is created on some randomly-generated data at some arbitrary distance, ε.

Definition 3.4.6. To form a Vietoris-Rips complex, $VR_\varepsilon(X, \partial_X)$, let (X, ∂_X) be a finite metric space and $\varepsilon > 0$ be a fixed value. The abstract simplicial complex is determined by the rules [59]:

1. The vertices, $v \in X$, form the vertices in $VR_\varepsilon(X, \partial_X)$.
2. A k-simplex is formed when $\partial_X(v_i, v_j) \leq 2\varepsilon,\ \forall i, j \leq k$.

Determining an optimal value for ε is not a trivial task, especially for data embedded in higher dimensions. Setting this value too low might result in a simplicial complex being disjoint, which would fail to capture the true underlying topology and shape of the data. Conversely, if the value is too large, features may be obscured. Such is the case if a manifold that data are sampled from contains a hole of diameter, d, then if $\varepsilon > d$, simplices will be formed spanning the hole. Complications further escalate when considering a data set with varying feature scales. Imagine a data set possessing two topologically-interesting features: one scaled around 10^{-3} and the other at 10^{3}, where the data points at each feature are distributed according to the scale. Identifying a *single* ε that aptly captures the characteristics of both is particularly difficult, or even impossible. A solution to all these predicaments is presented within Section 3.6, where persistent homology is discussed.

This simplicial complexes section has shown how to create abstract mathematical objects from discrete data. Hopefully, readers can appreciate how simplicial complexes act as the glue joining data and the toolboxes equipped within an algebraic framework. The following chapter demonstrates how a topological invariant of the simplicial complexes, called homology, is determined from the abstract simplicial complexes. Following this, homology is then attributed to the data, giving an insight into the shape of a data set at some arbitrary distance parameter.

3.5 Homology

Previously in this chapter, a framework for extracting mathematical objects from data was introduced; however, there is still the requirement to deduce shapes from these objects. Herein, *homology* provides a precise solution to this issue, by coalescing everything discussed up until now.

Homology provides a method of categorising seemingly qualitative topological features into algebraic objects called *homology groups*. The homology group, $H_k(X)$, is a topological invariant associated with the k^{th} dimension. A common interpretation of homology groups is that $H_k(K)$ encodes information that counts the number of k-dimensional holes in K. Under the rules of topology, discontinuities (or voids) cannot be created or destroyed under homeomorphisms. Therefore, a simplicial complex can be categorised by the properties underpinned by homology. The *simplicial homology* is then used to categorise and compare between simplicial complexes, and by extension, data sets.

However, the computational practicality of homology does not come for free. The definition of homology groups is less transparent than ones typically found within engineering, having to rely on abstract concepts within group theory. The involvement of the task in defining homology is unavoidable, potentially acting as a gatekeeper to topological data analysis. However, once this threshold is crossed, homology gives life to a rich and unique theory, easily extensible to data.

Homology is not a new analysis, consequently, many great introductions and summaries exist, both in the classical mathematical sense [48, 50, 51, 60], and the more modern computational approach [38, 46, 61, 62], and it suffices to say that this description of homology presented here cannot be singly attributed to any of them.

In addressing the definition of homology, a few subtleties need to be addressed which were previously overlooked, and deemed nonessential in describing earlier topics. One such requirement is having a mathematically-rigorous definition for the boundary of a simplex, which was qualitatively presented in Definition 3.4.2. Subsequently, these ideas lead to quantifiably defining the boundary of a simplicial complex, which requires a few more nuances involving simplex orientation.

Definition 3.5.1. The *boundary* of the simplex, Δ^k, with the vertices $[v_0, \ldots, v_k]$, is:

$$\partial\Delta^k = \bigcup_{i=1}^{k} [v_0, \ldots, \hat{v}_j, \ldots, v_k] \tag{3.8}$$

where the element $\hat{v}_j$ is omitted from that set [50].

At a glance, this definition is somewhat transparent, given that a simplex involves all vertices being fully connected. Excluding a single vertex then gives a fully connected simplex in a dimension reduced by one; this resultant structure is the face corresponding to that vertex, as discussed when setting a barycentric coordinate, λ_i, to zero. Then by performing this omission over the length of the simplex, all possible faces are created. For the case of a tetrahedron, $\Delta^3 = \{v_0v_1v_2v_3\}$, the boundary as per Definition 3.5.1, is $\partial\Delta^3 = \{v_1v_2v_3,\ v_0v_2v_3,\ v_0v_1v_3,\ v_0v_1v_2\}$. This result is the four triangles which are the tetrahedron faces, which collectively define its external surface or boundary.

Example 3.5.2. The boundary operator given in Definition 3.5.1 is valid for the case of individual simplices, but breaks down when applied to simplicial complexes. This shortcoming is evident when applied to perhaps the most rudimentary nontrivial simplicial complex, $K = \{v_0v_1, v_1v_2\}$. The geometric realisation of K is anything homeomorphic to a straight line with a vertex somewhere along its length, as shown in Figure 3.12.

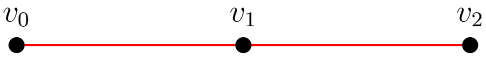


Figure 3.12: A simplicial complex on which the simplex boundary operator fails.

From an intuitive (and correct) standpoint, the boundary of the simplicial complex should be the two exterior points, $\partial K = \{v_0, v_2\}$. Despite this intuition, if the boundary operator given in Definition 3.5.1 is distributed over the simplices in K, the result is $\partial K = \{v_0, v_1, v_1, v_2\} = \{v_0, v_1, v_2\}$. This outcome is incorrect, since v_1, the midpoint of the line, is erroneously being considered as a part of the simplicial complex boundary; something that inherently represents the exterior. The amendment for this problem is by modifying Definition 3.5.1 to include simplex *orientation.* An orientation is assigned to each simplex according to how the vertices are ordered, creating what is termed an *oriented simplex.*

For an oriented simplex $\Delta^1_+ = [v_0, v_1]$, the orientation presented conveys a directionality from v_0 to v_1. Previously, this simplex could have also been written $\Delta^1_- = [v_1, v_0]$. However, now that orientation is being considered, these are different, as $[v_1, v_0]$ implies a simplex from v_1 to v_0, which implies opposing orientation. Orientation is not limited to Δ^1 and is extensible to any dimension. Consider, $\Delta^2_+ = [v_0, v_1, v_2]$ which is perceived as having a clockwise direction, implying that $\Delta^2_+ = [v_0, v_1, v_2]$, $[v_1, v_2, v_0]$, and $[v_2, v_0, v_1]$ all have the same orientation, as they are *even permutations* of one another, this is a rotation of the indices. Whereas, $\Delta^2_- = [v_0, v_2, v_1]$ has changed the orientation. Imagine that the indices, of the simplex form a unit of a repeating sequence, the sequences with positive orientation increase left to right, whereas the negatively-oriented simplices decrease. This notion of orientation and permuting can be generalised to higher dimensions, where an even permutation is defined as a permutation of the vertices that results in a simplex with the same orientation.

The *orientation* of the vertices of a simplex, Δ^k, is an equivalence class of orderings of the vertices under the equivalence relation that two orderings are the same if they differ by an *even permutation.* An even permutation can be expressed as a composition of even permutations [38]. There are only two possible equivalence classes in all dimensions, which are positive and negative.

However, a slight complication remains. Much like the requirement to convert geometric simplicial complexes into abstract simplicial complexes, without any loss of generality, the combinatorial form of an abstract simplicial complex can be converted into an algebraic counterpart via group theory. For each dimension k, in a simplicial complex, a group known as a *chain group* is used to represent the simplices.

Definition 3.5.3. To each standard simplex Δ_i^k of a simplicial complex K, a free Abelian group, $C_k(K)$, called a *chain group* is associated with it. The k^{th} chain group is the set of all combinations of k-simplices in the simplicial complex K [61].

Now consider that K contains l_k k-simplices for all possible values of k. The *k-chain group* of K, $C_k(K)$ is the free Abelian group generated by the oriented k-simplices of K. Consequently, any element $c_k \in C_k(K)$ can be expressed as,

$$c_k = \sum_{i=1}^{l_k} f_i \Delta_i^k, \;\; f_i \in \mathbb{Z} \tag{3.9}$$

provided the following criteria are satisfied [50],

1. A negation of simplices. $\Delta_i^k + (-\Delta_i^k) = 0, \;\forall i, k$.
2. A linearity over the elements. $\sum_{i=1}^{l_k} f_i \Delta_i^k + \sum_{i=1}^{l_k} g_i \Delta_i^k = \sum_{i=1}^{l_k} (f_i + g_i) \Delta_i^k$ where $f_i, g_i \in \mathbb{Z}$.

There are numerous intricate details within Definition 3.5.3. To aid comprehension, it is beneficial to elaborate on some more abstract points and revisit certain group definitions. Chain groups serve as crucial links, transforming the combinatorial abstract simplicial complexes into algebraic counterparts, thereby opening the door for the analysis of simplicial complex analogues with group theory. It is important to understand the meaning of how a free Abelian group generated by oriented simplices represent simplicial complexes, and what the elements of the group intuitively describe. An apt analogy might be the construction of a wireframe, such as a truss bridge, from toothpicks and marshmallows. In this case, each toothpick and marshmallow is considered a generator of the truss bridge, and by combining these elements in various ways, the entire wireframe structure, inclusive of its loops, hollows, and boundaries can be described. The free Abelian group generated by the simplices of a simplicial complex is almost exactly analogous to the truss bridge created from toothpicks and marshmallows. The chain groups offer an algebraic structure that captures the simplicial complex's topological aspects. By imposing that the simplices are generators of the free Abelian group, this means that every individual k-simplex is treated as an elementary piece that can be combined with the others to capture some aspect of the simplicial complex's shape. An element of the chain group C_1 might look something like $[ab] - 7[bc] + 2[cd]$, where $[ab]$, $[bc]$, and $[cd]$ are 1-simplices. Generally, the complete set of chain elements encompasses all integer-linear combinations of the simplices. Expressing chain elements in this form allows for the algebraic combinations of simplices, thereby providing a method of describing subsets within a topological space. Chains are particularly important in homology, as they enable the algebraic representation of boundaries and voids within a simplicial complex.

After presenting all the preliminary definitions for homology, following with a practical example calculation of homology groups for a simplicial complex will prove insightful. Such an illustration showcasing the process, presented in Example 3.5.1, helps give a deeper understanding than by theory alone.

After transitioning into an algebraic domain, a more robust definition for the boundary operator can now be established. With simplicial complexes now represented by chain groups, the problem has become entirely algebraic.

Definition 3.5.4. The *boundary operator*, ∂_k, maps between chain groups,

$$\partial_k \colon C_k(K) \to C_{k-1}(K) \tag{3.10}$$

Given an oriented simplex $\Delta^k = [v_0, \ldots, v_k]$, a positive sign is assigned to every member of the even permutation class of Δ^k and a negative sign to every member of the odd permutation class. The boundary operator must now obey the rules [50]:

1. For an oriented simplex,

$$\partial \Delta^k = \sum_{i=0}^{k} (-1)^i [v_0, \ldots, \hat{v}_i, \ldots, v_k] \tag{3.11}$$

   where $[v_0, \ldots, \hat{v}_i, \ldots, v_k]$ represents the face of the simplex with the i^{th} vertex omitted. Note that every successive omission changes the orientation of the face.

2. Thinking of a simplicial complex, K, as the sum[3] of all the standard simplices required to construct it, $K = \sum_{i,k} \Delta_i^k$ for $\Delta_i^k \subset K$. The boundary operator is a linear function over all the simplices in the simplicial complex,

$$\partial(K) = \partial\big(\sum_{i,k} \Delta_i^k\big) = \sum_{i,k} \partial(\Delta_i^k),\ \Delta_i^k \subseteq K \tag{3.12}$$

Despite the notion of chain groups, the mathematical definitions for the boundary of simplices and simplicial complexes are strikingly similar. However, given that chain groups fundamentally portray the simplicial complexes algebraically, this is merely a formality when developing intuition. Definition 3.5.4 is an ideological extension of Definition 3.5.1, much like simplicial complexes being an extension to simplices. The primary distinction between the two boundary definitions is that the simplicial complex boundary considers orientation, as detailed in the first item in Defintion 3.5.4. The second item emphasises that this process is distributed over all simplices in the simplicial complex.

Given the boundary map defined by $\partial_k \colon C_k(K) \to C_{k-1}(K)$, it is vital to comprehend the map's injective nature, wherein it is a mapping to a subspace rather than the entire space. While ∂_k does indeed map k-simplices to their $(k-1)$-simplex faces, it does not fully cover the entirety of $C_{k-1}(K)$. This is because some simplices within $C_{k-1}(K)$ are *not* faces of higher-dimensional simplices. Consequently, the boundary map's image is confined to a subset of the $C_{k-1}(K)$. Grasping this nuance is pivotal for a clear understanding of homology groups.

Example 3.5.5. Revisiting the simplicial complex in Figure 3.12 with this new definition for the boundary of $K = \{v_0v_1, v_1v_2\}$. The result $\partial K = (v_1 - v_0) + (v_2 - v_1) = v_2 - v_0$ is obtained. This now gives the correct, and intuitive answer for the boundary of K.

As stated in Definition 3.5.4, the boundary is a mapping between chain groups $\partial_k \colon C_k(K) \to C_{k-1}(K)$, and thus repeated applications between successive chain groups gives a *chain complex* [61],

$$\ldots \to C_k \xrightarrow{\partial_k} C_{k-1} \xrightarrow{\partial_{k-1}} \ldots \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} 0 \tag{3.13}$$

---

[3] The summation is now being used in place of the union, as it is required that two simplices with opposing orientations cancel each other out, as given in Definition 3.5.3 and set-theoretic notation does not easily allow for the cancelling of elements.

For a k-dimensional simplicial complex, this process is repeated k times, leading to the final result of 0, and with the initial condition $C_i(K) = 0$ for $i > \dim(K)$. Intuitively, $C_i(K) = 0$ for $i < 0$ as there are no simplices present in these dimensions. In all of these cases, 0 represents the zero (trivial) group.

There is a sense of nestedness associated with chain complexes, as shown in Figure 3.14, which arises by applying successive boundary operators. A particularly intriguing subgroup of the chain group is the kernel, which is the subgroup consisting of all the elements mapped to 0 over a single application of a boundary map.

Definition 3.5.6. For a simplicial complex K, elements of the chain group, $z_k \in C_k(K)$, are called *k-cycles* if $\partial z_k = 0$. The group of k-cycles, $Z_k(K)$, is given by the kernel of the boundary map,

$$Z_k(K) = \ker(\partial_k \colon C_k(K) \to C_{k-1}(K)) = \{z_k \in C_k \colon \partial_k z_k = 0\} \tag{3.14}$$

and $Z_k(K)$ is a subgroup of $C_k(K)$ [46].

The term k-cycle arises since these elements encapsulate the cyclic-looping structure of a simplicial complex in the k^{th} dimension. The group of k-cycles $Z_k(K)$, is defined as the group of all chains of k-simplices that have an empty boundary. This means that k-cycles represent holes, or regions without an exterior in dimension k, since, by definition they have no boundary. The most intuitive group to picture is $Z_1(K)$: this group denotes the collection of all linear combinations of edges (1-simplices), forming closed paths (cycles), without any interior; with the same reasoning extensible to higher dimensions.

Furthermore, it is worth considering which elements of the k^{th} chain group are boundaries of the $(k+1)^{\text{th}}$ chain group, that is the image of the $(k+1)^{\text{th}}$ chain group under the boundary map.

Definition 3.5.7. For a simplicial complex K, elements of the chain group $b_k \in C_k(K)$, are called *k-boundaries* if there exists a chain group $C_{k+1}(K)$, such that $\partial C_{k+1}(K) = b_k$. The group of k-boundaries $B_k(K)$, is given by,

$$B_k(K) = \operatorname{Im}(\partial_{k+1} \colon C_{k+1}(K) \to C_k(K)) = \{b_k \in C_k \colon \exists b'_k \in C_{k+1}, b_k = \partial b'_k\} \tag{3.15}$$

and $B_k(K)$ is a subgroup of $C_k(K)$ [46].

Finally, there is one last concept to understand before defining homology. When

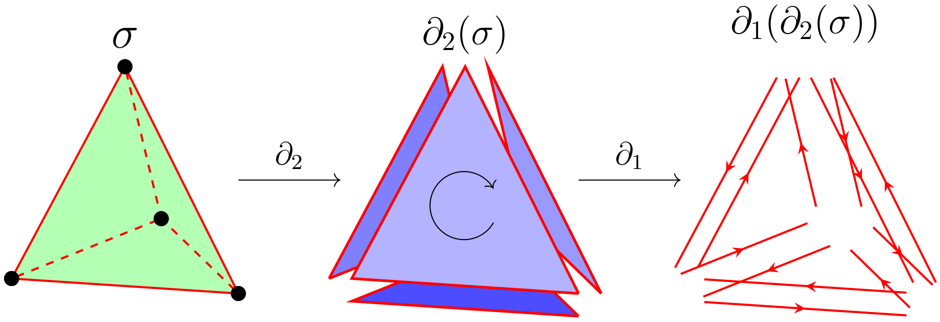


Figure 3.13: Visualisation of two successive boundary maps applied to the tetrahedron, where it can be seen that all matching edges have opposing orientations after two applications of the boundary map.

performing the composition of two successive boundary operations, the boundary operator exhibits a special characteristic: two successive applications give the zero group. In other words, the boundary of a boundary is empty,

$$\partial_{k-1} \circ \partial_k = 0, \ \forall k \tag{3.16}$$

This is a highly significant result, and its proof is supplied in Appendix C. Here, the general idea is demonstrated that a singular case of equation (3.16) holds true. To illustrate this result, the tetrahedron $\Delta_3 = \{v_0v_1v_2v_3\}$ is analysed. To aid understanding, Figure 3.13 is supplied to assist in visualising the concepts. The first application of the boundary gives the result,

$$\partial_3(\{v_0v_1v_2v_3\}) = \{v_1v_2v_3\} - \{v_0v_2v_3\} + \{v_0v_1v_3\} - \{v_0v_1v_2\} \tag{3.17}$$

The four faces of the tetrahedron are obtained, as expected. However, when compared to the previous boundary map of the tetrahedron, the orientation is now being captured in the signs of the vertex sets. Consequently, these faces no longer all belong to the same equivalence class. The second application of the boundary map gives,

$$\begin{aligned}\partial_2(\{v_1v_2v_3\}) - \partial_2(\{v_0v_2v_3\}) + \partial_2(\{v_0v_1v_3\}) - \partial_2(\{v_0v_1v_2\}) &= \\ [\{v_2v_3\} - \{v_1v_3\} + \{v_1v_2\}] - [\{v_2v_3\} - \{v_0v_3\} + \{v_0v_2\}]+ \\ [\{v_1v_3\} - \{v_0v_3\} + \{v_0v_1\}] - [\{v_1v_2\} - \{v_0v_2\} + \{v_0v_1\}] &= 0\end{aligned} \tag{3.18}$$

Over the two applications of boundary map, the orientation changes once for each simplex, resulting in the cancellation of all terms. This example verifies the result

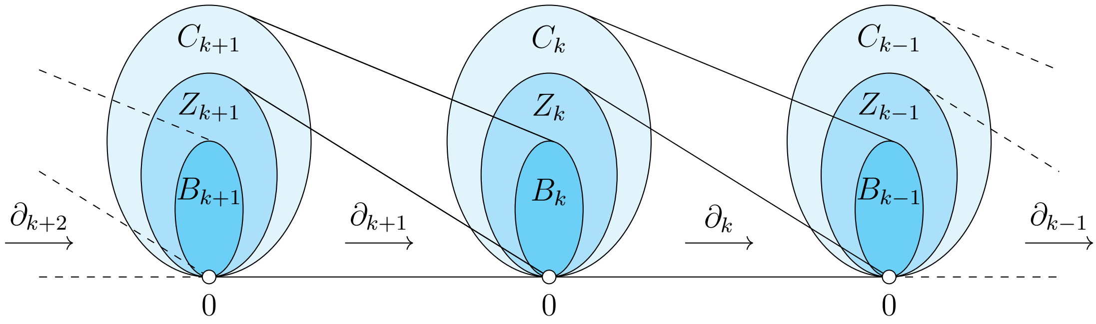


Figure 3.14: A chain complex with boundary maps between the chain groups, showcasing the nested relationship between C_k, Z_k, and B_k.

$\partial_2 \circ \partial_3(\Delta_3) = 0$. The general idea for higher-dimensional simplices follows a similar pattern where the orientation of each simplex changes once over the composition, leading to the cancellation of terms when summed. Whilst the example shown here is applied to a simplex, the theory and proof presented in Appendix C are valid for simplicial complexes. This is because the boundary is injective to the faces of simplices inside the lower chain group. Essentially, the extension to prove equation (3.16) for simplicial complexes is distributing (or repeating) the logic over all the simplices represented in the chain group.

Corollary 3.5.8. A very important result follows from equation (3.16). For a simplicial complex K, any element of the boundary group $b_k \in B_k(K)$ has the property $\partial_k b_k = 0$. Therefore, $B_k(K) \subseteq Z_k(K)$ where $Z_k(K)$ is the group of k-cycles, the group of all the elements mapped to 0. Since both $Z_k(K)$ and $B_k(K)$ are Abelian, a property inherited by being subgroups of $C_k(K)$. $B_k(K)$ is a normal subgroup of $Z_k(K)$. Consequently, the set of cosets $Z_k(K)/B_k(K)$, form a quotient group [50].

Corollary 3.5.8 brings together much of the background theory discussed in previous sections, albeit in a rather implicit manner. To better grasp Corollary 3.5.8, it is beneficial to dissect this information into logical steps.

1. The first stage is understanding why $\partial_k b_k = 0$. This is because $\partial_k b_k$ is essentially the application of two boundary operators since the element b_k arises from the boundary operation of a prior chain element. Hence, $\partial_k b_k = (\partial_k \circ \partial_{k+1})c_{k+1}$, and by equation (3.16), the result is zero.

2. Secondly, if all the elements in $B_k(K)$ are mapped to zero after the application of ∂_k then $B_k(K)$ must be a subset of all the elements that are mapped to zero.

According to Definition 3.5.6, this is the kernel $Z_k(K)$.

3. Thirdly, from Definitions 3.5.6 and 3.5.7, it is known that $Z_k(K) \subset C_k(K)$ and $B_k(K) \subset C_k(K)$. Given that chain groups $C_k(K)$, are defined as Abelian, then so are their respective subgroups $B_k(K)$ and $Z_k(K)$, since subgroups inherit the operator from the parent group, by Definition 3.2.3.

4. Finally, it has also been shown that $B_k(K) \subset Z_k(K)$, and since $B_k(K)$ is also Abelian, then $B_k(K)$ is a normal subgroup $B_k(K) \triangleleft Z_k(K)$. Therefore, the set of cosets $Z_k(K)/B_k(K)$ is a quotient group.

Definition 3.5.9. The *homology groups*, $H_k(K)$, are the quotient groups,

$$H_k = Z_k(K)/B_k(K) \tag{3.19}$$

Subsiding all this mathematical abstraction, the k^{th} homology group $H_k(K)$ can be perceived as the cycles in $C_k(K)$ that are not boundaries of the elements within $C_{k+1}(K)$. An element of C_k represents a cycle, implying it encloses a k-dimensional region. The fact that it isn't a boundary means the interior bound by the k-cycle is not included in the space; from here emerges the idea of counting the k-dimensional holes. A generalisation of the rule is that $H_k(K)$ pertains information to count the k-dimensional holes in the simplicial complex, K. For example, $H_1(K)$ precisely describes the inability of 1-cycles in K to bound 2-simplices in K; epitomising the exact idea of a 1-dimensional hole. Considering other visualisable homology groups,

$H_0(K)$ is the only exception to the rule, as this encodes information about the number of connected regions in K, i.e. the number of disjoint subsets of K.

$H_1(K)$ encodes information about $1D$ holes, these can be visualised as circular holes.

$H_2(K)$ encodes information about $2D$ holes, these can be visualised as cavities.

$H_k(K)$ encodes information about kD holes.

Before proceeding onto an example calculation for homology, there are a few extra, but very important definitions built atop homology. If $H_k(K)$ is a vector space, the elements are the *homology classes* of K. For a cycle $z_k \in Z_k(K)$, its homology class is the coset $c_k + B_k(K) = \{c_k + b_k \colon b_k \in B_k(K)\}$. Cycles are said to be *homologous* if they are in the same homology class [46]. In essence, a k-cycle that wraps around a hole once is homologous to a k-cycle that wraps around the same hole twice.

From the homology groups, the dimension of these spaces give topological invariants called the *Betti numbers*,

$$\beta_k = \dim(H_k(K)) \tag{3.20}$$

The k^{th} Betti number quantifies the k-dimensional holes present within a simplicial complex [62]. Throughout this thesis, Betti numbers serve as topological invariants to quantify the number of holes in a simplicial complex constructed from some data. Since Betti numbers are topological invariants, they can be used similarly to the other topological invariants listed in Section 3.3.3, in checking if two spaces are homeomorphic. A mismatch in the Betti numbers of two spaces signifies that the two spaces are not topologically identical. Unlike the previously listed topological invariants, Betti numbers can be determined computationally. However, when applying a homological perspective on data, the topological invariants are less clear-cut because of factors such as data sampling and noise from real-world observations. Hence, a probabilistic approach via the *Wasserstein-distance* is considered.

3.5.1 Calculating Homology

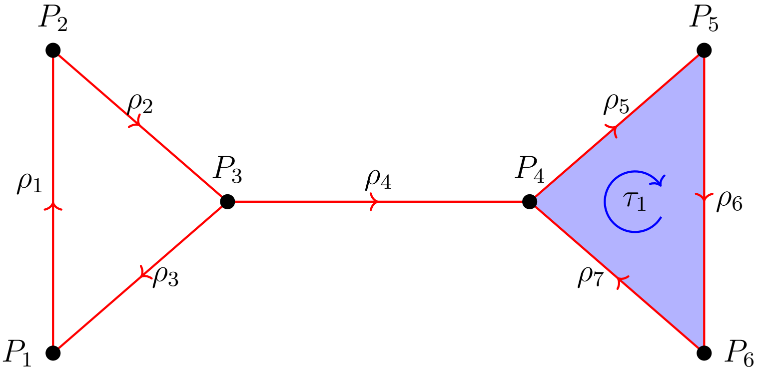


Figure 3.15: An oriented simplicial complex, K.

At last, the final stage in describing homology is to motivate with an example, tracing all the steps outlined along the way; all the way back from a geometric simplicial complex to the Betti numbers. Consider the simplicial complex presented in Figure 3.10, but imposing the notion of oriented simplices, this result is presented in Figure 3.15. This case is shown in its entirety to back up the inner workings of the definitions and showcase what truly is going on at the lowest level.

Given the geometric simplicial complex, K in Figure 3.10, the abstract simplicial complex is,

$$\tilde{K} = \{a, b, c, d, e, f, ab, bc, ca, cd, de, ef, fd, def\} \tag{3.21}$$

For ease of notation at a later stage, a new naming convention is used over all the simplices in K,

$$\begin{gathered} P_1 = a,\ P_2 = b,\ P_3 = c,\ P_4 = d,\ P_5 = e,\ P_6 = f \\ \rho_1 = ab,\ \rho_2 = bc,\ \rho_3 = ca,\ \rho_4 = cd,\ \rho_5 = de,\ \rho_6 = ef,\ \rho_7 = fd \\ \tau_1 = def \end{gathered} \tag{3.22}$$

The newly attributed names and the orientation associated with each simplex are shown in a geometric realisation in Figure 3.15. Using this new naming convention, the chain groups may be formed using summation notation[4],

$$C_0(K) = \left\{ \sum_{i=1}^{6} r_i P_i \ : \ \forall r_i \in \mathbb{Z} \right\} \tag{3.23}$$

$$C_1(K) = \left\{ \sum_{j=1}^{7} m_j \rho_j \ : \ \forall m_j \in \mathbb{Z} \right\} \tag{3.24}$$

$$C_2(K) = \{ n_1 \tau_1 \ : \ n_1 \in \mathbb{Z} \} \tag{3.25}$$

Now the chain groups have been defined, each homology class can be calculated. For intuition, as mentioned in the theory, $C_0(K)$ represents the set of all integer sums over P_i, that is the group is generated by the vertices of the simplicial complex. Similarly, $C_1(K)$ represents the integer sum over all the edges in the simplicial complex. However, since there is only a single 2-simplex, $C_2(K)$ is simply all integer scales of the generator simplex.

Calculating $H_2(K)$

Starting with $H_2(K)$, this case is trivial, since there is only a single 2-simplex, and it, therefore, cannot enclose k-dimensional space, therefore the answer is intuitively 0. However, in search of completeness and rigour, the second homology group is $\ker(\partial_2\colon C_2 \to C_1)$ modulo $\mathrm{Im}(\partial_3\colon C_3 \to C_2)$, and since there are no 3-simplices in K, the image is trivially the zero group; meaning $H_2(K)$ is just $\ker(\partial_2\colon C_2 \to C_1)$. To determine the kernel, consider the action of the boundary map over the simplex

---

[4] For completeness, a more compact way of representing the chain groups makes use of the direct-sum notation. However, this has not been introduced. The direct sum equivalents would be $C_1(K) = \bigoplus_{i=1}^{6} \mathbb{Z}_i P_i$, and $C_2(K) = \bigoplus_{i=1}^{7} \mathbb{Z}_i \rho_i$.

τ_1 via the application of Definition 3.5.4,

$$\begin{aligned}
\partial_2\tau_1 &= \partial_2(P_4P_5P_6) \\
&= P_5P_6 - P_4P_6 + P_4P_5 \\
&= \rho_6 + \rho_7 + \rho_5
\end{aligned} \tag{3.26}$$

When calculating the boundary of a chain element, the coefficient is distributed over all the edges, thus giving, $\partial_2 c_2 = n_1\rho_6 + n_1\rho_7 + n_1\rho_5$, where c_2 represents a chain element in C_2, where $n_1 \in \mathbb{Z}$. Therefore, since the application of ∂_2 results in a combination of simplices and does not map τ_1 to zero, then τ_1 is not in $\ker(\partial_2)$. Given that this is the only generator of $C_2(K)$, this means $\ker(\partial_2) = 0$, and thus $H_2(K) = 0$, giving the Betti number $\beta_2 = 0$, indicating that there are no $3D$ cavities in K, which is clear from Figure 3.15.

Calculating $H_1(K)$

Moving onto $H_1(K)$, recalling that $H_1(K) = \ker(\partial_1)/\mathrm{Im}(\partial_2)$, therefore, the two elements that need to be determined: the 1-cycles, which are the elements of $\ker(\partial_1\colon C_1(K) \to C_0(K))$, and the 1-boundaries, the elements of $\mathrm{Im}(\partial_2\colon C_2 \to C_1)$. A general element in the chain group is given as $c_{1_i} = \sum_{j=1}^{7} m_j\rho_j$, for the homology, its boundary is required; to perform this, consider each 1-simplex individually,

$$\begin{aligned}
\partial_1\rho_1 &= P_2 - P_1 \\
\partial_1\rho_2 &= P_3 - P_2 \\
\partial_1\rho_3 &= P_1 - P_3 \\
\partial_1\rho_4 &= P_4 - P_3 \\
\partial_1\rho_5 &= P_5 - P_4 \\
\partial_1\rho_6 &= P_6 - P_5 \\
\partial_1\rho_7 &= P_4 - P_6
\end{aligned} \tag{3.27}$$

Therefore, distributing the coefficients over the application of the boundary operator to the chain element gives,

$$\begin{aligned}
\partial_1 c_1 &= m_1(P_2 - P_1) + m_2(P_3 - P_2) + m_3(P_1 - P_3) + m_4(P_4 - P_3) \\
&\quad + m_5(P_5 - P_4) + m_6(P_6 - P_5) + m_7(P_4 - P_6) \\
&= P_1(m_3 - m_1) + P_2(m_1 - m_2) + P_3(m_2 - m_3 - m_4) \\
&\quad + P_4(m_4 - m_5 + m_7) + P_5(m_5 - m_6) + P_6(m_6 - m_7)
\end{aligned} \tag{3.28}$$

Grouping the coefficients is very informative for determining cycles, since it is desired to find the unique sets of coefficients that result in 0, thus giving the cycles described by the kernel of the map. There are two cyclic behaviours arising here, one emerging from the set of points $\{P_1,\, P_2,\, P_3\}$, and the second from $\{P_4,\, P_5,\, P_6\}$. Both of these form cycles that will be in the kernel, however only $\{P_1,\, P_2,\, P_3\}$, will contribute to the homology group as $\{P_4,\, P_5,\, P_6\}$ is part of the image from the higher order chain group. For completeness, both cycles will be considered to showcase the quotient group in action removing the cycles which are faces of simplices. Therefore, when considering the kernel, $\partial_1 c_1 = 0$, two distinct criteria satisfy this condition, which are,

1. $l_1 = m_1 = m_2 = m_3$ defining the interiorless loop.
2. $l_2 = m_5 = m_6 = m_7$, defining the boundary loop.

Therefore, these two conditions outline the requirements for the equivalence classes forming the kernel.

$$Z_1(K) = \{l_1\rho_1 + l_1\rho_2 + l_1\rho_3,\; l_2\rho_5 + l_2\rho_6 + l_2\rho_7 \;:\; \forall l_i \in \mathbb{Z}\} \tag{3.29}$$

Since there are two families of cycles described in $Z_1(K)$, this means that $Z_1(K) \cong \mathbb{Z}^2$. Now to determine the image in $C_1(K)$, the boundary group is required. This result has already been calculated in $H_2(K)$.

$$B_1(K) = \partial_2 C_2(K) = \{n_1\rho_6 + n_1\rho_7 + n_1\rho_5 \;:\; n_1 \in \mathbb{Z}\} \tag{3.30}$$

Where there is a single element contained in $B_1(K)$, and thus $B_1(K) \cong \mathbb{Z}$. Now, the interesting part arises, when forming the quotient group. From equation (3.29) and (3.30), it can be seen that when $l_2 = n_1$, these two cycles are the same, and this family of cycles was formed as a boundary from the previous chain group in the chain complex. Therefore, when calculating the quotient group this cycle is purged.

$$H_1(K) = Z_1(K)/B_1(K) = \{l_1\rho_1 + l_1\rho_2 + l_1\rho_3 \;:\; l_1 \in \mathbb{Z}\} \tag{3.31}$$

Meaning that the cycle $l_1\rho_1 + l_1\rho_2 + l_1\rho_3$ generates a nontrivial homology class, leading to the fact that $H_1(K)$ is generated by this cycle, giving $H_1(K) \cong \mathbb{Z}$. Therefore, the Betti number, $\beta_1 = \dim(H_1(K)) = 1$, stating that there is a single hole in the simplicial complex which is bounded by edge set $\{\rho_1,\, \rho_2,\, \rho_3\}$.

Calculating $H_0(K)$

For transparency, this calculation will go in full depth by considering general elements and the cosets, rather than 'hand-wavey' notions of dividing by the quotient group.

Since 0-simplices by definition have no boundaries, as they are the most fundamental bounding element, then $Z_0(K) = C_0(K)$,

$$Z_0(K) = \left\{ \sum_{i=1}^{6} r_i P_i \; : \; \forall r_i \in \mathbb{Z} \right\} \tag{3.32}$$

This means that a general 0-cycle $z_0 = r_1P_1 + r_2P_2 + r_3P_3 + r_4P_4 + r_5P_5 + r_6P_6$ is valid for all $r_i \in \mathbb{Z}$. This means there are six independent generators of $Z_0(K)$, and thus $Z_0(K) = \mathbb{Z}^6$.

The image was previously calculated in equation (3.28),

$$\begin{aligned} B_0(K) = \{P_1(m_3 - m_1) + P_2(m_1 - m_2) + P_3(m_2 - m_3 - m_4) \\ +P_4(m_4 - m_5 + m_7) + P_5(m_5 - m_6) + P_6(m_6 - m_7) \; : \; \forall m_i \in \mathbb{Z}\} \end{aligned} \tag{3.33}$$

But since the sum of all the coefficients in equation 3.33 equals zero, this means there are only five generators and thus $B_0(K) = \mathbb{Z}^5$, hence $H_0 = \mathbb{Z}^6/\mathbb{Z}^5 = \mathbb{Z}$. However, for full completeness consider a single coset and group the coefficients,

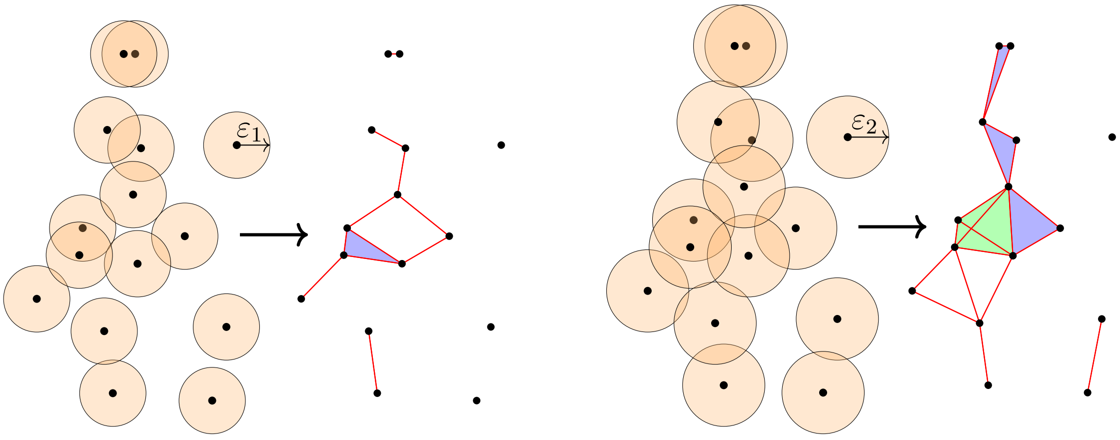
$$\begin{aligned} h_0 &= z_0 + B_0(K) \\ &= P_1(r_1 + m_3 - m_1) + P_2(r_2 + m_1 - m_2) + P_3(r_3 + m_2 - m_3 - m_4) \\ &\quad + P_4(r_4 + m_4 - m_5 + m_7) + P_5(r_5 + m_5 - m_6) + P_6(r_6 + m_6 - m_7) \end{aligned} \tag{3.34}$$

Here, h_0 represents a general form of a coset in $H_0(K)$. The 0^{th} homology group, $H_0(K)$ is the set of all such cosets. Now consider another coset h_0', which is formed by another 0-cycle, z_0'. However, the difference $z_0 - z_0'$ only differs by a 0-boundary, i.e, $h_0 = h_0'$ if and only if $z_0 - z_0' \in B_0(K)$. Since every combination of points P_i can be obtained as a 0-boundary, this means that any two 0-cycles z_0 and z_0', and therefore any two cosets h_0 and h_0' in $H_0(K)$ are homologous, therefore $H_0(K) \cong \mathbb{Z}$

This approach may appear on the surface very verbose, especially when compared to looking at the shape and stating these exact results via intuition. However, this conclusion was arrived at using a generalisable algebraic framework. It is also not too difficult to envisage how this could be implemented on computers, opening the door to truly unique analytical techniques – giving a fresh perspective on decades-old problems. However, there are still a few caveats when calculating the homology of a single simplicial complex, and this is where persistent homology rectifies such issues.

3.6 Persistent Homology

After all this meticulous outlining of an abstract method for quantifying the shape of data, when a homology approach is applied to real-world data, the process is not without flaws. There is also a very fundamental question overhanging: which length scale ε is most suitable for creating VR complexes that best represent the topology of a data set? Addressing this concern is crucial, as even minor variations in ε can yield drastically different simplicial complexes, leading to varying topological invariants, and data sets do not come with some implicit representative length scale. Figure 3.16 illustrates the difference between two simplicial complexes, even with just a 20% increase in ε. A quick qualitative assessment shows that Figure 3.16 (a) at ε_1 has six connected components, a single hole and a maximal dimension of two. In contrast, Figure 3.16 (b) has three connected components, two holes, and a maximal dimension of three. Determining the homology of these simplicial complexes gives discrepancies between the like Betti numbers, suggesting that these data sets are topologically different. Potentially, leading to the inference that these simplicial complexes represent different data, despite being formed atop the same set.



(a) ε_1 (b) $\varepsilon_2 = 1.2\varepsilon_1$

Figure 3.16: Two VR complexes, at different ε, built atop the same randomly-generated data.

Furthermore, manifolds may display characteristics across a variety of length scales. Figure 3.17 illustrates a noisy, artificially-created, continuous point cloud, containing three circular holes (shown as the hatched-green regions) with varying radii; implying $\beta_1 = 3$. Notably, the distribution of points along these holes' circumferences is roughly inversely proportional to their sizes. Open balls representing the formation of a VR-complex, are presented at three different ε, each revealing different aspects of the data manifold. While this data set was fabricated with 1D holes in mind and the discussion is limited to this dimension, the logic is extensible to higher-dimensional holes. Upon qualitatively examining each subfigure within Figure 3.17, the following observations are made,

(a) Only the small central hole is being captured, resulting in a simplicial complex with $\beta_1 = 1$. This means there are no topological insights into the medium and large holes since their perimeters are not encircled by connected paths.

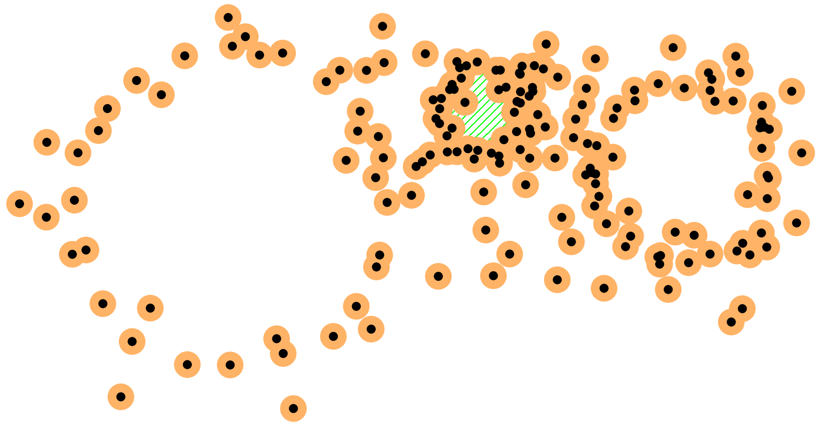
(b) At this ε, the small and medium holes are detected as intended. However, the large hole's perimeter remains as disjoint clusters. It is concerning that holes are emerging which are artefacts of noise and discrete sampling. Highlighted in red, these areas of *topological noise* have little purpose in describing the true topology of manifolds. Topological noise is an unavoidable consequence when creating simplicial complexes from discrete and noisy data. Additionally, the small hole has been partitioned into two sub-holes, as a result of the noise in the data. These artefacts introduced at this ε result in $\beta_1 = 8$, where the largest hole is still being neglected.

(c) Interestingly at this ε, only the medium and big holes are being captured. ε is now approximately equal to the radius of the small hole, and therefore data points on either side of the small hole are spanning its diameter. There are also four holes arising from topological noise, one of which is barely visible, giving $\beta_1 = 6$. The fact that one hole is barely visible and still counts towards β_1 highlights the topological philosophy, as the geometric notion of size has been stripped but the size-insignificant hole is still counted. A further intriguing observation is the different positions of topological noise in (b) and (c). This observation results in topological noise being less troubling than initially perceived.

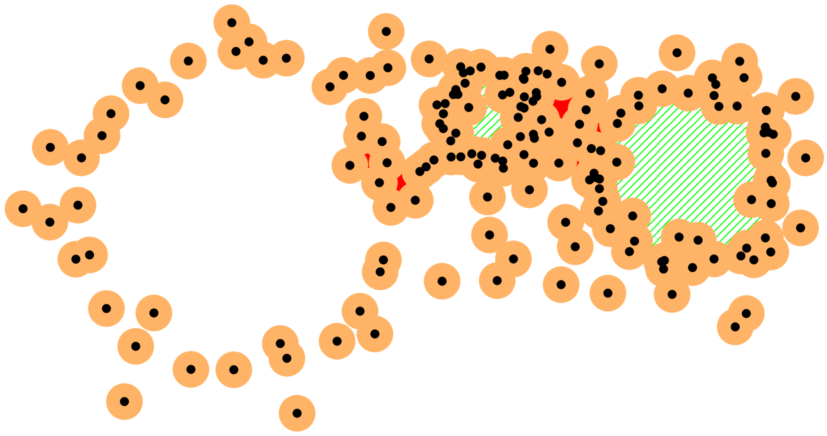
Each hole stands out as a prominent manifold feature, yet no single ε can adequately capture all three since the small hole is spanned before the large hole's perimeter is closed. If one tries to emphasise one hole at the expense of the other, determining which is the most prominent feature is debatable: while the leftmost hole is certainly the largest, it is also the most sparse. Thus, selecting an ε that highlights this feature may be considered a naive solution since this would overshadow information regarding the most densely-populated hole. Therefore, there may not always be a single ε that adequately describes the global topology of a point cloud. Moreover, the emergence of artefacts from topological noise only complicates matters. While only three circles were generated for illustration, the indiscriminate nature when forming the VR complexes[5], holes are unexpectedly forming as a result of the noisy discrete data. Real-world data will invariably be discrete and contain noise, therefore topological noise is an inevitable reality of topological data analysis. The prudent point to takeaway here is the potential inability of one length scale to accurately depict a data set within a traditional topological framework.

---

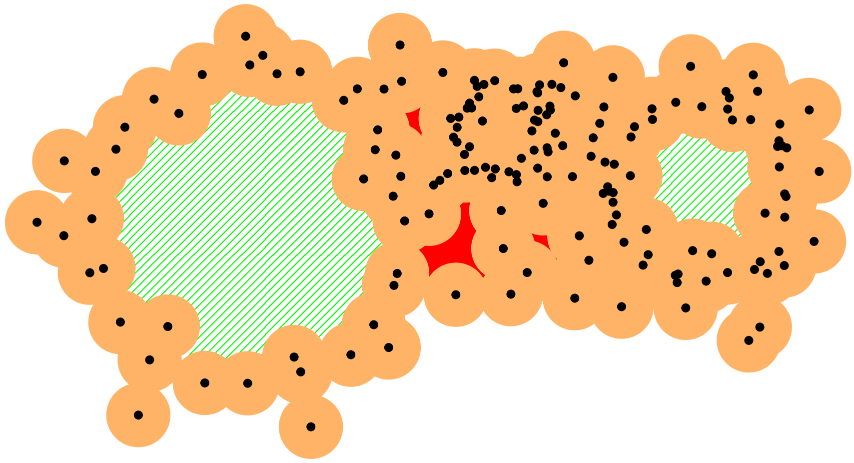
[5] Leopold Vietoris, fittingly linked with the concept of persistence, was also renowned as a supercentenarian. He lived for an impressive 110 years and 309 days, making him the oldest verified Austrian man on record.



(a) Small hole captured



(b) Small and Medium holes captured



(c) Medium and Large holes captured

Figure 3.17: Data sampled from a manifold with features of varying length scales. This discrete sampling shows the inability to capture all of the homological features with a single ε. Correctly identified holes are shown in green and topological noise in red.

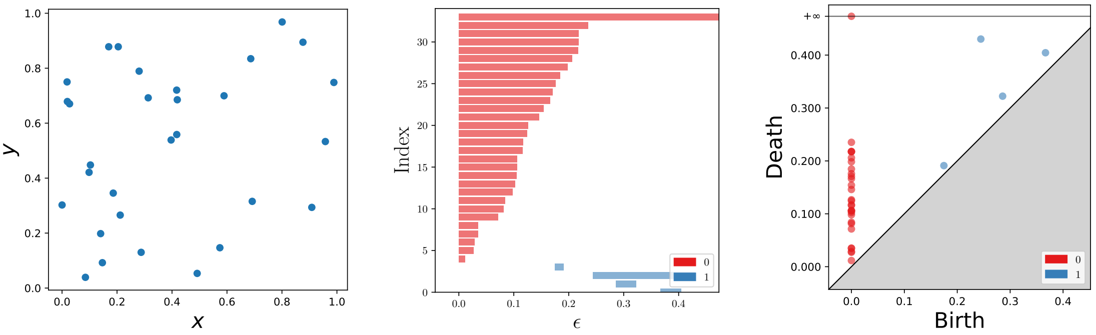
The solution to an appropriate topologically representative length scale is elegantly straightforward: simply consider *all* length scales. This idea is the philosophy underpinning *persistent homology*, where ε is considered over a range of values and determines how the topological features *persist*. This approach results in a filtration, a sequence formed by appending simplices, ordered by their creation with an association to the varying ε parameter. This process results in n disconnected points at the start of the filtration, where $\varepsilon = 0$, and a fully connected $(n-1)$-simplex at the end. The homology at each instance along the filtration is calculated, giving the Betti numbers associated with every ε. The necessity of analysing multiple length scales stems from the limited and potentially misleading inference when considering a single value. Varying ε does not bias any length scale, therefore offering a multi-scale descriptor of the data manifold's shape.

Persistent homology extends traditional homology to uncover the duration for which holes are present within a manifold, implying that these features eventually terminate. This phenomenon was shown in Figure 3.17 (c), wherein the small hole was spanned. The main interest from persistent homology isn't just in the number of holes, but the number of holes that *remain over nontrivial* ε *ranges*. When features persist over larger ranges of ε, the feature is considered more prominent, and it is likely to truly represent the manifold. Each homological feature is said to be *birthed* upon its creation at $\varepsilon_{\text{birth}}$ and *die* when enclosed at $\varepsilon_{\text{death}}$. The range $[\varepsilon_{\text{birth}}, \varepsilon_{\text{death}}]$, is called the *persistent interval* and is representative of a single homological feature. Since each persistent interval is associated with a specific feature in a specific dimension k, they are commonly depicted as pairs $(k, [\varepsilon_{\text{birth}}, \varepsilon_{\text{death}}])$. Hence, the set of all persistent intervals acts as a multi-scale multi-dimension descriptor representing the topology of a data set.

Persistent homologies exhibit stability [63], thus a small change in a point cloud aptly reflects a small variation in the persistent homology. Therefore, two spaces that exhibit similar topological features in their point clouds, will have similar persistent homologies. Thus, persistent homology can be used as a signature for a data set, by characterising the topological properties, much like other topological invariants discussed in earlier sections. Consequently, by defining suitable metrics over the space of persistent homologies [64], it becomes feasible to identify whether two point clouds share topological similarities. One such metric is the Wasserstein distance, which is introduced in Section 3.6.2. Such metrics allow for inference between data sets based entirely on their shape.

When Betti numbers are considered over their persistent intervals, a notion of size is bestowed on the topology. Hence, persistent homology, as an analytical technique, falls somewhere between geometry and topology. The inherent metric nature of persistence intervals introduces an ability to quantify the size of the holes present within a simplicial complex, a technique not possible when using traditional homology. For the case when constructing simplicial complexes from a VR complex, the radius of the hole will be approximately $\varepsilon_{\text{death}}$, where slight inaccuracies may arise from noise within the data, and holes not being perfect analogues of circles. Then the length of the persistent interval, $\varepsilon_{\text{death}} - \varepsilon_{\text{birth}}$, is akin to a weighting, or probability that this feature truly represents the manifold. Armed with this newfound knowledge, topological noise may be addressed, since it is not truly representative of the manifold structure. Therefore, topological noise typically appears over brief intervals. This phenomenon was demonstrated between Figures 3.17 (b) and (c), as there were no common regions of topological noise. Whereas the medium hole, which is a true feature, persisted over both ε values.

3.6.1 Visualising Persistent Homology



(a) (b) (c)

Figure 3.18: (a) 20 randomly-generated 2D points. The data's (b) persistence barcode and (c) persistent diagram.

There are two common ways to present the persistent intervals: *persistent barcodes* and *persistent diagrams*. Each format comes with merits and limitations. The persistent barcodes primarily provide qualitative insights, whereas the persistent diagrams are more geared towards quantitative analysis. Figure 3.18 (a) shows some point-cloud data, (b) showcases the persistent barcode, and (c) illustrates the persistent diagram. When plotting persistent intervals, there are some shared characteristics between the two plots,

1. Both are plotted in $\mathbb{R}^2$.
2. The colouring schemes in these diagrams convey the dimension of the persistent interval [65]. As a reference, this thesis consistently uses red, blue, and green to denote β_0, β_1, and β_2, respectively. This rationale influenced the colour scheme in previous simplicial complex plots.
3. In both diagrams, a single feature persists to infinity. This interval arises because at some distance parameter ε_{fc}, a point cloud will become a simplex rather than a simplicial complex. Then for $\varepsilon > \varepsilon_{\text{fc}}$, the simplex will only remain fully connected, and therefore this interval will continue to infinity. When performing calculations over persistent homologies, it is often required to remove this infinite interval, resulting in the *reduced persistent homology.*

Persistent Barcodes

Persistent barcodes offer a more intuitive visualisation of persistent intervals. In these persistent barcodes, the x-axis is represented by ε, while the persistent intervals are stacked in the y-axis, without significant structure. As a homological feature emerges, its barcode interval begins at $\varepsilon_{\text{birth}}$ and extends until the feature is spanned or merged into a like feature at $\varepsilon_{\text{death}}$. This process is repeated over the entire set of persistent intervals.

By arranging the persistent intervals in this manner, one can intuitively determine the Betti numbers at any instance, ε, along the filtration by drawing a line at $x = \varepsilon$. The number of intersections with red intervals is β_0 and the number of intersections with blue intervals is β_1, therefore mentally drawing vertical lines provides an intuition to the number of holes present within a simplicial complex at any given ε.

An example of a barcode depicting this process is shown in Figure 3.19, with vertical dashed lines showing the intersections with the intervals and the corresponding simplicial complex shown at its end. For instance, for $\varepsilon = 0.3$, the simplicial complex comprises one connected component, indicated by the single red intersection, and there are two holes, inferred by the two blue intersections. However, at $\varepsilon = 0.45$, all holes have been spanned and any simplicial complexes formed after this point will only exhibit increased connectivity, as shown in $\varepsilon = 0.50, 1.00$. Hence, there is no topological insight concerning holes after this point.

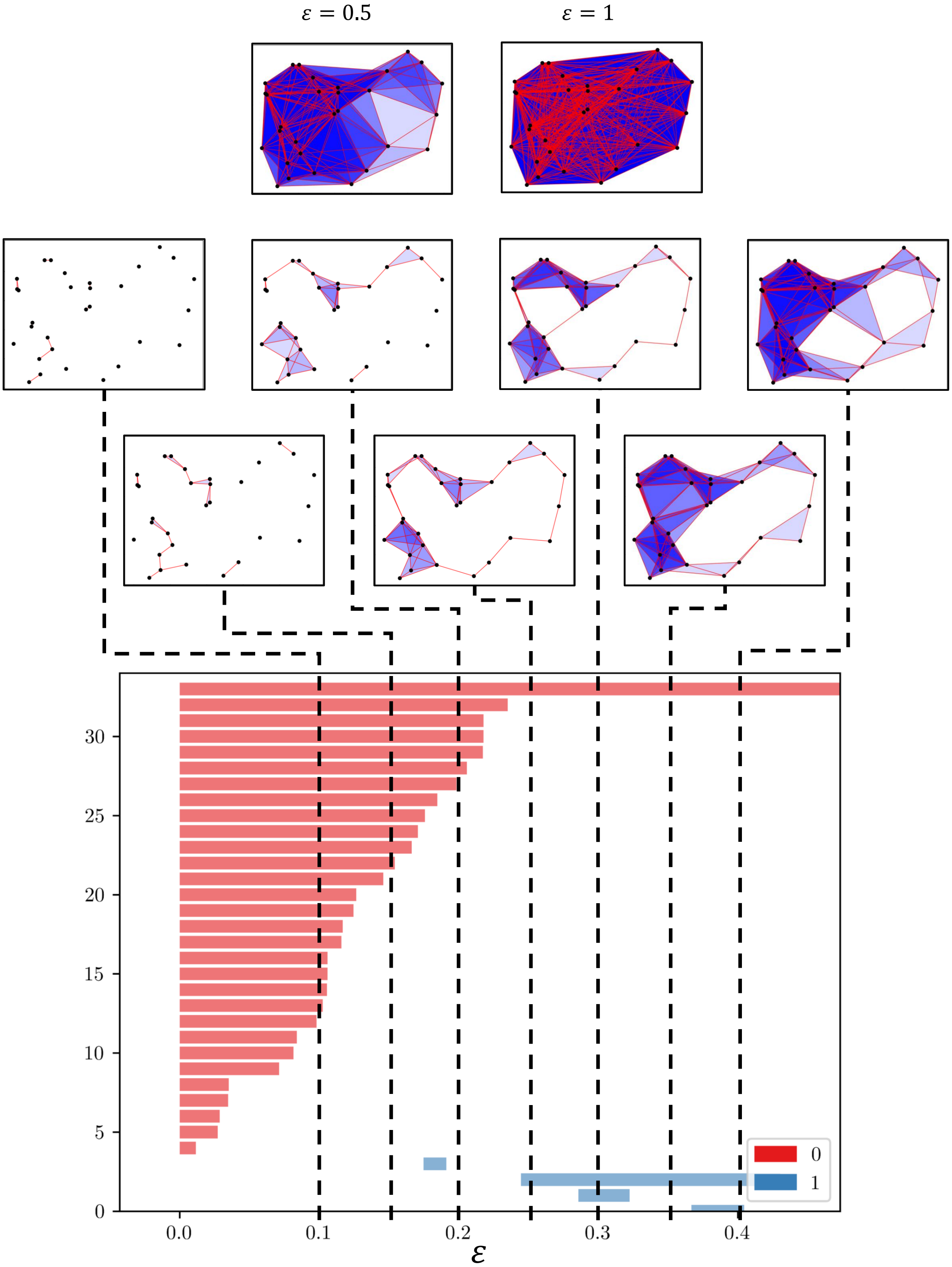


Figure 3.19: Simplcial-complex realisations formed over different ε, showing their topological features on the persistent barcode.

The barcode representation also better shows repeated intervals, as each persistent interval is a separate entity on the y-axis. In Figure 3.19 there are three near-identical persistent intervals over the range $[0, 0.22]$, and are easily distinguishable. Conversely, on the persistent diagram, these intervals will be displayed at the same point overlapping one another, potentially masking information in such situations. Although persistent barcodes excel at aiding understanding of the space, they lack the structure to be used quantitatively. Consequently, for practical-quantitative applications persistent diagrams are almost always used.

Persistent Diagrams

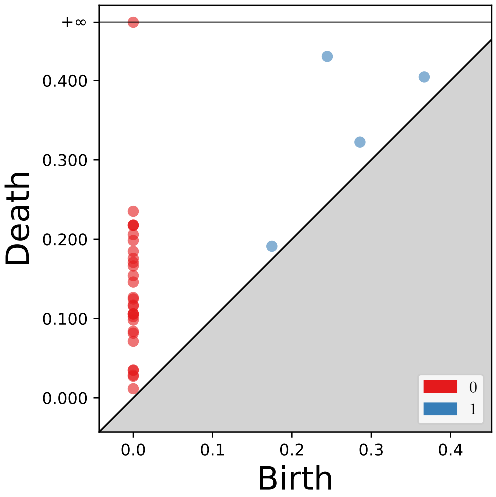


Figure 3.20: Enlarged persistent diagram of Figure 3.18 (c).

On the other hand, persistent diagrams, or birth-death diagrams, provide a more structured representation of persistent intervals. To plot persistent diagrams, the persistent intervals are shown as points in the xy-plane, where $\varepsilon_{\text{birth}}$ is plotted on the x-axis and $\varepsilon_{\text{death}}$ is plotted on the y-axis. An example of a persistent diagram can be seen in Figure 3.18 (c), with an enlarged version in Figure 3.20. Each persistent interval is represented by the coordinates $(\varepsilon_{\text{min}}, \varepsilon_{\text{min}})$, with the colour denoting the corresponding dimension. Persistent diagrams show a boundary line, defined by $y = x$, below which points will never be plotted. This line specifies the principle that a feature must first exist before it can die. Subsequently, the difference between the line $y = x$ and the y-coordinate of a point is analogous to the length of the persistent interval. The further a point is from the line $y = x$, the longer the feature persists, implying that features close to $y = x$ are likely to originate from topological noise.

At the expense of intuition, persistent diagrams exhibit an easily analysed structure. From persistent diagrams, understanding individual simplicial-complex states along the filtration is much more challenging. To obtain the Betti numbers for a simplicial complex, one would have to visualise a line moving perpendicularly to $y = x$, then the y intersect of this moving line denotes ε, and the count of points above this line would be the Betti numbers. Needless to say, this approach is much less intuitive than the barcode representation. Nonetheless, as previously described, this form exhibits a superior structure, making it the preferred choice for quantitative analysis. The more natural geometric structure of persistent diagrams naturally facilitates the matching of features between persistent homologies.

Other representations of persistent homology are used in this thesis, specifically vector-stable forms, which allow for persistent homologies to be passed as features into a machine-learning pipeline. While these representations are not covered in this section, they are introduced in Chapter 5, where their application to the Z24 bridge forms a standalone chapter.

3.6.2 Wasserstein Distance

This chapter has emphasised that the persistent homology space allows for the definition of metrics [66], with the implication that proximity in the persistent homology space arises from the topological similarity between data sets. Using persistent homologies as topological invariants in this way is less clear-cut than when compared to purely abstract mathematical topological invariants. Therefore, to define the similarity between point clouds' shapes, a more complicated measure is required, in the form of the *Wasserstein distance* [67].

Rather than acting over the multi-set persistent interval form of persistent homologies, the Wasserstein distance acts over the structured space of persistent diagrams. Thereby, equipping the space of persistent homologies with a metric. Numerous works [7, 63, 68, 69] have proven the stability of metrics over the space of persistent homologies, indicating that for small perturbations between point clouds, there are proportionally small changes in the persistent homologies. Using this insight, the Wasserstein distance can quantify the topological similarity between point clouds.

The Wasserstein distance aims to find the set of optimal matchings[6] between the

---

[6] When determining the optimal matching between the points, the concept of optimal transport [70, 71, 72] is heavily relied upon.

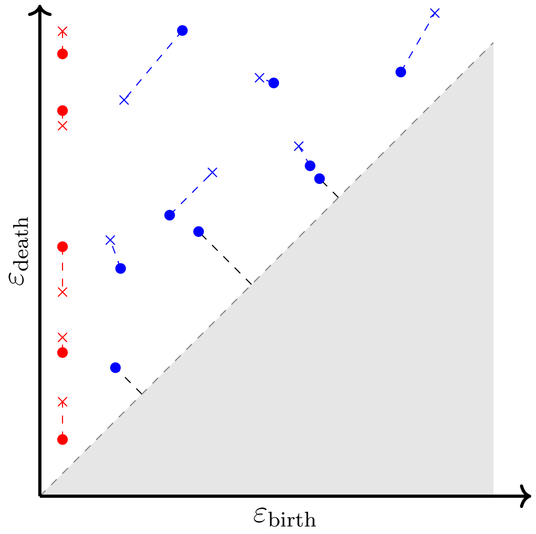


Figure 3.21: A matching between two persistent homologies, with colour denoting homology dimension, and circles and crosses denoting different persistent homology sets.

points of two overlayed persistent diagrams, which minimises the cost associated with the calculation. An example of some optimal matchings between two persistent diagrams is shown in Figure 3.21. Consequently, the Wasserstein distance is defined as the weighted sum of all the matching lengths.

Definition 3.6.1. Given two persistent diagrams D_1 and D_2. For a positive real number p representing the weighting given to the matching between persistent intervals, the *p-Wasserstein distance*, W_p, is given by,

$$W_p(D_1,\, D_2) = \inf_{\phi} \left(\sum_{u \in D_1} d(u,\, \phi(u))^p \right)^{\frac{1}{p}} \tag{3.35}$$

Here, u is a persistent interval from D_1, while $\phi(u)$ is its optimally-matched counterpart, such that $\phi(u) \in D_2$. The function d is the Euclidean distance. When $p = 1$, this is the sum of all the matching distances [38].

For an analogy of the Wasserstein distance, consider two piles of soil at different locations. The Wasserstein distance then provides a way to measure how much effort is required to move one pile to the other. This analogy gives rise to the name Earth mover's distance, which is the 1-Wasserstein distance. Here, the piles of soil are equivalent to the optimally-matched points, and the effort mirrors the matching length. Essentially, what the Wasserstein distance quantifies is the required 'effort' to morph one persistence diagram into another.

Before calculating the Wasserstein distance between persistent homologies, a small but important modification is required. The persistent homologies are densely populated with points along diagonal $y = x$, with infinite multiplicity [73]. This adjustment arises from the need to ensure that every point finds a match, as there is an infinite cost associated with unmatched pairs. By including the diagonal line, it guarantees that every point is matched. This is especially important when matching persistent diagrams with varying persistent interval counts. Consequently, the inclusion of the diagonal also ensures that topological noise, which is typically concentrated around $y = x$, is matched at a small cost. Therefore, the influence of topological noise is minimal when comparing persistent homologies.

3.6.3 Flaws of Persistent Homology

Persistent homology offers a rich and unique analysis, distinguishing itself far from other techniques within data science by utilising an abstract notion of shape. This process is motivated by some very elegant and abstract mathematics. Nevertheless, no matter the level of abstraction and innovation, topological data analysis is a relatively-young field and is still exhibiting some teething problems, some unavoidable and others awaiting solutions. This section will discuss some of the more pressing, noteworthy, and subtle pitfalls of persistent homology.

Arguably, the most restrictive challenge that persistent homology faces is the computational expense required for calculations. Whilst the theoretical framework is fully generalisable to many points and dimensions, the reality is that practical applications often fall short. Scalability is a monumental challenge for persistent homology, resulting from the need to check $k - 1$ distances over k points, and then repeat this over ranges of ε when forming filtrations from data. Given the current state of the algorithms and computational power, the practical limits of persistent homology often restrict calculations to a subset of the homology groups, usually H_0 and H_1, only considering a few thousand data points embedded in single-digit dimensions. Several packages exist for calculating persistent homology, across various programming languages, including Ripser [55], GUDHI [54], PHAT [74], DIPHA, and Dionysus. The current general consensus [55, 75] suggests the C++-based Ripser is the most efficient, often being orders of magnitude faster than other packages. However, while Ripser is very efficient, it only focuses on VR complexes; it only does one thing, but it does it *very* well. Whereas, other packages offer more versatility and a wider

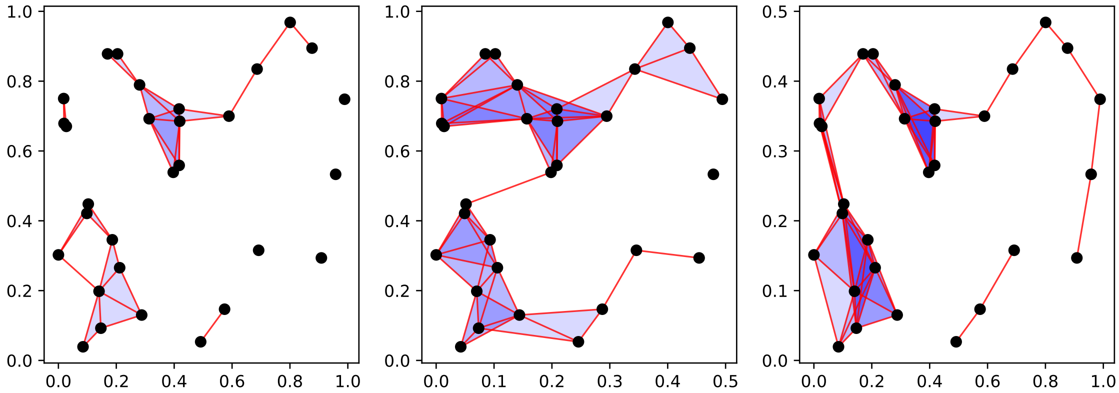


Figure 3.22: Data scaled in different axes showing different simplicial complexes at the same ε. Note the difference in scale along the axes.

range of analytical features. Despite this, new faster algorithms will inevitably be developed in the future allowing for calculations over larger and higher-dimensional point clouds. Notably, some researchers are already developing algorithms for use on quantum computers [76].

Another unavoidable consequence of this analysis is that constructing simplicial complexes from data is a metric process, which induces a scalar functional dependency on the persistent homology. When considering a VR complex, an open ball is centred at each vertex, the open balls are defined by the metric distance ε. Given some data where all the axes are on different scales, there will be a bias for forming simplices along the axes with smaller scales. This phenomenon is shown in Figure 3.22, where the second and third plots show a preference for forming simplices in the x and y axes, respectively. Fortunately, this isn't too problematic, as it proves good practice to normalise the data in each dimension prior to calculating the persistent homology. This normalisation procedure is performed throughout the remainder of this thesis. It is essential to highlight this subtle point, as topology is a scale-invariant study, but topological data analysis is fundamentally limited by the need for a metric when defining complexes on data.

While persistent homology is independent of length scales, this might falsely give the impression of parameter independence. However, the persistent homology calculations are influenced by various hyperparameters. Some hyperparameters include the filtration type, homology coefficient field, minimum persistent interval length, maximal embedding dimension, and maximal value of ε (set to infinity in all of this work). The choice used for these hyperparameters affects the results, and there is no

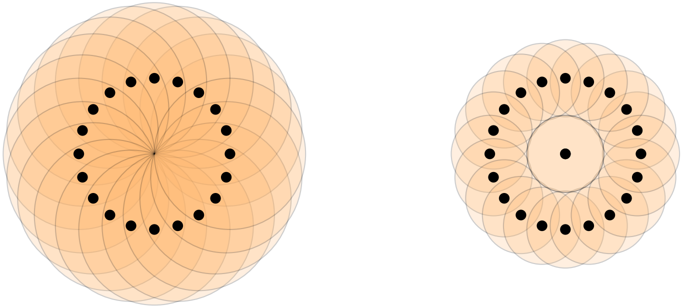


Figure 3.23: Noisy data spanning the hole earlier than expected.

universally-accepted protocol for choosing their values. While most of these hyperparameters are finding a balance between information and computational expense, others can give entirely different results.

How persistent is persistent enough? There is no universal threshold or formula for determining when homological features represented by persistent intervals are true to the topology of the manifold, or are artefacts of topological noise. It might be the case that there is no firm boundary between the two distinctions, but rather a probabilistic-fuzzy boundary. This ambiguity is only an issue when wanting to understand a point cloud's absolute topology. In contrast, the bulk of this research and applications within the literature concern the relative comparison of persistent homologies, primarily via the Wasserstein distance.

While calculating persistent homology is resilient to small noise levels, greater noise levels can introduce scenarios which significantly alter the persistent diagrams. As an example, consider two circles with the same radii, with one of the circles containing a data point at its centre arising from noise, as illustrated in Figure 3.23. The inclusion of this central 'noise' point reduces the ε required to span the hole. Consequently, the persistent interval for this hole is significantly shortened, thereby implying that this feature isn't as prominent. Admittedly, this is an exaggerated example, but the principle applies to toned-down cases. Such an occurrence was observed in Figure 3.17, where the small hole was prematurely spanned, which can be primarily attributed to a single noisy data point inside the hole. While it is clear in 2D, with the aid of plots, recognising this case in higher dimensions is much more challenging. Data preprocessing can help mitigate such issues; however, this introduces a trade-off between preserving genuine topological features and smoothing the data.

Integrating persistent homologies into broader analytical frameworks is tough, because of the lack of a canonical form. Two representations of persistent homology have already been shown in this chapter, and more vector-stable representations are discussed at a later stage. The fundamental issue lies in the unstructured persistent intervals, which cannot be readily passed as a feature into other data analytical toolboxes, such as machine learning. The current solution is to manipulate the persistent intervals in some way to ensure they are in the form required for the succeeding analysis. In doing so, one increases complexity, data processing time and cost. Additionally, multiple vectorised representations of persistent homology exist, all of which emphasise different features and give incoherent results with each other.

Finally, to end on a slightly more philosophical note, there is an ambiguity concerning what the homological features of data represent. Establishing a correlation between the holes present in some data to real-world phenomena remains an open challenge. This ambiguity intensifies when considering already complex data sets, such as those found in SHM. Does a void form from a temperature change, the inclusion of damage, traffic going over a bridge, a combination of these, or a variable that is not being monitored? Additionally, there is also an ambiguity regarding feature importance. It is not necessarily the case that the most dominant homological feature relates to the most influential in the real world. It is conceivable that the most influential variables in SHM do not permit a topologically-interesting feature. An argument of this nature was supplied when analysing Figure 3.17, where one hole was larger and one was more densely populated, there is no obvious way to attribute an order of importance between these features.

Despite all these flaws and its infancy, persistent homology has seen applications spanning a wide range of fields and has opened the doors to analytical techniques to consider aspects of data that were previously overlooked. In fields such as SHM, when working with expensive, unique, often safety-critical problems, there is no good excuse for overlooking potentially informative aspects of data.

3.7 Time-Delay Embedding

In the analytical chapters of this thesis, there is a requirement to represent and calculate the topology of 1D time series. There is not much that can be described as 'holey' about a 1D time series, other than perhaps discontinuities along the series. A solution to describing the topology of 1D time series is proposed via *time-delay embeddings*, where higher-dimensional manifolds are constructed from time series. By this approach, trends in the time series present themselves very naturally as homological features, making them capturable with TDA. This method thereby facilitates topological inference over 1D time series.

Often, data are collected as a 1D time series, rather than as a set of variables, which can represent points of a manifold. Additionally, in some cases, a single time series might be output from an analytical process, such as the most stationary residual series in cointegration. Time-delay embeddings broaden the scope of applying persistent homology to 1D data sets, without the need for having k data series embedded in $\mathbb{R}^k$.

Time-delay embeddings trace their origins to work by Packard in [77] as a method of inducing geometry from a one-dimensional time series. Takens later showed that a manifold representing the full state space of an attractor can be derived from one of its time series [78]. The important part of this result – relating to this thesis – is the fact that the topologies of the reconstructed and complete attractors are equivalent when using a time-delay embedding [78]. Essentially, time-delay embeddings enable a 1D time series to assume an nD topology that is identical to the topology of the attractor's full-scale state space. Although Taken's embedding theorem was proven strictly for topological reconstructions of attractors, the adaptability of time-delay embeddings has been extended beyond this scope. Time-delay embeddings act as a consistent and reliable technique to represent the topology of 1D time series and consequently have seen applications in topological-time series analysis [79, 80, 81].

To determine the time-delay embedding for a given time-varying series $f : t \to \mathbb{R}$, the series are stacked d times to form a d-dimensional point cloud. Every successive instance is shifted by integer multiples of the delay α. Formally, the embedding $\Phi(f,\, \alpha,\, d) : \mathbb{R} \to \mathbb{R}^d$ is represented by,

$$\Phi(f,\, \alpha,\, d) = [f(t),\, f(t+\alpha),\, \ldots,\, f(t+(d-1)\alpha)]\,. \tag{3.36}$$

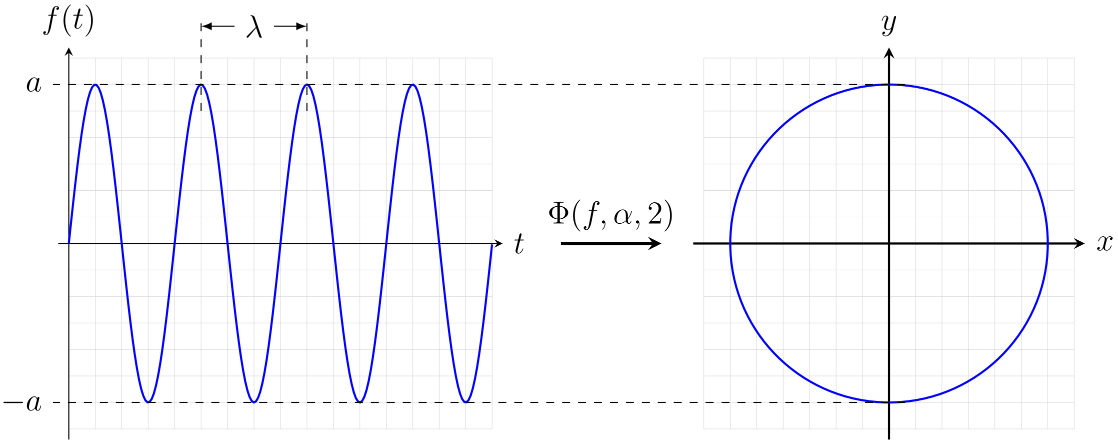


Figure 3.24: A 2D time-delay embedding of a cyclic trend.

The interest in using time-delay embeddings in TDA is that they can represent time series' features as homological features. For example, a cyclic trend in the time series manifests as a loop in the embedding space. When multiple independent cyclic trends are present within a time series, which is very often the case in SHM, these will begin to trace out a complex higher-dimensional manifold. Then the shapes of these manifolds are quantified with TDA and used to form decisions over 1D time series, based on their shape.

Time-delay embeddings are interesting as they encapsulate the local history of a time series in higher dimensions. By locally shifting the time series by α, and then representing this as a point in d-dimensional space, both the current state and the history at integer multiples of α are captured. Hence, given appropriate values for the embedding delay and dimension, the reconstructed manifold approximates the full state of a dynamical system. It is vital to note that the shape of the reconstructed manifold is heavily influenced by the choice of α and d. Consequently, later in this thesis, a consistent method for determining optimal time-delay embedding parameters considering persistent homology is presented.

Chapter 4

DETECTING NOVELTY WITH TDA

When considering Rytter's hierarchy [2], the foundational task is the detection of novelty. Novelty detection is undoubtedly the easiest task to consider in Rytter's hierarchy, which is further simplified by knowing the exact point when a structure becomes damaged, i.e. having labelled damage data. However, since TDA has never been applied in an SHM context, novelty detection with labelled data is a promising starting point. The perfect candidate for this case study is the heavily-researched Z24 bridge [82], as the damage was artificially induced, therefore the exact instance is known when damage became present in the structure. Additionally, the data set contains a collection of monitored environmental variables; the most interesting relating to this work is temperature.

Although novelty detection is situated at Level One – the most rudimentary tier of Rytter's hierarchy – it should not be misconstrued as straightforward. In real-world scenarios, including this one, detecting damage within structures is not easy, because of the external influence of Environmental and Operational Variations (EOVs) on damage-sensitive features, such as natural frequencies. Although EOVs are expected, and typically harmless to the structure, they often have a more pronounced effect on damage-sensitive features than damage, presenting significant challenges when analysing SHM data. This happening is exactly the situation with the Z24 bridge, where temperature fluctuations alter the natural frequencies by up to 30%, in contrast to a maximum change of 7% caused by damage.

As mentioned in the previous section, the analysis here will be conducted solely within the confines of persistent homology, where the Wasserstein distance is used as a metric to differentiate between data. Hereby, using only the data's shape to form decisions regarding the health state of the bridge. The general approach involves partitioning the data based on temperature – the most dominant EOV – and then calculating the persistent homologies and Wasserstein distances over these embedded manifolds. This analysis seeks to exploit the idea that detectable topological alterations occur in the data when damage is present, setting them apart from temperature-induced effects. The aim is for TDA to identify these topological changes in the damage manifold, and distinguish them from the norm, even with changes brought about by the more impactful temperature variations.

In this analysis, the Z24 data are presented in their most natural form, a manifold embedded in $\mathbb{R}^4$, based on the first four natural frequencies. Some supplementary cases are also provided, showcasing the stability of TDA against sampling mechanisms, reconstruction procedures, and dimension reduction. These additional cases underscore the preservation of topologically interesting features over data preprocessing techniques. To emphasise the robustness of TDA, the three case studies are:

1. Manifolds embedded in four-dimensional space, where each of the axes corresponds to a natural frequency: ω_1, ω_2, ω_3 and ω_4.

2. Manifolds embedded in three-dimensional space, where the nonlinear ω_2 feature has been omitted. Showing the resilience of shape, even when features with dominating effects are omitted.

3. The third case shows the robustness of the manifolds to a linear dimension-reduction algorithm. In this case, the four-dimensional embedding of the data will be compressed down to two and three dimensions [83]. The reduced-dimension manifolds are then analysed with TDA.

However, before heading into the analysis section, a greater background in understanding the Z24 bridge data is provided, by giving a very brief history of the bridge and listing some quirks of the data.

4.1 Z24 Bridge

The Z24 was a prestressed, three-span bridge with two lanes and a main span of 30m between the two regions, Koppigen and Utzenstorf, in Switzerland. The bridge was built in 1963 and eventually demolished in 1998, as a new railway was being built which required a bridge with a larger span [84]. Knowing that the bridge needed to be demolished, data were collected leading up to the bridge's dismantlement. A sensor network was placed over the bridge to collect modal parameters, as well as sensors for measuring air temperature, soil temperature and humidity local to the bridge [84]. Modal analysis [82, 85] was then used to extract the natural frequencies from the accelerometer data, the first four of which are analysed in this work.

Since the data were captured over a whole year, the extreme Swiss seasonality was captured, with the air temperature recorded as low as $-9\,^{\circ}\mathrm{C}$ and as high as $36\,^{\circ}\mathrm{C}$. These temperature effects caused significant changes to the calculated natural frequencies [86]. Towards the end of the year-long monitoring campaign, but shortly before the bridge's destruction, controlled damage was introduced to the system [84], the effects of which present themselves in the set of natural frequencies.

As previously mentioned, the temperature change is the biggest contributing factor to the natural frequency variations, with a magnitude of around 30%, whilst the damage effects result in only a 7% change. For this reason, the change in the magnitude of the natural frequencies offers little insight into the presence of damage, and more elaborate data analysis techniques are required to distinguish damage and temperature variations. The main problem here is to isolate the less prominent effects of damage, from the sets containing the more prominent temperature effects.

In order to aid the novelty detection, the natural frequencies were partitioned into four categories based on the air temperature at the instance the data were captured. The data set is broken down into four classes based on the air temperatures at the time of the measurements, or whether damage was present (irrespective of temperature), more information regarding these partitions is presented in Table 4.1. Figure 4.1 shows the temperature readings and the first four calculated natural frequencies plotted against the measurement index, where the corresponding colours refer to the data partition class.

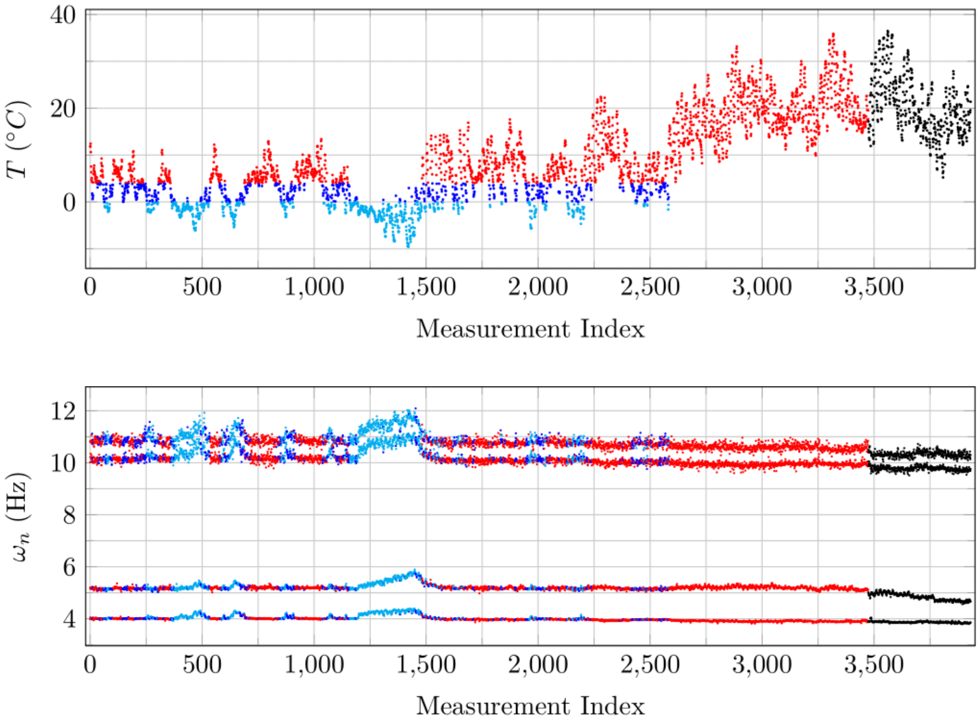


Figure 4.1: Air temperature and the first four natural frequencies, where the colour of the points refer to the class associated with the data point.

Region	Colour	Temperature Range	Data Count
Freezing, F	Light Blue	$T < 0\,°\mathrm{C}$	720
Cold, C	Dark Blue	$0\,°\mathrm{C} \leq T < 4\,°\mathrm{C}$	666
Warm, W	Red	$4\,°\mathrm{C} \leq T$	2089
Damage, D	Black	N/A	457

Table 4.1: Information regarding the four Z24 data partition classes.

Qualitatively analysing Figure 4.1 shows that there is a significant increase in all the natural frequencies over the region of sustained freezing, approximately over the indices 1200 to 1500. Over this region, it is speculated that the water trapped within the bridge deck and soil froze, resulting in the bridge displaying heightened rigidity, thus increasing the natural frequency [86].

Intriguingly, the natural frequencies are affected differently, where ω_1, ω_3, ω_4 show a linear relationship with temperature, but ω_2 displays a nonlinear relationship. This nonlinear relationship will be explored in more detail in Chapter 7, when considering cointegration. Also, it can be seen visually that after the onset of damage, ω_1, ω_3, and ω_4 appear to be relatively stable, whereas ω_2 is decreasing over the same interval.

For the succeeding analysis sections of this chapter, the first case will demonstrate a full walk-through of the calculation procedure and motivate a normalisation scheme. For the following cases, only the results are presented to limit repetition; however, where elements are essential to the new cases, these points will be explicitly stated.

4.2 Embedding in $\mathbb{R}^4$

The first four natural frequencies of the Z24 bridge can be represented as a manifold M_{Z24} embedded in $\mathbb{R}^4$. In this four-dimensional space, each set of four natural frequencies from a measurement instance forms a point, with the natural frequencies defining the coordinates. By plotting all the data points in the partitions of Table 4.1, the partitions become distinct manifolds embedded in $\mathbb{R}^4$.

Both temperature and damage impact the natural frequencies, implying that the manifolds are also influenced by these factors. Consequently, shifts in temperature or the bridge's health state will alter the manifolds' shapes. TDA allows for the quantification of these shape variations, even if minor. It is conjectured that temperature variations will result in a smooth yet nonlinear shape change, especially when $T < 0$. On the other hand, damage will cause an abrupt change, resulting in a distinct shape when compared to nondamaged counterparts. This difference in shape is used to indicate some novelty in the data, different from the smooth effects arising from temperature. The dramatic shape change introduced by damage serves as an indicator of novelty in the data, distinguishing it from smooth effects arising from temperature; thus acting as a topological-based novelty detection method, albeit on labelled data.

Several factors influence the manifold shape, with some being temperature-related and others not. For those factors correlated with temperature, their effects on the manifold's shape will be specific to each manifold. For uncorrelated factors, their effects will be consistent across all manifolds, ensuring that there is no shape biasing between partitions. The Z24 natural frequency manifolds data will include effects from all EOVs, with each imparting a unique effect on the natural frequencies. Given that the data span an entire year, seeing the full seasonal range, various EOV interactions will be observed, collectively contributing to the manifold's shape.

4.2.1 Data Nomalisation

As highlighted in Section 3.6.3, because of the metric characteristics of constructing VR complexes, it is beneficial to normalise along each axis. In this context, the ω_1 axis would inherently receive preferential treatment when forming simplices because it is by definition the smallest natural frequency. Normalising the data by the maximum value in each time series ensures that the simplex formation is fair along each axis. Subsequently, changes in the persistent homologies of the manifolds arise from the relative positioning of points rather than their absolute distribution.

However, there is another normalisation procedure required that is even more influential than the relative axis scales. When partitioning the data into classes, the relative sizes of the partitions were not considered, resulting in W having many more points than the others. Yet, the partition size plays a significant role in this context. For well-sampled manifolds, their persistent homologies will show little variation based on the number of points. However, the persistent homologies will be more densely populated with persistent intervals. Directly as a result of the persistent-interval count, there are more points to match within the Wasserstein distance calculations, contributing to increases in their value. Consequently, the Wasserstein distances associated with the warm partition are likely to be larger than those from other manifolds with fewer points. Addressing this issue is somewhat tricky, as more data points also provide a better depiction of a manifold's shape.

To investigate the relationship between the Wasserstein distance and the number of points in a manifold, W is divided into two randomly-allocated-equal subsets, W_1 and W_2. Both subsets should exhibit a very similar topological structure since they are sampled from the same manifold. Whilst there might be minor variations from topological noise from their distinct samplings, these discrepancies are considered inconsequential regarding their true topology. These new subsets are then included in the analysis. By incorporating these subsets, into this analysis, the aim is to highlight the proportionality between the number of points and the Wasserstein distance.

The persistent homologies are computed for all the manifolds, focussing solely on the first two Betti numbers, β_0, β_1[1]. Subsequently, the Wasserstein distances are

---

[1] This approach was taken entirely for computational reasons, as calculations of β_2 for W would crash because of its large number of points.

	F	C	W	W_1	W_2	D	S_M
F	0.00	1.08	2.91	1.47	1.45	1.21	8.10
C	1.08	0.00	2.89	1.11	1.08	0.64	6.79
W	2.91	2.88	0.00	1.92	1.98	3.17	12.86
W_1	1.47	1.11	1.92	0.00	0.21	1.34	6.05
W_2	1.45	1.08	1.98	0.21	0.00	1.28	6.00
D	1.21	0.64	3.17	1.34	1.28	0.00	7.63

Table 4.2: Wasserstein distances over each partition, the row sums in the final column.

calculated over all the persistent homologies, giving a result akin to an outer product. The Wasserstein distances are presented in Table 4.2. However, these values are relatively uninformative in their raw form, because of their dependence on the number of points within the manifolds. Nonetheless, Table 4.2 effectively showcases the relationship between Wasserstein distance and the count of data points. This is evident as the Wasserstein distances associated with W_1 and W_2 are consistently about half (with minor fluctuations owing to topological noise induced from sampling) of those associated with W.

To better quantify the topological uniqueness of a manifold, the Wasserstein distances are summed along the rows, which is denoted S_M for a manifold M. This sum is shown in the final column of Table 4.2. A larger value indicates that the partition is less topologically similar to the others. However, this calculation still has a dependency on the number of points, exemplified by the fact the sum associated with W is still approximately double W_1 and W_2. To counteract this dependency, one can normalise by the number of points in the manifold,

$$\hat{S}_M = \frac{S_M}{n_M} = \frac{1}{n_M} \sum_{i \in Q} W_p(M, i) \tag{4.1}$$

Where $\hat{S}_M$ is the normalised Wasserstein-distance sum of the manifold M, n_M is the number of points in M, and Q is the set of all manifolds. The results of the normalised Wasserstein-distance sums are present in Table 4.3, where it is evident that when normalising by the number of points, the Wasserstein distance sums for W, W_1, W_2 converge. This is the desired behaviour, by suggesting that these partitions, which were sampled from the same manifold, exhibit minimally-different levels of uniqueness when compared to others, despite their varying sample sizes. Consequently, this normalisation process produces a measure which describes the

M	S_M	n	$\hat{S}_M$
F	8.10	720	0.0113
C	6.79	666	0.0102
W	12.86	2089	0.0062
W_1	6.05	1044	0.0058
W_2	6.00	1045	0.0057
D	7.63	457	**0.0168**

Table 4.3: Summed and scaled Wasserstein distances for each manifold partition.

distinctiveness of a manifold's shape, without being biased by the number of points.

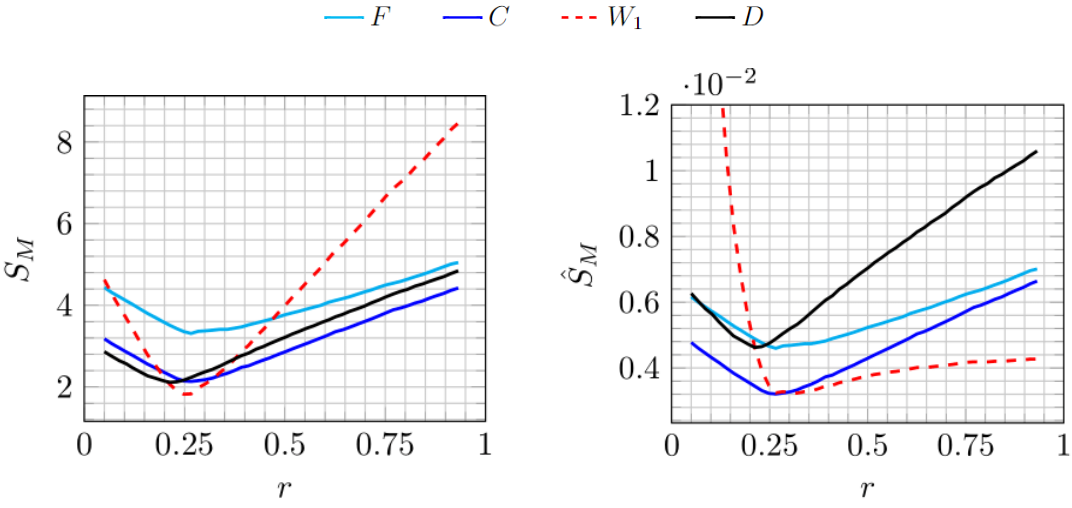
It is crucial to acknowledge that this approach is only valid when there are enough points to adequately represent the manifold's true topology. If there is an insufficient number of points, significant artefacts will be introduced into the persistent homologies because of the undersampling, resulting in unexpectedly-large Wasserstein distances. Furthermore, the already large Wasserstein distance values are then divided by a small number, which will result in asymptotic limiting behaviour because of the reciprocal relation with n_M in equation (4.1).

Whilst this analysis provides insights, basing conclusions on a single equal split is not comprehensive enough. A more holistic approach involves examining this analysis across a range of split ratios, denoted as r. Given $0 < r < 1$, the two disjoint subsets are formed $W_1 = rW$ and $W_2 = (1 - r)W$, such that $W = W_1 \cup W_2$. To validate this normalisation technique, r is varied and the Wasserstein distance sums along the rows of the outer product are calculated. Essentially extending the previous analysis over multiple r values.

To demonstrate the effects of varying r, two scenarios are examined. The first scenario considers F, C, W_1, and D, highlighting the threshold at which a manifold contains a sufficient number of points. The subsequent scenario includes F, C, W, W_1, and D, showing that W_1 converges to W after normalising. Symmetric equivalents of these cases, where W_2 is also included, are provided as supplementary material in Appendix D.

F, C, W_1, and D

In this case, only the subset W_1 is considered, ensuring only one variable with respect to r and no biasing from the inclusion of multiple warm manifolds. This approach



(a) Raw (b) Scaled

Figure 4.2: Size of the Wasserstein distances depending on the warm partition size.

reveals the number of points necessary to adequately represent the W manifold. Figure 4.2 (a) shows an inflection point at $r = 0.25$ for W_1. In the interval $0 < r \leq 0.25$, the manifold is so under-sampled that its true topological features cannot be identified, and the topological noise is dominating. This undersampling results in artefacts within the persistent homology and a likeness between the manifolds cannot be established, resulting in large Wasserstein distances. On the other side of the inflection point, in the range $0.25 < r < 1$, the true topology is adequately represented. As r increases, more points are included in W_1, resulting in a linear increase in the persistent intervals, and finally a linear increase in the Wasserstein distance sum.

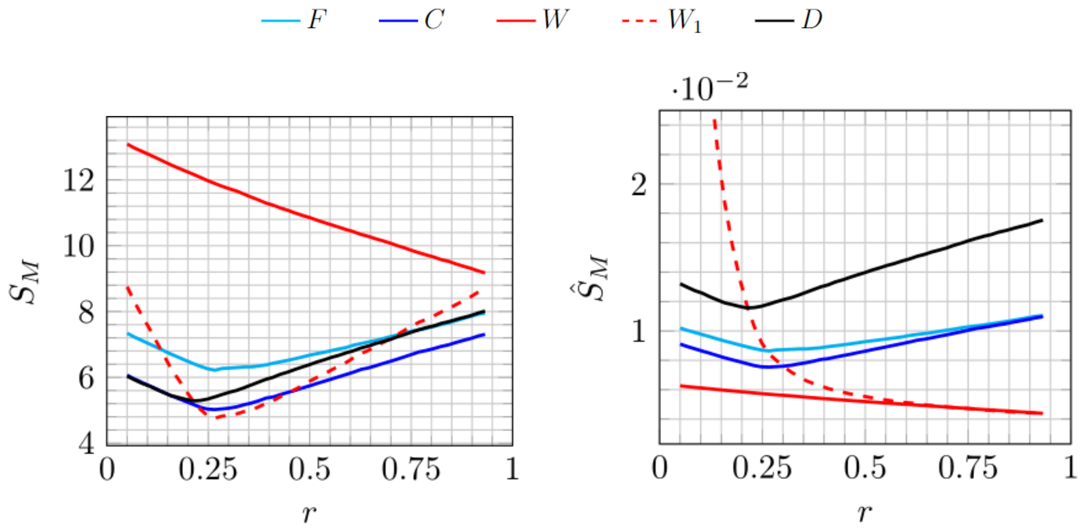
Figure 4.2 (b) shows the Wasserstein distance sums post-normalisation. As anticipated, within the under-sampled interval $0 < r \leq 0.25$, the result for $\hat{S}_{W_1}$ shows asymptotic limiting behaviour approaching small r. This region, however, is not insightful since the topological features primarily arise from noise. The region beyond the inflection point is more informative, where the manifold's true topology is adequately depicted. In this range, $\hat{S}_{W_1}$ remains more stationary over r, especially when compared to its unnnormalised counterpart.

A notable observation in Figure 4.2 (b) is the linear increase for other $\hat{S}$ values. This trend emerges because they are not being normalised by the changing number of points, thus still exhibiting a dependency. This artefact doesn't exist when considering the complement subset W_2 in Appendix D, as $n_{W_1} + n_{W_2} = 2089$ for all r.

F, C, W, W_1, and D

Having established that normalising by the number of points gives a descriptor of topological uniqueness that is invariant over r, it is important to demonstrate that this normalisation procedure yields consistent results given different samplings from the same manifold, given an adequate number of samples. To demonstrate this desired quality, both W and W_1 are now considered.

This methodology is the same as the previous section but with the inclusion of W. The results for the unnormalised Wasserstein distance sums are presented in Figure 4.3 (a). The observed linear decrease in W as r increases is expected, since the only variable in the system is W_1, and as r increases, W_1 is becoming increasingly similar to W, hence their Wasserstein distance decreases.



(a) Raw (b) Scaled

Figure 4.3: Size of the Wasserstein distances depending on the warm partition size.

Figure 4.3 (b) shows the normalised equivalents of Figure 4.3 (a). The important takeaway from this analysis is the evident convergence between $\hat{S}_W$ and $\hat{S}_{W_1}$ after being normalised. This convergence underscores the validity of the normalisation procedure in providing a more consistent, point-count invariant descriptor of manifold topological uniqueness.

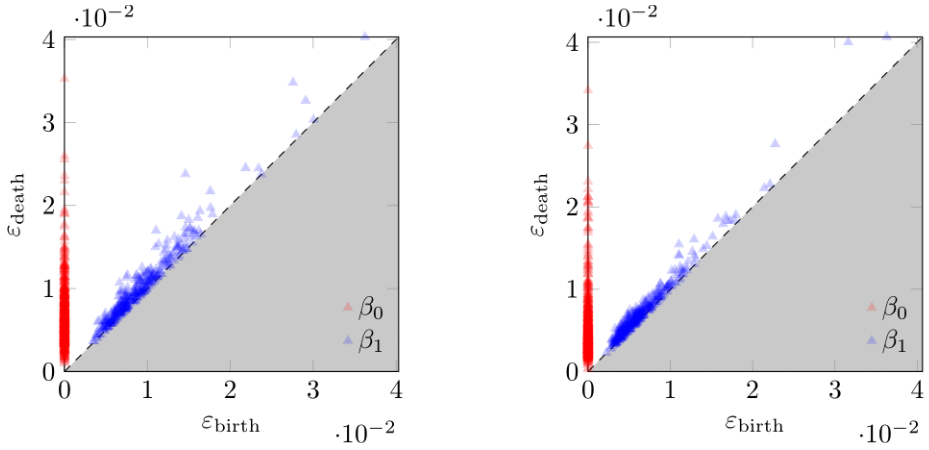
The rationale behind this normalisation procedure is that the number of persistent intervals is proportional to the number of points within a manifold. Furthermore, the number of persistent intervals is proportional to S_M. Thus establishing a transitive proportionality, $S_M \propto n_M$. Dividing by n_M then gives the constant of proportionality, $\hat{S}_M$, which in this case is dependent on the relative persistent homologies of the manifolds.

4.2.2 Results

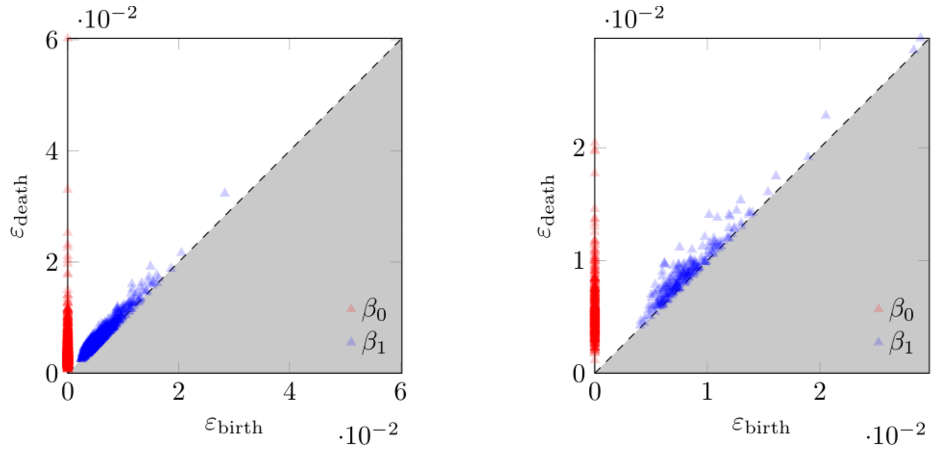
Having established a normalisation procedure that accounts for the varying number of points between manifolds, a more refined descriptor of topological dissimilarity is available, thus allowing for quantitative novelty detection. It is also logical to exclude W_1 and W_2 from further analysis, as after normalising they yield the same results as W. Additionally, the inclusion of W_1 and W_2 also biases the warm partition by double-counting the warm data points.

For a qualitative understanding of the $\mathbb{R}^4$ manifolds, the persistent diagrams for each partition are depicted in Figure 4.4. These diagrams offer a visual representation of the aforementioned smooth transition between the temperature-based partitions. There is a gradual evolution: as temperature increases, the persistent diagrams become more compact, indicating the features persist less. Specifically, the freezing persistent diagram shows several β_1 features persisting in the range $0.015 < \varepsilon_{\text{birth}} < 0.04$. This effect is less noticeable in the cold partition and is arguably nonexistent in the warm.

However, the crux of this analysis lies in distinguishing the damage manifold from the others. The fairest comparison for the damage persistent homology is with the warm manifold, given that the damage data were all collected in warm conditions, as shown in Figure 4.1. When comparing these two, the damage β_1 features are much more dispersed, therefore suggesting a topological change resulting from the inclusion of damage. Another distinguishing characteristic of the damage persistent homology is that all the features are on a smaller scale. The β_1 features predominately appear in the interval $0.004 < \varepsilon_{\text{birth}} < 0.015$, whereas the same features in the other manifolds span broader ranges. Therefore, when matching within the Wasserstein distance, longer matchings are required, resulting in a greater value.



(a) Freezing (b) Cold



(c) Warm (d) Damage

Figure 4.4: Persistent diagrams for data partitions of manifolds embedded in $\mathbb{R}^4$.

	F	C	W	D	S_M	$\hat{S}_M$
F	0.00	1.08	2.91	1.21	5.19	0.0072
C	1.08	0.00	2.89	0.64	4.60	0.0069
W	2.91	2.88	0.00	3.17	8.96	0.0043
D	1.21	0.64	3.17	0.00	5.02	**0.0110**

Table 4.4: Wasserstein distances and scaled sums with the warm subsets not included.

Shifting to a more quantitative perspective, the Wasserstein distances over these persistent diagrams are considered, following the same procedure as in Section 4.2.1, the results of which are shown in Table 4.4. The normalised Wasserstein distance sum, $\hat{S}_M$, has been described to quantify the topological uniqueness of the manifold M. Consequently, distinct increases in $\hat{S}_M$ imply some substantive change in topology when compared to other manifolds. The manifold with the largest $\hat{S}_M$ stands out as the most topologically distinct. As Table 4.4 shows, $\hat{S}_D$ is significantly larger than the others, hinting at potential novelty in the damage manifold; a known fact from the labelled data. Following far after is $\hat{S}_F$, which is consistent with the freezing effects observed in the natural frequencies; this is closely followed by $\hat{S}_C$.

4.3 3D Shadow: ω_1, ω_3, and ω_4

In this scenario, the focus shifts to a lower-dimensional representation, specifically a shadow embedded in $\mathbb{R}^3$. To form this shadow, ω_2, the temperature-nonlinear natural frequency is omitted. Since the data are now in three dimensions a direct visualisation is possible, as shown in Figure 4.5. The topological information is shown to give consistent results over reduced dimensional representations. While this statement is not a universal characteristic of TDA, this does underscore its robustness and adaptability via this type of projection.

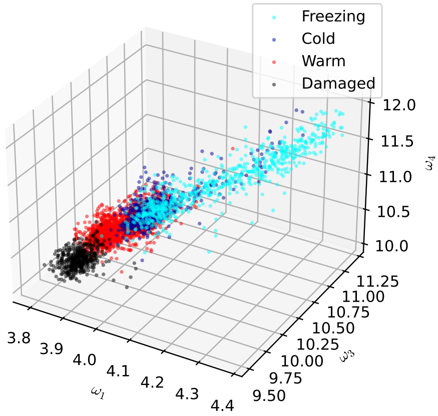


Figure 4.5: Manifold partitions in ω_1, ω_3, ω_4, before normalising.

The decision to omit ω_2 acts as a double-edged sword. On one hand, it removes the nonlinear freezing-temperature relation, but on the other hand, ω_2 is particularly sensitive to damage. Therefore, removing ω_2 indiscriminately removes the undesirable nonlinear EOV relation at the cost of a diminished damage response.

Counterintuitively, the exclusion of an axis actually may aid matters. This phenomenon can be attributed to the 'curse of dimensionality'. For example, consider a

	F	C	W	D	S_M	$\hat{S}_M$
F	0.00	0.64	1.52	0.91	3.08	0.0043
C	0.64	0.00	1.46	0.47	2.57	0.0039
W	1.52	1.46	0.00	1.74	4.72	0.0023
D	0.91	0.47	1.74	0.00	3.13	**0.0068**

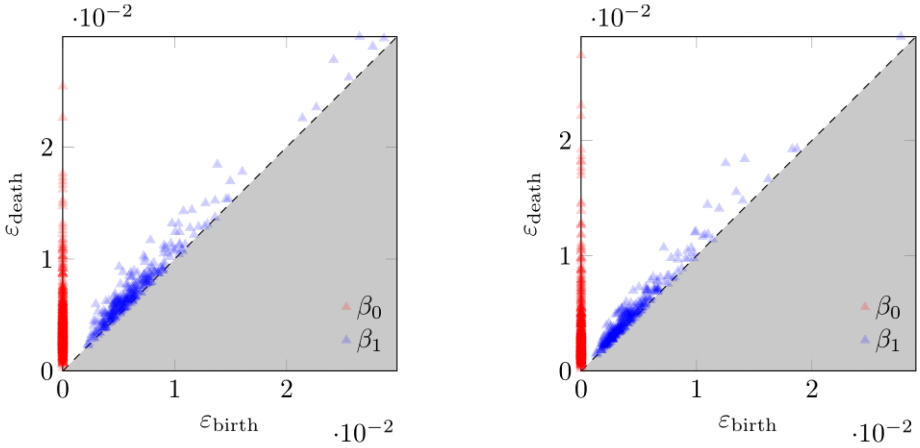
Table 4.5: Wasserstein distances with the highly nonlinear ω_2 removed.

densely-populated 1D line with n points along its length. For an equivalently-dense 2D square n^2 data points are required, and for a 3D cube, n^3 points are required, and so forth. As a greater number of dimensions are considered, exponentially-many points are required to maintain the same descriptive power.

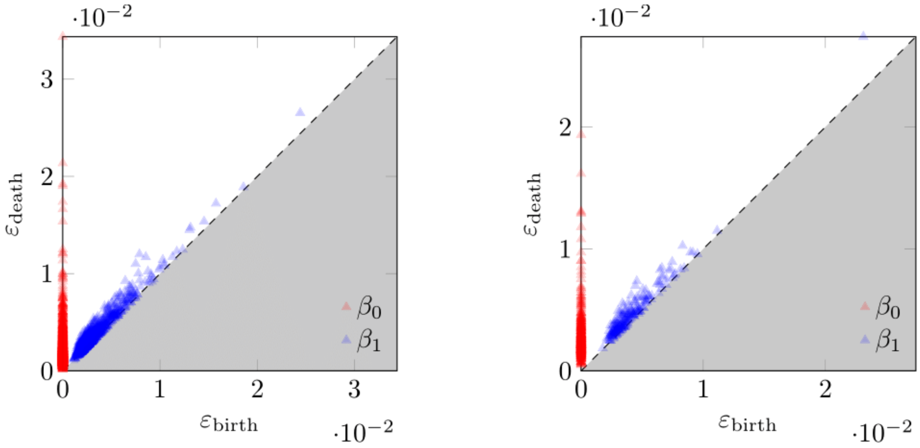
Given that the number of data points remains constant across different dimensional representations, the exclusion of a dimension will lead to a more densely-sampled and better-described manifold. However, this benefit comes at the expense of potentially informative degrees of freedom from omitted dimensions. The trade-off between clarity and information richness is common across all data-science disciplines, highlighting the need for thoughtful dimensionality-reduction processes. As an extension to this case, the subsequent section presents a less naive approach for reducing the dimension of the manifolds.

The results from the lower-dimensional shadow, as tabulated in Table 4.5, again show that the damage manifold is the most distinct, as $\hat{S}_D$ remains the largest value, even after omitting all information regarding ω_2. However, in this case, $\hat{S}_F$, $\hat{S}_C$, and $\hat{S}_W$ are closer in magnitude suggesting the structures of these manifolds are more similar in this space. This is likely arising from the removal of the nonlinear temperature effects present in ω_2. F, C, and W being similar topologically in this space further emphasise the damage manifold as being the most distinct, resulting in a relatively larger jump from $\hat{S}_F$ to $\hat{S}_D$ than when compared to the $\mathbb{R}^4$ embedding. This alignment between F, C, and W is also visible in their persistent diagrams, shown in Figure 4.6, as they visually vary less than when compared to the equivalents in Figure 4.4.

This example shows that TDA can still discern the damage partition as the most distinct, even in the presence of arguably the most significant feature being omitted. However, this approach to dimensionality reduction was based on prior engineering knowledge about the Z24 bridge. A more objective approach to dimension reduction is now to be considered via *principal component analysis.*



(a) Freezing (b) Cold



(c) Warm (d) Damage

Figure 4.6: Persistent diagrams for data without ω_2.

4.4 Principal Component Analysis

In this section, a linear dimension-reduction algorithm called *principal component analysis* (PCA) [83] is applied to the $\mathbb{R}^4$ embeddings of the Z24 bridge. While dimension reduction will inevitably result in information loss, PCA is a more refined approach than the previous example. PCA identifies orthogonal projections that retain as much data variance as possible. The first principal axis denoted PC_1, captures the most variance, with each subsequent axis capturing progressively less.

This case study examines the effects of taking the first three, and then the first two, principal components of the manifolds embedded in $\mathbb{R}^4$, and investigating changes in the topological structure along the way. The goal is to further understand the trade-off between preserving information and simplifying the data representation, a concept touched upon in the prior section. This demonstration will show how the data's topological structure is preserved over linear transformations, and how the accuracy degrades as the dimension reduction becomes more distant from the true embedding dimension.

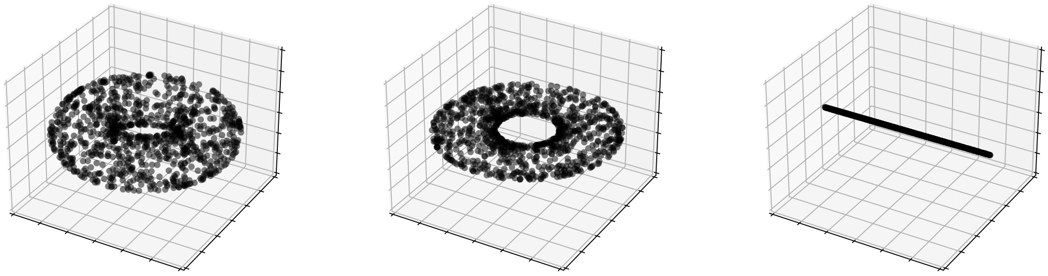

(a) $3D$ (b) $2D$ (c) $1D$

Figure 4.7: A torus (a) decomposed into $2D$ (b) and $1D$ (c).

To illustrate the effect of PCA on topology, consider a torus shell example, shown in Figure 4.7 (a). The original torus has Betti numbers of 1, 2, 1 for β_0, β_1, and β_2, respectively. Now consider the first two principal components, as shown in Figure 4.7 (b), the enclosed volume is lost. However, the central hole that characterises the torus remains. The first two principal axes then result in Betti numbers of 1, 1, 0 for β_0, β_1, and β_2, respectively. Finally, when considering only one principal component, as depicted in Figure 4.7 (c), the embedding is simply a straight line. Consequently, no interesting homological features are retained from the original manifold.

To clarify the procedure used on the Z24 data set: the data were partitioned into the appropriate classes. Each class was then normalised along each axis, by the largest value. Finally, the principal components of each class were determined individually.

First Three Principal Axes

This initial example maps the manifolds from four-dimensional to three-dimensional embeddings, aiming to retain as much information as possible. The exact amounts of explained variance for the principal component axes for each manifold are shown in Table 4.6. The persistent diagrams for the partitions are presented in Figure 4.8. When analysing these diagrams, the dependency of the β_1 features on temperature is more evident, mirroring observations from Section 4.2.2. This observation suggests that these patterns stem from the nonlinear temperature effects within ω_2, which were absent in Section 4.3 because of ω_2's omission. Hence, the consideration of ω_2 leads to notable alterations in the manifold's structure as temperatures vary.

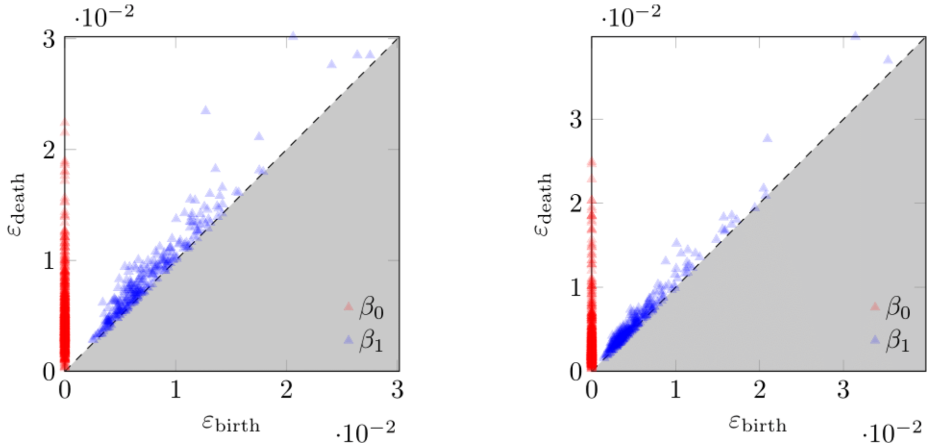
M	1	2	3
F	90.3%	95.6%	98.9%
C	77.7%	89.5%	97.2%
W	61.3%	79.3%	93.9%
D	76.6%	90.0%	96.0%

Table 4.6: Cumulative percentage explained variance for successive principal components.

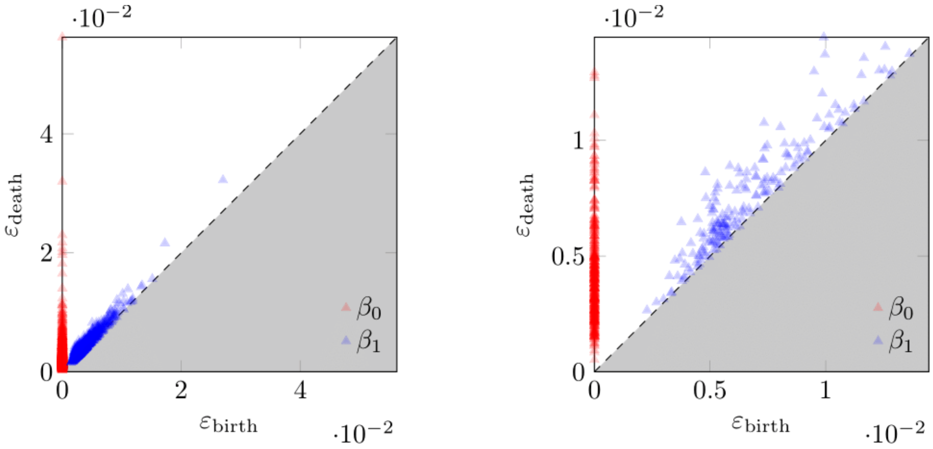
Furthermore, there has been a noticeable shift concerning the damage persistent diagram. While Figure 4.6 (d) showed the β_1 cluster beginning at $\varepsilon_{\text{birth}} = 0.02$, when considering effects from ω_2 in Figures 4.8 (d) and 4.4 (d), the β_1 cluster begins at around $\beta_1 = 0.05$. This shift suggests that the inclusion of ω_2 impacts the manifold's scale, observable in both 3D and 4D. Additionally, the β_1 features exhibit greater persistence with the inclusion of information from ω_2.

Assessing the Wasserstein distance sums for the first three principal components, as depicted in Table 4.7, it is evident that the damage remains the most distinctive aspect. However, its prominence has decreased compared to earlier observations. In this PCA case, the percentage difference between $\hat{S}_F$ and $\hat{S}_D$ is 37%, whereas in the

$\mathbb{R}^4$ embedding case, the equivalent percentage difference was 53%; indicating that damage sensitivity is being lost over dimensionality reduction.



(a) Freezing (b) Cold



(c) Warm (d) Damage

Figure 4.8: Persistent diagrams of the first three principal components, for the Z24 data partitions.

	F	C	W	D	S_M	$\hat{S}_M$
F	0.00	0.90	2.05	0.93	3.87	0.0054
C	0.90	0.00	1.87	0.55	3.32	0.0050
W	2.05	1.87	0.00	1.92	5.83	0.0028
D	0.93	0.55	1.92	0.00	3.40	**0.0074**

Table 4.7: Wasserstein distances of the first three principal components.

First Two Principal Axes

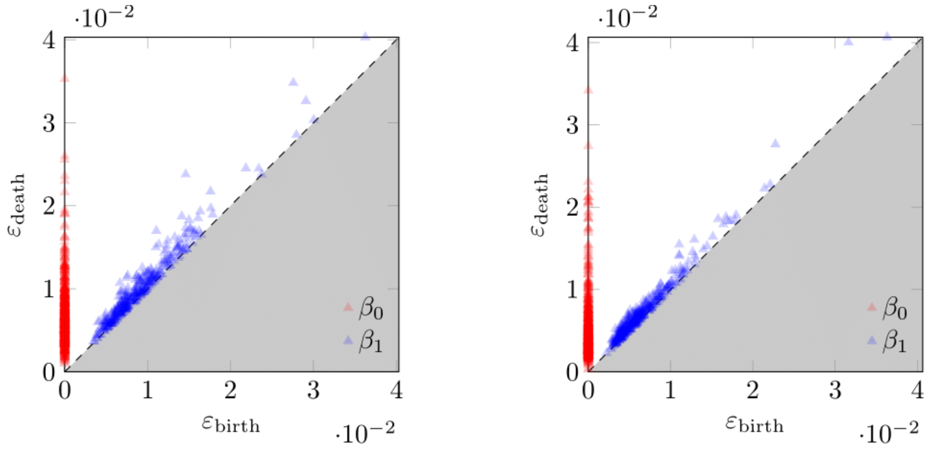
Now for a larger dimension reduction, specifically from 4D to 2D embeddings. This case still preserves the topology, although damage sensitivity is being lessened and temperature effects are becoming more dominant when compared to the $\mathbb{R}^4$ and $\mathbb{R}^3$ cases. Hence the nuances of the topology are being lost, much like in the Torus example, which in turn is affecting the ability to discern unique attributes between the manifolds. The loss of topological information can be explained by the reduction in the explained variance in two principal component axes, as indicated by Table 4.6.

Reducing the manifolds to such a degree overlooks intricate topological details, as the data are compressed down to only the first two principal components. Therefore, the persistent homology is not capturing intricacies, as it was in previous cases. Consequently, the gap in the Wasserstein distance sum between freezing and damage has narrowed, as shown in Table 4.8. Yet, despite this information loss and distinctiveness, the ordering between $\hat{S}_M$ remains unchanged, with the damage manifold being quantified as the most unique.

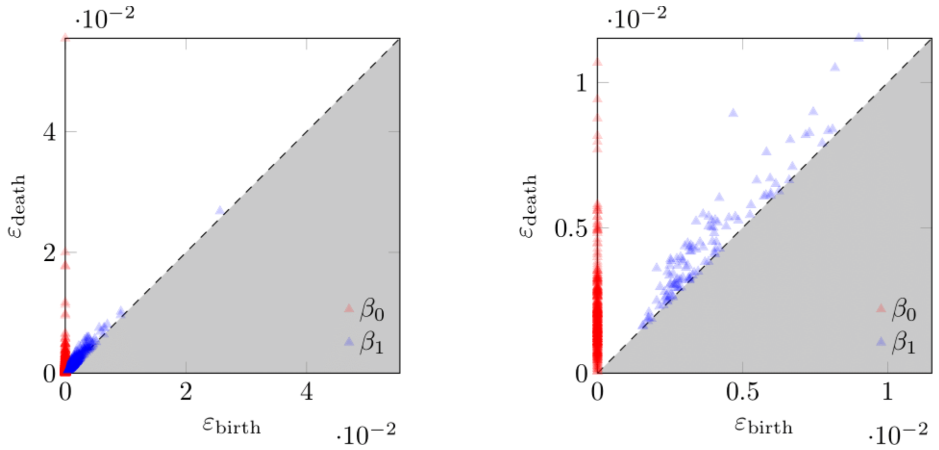
	F	C	W	D	S_M	$\hat{S}_M$
F	0.00	0.51	0.91	0.47	1.89	0.0026
C	0.51	0.00	0.73	0.39	1.63	0.0024
W	0.91	0.73	0.00	0.68	2.32	0.0011
D	0.47	0.39	0.68	0.00	1.53	**0.0033**

Table 4.8: Wasserstein distances of the first two principal components.

This case emphasises that dimensionality reduction invariably leads to information loss, and when embedding the Z24 manifolds in two dimensions via PCA, the result proves too much of a compromise to give results for viably detecting novelty.



(a) Freezing (b) Cold



(c) Warm (d) Damage

Figure 4.9: Persistent diagrams for the first two principal components.

4.5 Conclusion

This chapter introduced some aspects of the Z24 Bridge data set, which will be used throughout this work. However, the primary objective of this research was to propose a topological approach to detect damage or novelty within data. This objective is especially challenging as the EOVs present in the Z24 bridge exert a more pronounced effect on the raw data than the damage itself. Despite this, TDA was able to identify topological nuances between different data partitions and identified the damage case as the most distinct. These findings were grounded solely on topological reasoning, introducing novel concepts to the field of SHM.

To ensure unbiased comparisons of manifolds, a normalisation scheme was presented, to aid novelty detection. This normalisation procedure was designed to account for manifolds with varying numbers of data points, which was motivated by a case study where the natural frequencies were embedded in $\mathbb{R}^4$. Subsequent case studies further validated this approach.

Upon defining the normalisation scheme, some additional cases were presented, which demonstrated the effectiveness of TDA to preserve data integrity when considering lower-dimensional embeddings. An argument was presented showing that topological information is lost when compressing data into successively fewer dimensions.

Despite these inherent complexities, all cases using this normalisation scheme for evaluating manifold uniqueness consistently identified damage as the most distinctive feature. This is noteworthy, especially considering that temperature-induced effects have a more pronounced impact on the natural frequencies than the damage.

Chapter 5

VECTOR REPRESENTATIONS

The form of the persistence intervals makes them problematic for analysis with common data science techniques, notably *machine learning* (ML). Consequently, this chapter focuses on techniques to bridge the TDA-ML gap by interfacing persistent homology with ML. There is a growing interest in such procedures where the persistence intervals are transformed into a stable vector form, thereby streamlining the integration of topological feature extraction into ML pipelines. This ML preprocessing approach paves the way for directly incorporating data shape as a distinct feature in ML. Although this study exclusively uses vectorised forms of the persistent homology to classify data, indicating that shape is enough to distinguish data, this is not a strict necessity. The vectorised persistent homology features can be used in conjunction with more traditional features to supplement ML decision-making.

The inherent structure of persistent intervals is awkward to interface with ML because they are not represented as fixed-length vectors. Whilst the count of persistent intervals is somewhat influenced by the number of points and embedding dimension, there is no clear formula for the output number of persistent intervals. In contrast, ML requires inputs of known-fixed length. Moreover, persistent intervals are represented as unordered sets of intervals of the form $(\beta_n, (\varepsilon_{\text{birth}}, \varepsilon_{\text{death}}))$. This format deviates from the consistent-ordered vector pattern that ML demands.

Central to this work are two vector representations: Betti curves [87] and persistence images [88]. These vector representations are applied to persistent homologies derived from the Z24 bridge's natural frequencies. Via these representations, a classifier is developed to distinguish samples exclusively by their persistent homologies. The methodology for this analysis deviates slightly from more conventional classification strategies associated with the Z24, as to derive a meaningful topology, multiple data points are required, whereas a single instance usually represents a sample. Consequently, this analysis considers a sliding window of points and computes the persistent homology over this window, which represents a sample. As this work serves as the first application of a TDA classifier in the SHM literature, this study adopts a slightly simplified problem as compared to Chapter 4, where only two classes are considered, these being normal condition and damage.

In order to classify the vector representations, *logistic regression* is used in a supervised setting. The simplicity of logistic regression makes it an ideal choice for illustrating the connection between TDA and ML, by keeping the focus on the vector representations by not imposing complex ML algorithms. However, the extracted features from TDA are in no way exclusive to logistic regression.

5.1 Logistic Regression

Before jumping into the details of the vector representations, a brief introduction to logistic regression is provided, as this is used to distinguish between the persistent homology vector representations being discussed. Nonetheless, this is a passing overview in order to shed light on the classification procedure but keep the focus on the persistent-homology representations. For a thorough understanding of logistic regression, especially when motivated by Bayesian statistics and a detailed look at the training process, readers are directed to works such as [89, 90, 91].

Logistic regression estimates the probability of a binary outcome, determining whether a sample does or does not belong to a specific class, based on some features $\mathbf{x}$. In this work, the features are encapsulated in the vector representations of persistent homology.

Suppose there are two classes A and B, a set of p features $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_p \end{bmatrix}$, and a corresponding set of weights $\mathbf{w} = \begin{bmatrix} w_1 & w_2 & \cdots & w_p \end{bmatrix}$, where the weights are derived from some training data. The general idea of logistic regression is to fit a linear model to the training data, dictated by $\mathbf{w}$ and then apply a function to squeeze the output of the model between 0 and 1. The function used to squash the

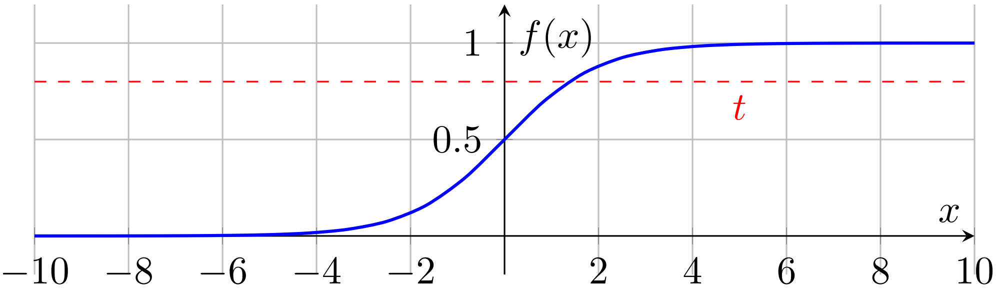


Figure 5.1: Logistic Function, with some arbitrary threshold t.

linear model is called the *logistic function*, and is defined by,

$$\sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}} \tag{5.1}$$

Since $\sigma(\mathbf{w}^T\mathbf{x})$ has two horizontal asymptotes at $y = 0$ and $y = 1$, its value is bounded in this interval for all $\mathbf{w}^T\mathbf{x}$, thus the results from the logistic function may be interpreted as probabilities of belonging to a class. Consequently, if $\sigma(\mathbf{w}^T\mathbf{x})$ is less than some threshold t, $\mathbf{x}$ is assigned the predicted label A; otherwise, if $\sigma(\mathbf{w}^T\mathbf{x}) \geq t$, it is labelled B. Figure 5.1 demonstrates this idea, where any points on the logistic function below t are predicted to belong to A, and any points above are predicted to belong to B.

5.1.1 Classification Metrics

To assess how well the classification has performed, four metrics are considered, called: *accuracy*, *precision*, *recall*, and F_1 *score*. These classification metrics are all built on top of a *confusion matrix*, shown in Table 5.1. The confusion matrix shows all possible classification outcomes, by comparing the actual versus the predicted classifications. In binary classification, there are four types of classification in the confusion matrix,

		Predicted	
		Normal	Damage
Actual	Normal	TN	FP
	Damage	FN	TP

Table 5.1: Example Confusion matrix.

1. True Negative (TN): Correctly predicted negative cases.
2. False Positive (FP): Incorrectly predicted positive cases.
3. False Negative (FN): Incorrectly predicted negative cases.
4. True Positive (TP): Correctly predicted positive cases.

From the confusion matrix, metrics can be defined that quantify a classifier's performance [92].

Accuracy is the fraction of correctly-predicted instances. From the confusion matrix, accuracy is determined by,

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{5.2}$$

Accuracy is intuitive, assessing overall correctness, but can be misleading, especially with imbalanced classes like these for the Z24. A high accuracy could arise by correctly predicting the size-dominant class. The accuracy also does not give any insight into the types of misclassifications, therefore other measures are used to complement it.

Precision is the ratio of correctly predicted positive observations to the total predicted positives, determined by,

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5.3}$$

Precision assesses how well the classifier is predicting the positive class. However, this does not consider true negatives, so is used in conjunction with *recall.*

Recall is the ratio of correctly-predicted positive observations to all actual positive occurrences, determined by,

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5.4}$$

In SHM, high recall is essential because a classifier with low recall indicates an inability to detect damage, potentially leading to severe consequences. However, there exists a trade-off between precision and recall, making it crucial to find a balance between them.

F_1-score represents the harmonic mean of precision and recall, providing a combined measure of the two, mathematically stated as,

$$F_1 = \frac{2}{\dfrac{1}{\text{precision}} + \dfrac{1}{\text{recall}}} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = \frac{\text{TP}}{\text{TP} + \dfrac{\text{FN} + \text{FP}}{2}} \tag{5.5}$$

The F_1-score offers a more reliable measure than accuracy, especially in imbalanced data sets like the Z24.

The confusion matrix is also used for determining an optimal classification threshold, t. By adjusting the threshold, the optimal value for this analysis is established by,

$$\max_t \left(\frac{\text{TP}}{\text{TP} + \text{FN}} - \frac{\text{FP}}{\text{FP} + \text{TN}} \right) \tag{5.6}$$

5.2 Betti Curve

Arguably the simplest vectorisation of persistent homology is the Betti curve. The first application [87] of the k^{th}-Betti curve counted the number of k dimensional persistent intervals present at any ε in the filtration[1]. Over time, there has been progress in associating a weight function w to each persistent interval, parameterised by the interval's birth and death values [53]. The Betti curve relating to the homology dimension k is given by,

$$\beta_k(\varepsilon) = \sum_{[\varepsilon_{\text{birth}},\, \varepsilon_{\text{death}}] \in \text{PH}_k} w(\varepsilon_{\text{birth}},\, \varepsilon_{\text{death}}) \cdot [H(\varepsilon - \varepsilon_{\text{birth}}) - H(\varepsilon - \varepsilon_{\text{death}})] \tag{5.7}$$

---

[1] This application essentially has a weight function $w(\varepsilon_{\text{birth}},\, \varepsilon_{\text{death}}) = 1$.

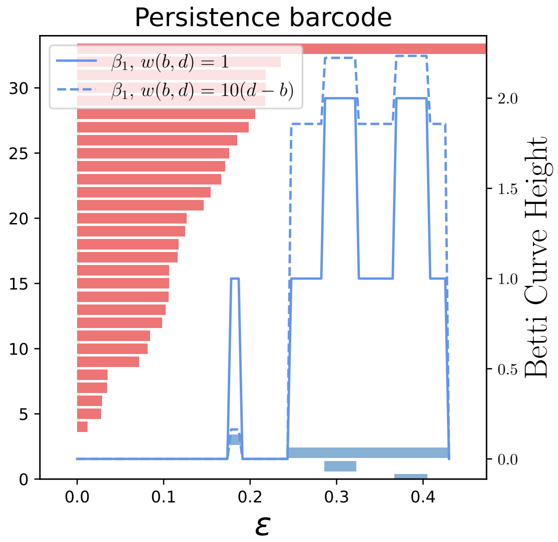


Figure 5.2: Two Betti curves shown on top of their barcode: one with a weight function and one without.

where H is the Heaviside step function,

$$H(\varepsilon) = \begin{cases} 0, & \text{if } \varepsilon < 0 \\ 1, & \text{if } \varepsilon \geq 0 \end{cases} \tag{5.8}$$

Betti curves serve as a one-dimensional vector representation of persistent homology, capturing the persistent homology information across the span of ε. This span is divided linearly into p segments, with the curve being evaluated at each segment. The values at each of the p segments then define the p features input into the logistic regression classifier. By keeping a consistent p value for all sliding windows, persistent homologies are represented by vectors of uniform length, irrespective of the number of persistent intervals.

Two example Betti curves are displayed in Figure 5.2, both derived from the same persistent homology. The first Betti curve, represented by the solid line, counts the number of persistent intervals at ε. The second, depicted with a dashed line, incorporates a weight function[2] $w(\varepsilon_{\text{birth}}, \varepsilon_{\text{death}}) = 10(\varepsilon_{\text{death}} - \varepsilon_{\text{birth}})$. The factor of 10 is used for illustrative purposes, ensuring the two Betti curves in Figure 5.2 are

---

[2] Providing a greater weight to longer persistence intervals may seem intuitive, but it should be noted that one work [40] found that medium-length persistences were most descriptive for their application. Therefore, a monotonically-increasing function might provide suboptimal results in such cases.

displayed on a comparable scale.

Incorporating such a weight function ensures a greater influence from the longstanding persistent intervals, which are more likely to represent true topological features of the manifold, while simultaneously limiting the effects of shorter persistent intervals, likely arising from topological noise. Mitigating effects from topological noise is essential in this work, given the subsampling into sliding windows.

In this classification procedure, only the $\beta_1(\varepsilon)$ Betti curves are considered. Consequently, the Z24 natural frequency sliding windows are classified entirely by the quantity and size of their 2D holes. Additionally, this study uses the persistent interval length as the weight function, i.e. $w(\varepsilon_{\text{birth}}, \varepsilon_{\text{death}}) = \varepsilon_{\text{death}} - \varepsilon_{\text{birth}}$.

5.3 Persistence Image

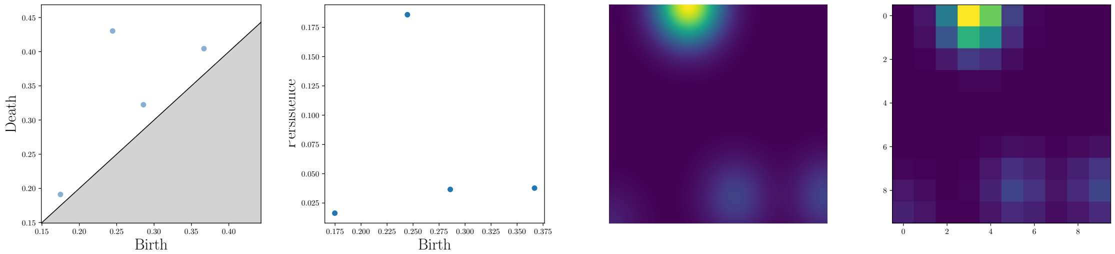
Persistence images, much like Betti curves, imbue a stronger structure to persistent homology, than when compared to its standard multi-set representation. This added structure ensures the persistent images are passable as features in a logistic regression classifier. Persistent images represent the persistent homology in a 2D plane, allowing for an intuitive pictorial representation. Furthermore, persistence images are provably stable to small perturbations in the persistent homology [88]. This stability is essential in their application to real-world data, where the presence of noise is inevitable.

To convert the persistence homology into a persistence image, it is standard practice to omit the connectedness features, β_0, focussing instead on higher-dimensional homology features. However, in this study, similar to the approach with Betti curves, only β_1 persistent images are considered.

Given a persistence (also known as birth-death) diagram D, each persistent interval is represented as a point $(\varepsilon_{\text{birth}}, \varepsilon_{\text{death}})$. To prepare the birth-death diagram for a persistence image, the linear map $T : \mathbb{R}^2 \to \mathbb{R}^2$ is used to convert it into a *birth-persistence* format, as described by

$$D' = T(D) = \{(\varepsilon_{\text{birth}}, \varphi) \ : \ \forall(\varepsilon_{\text{birth}}, \varepsilon_{\text{death}}) \in D\} \tag{5.9}$$

where D' is the birth-persistence diagram equivalent of D, and the persistence is denoted $\varphi = \varepsilon_{\text{death}} - \varepsilon_{\text{birth}}$. In D', the x-axis still indicates the length scale of



(a) (b) (c) (d)

Figure 5.3: The development of the persistence image: (a) The birth-death diagram of H_1 for some data. (b) The birth-persistence diagram, $T(H_1)$. (c) The persistence surface with a linear weight function over the persistence interval length. (d) The persistence image with 10×10 resolution sampled from the persistence surface.

homology feature formation, but the y-axis is now its persistence.

Now, let $\phi : \mathbb{R}^2 \to \mathbb{R}$ be a differentiable probability distribution, typically a 2D Gaussian $g_{p'}$. The probability distribution is centred at each point p' in the birth-persistence diagram, serving as a smoothing function. This work uses a Gaussian, which is defined as,

$$\phi_{p'}(x,\, y) = g_{p'}(x,\, y) = \frac{1}{2\pi\sigma^2} \exp\left\{ -\frac{1}{2\sigma^2} (x - \varepsilon_{\text{birth}})^2 + (y - \varphi)^2 \right\} \tag{5.10}$$

where $\varepsilon_{\text{birth}}$ and φ are the coordinates of p' [53, 88].

Similarly to the Betti curve, a weight function w modifies the magnitude of each point according to its birth and death length scales. This weight is applied to each p', scaling the probability distribution centred at that point, resulting in a continuous *persistence surface* ρ [88],

$$\rho(x,\, y) = \sum_{p' \in T(D)} w(p') \cdot \phi_{p'}(x,\, y) \tag{5.11}$$

The persistence surface provides a continuous 2D scalar field representation of the persistent homology. To make the persistence surface suitable for machine learning applications, it must be discretised. The surface is divided into a grid with m segments in the x-axis and n in the y-axis, the scalar field is then integrated over each grid cell to produce an $m \times n$ *persistence image* $I(\rho)$, mathematically given as,

$$I(\rho) = \iint_{m,\,n} \rho \,\mathrm{d}x\mathrm{d}y \tag{5.12}$$

Figure 5.3 shows each stage in forming a persistent image from a persistence diagram.

Persistence images, though insightful in their 2D form, are transformed into a vector format for computational applications. The $m \times n$ matrices are flattened into a $p \times 1$ feature vector which is representative of the image and persistent homology, where $p = m \cdot n$. In machine learning contexts, this vectorised matrix serves as a consistent feature, given fixed values of m and n. Much like the Betti-curve representation, this vectorised persistent image structure format provides a greater structure and consistency than the traditional multi-set form of persistent homology.

In the creation of persistence images from birth-death diagrams, there are three primary hyperparameters,

Resolution refers to the granularity of the discretised persistence surface. Determined by the product of the m columns and n rows, giving the number of features of this representation.

Distribution dictates the type of probability distribution used as a smoothing kernel over the birth-persistence diagram points. Commonly, as in this work, a Gaussian distribution is chosen. This selection introduces a secondary hyperparameter, the *bandwidth* σ, which governs the spread of the Gaussian. A smaller σ yields a distribution more focused on the point.

Weight Function assigns a magnitude to each persistence interval, giving varying emphasis based on $\varepsilon_{\text{birth}}$ and $\varepsilon_{\text{death}}$.

5.4 Application to the Z24 Bridge

Topological data analysis offers a fresh perspective on classification within machine learning, with a focus on shape-based classification. In contrast to conventional classification methods, which usually take a single point as a sample, topological approaches require a collection of points to determine a shape via their relative positioning.

In this application, a sliding window of points examines the Z24 Bridge's natural frequencies. Each sliding window serves as the foundation of a sample in this classification task. To attribute shape to each window, the four natural frequencies are embedded as points in $\mathbb{R}^4$. Then the shape of each sliding window in this embedding space is quantified via persistent homology, similar to the analysis in Section 4.2. However, in this case, there is an additional preprocessing step, of extracting the vectorised form from the persistent homology, which is subsequently integrated into a machine-learning framework.

Introducing a sliding window of points complicates the labelling process. With the labelling criteria in Section 4.1, centred around the temperature of each data point, sliding windows are likely to contain a mixture of labels, making window labelling nontrivial. Consequently, this classification problem is reduced to a binary one, by separating instances into normal condition or damage data classes. However, some overlaps still exist. For instance, windows containing the damage index 3475 will have both normal condition and damage data. In such cases, a conservative labelling methodology is applied: if any instance within a window has a damage label, the whole window is subsequently labelled as damaged. Such a conservative strategy is essential when concerning safety-critical applications, even if at the expense of classification precision.

The structured outline of this classification procedure is as follows,

1. **Standardise:** The Z24 natural frequencies are standardised to ensure simplex formation is fair along each axis.
2. **Window:** Segment the data using a sliding window of predetermined and consistent length w_l, creating a $w_l \times 4$ array of points.
3. **Label Window:** Assign a label to each sliding window, determined by:
    (a) **Normal Condition:** If all instances within the window do not belong to the damage class.
    (b) **Damage:** If any instance in the window is labelled damage, the entire window is designated as damage.
4. **Persistent Homology:** Embed the sliding window in $\mathbb{R}^4$, and compute its persistent homology.

5. **Representation:** Extract a consistent, structured vectorised representation from the persistent homology, forming a single sample permissible for machine learning, with all information based solely on the persistent homology.

6. **Form All Samples:** Repeat steps 2 to 5, considering the next sliding window from the Z24 natural frequencies. This iteration continues over the length of the data set, forming a set of vectorised persistent homology samples, with corresponding labels.

7. **Train-Test Split:** Partition the sample set into training and test subsets.

8. **Train Classifier:** Train the classifier in a supervised manner using the samples from the training dataset. The outcome is a classifier with the ability to classify unseen samples of the vector representations.

9. **Assess Classifier:** Evaluate the classifier using the test samples to gauge the model's performance.

The choice of the sliding window size w_l is a vital consideration. A smaller w_l is likely to lead to a persistent homology dominated by noise, and lacking sufficient topological information indicative of normal condition and damage. Conversely, larger w_l provide a better-described topology, at the expense of fewer samples, reduced sensitivity to the onset of damage, and an increase in computation time. In addition, when segmenting the data into windows, reducing overlaps can be advantageous. If windows were formed by sliding down just one position, consecutive windows would only differ by a single point, meaning their persistent homologies would be near-identical. Hence, a sliding window step s is introduced to decrease the overlap between the adjacent windows. In essence, s determines the number of distinct points between neighbouring sliding windows. The implications of w_l and s are discussed more comprehensively in Section 5.4.3. Taking into account these factors, $w_l = 300$ and $s = 5$ were selected for this study, leading to 635 samples under normal circumstances and 92 samples indicating damage.

This study examines two persistent homology vector representations: the Betti curve and the persistence image, assessing their capability to distinguish data between normal and damaged conditions. The results are detailed in Sections 5.4.1 and 5.4.2. To ensure fairness and consistency, the same hyperparameters were maintained across both vectorisations, including window size, weight function, and the number of features. The number of features for both was set to $p = 16$, leading to an 8×2

discretisation of the persistence surface. Furthermore, the weight function used for both representations is $w(\varepsilon_{\text{birth}}, \varepsilon_{\text{death}}) = \varepsilon_{\text{death}} - \varepsilon_{\text{birth}}$, representing the persistence of the feature.

As alluded to in Sections 5.2 and 5.3, only the β_1 features are considered for simplicity and reduced computational time. Therefore, the vector representations encode information related to the number and size of the 2D holes within each window, making these topological characteristics the sole determinants for classification. Nonetheless, both representations are readily extensible to include higher-dimension homology features. Moreover, via the use of sliding-window subsampling, calculating these advanced features becomes more manageable and efficient.

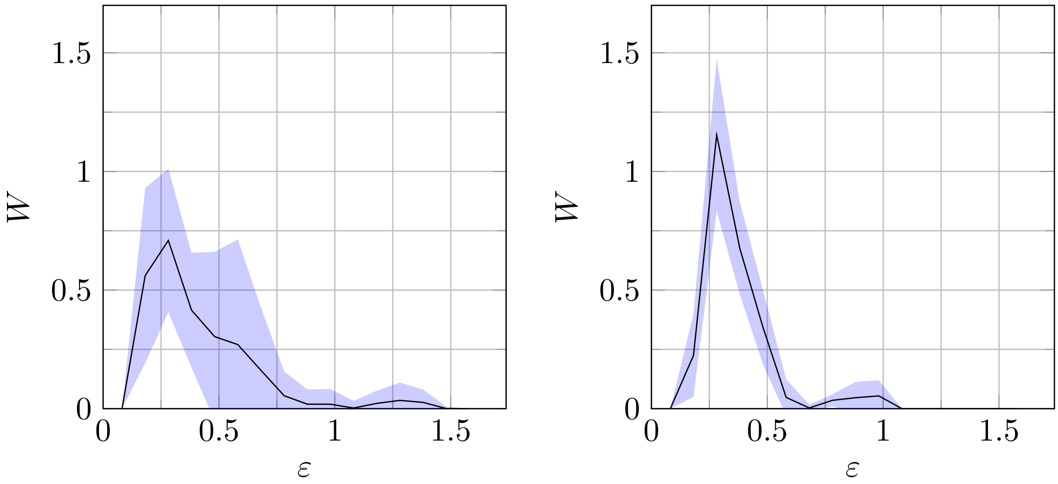
Maintaining consistent hyperparameters across the two representations ensures no undue preference for one over the other, permitting an objective comparison of their effectiveness to classify damage in the Z24 bridge's natural frequencies. The only difference is how the two representations encode the persistent homology information.

For more robust classification metrics, the procedure was executed 200 times, with averages subsequently calculated. Furthermore, windows were assigned randomly, adhering to a 60%–40% training-test split.

5.4.1 Betti Curve

To classify the sliding windows based on their persistent homologies, the Betti curve vectorisation serves as the initial application. Figures 5.4 (a) and (b) display the averaged Betti curves for the normal and damage conditions, respectively. Notably, the averaged Betti curve for the normal condition shows a much greater variance than the damage condition equivalent. This finding is consistent with expectations, given that there is no consideration of temperature in this binary classification. The normal condition spans sliding windows inclusive of freezing, cold, and warm regions, capturing the full spectrum of natural frequency variations because of temperature.

As a case in point, Figure 5.5 offers additional insights by averaging the Betti curves within specific areas of interest. Figure 5.5 (a) shows the average Betti curve for sliding windows including the sustained freezing region. Within this figure, there is a significant peak at $\varepsilon = 0.58$, a feature also evident within the variance of Figure 5.4 (a). This like-for-like peak surge suggests that the variance in Figure 5.4(a)

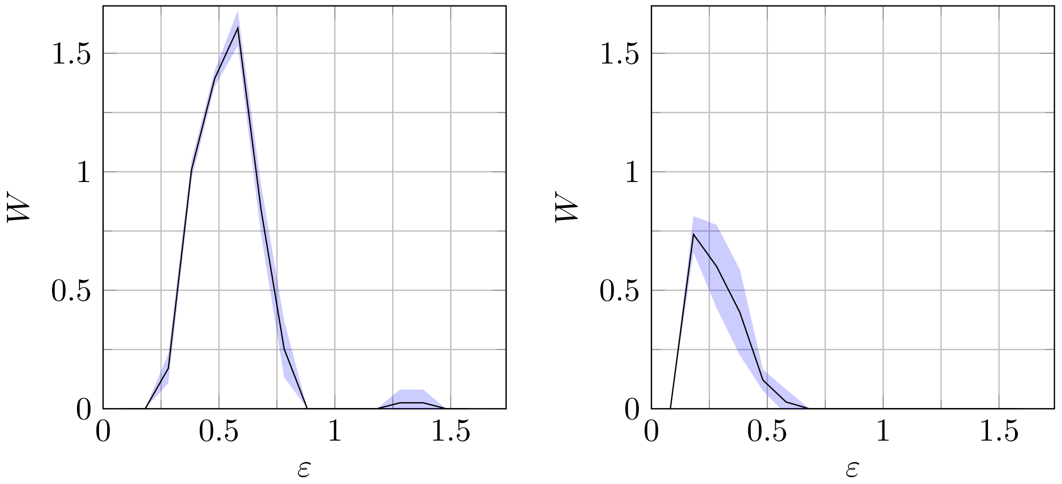


(a) Normal (b) Damage

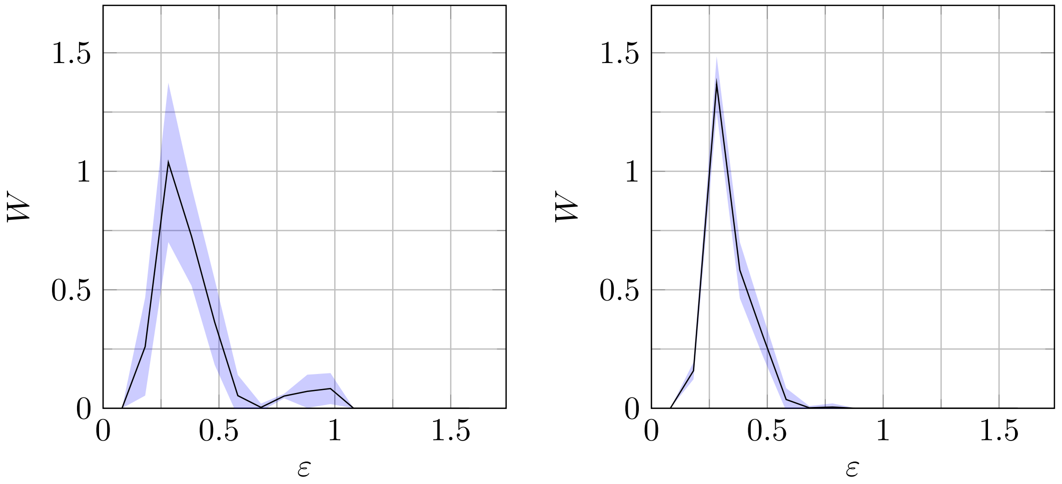
Figure 5.4: Average Betti curves of the two classes.

results from the manifold shape alterations influenced by temperature.

For the damage condition, the temperature has a muted effect on the natural frequencies because all data points in the sliding windows are sampled from relatively-high temperatures. Nonetheless, there is still significant variance visible in Figure 5.4 (b). This variance arises from the gradual transition from the warm-undamaged-state Betti curve to the purely damaged Betti curve. At the onset of damage, there are very few damage points in the sliding window, but each successive sliding window includes an increasing number of damage instances until the window is fully comprised of damage data points. This evolution is demonstrated in Figures 5.5 (b)-(d): Figure 5.5 (b) shows the average warm undamaged curve, just before damage onset. Figure 5.5 (c) shows a mix of warm and damage data points, and finally Figure 5.5 (d) is the average of the Betti curves created from sliding windows fully comprised of damage data points. Over these figures, there is an evolution from the undamaged to the damaged state. When more damage points are included in the sliding window, the Betti curve more closely represents the exclusively damaged case.



(a) Freezing (b) Warm only



(c) Warm-damage mix (d) Damage only

Figure 5.5: Average Betti curves over the large freezing region.

		Predicted	
		Normal	Damage
Actual	Normal	233.7	21.0
	Damage	1.0	35.2

Table 5.2: Averaged confusion matrix for Betti curve classification.

The averaged results for the confusion matrix are shown in Table 5.2. Derived from this confusion matrix, the classifications metrics are: accuracy = 92.4%, precision = 63.3%, recall = 97.3%, F_1 score = 76.3%.

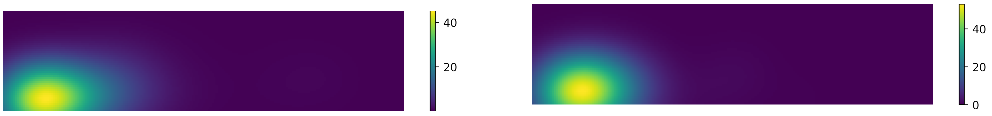
For SHM systems, it is informative to understand which regions are being misclassified. Considering the temperature labels as discussed in Section 4.1, these can help identify regions where classification is failing. The most significant region of misclassification occurs with false positives, where undamaged windows are being wrongly predicted as damaged. Out of the average 21.0 misclassifications, 20.3 arise from sliding windows containing a mix of freezing, cold, and warm data. Including data over such a broad temperature range will inevitably result in a correspondingly large natural frequency range within the window. The classifier might be misinterpreting this large natural frequency variation as the large variation stemming from the onset of damage.

For false negatives, which are of greater concern for an SHM classifier, all but one of these false negatives arose from sliding windows consisting of warm and damage instances over the 200 repetitions. These errors predominately occur at the damage onset when there are insufficient data points to depict the topological features associated with damage. This outcome is anticipated given the conservative labelling strategy, where just one damage instance within the 300-point window assigns it a damage label. Notably, there was an exceptional case of a window entirely composed of damage data that was mistakenly labelled as undamaged. However, this rarity only contributes 0.005 to the overall average.

5.4.2 Persistence Images

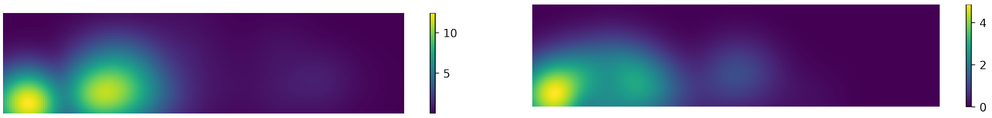
Persistence images offer an alternative means to vectorise persistent homology, by more freely being described in 2D, though they come with complexities, primarily from the additional hyperparameters of discretisation and probability distribution. An inappropriate selection of hyperparameters can render many of the features unhelpful for classification. If σ is too large, the persistent images can merge into an uninformative continuous image. Conversely, small σ require more features to capture the finer details in discretisation.

In this work, as detailed in Section 5.3, a 2D Gaussian is selected, with its bandwidth set to $\sigma = 0.1$. For the discretisation, the number of rows and columns was chosen to match the aspect ratio of the birth-persistence diagrams, ensuring that



(a) Normal average

(b) Damage average



(c) Normal standard deviation

(d) Damage standard deviation

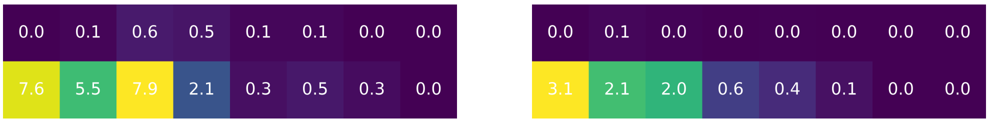
Figure 5.6: Average persistence surfaces over both classes.

0.0	0.1	0.2	0.1	0.0	0.0	0.0	0.0
17.5	36.9	10.8	1.8	0.3	0.3	0.1	0.0

(a) Normal average

0.0	0.2	0.1	0.0	0.0	0.0	0.0	0.0
13.8	45.0	10.1	1.0	0.4	0.2	0.0	0.0

(b) Damage average



0.0	0.1	0.6	0.5	0.1	0.1	0.0	0.0
7.6	5.5	7.9	2.1	0.3	0.5	0.3	0.0

(c) Normal standard deviation

0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0
3.1	2.1	2.0	0.6	0.4	0.1	0.0	0.0

(d) Damage standard deviation

Figure 5.7: Average persistence images for both classes.

		Predicted	
		Normal	Damage
Actual	Normal	242.9	11.9
	Damage	2.2	34.1

Table 5.3: Confusion matrix for persistence image classification.

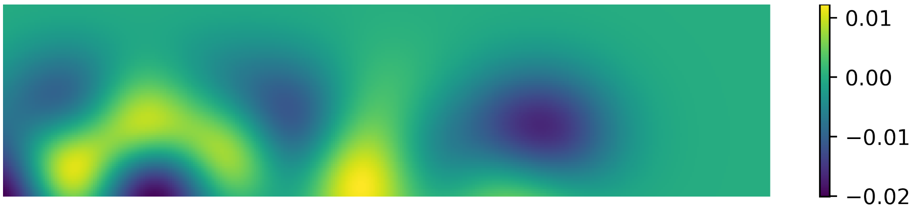


Figure 5.8: Persistence image classification weights.

the persistence images were not biased towards one axis. This approach determined that the $\varepsilon_{\text{birth}}$ range is roughly four times greater than the persistence range, resulting in an 8×2 discretisation; thus, preserving the number of features over the two vector representations.

Had this method not been employed, a compromise would be needed between either skewing the data or increasing the feature count. For comparison, a square persistence image with the same column resolution, and uniform axis scales would require an 8×8 discretisation, leading to 64 features, where 48 features (represented by the top 6 rows) would be largely redundant, offering little to no information for classification.

The mean persistence surfaces for both classes are shown in Figures 5.6 (a) and (b). Directly below, their respective standard deviations are depicted in Figures 5.6 (c) and (d). The corresponding discretisations of these surfaces are provided in Figure 5.7. From these persistent images, classification was performed, resulting in the averaged confusion matrix presented in Table 5.3. From this confusion matrix, the classification metrics are as follows: accuracy = 95.2%, precision = 75.3%, recall = 94.1% and an F_1 score = 83.2%.

Similar to the Betti-curve classification, a significant contribution of 10.2 of the false positives stems from the combination of the freezing, cold and warm regions. The remaining 1.7 originates from purely-warm sliding windows. Most importantly, the persistence image results in fewer than half the false positives compared to the Betti curve. However, this significant improvement in precision does compromise the recall, illustrating the precision-recall trade-off. An average of 2.2 damage sliding windows are now misclassified as undamaged. All of these false negatives arise at the damage onset, where there is a window mixing of warm and damage data points.

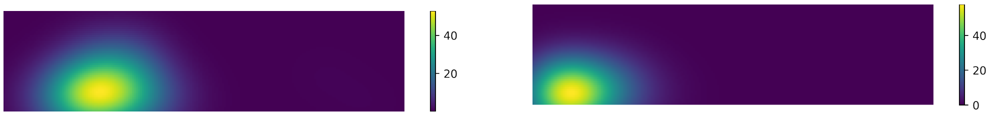
When conducting the persistence image classification procedure at a high resolution, the weights used for classification can be visualised, akin to the persistence surface, offering insight into damage-sensitive topological features. Given that the points in a persistence diagram, the Gaussian distribution, and the weight function are always positive, the classification weight's sign indicates which areas of the persistence surface signify damage. Figure 5.8 shows the weight surface, where negative values suggest regions of topological features relating to normal conditions, whereas positive values hint at damage. The magnitude of these values indicates the confidence level of the region corresponding to a particular class. Thus, a more pronounced yellow hue in Figure 5.8 signifies areas of topological features with a stronger association with damage.

5.4.3 Discussion

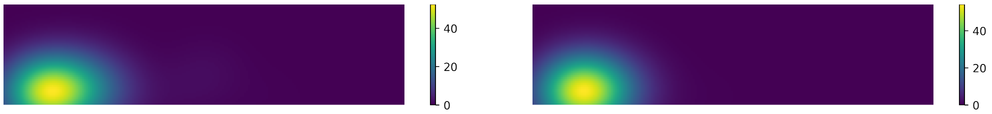
Both the Betti curve and persistence image provide encouraging outcomes for damage classification based on topological features. As the hyperparameters were consistent for both methods and thresholds were systematically determined, the difference in classification performance is intrinsic to their respective representation styles.

The key difference between the two vector representations is that Betti curves are inherently one-dimensional, whereas persistent images are two-dimensional. This dimensional discrepancy comes with trade-offs. Betti curves offer a more compact representation, but risk masking features because of the compression of information into a single dimension. Even with a weight function, a similar rise in a Betti curve could occur from a highly persistent feature or many localised smaller features. In contrast, the added spatial dimension in persistent images can differentiate between these scenarios by shifting the highly persistent feature along the y-axis. However, the extra dimension does not come for free and requires more features to maintain the granularity available from the Betti curve. In this application, which is dominated by minimally persistent features, this shifting in the y-axis is subtle. Nevertheless, other applications, with more dominant homological features might harbour better results with persistent images than Betti curves.

In this Z24 study, a typical persistent image often has vast areas of empty information, resulting in many nondescript features. However, different regions of the data tend to activate different areas of the persistent image, which aids the classification process. Therefore the added dimension offers a more intuitive mapping from features to classification than the Betti curves, as shown in Figure 5.8. To highlight this, the average persistence surfaces for the various regions of interest are shown in Figure 5.9, and the associated standard deviations are shown in Figure 5.10. The

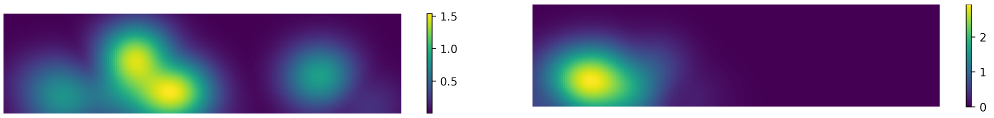


(a) Freezing (b) Warm

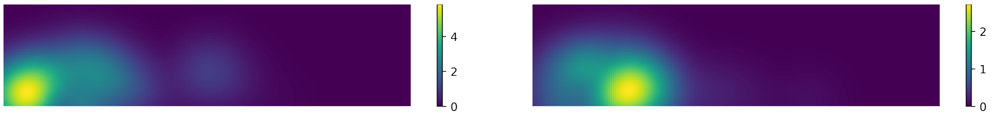


(c) Warm-Damage (d) Damage

Figure 5.9: Average persistence surfaces for supplementary conditions.



(a) Freezing (b) Warm



(c) Warm-Damage (d) Damage

Figure 5.10: Standard deviation of the persistence surfaces for supplementary conditions.

persistence images associated with the freezing region most noticeably make use of the added dimension, with the effects most clear in Figure 5.10 (a). The persistence image's ability to model the freezing topological features into an extra dimension might account for its superior precision over Betti curves.

There are however some challenges with this application of vector representations, that cannot go unstated. In order to form a descriptive topology of the sliding windows, a considerable number of data points are required. For this work, a sliding window length $w_l = 300$ was used, meaning 300 data points are required for a single classification sample. Notably, other studies within the SHM literature, which consider more traditional feature representations, have achieved better results, when considering only a single data instance [93, 94, 95].

To gauge the effect of sliding window size on the classification metrics, sliding windows of various sizes were considered while maintaining a consistent step size of 5 between each window. Figure 5.11 showcases the outcome of this analysis for the Betti-curve representation and suggests that distinguishable topological features are

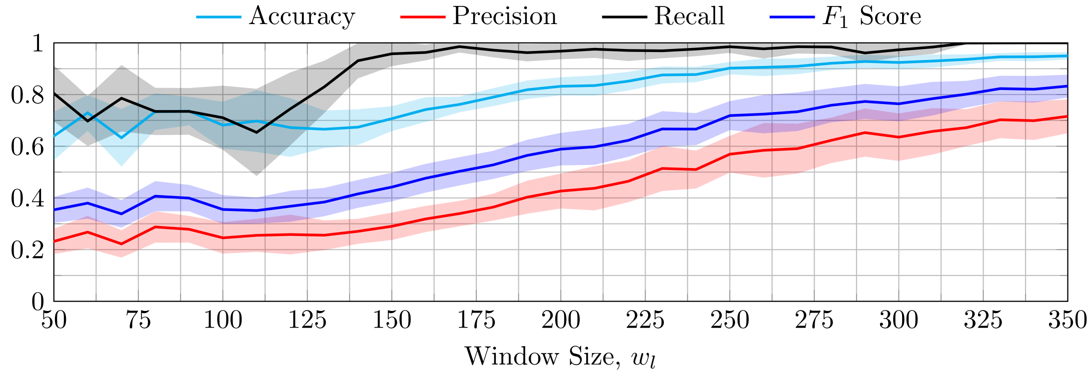


Figure 5.11: Effect of varying window size over classification measures for Betti curve, 100 repeats taken at each w_l.

not present until $w_l = 120$, where before this point the windows are dominated by uninformative topological noise. After $w_l = 120$ there is a gradual increase in the classification metrics. As the topologies within the sliding windows become more accurately depicted, the Betti curves can better capture the topological nuances between normal and damaged conditions, leading to enhanced classification.

The enhanced topological descriptive power of larger w_l comes at the expense of damage sensitivity. Specifically, let n_{dam} denote the number of damage points within a mixed window of length w_l. The proportion of damage points in the window is n_{dam}/w_l. Consequently, with a fixed n_{dam}, as w_l increases, the proportion of damage points decreases. As a result, at damage onset, the window's topological characteristics lean more toward the warm data points. With a larger w_l, this classification scheme is more sluggish to identify damage at its onset, hence possibly reducing the classification's precision.

Arguably, the most significant cause for concern is the significant overlap between sliding windows in this analysis. Given a step size of 5 and a window length of 300, which this study employs, adjacent sliding windows differ by merely 5 points. Consequently, adjacent windows' persistent homologies are likely to resemble each other. A situation then arises where one sliding window could be included in the training set while the following window is in the test set, or vice versa. Although their persistent homologies are not identical, they are likely to bear strong similarities, and consequently, their vector representations are very similar. Therefore, this substantial 98% overlap between successive windows may be inflating the classification performance. Increasing the step size between windows might appear to be a logical solution, however, this quickly decreases the number of samples, resulting in insufficient data to adequately train the classifier. An alternative solution might be to apply this classification approach to a more extended monitoring scheme than the Z24. Nevertheless, this vector representation method for topological-based classification showcases the potential of topological features in SHM classification.

5.5 Conclusion

This chapter marked an initial attempt at classifying the Z24 Bridge using persistent homology. This chapter presented a novel method that uses β_1 persistent homology features for detecting damage in sliding windows of data. This approach enabled classification based entirely on the quantity and size of 2D holes present in the sliding windows. The results demonstrated a fair classification of the sliding windows based on their persistent homologies, but this achievement required a considerable number of data points to establish a discernible topology.

However, more foundationally, a method of converting persistence homology into a format permissible for machine learning was introduced to the SHM community. These vector representations are versatile and can be adapted beyond just classification, logistic regression, Betti curves, or persistence images. These vector representations not only open doors from topological data analysis to machine learning but also offer insightful pictorial representations of persistent homology. Notably, both the Betti curves and persistence images exhibited marked differences across the freezing, warm, and damage regions, indicating these effects reflect uniquely into the persistent homology.

While the objective of this work was not to revolutionise damage detection within SHM, the intention was to spark a discussion on the application of topological features and their capability to discern between datasets.

Following on from this work, several potential directions for future research have been identified, including:

1. Consider a multiclass setting by reintroducing the temperature-based classes in Chapter 4.
2. Implementing a regression-based approach, to predict the proportions of points from freezing, cold, warm, and damage instances within the sliding window based on the window's persistent homology vector representation.

3. Investigate other persistent homology vector representations, such as complex polynomials, entropy, landscapes, and silhouettes.
4. Further assess the impacts of w_l, step size, number of features, and the inclusion of higher-dimensional homological features via cross-validation.
5. Transitioning from the supervised model to an unsupervised one, aiming to predict the onset of damage without labels.
6. Rather than including the mixing windows in the training set, exclusively train the classifier on homogeneously-labelled sliding windows, but then test over sliding windows with label mixing.
7. Investigate the impact of window size on early damage detection. This exploration might involve introducing a parameter that denotes the proportion of damage data points within the sliding window. Subsequently, evaluate how consistent this proportion is in identifying damage across varying window lengths.
8. A hybrid classification scheme, that makes use of topological and more traditional machine learning features. Showing if heightened results can be obtained from the inclusion of topological features.

Chapter 6

ATTRACTORS

Throughout this thesis, topological noise has often been regarded as uninformative and, to some degree, problematic. However, this perception is not entirely accurate; as this chapter presents, there exists meaning in the seemingly inconsequential. This chapter explores deeper into extracting information from topological noise by using it to calculate the *fractal dimension* of *strange attractors*. Doing so reveals that topological noise carries information that describes the inherent properties of embedded manifolds. Although topological noise might not be as enlightening as large-scale persistent features, this approach does convey that valuable information can be extracted from within. This approach of extracting insights from noisy regions of data is mirrored in the topological analysis of cointegrated residuals in Chapters 7 and 8.

Following the fractal-dimension calculations, certain reconstructed attractors derived from time-delay embeddings of data along a single axis, are evaluated using persistent homology. This process introduces a novel method to identify the most topologically similar time-delay embeddings for attractors when restoring them to their original dimensionality. This work serves as a nod to the pioneering use of time-delay embeddings in work by Takens [78]. In Takens' work, it was proven that the topology of a continuous attractor could be perfectly replicated by one of its axes via time-delay embedding. Unlike Takens' method, which requires large numbers of dimensions and continuous time series, this computational approach handles discrete data and embeds only into the attractor's original dimension. In this context, the Wasserstein distance between the persistent homologies of the original and reconstructed attractors is calculated over various α, determining a delay for a most-topologically-alike reconstruction.

6.1 Fractal Dimension

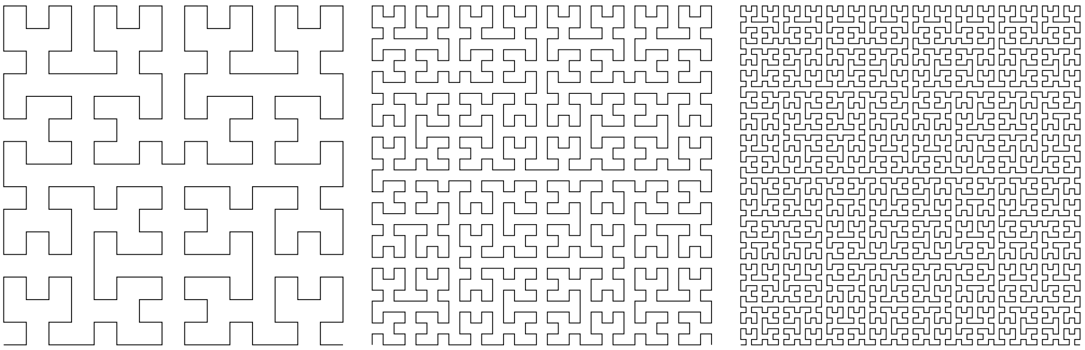

(a) 4 (b) 5 (c) 6

Figure 6.1: Different iteration depths of the Hilbert curve.

Fractal dimension is a measure of how much space a set occupies. For an intuitive grasp of fractal dimension, consider the following: a single point is zero-dimensional, a line connecting two points is one-dimensional, and the area spanning three points is two-dimensional, and so forth. However, these whole-number dimensions lack granularity, especially in edge cases where the dimensionality of curves blurs the lines of integer dimensions. Now, picture a curve of infinite length made of one-dimensional elements, yet confined to a limited region. This infinite-length curve inherently occupies more space than a straight line, but also does not completely fill this finite region. Hence, it is reasonable to suggest that this curve's dimension is somewhere *between* one and two; curves closer to one occupy less space, and the opposite is true for curves closer to two. This non-integer measure of dimension termed the *fractal dimension*, indicates the efficiency with which a curve fills space. This concept is demonstrated in Figure 6.1, where three examples of the Hilbert curve are shown at 4, 5, and 6 iterations. As the iterations of the Hilbert curve approach infinity, it tends to densely cover the 2D space, such that any small region of the space will contain a portion of the 1D curve.

The *box-counting method* is a widely recognised computational method to determine fractal dimension. This technique exploits the dimensional relationship between scalability and space occupied at varying scales. To calculate the fractal dimension through this approach, consider a space X. The smallest number sets with a diameter ϵ needed to cover X is denoted as $N(\epsilon)$. The choice of scale ϵ influences $N(\epsilon)$, while the dimension of X dictates the rate of change of $N(\epsilon)$. As the scale is altered, it is expected that the number of boxes scales with the dimension of the X [96],

$$N(\epsilon) \simeq c\epsilon^{-d_B} \tag{6.1}$$

for positive constants d_B and c. Here, X is said to have a fractal dimension of d_B. To deduce the dimension of X, rearrange the relation in equation (6.1) and take its limit, yielding,

$$d_B = \lim_{\epsilon \to 0} \frac{\log(N(\epsilon))}{-\log(\epsilon)}, \tag{6.2}$$

with the constant term, c, disappearing in the limit [97].

Fractals, such as strange attractors, are intrinsically complex objects that exhibit self-similarity, often over an infinite scale. However, there is a challenge when dealing with finite point-data sets, as they can only capture a limited amount of information. Upon excessive magnification, the distribution of points becomes sparser, leading to less-precise approximations. In computational reality, when working with finite sets, self-similarity is only visible over a few scales. This phenomenon is illustrated later in Figure 6.6, where the self-similarity is clear across two distinct scales, but beyond that, the distribution of points begins to thin out.

6.1.1 A Persistent Homology Approach

The box-counting method and persistent homology share a commonality in considering multiple length scales. Consequently, persistent homology offers a method to compute the fractal dimension of data sampled from a fractal. The roots of this method lie in the *minimal spanning tree* (MST) concept [98, 99]. An MST is a structure that connects all data points with edges such that the combined length of these edges is minimal. The MST approach was validated as an effective means to determine the fractal dimension of a set [100]. Subsequent studies have highlighted the equivalence of the MST method and the use of the zeroth homology group in calculating fractal dimensions [100, 101, 102, 103]. However, an advantage of persistent homology over the MST method is its ability to generalise to higher-order homology groups, providing a more abstract formulation of the fractal dimension.

Often when analysing the persistent homology, the smaller intervals are discarded as noise, as this means that this specific feature only persists for a short while. However, in determining the fractal dimension, every persistent interval contains valuable information. The goal is to deduce the fractal dimension of a sample space X_n from X, where the number of points $n \to \infty$ and observe how the persistence interval lengths fluctuate across each homology group. Notably, the shorter intervals provide a good measure for the localised geometry present in the sample.

To calculate the fractal dimension from the persistent homology, compute the sum of the persistence interval lengths for a specific homology dimension i, each with an exponential weighting $\alpha > 0$, defined as,

$$E_\alpha^i(X_n) = \sum_{(\varepsilon_{\text{birth}}, \varepsilon_{\text{death}}) \in \mathrm{PH}_i(X_n)} (\varepsilon_{\text{death}} - \varepsilon_{\text{birth}})^\alpha \tag{6.3}$$

where X_n represents the space of n points sampled from X, most interestingly, one that exhibits fractal behaviour. The term $\mathrm{PH}_i(X_n)$ represents the i^{th} dimension persistent homology group of X_n. The difference $\varepsilon_{\text{death}} - \varepsilon_{\text{birth}}$ is the length of an interval from $\mathrm{PH}_i(X_n)$. The alpha-weighted sum $E_\alpha^i(X_n)$ models the decay rate of topological noise. Analogous to the box-counting method, this relationship can determine the fractal dimension [104].

Definition 6.1.1. Let X_n be a bounded subset of a metric space X, a homology dimension $i \in \mathbb{N}$, and a persistent interval weight $\alpha > 0$. The fractal dimension deduced via persistent homology is given by,

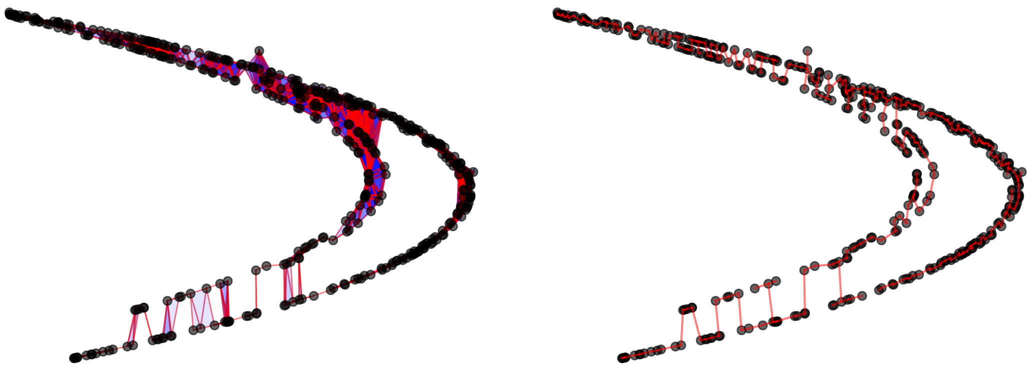
$$\dim_{\mathrm{PH}_i^\alpha}(X_n) = \frac{\alpha}{1-\beta}, \tag{6.4}$$

where,

$$\beta = \lim_{n\to\infty} \sup \frac{\log(\mathbb{E}(E_\alpha^i(X_n)))}{\log(n)} \tag{6.5}$$

In this context, sup is the supremum, denoting the largest value in the set. The operator $\mathbb{E}$ represents the expected value of a random variable. Finally, $E_\alpha^i(X_n)$ is the alpha-weighted cumulative sum of persistence intervals. The manifold's fractal dimension is calculable if $E_\alpha^i(X_n)$ scales with $n^{\frac{d-\alpha}{d}}$ [102, 104].

When $i = 0$, which refers to the calculation over the connectedness homology group, the persistence intervals are equivalent to the lengths of the edges in the MST. The MST approach formulates an analogous solution in terms of graph theory. In this



(a) VR Complex (b) MST

Figure 6.2: Hénon attractor constructed with the two methods of calculating fractal dimension.

setting, V represents the set of *vertices*, while E denotes the set of *edges* linking these vertices. Two vertices are called *connected* if a *path* connects them. A *path* describes the continuous linkage of adjacent edges from one vertex to another, allowing for unimpeded traversal between vertices that lie on a path. When edges form a closed loop, the structure is called a *circuit.* A graph without a circuit is called a *tree.*

The *minimal spanning tree* connects all vertices whilst ensuring the total edge lengths are minimal. Thanks to highly-optimised algorithms for MST computation, it is feasible to achieve efficient outcomes when calculating fractal dimensions [104], which are mathematically identical to ones from the 0^{th}-dimension homology group. By examining the edge lengths within an MST, it is possible to estimate the dimension of the manifold from which these vertices originate. However, the MST corresponds solely to the zeroth homology group. For insights using higher-dimensional homological features, slower, but more informative persistence algorithms are necessary. Figure 6.2 displays the VR complex at the smallest ε for which $\beta_0 = 1$, along with the MST for an attractor. This figure shows that the MST is a subset of the simplicial complex, focusing solely on the edges that first appear as the ε-neighbourhoods expand.

6.2 Attractors

Attractors often naturally arise in the phase space of many nonlinear dynamical systems. Attractors represent the state to which a dissipative dynamical system converges over time, regardless of the initial conditions. If an attractor exhibits a fractal structure it is termed *strange* and can be derived from chaotic nonlinear deterministic systems. Being geometric entities in Euclidean space, strange attractors allow for analyses using a Euclidean metric and, consequently, topological data analysis. Given the strange attractors' fractal nature, the fractal dimensions of strange attractors can be determined via topological approaches.

The three strange attractors examined in this study are frequently encountered in engineering, with each offering unique insights. The *Hénon* attractor is two-dimensional and known for its computational efficiency because of its iterative nature. In contrast, the *Lorenz* and *Rössler* attractors are three-dimensional and require computational solutions of *ordinary differential equations.*

Dynamical systems that display strange attractors, stemming from chaotic systems, inherently demonstrate a high sensitivity to initial conditions. Even minuscule changes can lead to dramatically different outcomes, a phenomenon known as the *butterfly effect.* For this reason, the attractors discussed in this section consider constant parameters. The specific values are the ones most prevalent within the literature, because of the intriguing properties these values display.

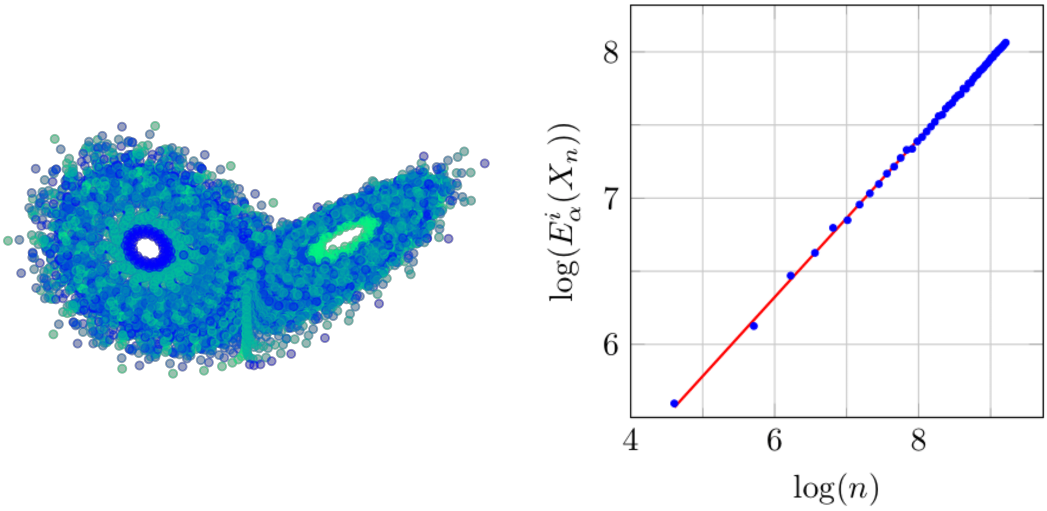
In all the provided examples, the exponential weighting is set to $\alpha = 1$. However, this is a hyperparameter in the analysis, where varying α can give better results for a fewer number of samples [104]. Additionally, only the zeroth dimension persistent homology features are used, to make use of the MST approach. The code for these calculations originates from Schweinhart's research [104, 105, 106].

6.2.1 Lorenz Attractor

The Lorenz attractor was first discovered by Lorenz when studying non-periodic turbulent flows [107], and is defined by the differential equations,

$$\begin{cases} \dot{x} = -\sigma \\ \dot{y} = \rho u - v - uw \\ \dot{z} = -\beta w + uv \end{cases} \tag{6.6}$$

where the parameters used in this work are $\rho = 28$, $\sigma = 10$, and $\beta = \frac{8}{3}$. This specific configuration for the Lorenz attractor shows two attracting points, each shown as a hole in the attractor. The presence of these holes intrinsically suggests an interesting topology, therefore further prompting some traditional topological



(a) Dense sampling (b) log-log plot

Figure 6.3: Dense Lorenz attractor, sed for embedding dimension calculations

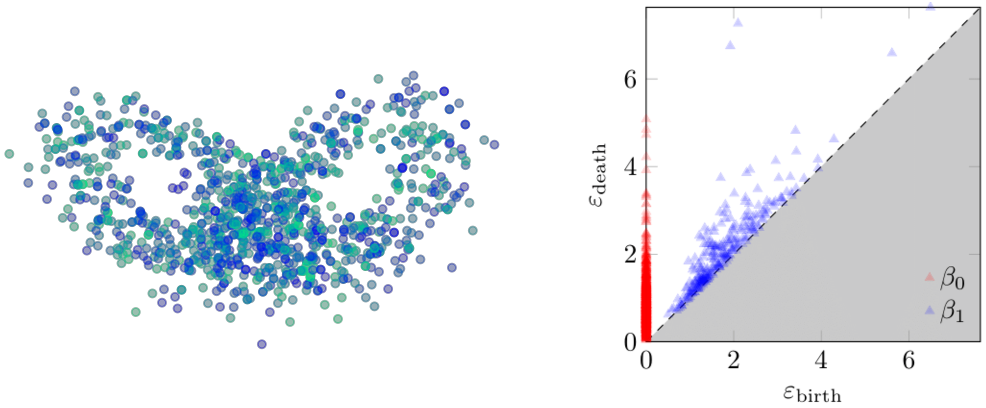
.

analysis. Furthermore, the Lorenz attractor is strange, displaying a Cantor-set-like behaviour over its cross sections [108].

To determine the fractal dimension of the Lorenz attractor, a large number of points must be sampled from the attractor, resulting in a dense embedding that reveals finer details, including the Cantor-like fractal cross sections. For this dense embedding, 10,000 points were used. The equivalence between the MST and β_0 persistent homology allows for the use of a significantly larger number of points in calculating the fractal dimension [104, 105, 106].

Since the Lorenz attractor is the first practical application of calculating the fractal dimension via persistent homology in this work, a full walkthrough is undertaken for clarity.

1. Obtain a dense point embedding of the attractor, in this case with 10000 points.
2. Randomly subsample this densely sampled attractor at logarithmically-spaced steps, where the number of points n in each subsample corresponds to the x-axis of Figure 6.3 (b).
3. For each subsample, compute the alpha-weighted sum $E^i_\alpha(X_n)$. The alpha-weighted sum then corresponds to the y-axis of Figure 6.3 (b).



(a) Sparse sampling

(b) Persistence homology

Figure 6.4: Sparse Lorenz attractor, used for persistence homology calculations.

4. Take logarithms of the values obtained in (2) and (3), giving a log-log plot of $\log(n)$ vs $\log(E^i_\alpha(X_n))$. When plotted in this manner, the data points can be adequately modelled by a linear approximation of the form $y = mx + c$.
5. In this log-log plot, the gradient m of the straight line is equivalent to β in equation (6.5).
6. Using this information, the persistent homology fractal dimension is estimated via equation (6.4).
7. Repeat this process for different random subsamples to obtain statistics regarding the fractal dimension estimation.

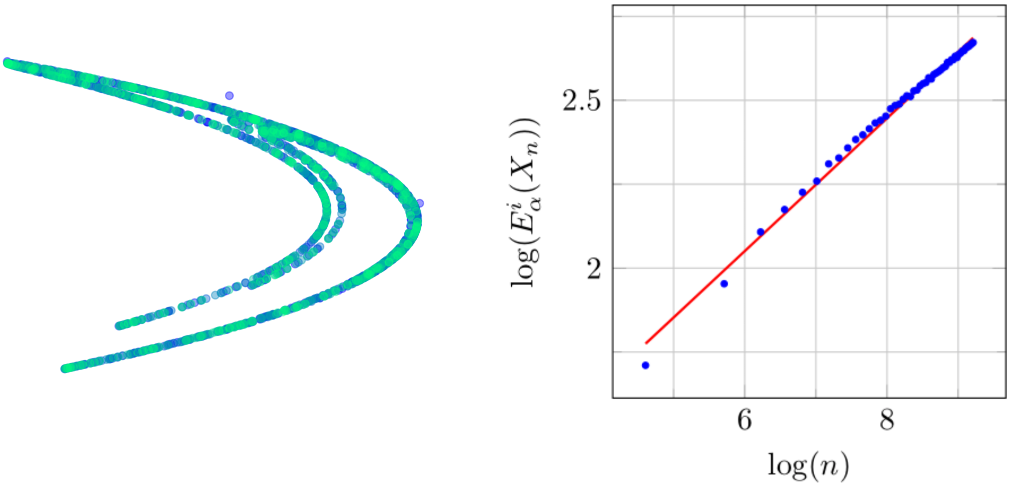
The fractal dimension of the Lorenz attractor for this specific parameter set, as determined by an alternative method, was 2.063 [108]. Employing the topological approach on the dense sampling of the Lorenz attractor gave an approximation to the fractal dimension of 2.0826 ± 0.03603. This dimension is derived from the gradient of the log-log plot comparing the scaling behaviour of the MST edge-length sum with the number of points sampled.

To capture the global topology of the Lorenz attractor using a limited number of points suitable for persistent homology calculations, a sparser Lorenz attractor is sampled with 1000 points, as shown in Figure 6.4. With the point count fixed at 1000, samples were obtained over the time interval $t = [0,\, 2500]\,\mathrm{s}$, resulting in large time gaps between samples. This wide gap ensures a uniform and distributed representation over the long-scale dynamics of the attractor. In essence, this approach models the attractor's overall form with fewer points, focussing on capturing its overarching shape. This sparse form over the long-lasting dynamics of the attractor is ideal for calculating the global topology of the attractor.

The persistence diagram of the sparse Lorenz attractor is displayed in Figure 6.4 (b). The red points, denoting the β_0 persistence, show no large differences, implying that the Lorenz attractor comprises a single connected component. This finding aligns with expectations for a structure emerging from a continuous-time sampling process.

Regarding the first homology group, the diagram shows a concentration of topological noise around $y = x$. Nevertheless, two distinct features stand out, with the coordinates (1.97, 9.17) and (2.59, 7.29). These two persistent β_1 features represent the holes present at the two attracting points within the manifold, and are evident in Figure 6.4 (a).

6.2.2 Hénon Attractor

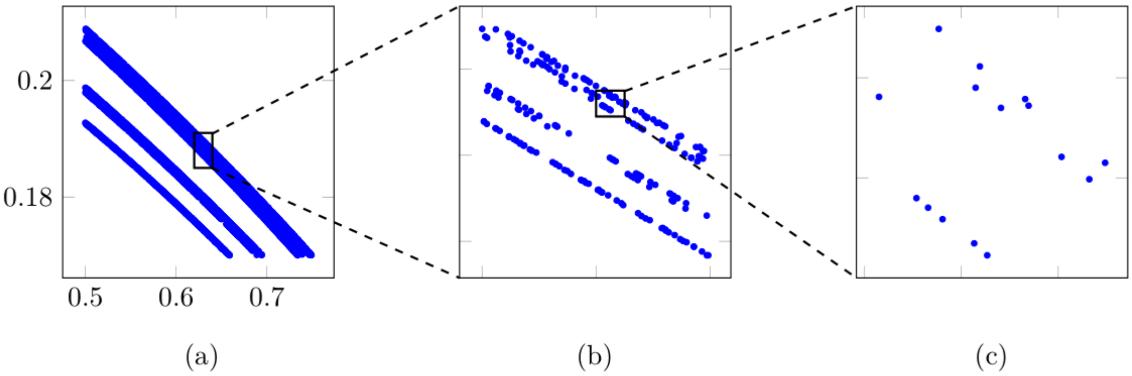


(a) Dense sampling

(b) log-log plot

Figure 6.5: Dense Hénon attractor, used for embedding dimension calculations.

The Hénon attractor, a two-dimensional quadratic map with a constant Jacobian, was originally introduced as a simplified discrete representation of the Lorenz system. Calculated via iteration rather than solving ODEs, the Hénon attractor computations are fast compared to other attractors. Consequently, this attractor has become



(a) (b) (c)

Figure 6.6: Points from the Hénon attractor sampled at three distinct scales, revealing the Cantor cross-section. (a) $x \in [0.50, 0.75]$, $y \in [0.15, 0.21]$, (b) $x \in [0.620, 0.640]$, $y \in [0.185, 0.191]$, (c) $x \in [0.6300, 0.6325]$, $y \in [0.1889, 0.1895]$.

a common object for study in dynamical systems when wanting to generate large numbers of points [109]. The Hénon attractor used in this work is defined by,

$$x_{n+1} = 1 - ax_n^2 + y_n \tag{6.7}$$

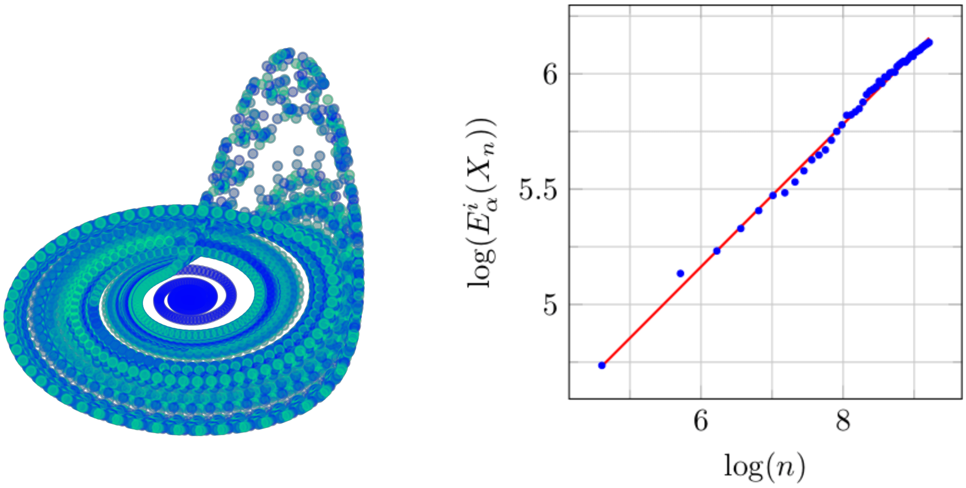
$$y_{n+1} = bx_n \tag{6.8}$$

In this case, the well-studied parameters $a = 1.4$, $b = 0.3$, and an initial point of (0.1, 0.3) were chosen, which result in a convergent solution.

After sampling 10000 points from the Hénon attractor, the fractal dimension was determined using the persistent homology approach, yielding a value of 1.2558 ± 0.04476, with the log-log plot present in Figure 6.5 (b). For comparison, a previously-calculated value using a different fractal dimension calculation procedure, determined the attractor's dimension to be 1.26 [110].

To visually uncover the fractal structure present within strange attractors, the Hénon attractor's rapid computation allows it to generate a sufficient number of points, revealing its multiscale fractal features [109]. To demonstrate the Cantor-like traversal structure, 10,000,000 samples were taken from the attractor. Figure 6.6 shows the self-similarity over progressive zooms into sections of the attractor, maintaining a consistent structure across scales. The dimensions of (a), (b), and (c) in Figure 6.6 are 1.233, 1.245, and 1.526 respectively; indicating that, as the data become more sparse, this method of estimating fractal dimension becomes less precise at reduced length scales.

6.2.3 Rössler Attractor



(a) Dense sampling

(b) log-log plot

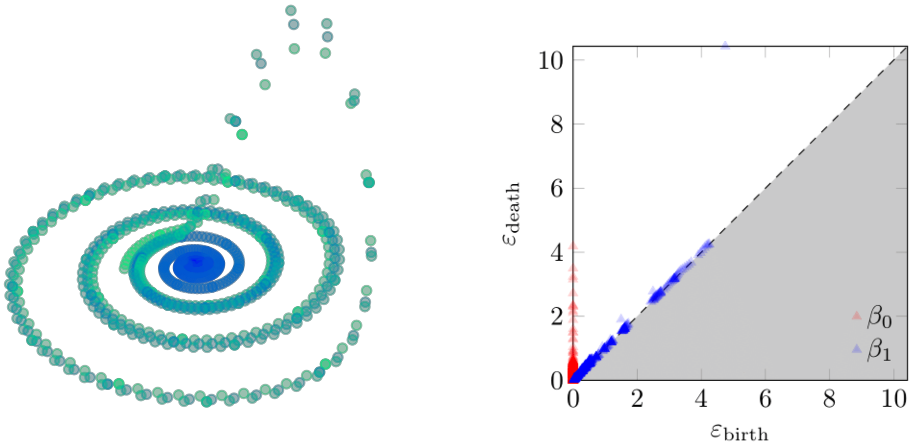
Figure 6.7: Rössler attractor with 10000 points, used to calculate the embedding dimension.

The final strange attractor explored in this work is the Rössler attractor, which was first formulated in [111, 112]. The Rössler attractor, much like the Lorenz attractor, is present in the phase space of the solutions of the system of differential equations,

$$\begin{cases} \dot{x} = -y - z \\ \dot{y} = x + ay \\ \dot{z} = b + z(x - c) \end{cases} \tag{6.9}$$

For this study, the system parameters were set at $a = 0.2$, $b = 0.2$, $c = 5.7$, starting from the initial point $p_0 = (0, 0, 0)$. With 10000 points sampled from the Rössler attractor, the dimension estimate was found to be 2.025 ± 0.0246, with the log-log plot shown in Figure 6.7 (b). This persistent homology fractal dimension aligns closely with a previously-determined fractal dimension of 2.0160 with these parameters [113].

In order to study the persistent homology, of the Rössler attractor, a sparse embedding of 1000 samples is used. From the persistence diagram in Figure 6.8 (a), it can be determined that the Rössler attractor forms a connected manifold, as all the β_0 features show minimal variation. The small range in the β_0 features arises



(a) Rössler attractor

(b) Persistence homology

Figure 6.8: Rössler attractor, with 1000 points, used to analyse the topology.

from points along the attractor's 'flick'. Given the brief time interval during which the 'flick' occurs, a constant time-step between points results in a sparser sampling in this discrete solution of the differential equations; these features would not be present in a continuous solution.

The characteristic 'flick' in the Rössler attractor is essentially a loop, giving rise to interesting H_1 information, shown in Figure 6.8. In this case, the topological noise is squashed relatively close to $y = x$, being overshadowed by the very persistent β_1 feature which represents the attractor's 'flick'. With coordinates (4.7, 10.4) and a persistence length of 5.7, this feature is unmistakably a prominent feature of the attractor.

6.3 Best Topological Reconstruction

This study compares the topologies of reconstructed phase spaces – derived from a 1D time series of the attractor – with the original point cloud. Persistent homology is used to evaluate the best embedding delay, resulting in the most topologically-similar reconstruction via time-delay embedding. The goal is to achieve a persistent homology from the reconstructed attractor that gives the smallest Wasserstein distance, hence the most topologically similar.

Takens' theorem shows that under certain circumstances, reconstructed attractors

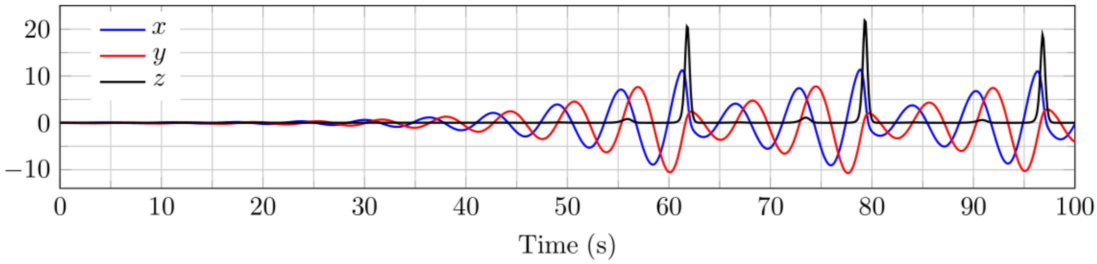


Figure 6.9: Individual Rössler axes as time series.

from the time-delay embedding can be homeomorphic to the original attractor [78]. Skraba et al [80], then showed that persistent homology could be used with the time-delay embedding to yield useful topological results. In usual applications, for the reconstructed attractor to be topologically identical to the original, Takens proved that the embedding dimension needs to be sufficiently large. More recent empirical studies have shown that the embedding dimension should typically be at least double that of the original attractor [114].

For this work, the Rössler attractor is reconstructed from its y-coordinates. The individual series relating to each axis are plotted against time in Figure 6.9. The x and y time series share notable similarities, with their main distinction being a constant offset. Both the x and y series show a constant increase until the sudden rise in the z-axis, which produces the 'flick' in the phase space, at this point the magnitudes of x and y coordinates are reduced, illustrating the attracting nature of this dynamic. This process occurs periodically in both the x and y series. However, when isolating the z-time series, it becomes clear that it does not contain enough information to reconstruct the original attractor, as shown in Figure 6.9.

Reconstructing attractors requires careful consideration of the delay. If the delay is too small, the embedded outcome will closely resemble a straight line. For example, when $\alpha = 0$, the time-delay embedding would plot the data points along the straight-diagonal line as $x = y = z$. Therefore, relatively-small α values will show topologies minimally evolved from a straight line. Hence, by increasing the delay, the prominence of the homological features in the reconstruction is also increased. On the other hand, using too-large α values will give incoherent results, as the lag is not considered local. Using persistence homology and the Wasserstein distance to topologically measure the reconstructed attractors identifies the most appropriate α, giving the most topological likeness to the original attractor.

Computing multiple reconstructions over a range of delays, along with their corre-

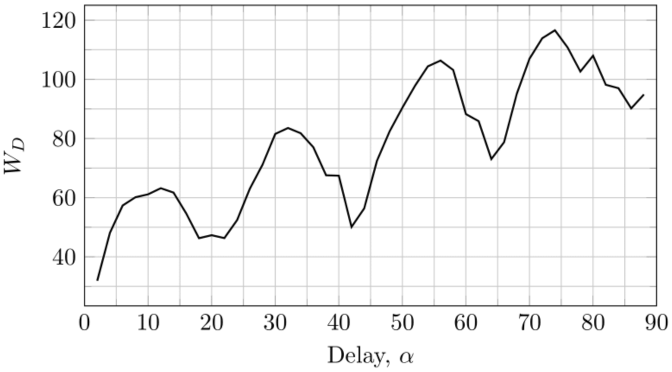


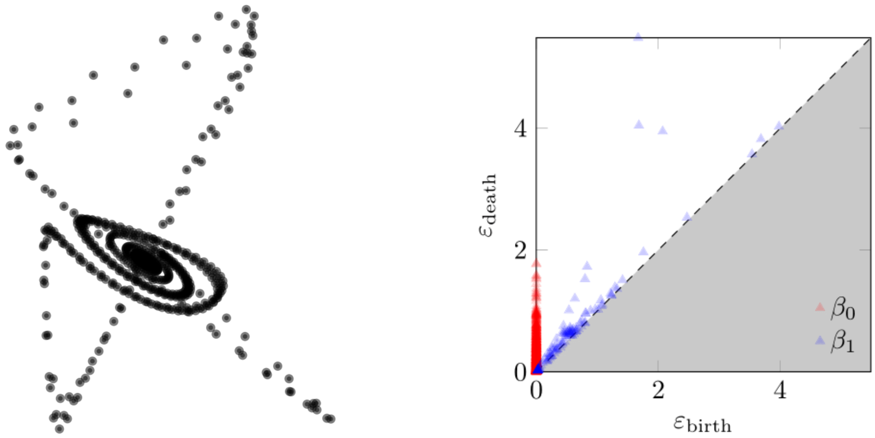
Figure 6.10: The Wasserstein distance between the complete Rössler attractor and its time-delay embedding reconstruction over a range of delays, α.

sponding persistent homologies, offers a method to identify the optimal topological reconstruction. The reconstruction that gives the smallest Wasserstein distance to the original attractor is by definition the most topologically alike. Figure 6.10 shows the Wasserstein distances between the original Rössler attractor and reconstructed attractors as α varies. The delay that yields the smallest Wasserstein distance indicates that the topologies of the original and the reconstructed attractors closely align. Consequently, such an α is deemed optimal in this case.

Figure 6.10 shows a clear periodic pattern emerging between the Wasserstein distances and α, suggesting that the time-delay embedding topologies periodically align with the original attractor. Additionally, as the delay increases, a linear increase in the Wasserstein distance is also evident. This linear-increasing trend arises as a larger α reduces the locality of the lag, introducing artefacts in the reconstruction.

(a) $\alpha = 1$ (b) $\alpha = 57$

Figure 6.11: Examples of bad Rössler attractor reconstructions.



(a) $\alpha = 20$

(b) Persistence homology

Figure 6.12: The optimally reconstructed Rössler attractor, based on the Wasserstein distance.

The topology of the reconstructed attractor periodically aligns with the original, as shown by the troughs in Figure 6.10. However, subsequent troughs after the first indicate a phase offset by an integer number of periods.

In actuality, the smallest Wasserstein distance in Figure 6.10 is when $\alpha = 1$. This low Wasserstein distance stems from the lack of homological features in the reconstruction, as the topology is too similar to the diagonal line, and the Wasserstein distance is dominated by the cost of matching the persistent homologies to the diagonal in the persistence diagram. The plot of the reconstructed phase space at $\alpha = 1$ is shown in Figure 6.11 (a). Therefore, only α values beyond the first peak at $\alpha = 13$ are considered, to ensure the formation of a nontrivial topology. On the other hand, using a too-large value for the delay gives a deformed manifold as the delayed coordinates are no longer local, revealing artefacts in the embedding as shown in Figure 6.11 (b).

From this reasoning, the best topological reconstruction using persistent homology and the Wasserstein distance corresponds to $\alpha = 19$, evident from the lowest point in the first trough. When using topological data analysis, there is an inherent metric flavour because of the Vietoris-Rips complex formation. Consequently, the time-delay embedding into 3D with $\alpha = 19$ shares a geometric resemblance to the original attractor as shown in Figure 6.12 (a). Regarding geometry, a reconstruction from the y-coordinates cannot capture a 'flick' as pronounced as the original attractor, as the maximum value in the y-axis is 7.78, whereas the maximum value in the z-axis is 21.80. Consequently, the persistent homology feature associated with the flick will always be less defined than in the original attractor, as illustrated in Figure 6.12 (b).

6.4 Conclusion

The work in this chapter, though perhaps less impactful or novel than others, plays a vital role in framing and constructing the overarching narrative of the thesis.

Firstly, this chapter presented an established method, demonstrating the utility of persistent homology in determining an attractor's fractal dimension. By this procedure, the chapter shows that even the small-scale persistent homology features, often labelled as topological noise, contain valuable information. Such insights set the stage for later sections of the thesis, where analyses focus on the persistent homology of time series associated with Gaussian white noise, inherently producing significant amounts of topological noise in their persistent homologies.

Secondly, but most importantly, this chapter emphasises the power of time-delay embeddings and Takens' theorem for creating reconstructed spaces that embody the dynamics of the original system. A new persistent-homology-oriented method for determining the delay that best represents an attractor for reconstruction was presented. Central to the overarching theme of this thesis, the reconstructed topology via time-delay embedding is reminiscent of the original, even when derived from a single 1D time series. This application of time-delay embeddings for space reconstructions from a 1D time series is a recurrent theme in later sections of this thesis.

Chapter 7

ASSESSING COINTEGRATION USING TDA

This chapter further explores the Z24 bridge dataset, focussing on the application of cointegration to its natural frequencies. Cointegration is used to eliminate correlated variations between nonstationary time series, yielding a stationary *residual*. At its core, in an SHM context, cointegration aims to account for nonstationarities originating from EOVs without factoring in damage effects. The idea is to devise a scheme that addresses these EOV nonstationarities, resulting in a stationary residual under such conditions. Since this cointegration scheme is only trained to remove EOV nonstationarities, any anomaly – like damage – that deviates from this setup would make the expected stationary residual nonstationary, implying novelty within the data.

For datasets where nonstationary time series are linearly correlated, this problem is well-defined, thanks in part to ideas adapted from the economics literature. However, a notable challenge arises in the context of the Z24: while ω_1, ω_3 and ω_4 show a linear relationship with each other, ω_2 exhibits nonlinear behaviour; as shown in Figure 7.1. Consequently, not all natural frequencies correlate linearly, making traditional linear cointegration unsuitable.

The Z24 introduces the need for a nonlinear cointegration approach. To do so, one can detrend via a Gaussian Process's (GP) mean function, where the GP has been trained on the Z24's data, removing the baseline nonstationarity [115]. But even

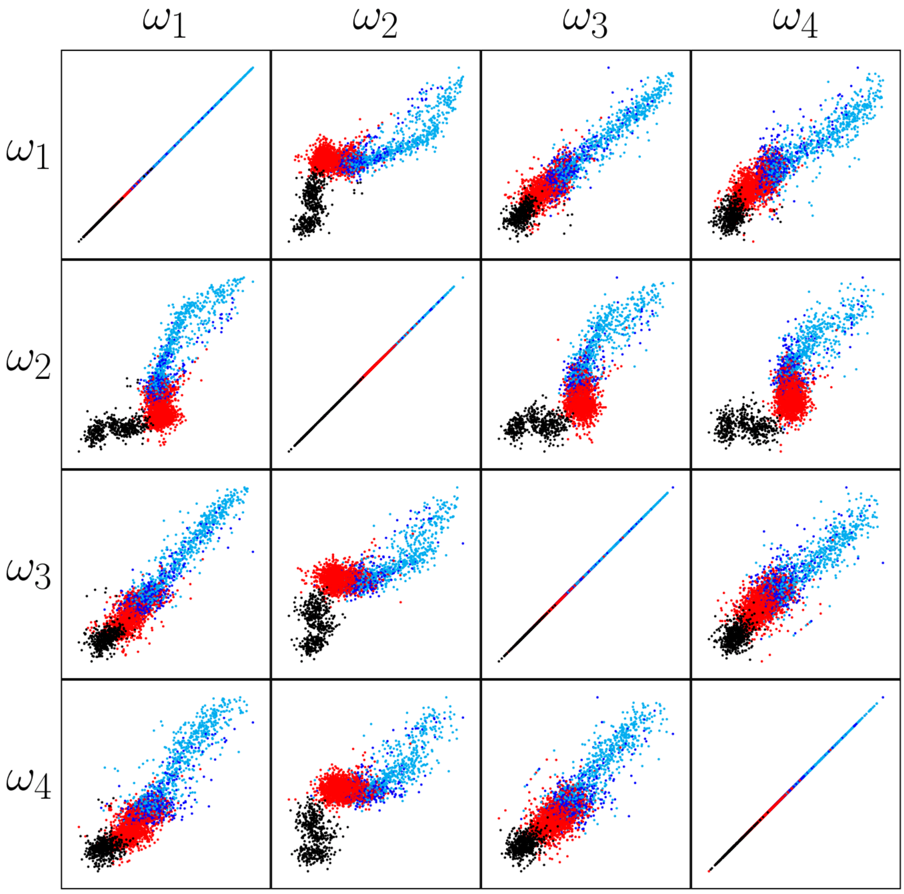


Figure 7.1: Correlations of all the Z24 natural frequencies, showing pronounced nonlinearity for ω_2

.

here, challenges arise because of the various behaviours exhibited in different regions over the year timeframe that the data were collected. The nonlinear relation occurs in the freezing region, if GPs are not tailored to this region, they inadequately model the nonlinear temperature effects, causing them to break through into the residual series.

The residual series can be embedded into higher dimensions using a time-delay embedding, and since residual series are assumed to be sampled from a white noise process, their n-dimensional embeddings are likewise expected to be n-dimensional Gaussian clusters. The time-delay embedding emphasises the breakthrough features in the residual series after applying nonlinear cointegration, as shapes in the higher-dimensional embedding. These shapes can then be quantitatively described using TDA. Consequently, if a particular feature or anomaly emerges in the residual series, it manifests as a specific shape in the embedding space. TDA can then be used to assess which shapes are present in the embedded residual, assessing the efficacy of the cointegration. In a nutshell, this process acts as a method to quantify and discern between features that remain in the residual after using a cointegration scheme.

Before getting into the topological data analysis, a quick introduction to cointegration theory, both linear and nonlinear, is supplied. The inability of linear cointegration to handle problems such as the Z24 is then highlighted. Thereafter, moving on to topological-based reasoning when applied to the embedded residual series from cointegration.

7.1 Primer on Linear Cointegration

This section provides a brief overview of the linear cointegration procedure and its key aspects. Cointegration seeks to remove latent trends present in multiple nonstationary time series by considering their weighted sum. In SHM, cointegration is used to mitigate benign effects in data arising from EOVs, such as temperature and humidity, that may alter damage-sensitive variables.

Over the past decade, cointegration has been applied in SHM to remove the effects of EOVs shared over multiple similar time series, where the first application was in removing EOVs from the DAMASCOS data set [116], proving its use in novelty detection. For greater detail on cointegration theory, readers are redirected to further references [117, 118, 119].

In this research, the stationarity of time series is assessed using the *Augmented Dickey-Fuller* (ADF) test. If the time series exhibit the same degree of nonstationarity via the ADF test, they are then passed into the *Johansen procedure.* The Johansen procedure outputs the *cointegrating vectors*, which encode the coefficients for the linear combination of the nonstationary time series, resulting in the best approximation of a stationary residual.

In search of brevity, some of the more foundational time-series analysis techniques are omitted in the main matter. For ideas regarding *stationarity*, *differencing*, and *Autoregressive Moving Average* (ARMA) models, clarification is presented in Appendix E.

7.1.1 Order of Integration

Traditional cointegration requires that all nonstationary variables under analysis be nonstationary to the same degree; that is, they are *cointegrated.* This characteristic is assessed by ensuring the order of integration for all the nonstationary time series is consistent. The order of integration is determined by the number of *differences* required to transform a time series from nonstationary to stationary. For instance, if four differences are required to reduce the time series to stationary, it is integrated of order four, denoted $y \sim I(4)$. The order of integration here mirrors that of the iterated integrals in Chapter 2, where the order of integration of a time series can be visualised as a number of integrals on a stationary time series.

Time series are said to be *cointegrated* when they share the same order of integration. For a collection of time series with this trait, there exists at least one linear combination that yields a stationary series, known as the *residual series.* The combination of time series that results in a stationary time series is encoded in the *cointegrating vectors.* For a matrix of nonstationary time series $Y(t)$, these are cointegrated if,

$$\epsilon(t) = \beta^T Y(t) \tag{7.1}$$

where β is the matrix of cointegrating vectors and $\epsilon(t)$ are the stationary residual time series.

7.1.2 Augmented Dickey-Fuller Test

The first step in cointegration is ensuring that all time series are integrated to the same order. In doing so, the time series must be tested for stationarity after each differencing. If a time series is deemed statistically stationary, the number of differences required to reach this state is its order of integration. The *Augmented Dickey-Fuller* test [118, 120] is one such test statistic.

To perform the ADF test, time series are modelled with *error-correction models*, using a least-squares approach, shown below [116],

$$\nabla y_i = \rho y_{i-1} + \sum b_j \nabla y_{i-j} + \epsilon_i \tag{7.2}$$

where ∇ is the difference operator, and ϵ_i is the time-series residual, assumed to be sampled from a white-noise process. After differencing a nonstationary time series, the ADF test is employed to confirm stationarity. This test revolves around the values of ρ in the model depicted in equation (7.2), forming a test statistic,

$$t_p = \frac{\hat{\rho}}{\sigma_\rho} \tag{7.3}$$

where $\hat{\rho}$ is the least squares estimate of ρ, and σ_ρ being the variance of the estimate. The ratio of $\hat{\rho}$ and σ_p give the test statistic t_p, which is compared against reference values [118, 120].

The ADF test essentially checks for the proximity of ρ to 0. If $\rho = 0$, the time series has a unit root and is inherently nonstationary [116]. Furthermore, if the test statistic determines that ρ is statistically close enough to 0, the null hypothesis is rejected, suggesting that the time series is nonstationary. This procedure is repeated until the null hypothesis is accepted. The number of differences required for the time series to become stationary is its order of integration.

7.1.3 Johansen Procedure

In the context of SHM, there is an emphasis on determining the cointegrating vectors [121]. These vectors are derived via the *Johansen procedure*, and encode the coefficients of the sum to determine the residuals; which are subsequently analysed with TDA in this study. The Johansen procedure provides a method of determining multiple cointegrating vectors for m nonstationary time series which are cointegrated with $y_i \sim I(1)$ [122]. Essentially, the Johansen procedure is an eigenvalue problem, with the most stationary residual linked to the cointegrating vector associated with the largest eigenvalue.

The set of input time series is modelled by a *vector auto-regressive* (VAR) model of the form [123],

$$\begin{aligned} Y_t &= \mu_t + \Phi_1 Y_{t-1} + \cdots + \Phi_p Y_{t-p} + \epsilon_t \\ &= \mu_t + \epsilon_t + \sum_{i=1}^{p} \Phi_i Y_{t-i} \end{aligned} \tag{7.4}$$

where Y_t is an m-dimensional vector representing the m input time series, and Φ_i are $m \times m$ coefficient matrices for each lag i; μ_t is an m-dimensional constant vector, and ϵ_t is an m-dimensional vector Gaussian noise series, $\epsilon_t \sim \mathcal{N}(0, \Omega)$, representing the errors of fitting the VAR to data. The quality of the VAR model is assessed by considering the Akaike information criterion [124]. From this VAR, a corresponding *vector error correction model* (VECM) can be formed by making the substitutions, such that,

$$Y_{t-i} = \nabla Y_{t-i} + Y_{t-i-1},\ \forall i \in 0,\ 1,\ \dots,\ p \tag{7.5}$$

and then further rearranging into,

$$\nabla Y_t = \mu_t + \epsilon_t + \Pi Y_{t-1} + \sum_{i=1}^{p-1} \Psi_i \nabla Y_{t-i} \tag{7.6}$$

where $\Pi = \Phi_1 + \cdots + \Phi_p - I$ is the adjustment regarding long-run information. On the other hand, the local information is captured by $\Psi_j = -(\Phi_{j+1} + \cdots + \Phi_p)$. Notably, the matrix Π is of significant importance as it holds information related to the cointegrating vectors.

$$\Pi = \alpha \beta^T \tag{7.7}$$

Importantly, α is the $(m \times r)$ *adjustment matrix*. However, of greater significance is β, which is an $(m \times r)$ matrix of cointegrating vectors, where r is the rank of the matrix Π, and is also the number of attainable cointegrating vectors.

To determine the optimal cointegrating vectors, which yield the most stationary residuals, a log-likelihood regression scheme is implemented. The aim is to then determine the parameters which maximise the log-likelihood relation, consequently, the most stationary residuals are obtained [121]. The details of this process are omitted for brevity; however, readers can refer to [121, 122, 123] for a more comprehensive understanding. Fundamentally, the likelihood-optimisation process simplifies down to solving an eigenvalue problem of the form,

$$|\Lambda S_{11} - S_{10} S_{00}^{-1} S_{01}| = 0 \tag{7.8}$$

where S_{01} is the covariance matrix between ΔY_t and Y_{t-1}, S_{11} is the covariance matrix of Y_{t-1}, and Λ is the diagonal matrix of eigenvalues, λ_i. The matrix of cointegrating vectors β, is composed of the eigenvectors from equation (7.8), such that,

$$\beta = \begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_r \end{bmatrix} \tag{7.9}$$

where each column of the matrix represents a cointegrating vector. It is known that β_1, which is associated with the largest eigenvector, gives the most stationary residual [123]. In this thesis, all the residual series are analysed, which are obtained by projecting the m input nonstationary time series with β,

$$\epsilon = Y_t \cdot \beta \tag{7.10}$$

Herein, the residual ϵ forms the basis of capturing all the information that has 'slipped through' when performing cointegration. If information finds its way into ϵ, it highlights the limitations of cointegration in filtering out these features. Such information seeping through into the residual is then analysed using TDA.

7.2 Nonlinear Cointegration

For the Z24 data, previous research [121] has shown that the four natural frequencies are all nonstationary, and all exhibit stationarity after one difference with 95% confidence. This means that the Z24 natural frequencies are suitable for the Johansen procedure. However, the natural frequencies are not linearly correlated, as shown in Figure 7.1. As a result, a linear sum cannot remove their nonlinear relationship. Consequently, an alternative cointegration approach is necessary in such cases – one capable of nonlinear relations between nonstationary time series; as observed with the Z24.

While numerous nonlinear cointegration schemes have been suggested in the literature [115, 125], this study focusses solely on detrending with Gaussian Process (GP) regressions [115]. In this application, GPs are used to estimate the *cointegrating function*, which captures details regarding the perturbations in the system from EOVs.

7.2.1 Gaussian Processes

In this work, a succinct introduction to Gaussian Processes (GPs) is provided, as they serve as the method of encapsulating the cointegrating functions. However, the primary interest in this work lies with analysing the residuals, rather than digging deeply into this well-established methodology [115, 125, 126].

GPs are employed as a Bayesian curve-fitting technique. The process begins by establishing an initial curve or function, termed a *prior*. This prior is then updated based on some training data introduced into the GP regression framework [89]. Unlike curve-fitting techniques which produce deterministic solutions, GPs offer uncertainty bounds.

Conceptually, GP regression can be viewed as generalisations of Gaussian distributions, but applied to functions. Just as a Gaussian distribution is characterised by its mean, μ, and variance, σ^2, a GP regression is characterised by a mean function, $\mu(x)$, and a covariance (or kernel) function, $k(x, x')$. Here, the mean function, $\mu(x)$, offers the best prediction for an input x. Meanwhile, the kernel, $k(x, x')$, quantifies the uncertainty or correlation between two points, x and x', in the input domain [127].

Whilst in previous work [121], there was a greater emphasis on the Bayesian aspect of cointegration, by propagating confidence bounds from the covariance function; this work is primarily concerned with the mean function from GPRs, and its inability to remove all EOV trends from data.

7.2.2 Detrending with GPs

Within this work, GPs are employed to estimate a *nonlinear cointegrating function*, $f(x_t)$, which models the nonstationary behaviour of a time series y_t regarding EOVs, such that,

$$y_t = f(x_t) + \epsilon_t \tag{7.11}$$

where ϵ_t is the residual, capturing any behaviour not described by the cointegrating function.

In the case of the Z24, ω_2 is nonlinear relative to the other natural frequencies. Hence, a GP is used to model the nonlinear relationship of ω_2. This GP is trained on the linearly correlated natural frequencies: ω_1, ω_3, and ω_4. Where this relationship is denoted,

$$\omega_2^* \mid \omega_1, \omega_3, \omega_4 \sim \mathcal{GP}(\mu_{\omega_2}, k_{\omega_2}) \tag{7.12}$$

where ω_2^* is the GP's prediction of ω_2. The mean function μ_{ω_2}, gives the best approximation to ω_2 given the other natural frequencies, thereby acting as the nonlinear cointegrating function. Now, incorporating these ideas into equation (7.11), the residual series is given by,

$$\epsilon_t = \omega_2 - \mu_{\omega_2} \tag{7.13}$$

Where the mean function removes nonstationary effects from EOVs in the nonlinear ω_2, resulting in a detrended stationary residual series.

A framework for conducting a GP approach to nonlinear cointegration is then formalised via a series of steps, as outlined in [121].

Selection of Relevant Variables: Much like linear cointegration, an appropriate set of monitored variables must be selected. Ideally, these variables should share the same latent influence and be of a similar type, e.g. a set of natural frequencies or a set of cable tensions. Subsequently, the ADF test is applied to these time series, ensuring they have consistent integration order.

Training-Test Split: The time series are partitioned into training and test data sets. The training set is used to train the GPs, whereas the test set is for SHM purposes. As cointegration aims to remove nonstationary effects, the training set must contain no damage-labelled instances. The goal of nonlinear cointegration is to detrend every attribute apart from damage.

Residual Calculation: The GP is trained using the damage-free training data. The mean function from the resulting GP is subtracted from the test set, the difference being the residual. Subsequently, the ADF test is applied to the residual series to examine if the integration order has reduced. If the integration order has decreased, the common trends have been purged from the residual series.

Specific features in the data, such as the region of sustained freezing, need to be adequately represented within the training set. If not, the GP's mean function, which determines the cointegrating function, might fall short when describing these nuances. Consequently, an inadequate representation of the EOVs in extreme cases may lead to these features inadvertently remaining in the residual series post-cointegration.

To accentuate the bleed-through features, the residual series undergoes time-delay embedding, attributing the time series with nontrivial topology. The persistent homology is then used to quantify the shapes formed by the breakthrough features in the embedded manifolds.

7.3 Validity of Cointegration in Engineering

When considering the use of cointegration, particularly in the nonlinear setting, the question arises that nonlinear observables may share some states that are cointegrated, while others might not. Without complex nonlinear system identification, only the monitored variables may be tested against. For this work, ensuring that the monitored data are cointegrated is sufficient. This problem is best discussed in previous research [128] where the authors argue that if the monitored variables are adequately represented by their error corrections models, regardless of philosophical quandaries regarding the data being sampled from a true unit root process, the shared nonstationarities are eliminated over the cointegration process.

Another argument arrived at in the paper [128] is one that with a sufficiently long observation window, the observations will tend towards a stationary process, as the yearly cyclic trends will centre around a mean value. However, to understand the structural dynamics on shorter time scales, the use of nonstationary theory is required. Hence, cointegration and nonlinear cointegration are required to detrend nonstationary time series. This requirement is for the application of a swift-acting novelty detector for infrastructure, rather than relying on multiple cycles (years) of data.

7.4 Natural Frequency Time-Delay Embeddings

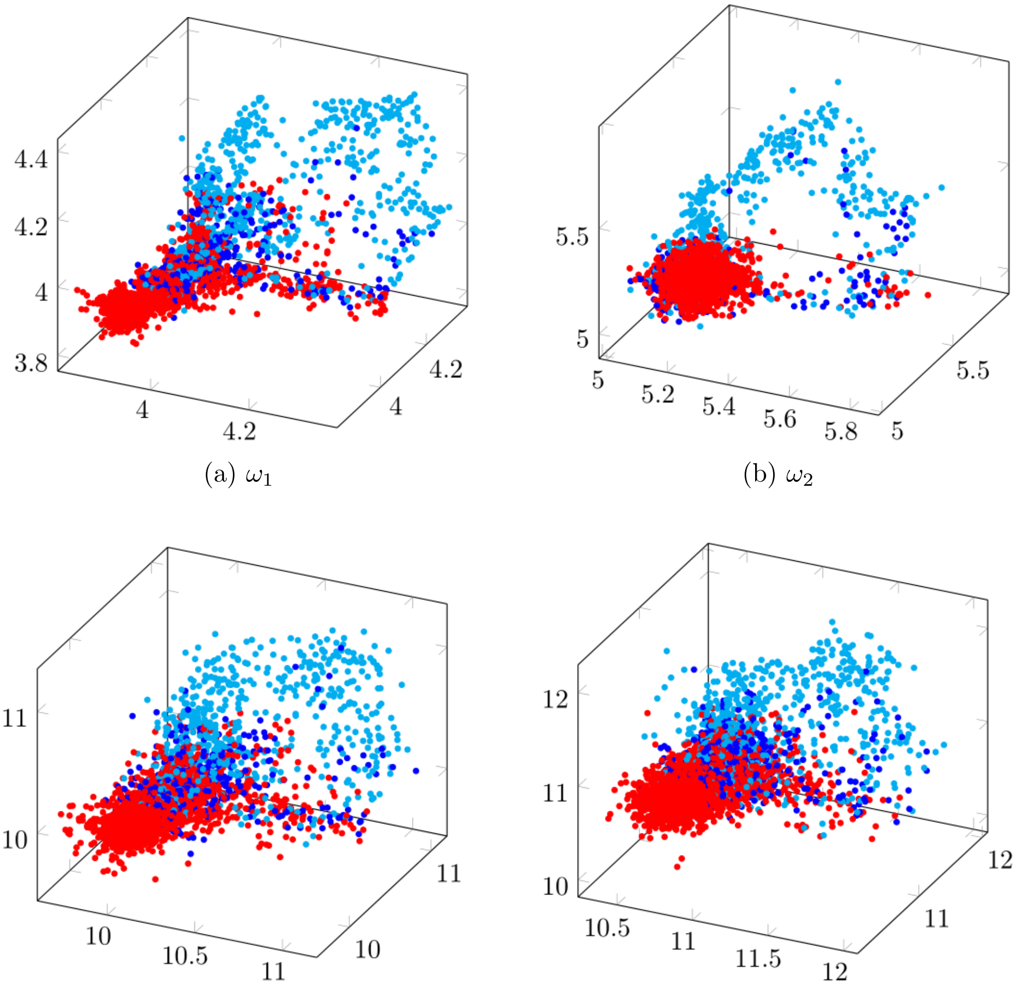
Before embedding the residuals, it is useful to first understand the shape of the unprocessed natural frequencies after time-delay embedding. Understanding the form of the unprocessed embeddings facilitates a better understanding of which information permeates into the residual series.

While both the raw and residual time series are depicted as 3D manifolds for visual clarity, this is not mandatory. Using higher dimensions could offer more freedom in representing the embedded manifolds. Additionally, a delay of $\alpha = 75$ is consistently used for all embeddings. Keeping a consistent α across all embeddings is key, as varying α would accentuate different features. This particular value was selected as it effectively provides pronounced features across all embeddings, thereby enhancing topological features for analysis with persistent homology.

The time-delay embeddings of the Z24 natural frequencies, depicted in Figure 7.2, are coloured based on the class of the unlagged state[1]. As explored in Chapter 6, the time-delay embedding causes a local shift in the data. Since this shift is a localised change and temperature variations are typically gradual over seasons, using the label

---

[1] The colouring scheme is the same as in Chapter 4, where light blue denotes freezing, dark blue represents cold, red indicates warm, and black signifies damage.



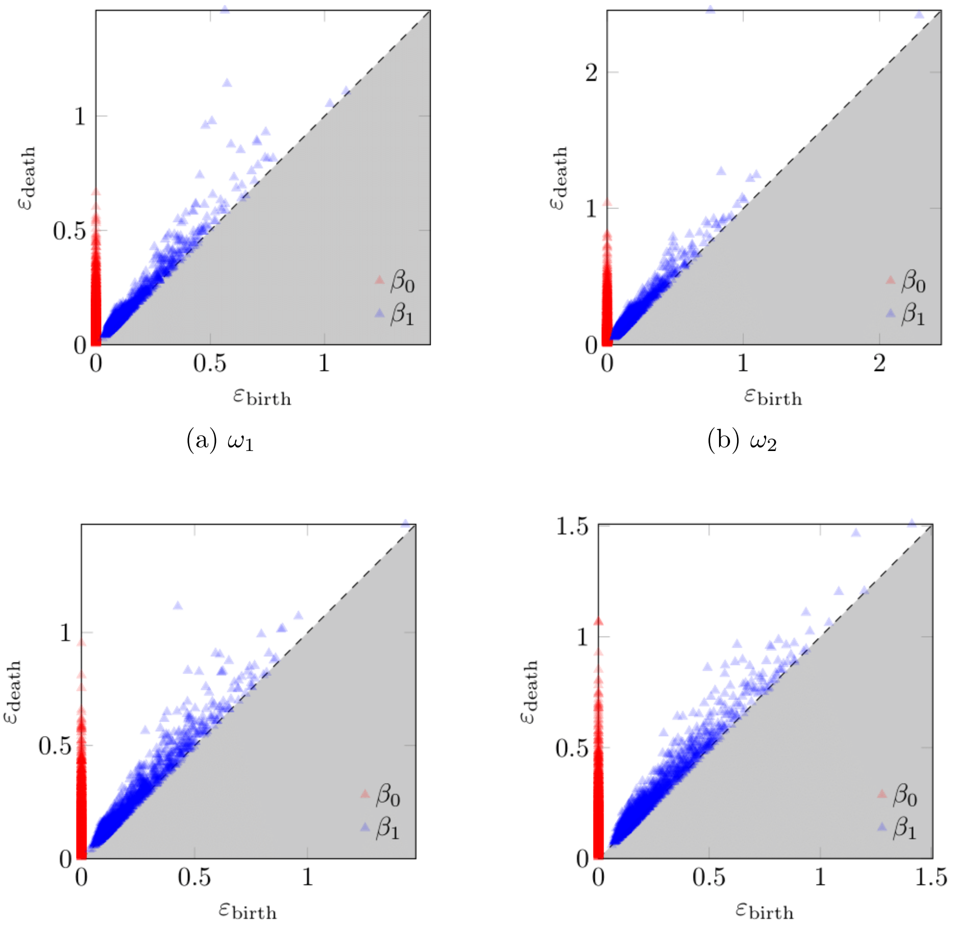
(a) ω_1 (b) ω_2

(c) ω_3 (d) ω_4

Figure 7.2: 3D time-delay embeddings for each natural frequency at $\alpha = 75$.

of the unlagged state to represent a point offers a fairly accurate class representation. Although not every delayed axis may portray the same class, the delay value of $\alpha = 75$ is relatively small considering the dataset's size of 3792 data points. As a result, even if the colour representation is not exact across every axis, it generally provides a near-accurate depiction of the class.

With this in mind, it is evident that certain topological features within the embedding manifolds are influenced by the conditions under which the data were sampled. There is a pronounced dependence of the topological features on the freezing data. For the freezing instance, the linearly-related natural frequencies, ω_1, ω_3, and ω_4,



(a) ω_1 (b) ω_2

(c) ω_3 (d) ω_4

Figure 7.3: Persistent homologies of each natural frequency, at a delay $\alpha = 75$, all in $3D$.

exhibit cavities[2], whereas a loop is formed in the ω_2 time-delay embedding at this α. The freezing data present the most pronounced topological feature within the embedding. The data points linked to the warm condition give a trivial topology in the embedding; a line. Meanwhile, the cold data serve as a topological-transitional phase, leading from the uninformative line into the topologically-interesting features.

These topological features shown in Figure 7.2 are more quantitatively expressed in their persistence diagrams in Figure 7.3. From these diagrams, the distinction of ω_2 from the others is clear, being primarily characterised by a single β_1 feature that persists over the interval [0.76, 2.45]. This feature is significantly more prominent

---

[2] Although cavities are present, the β_2 features cannot be determined because of the dataset sizes. Therefore, all Wasserstein distances will only account for up to β_1. Whilst not as informative as β_2, the persistence of β_1 is different for ω_2 when compared to the others.

than those in the other time-delay embeddings. The topological distinction of ω_2 in the freezing regions suggests that it behaves differently from the others under this condition, a distinction now quantitatively captured in the persistence diagram. On the other hand, the linearly-correlated natural frequencies show many β_1 features, though they do not persist as extensively. These features arise from holes forming over cavities' surfaces.

7.5 Results

The natural frequencies of the Z24 are not all linearly related, as illustrated in Figure 7.1. Consequently, a nonlinear cointegration scheme is necessary for these time series. Although the Z24 natural frequencies do not satisfy this criterion for linear cointegration, it remains possible to calculate the residuals via the Johansen procedure. Yet, one should not anticipate optimal outcomes; as the age-old maxim cautions, "garbage in – garbage out". The linear case study serves to demonstrate the emergence of topological features breaking through into the cointegrated-residual series embeddings and also acts as a benchmark for the nonlinear cointegration scheme.

In the context of nonlinear cointegration, this is only performed on ω_2 because of its challenging nature. Two different GPs are trained to model ω_2: the first without training data over the sustained freezing region, while the second includes it. This approach aims to demonstrate, via topological reasoning, the importance of appropriate training data selection.

7.5.1 Linear Cointegration

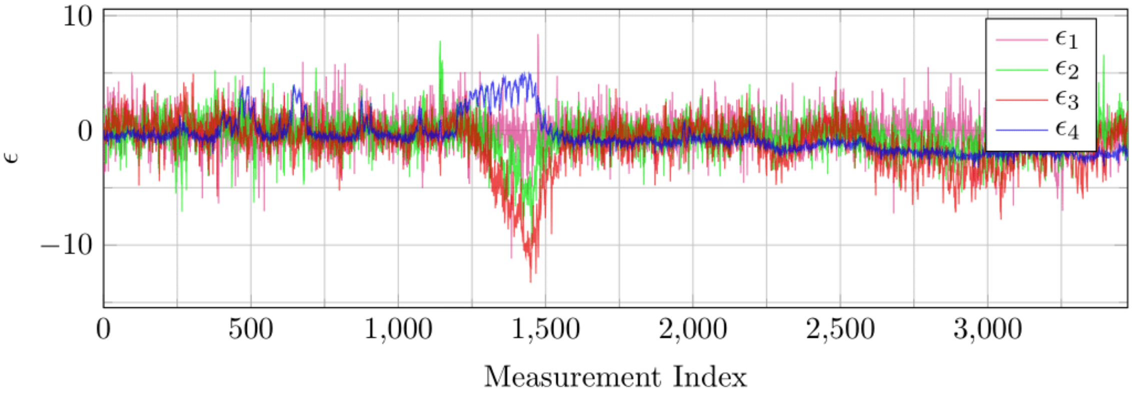


Figure 7.4: The linear residual series calculated from the first four natural frequencies.

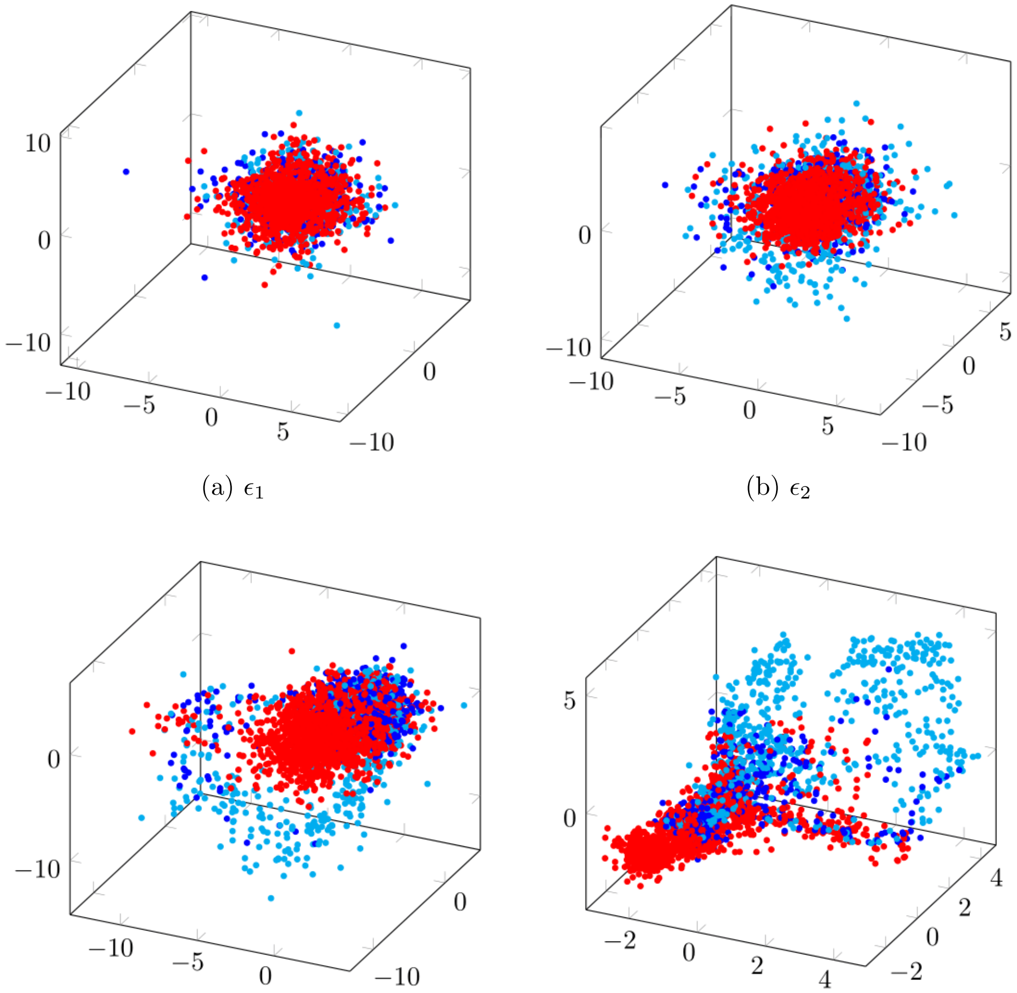
The motivation behind this case stems from the assumption that the residual series are derived from Gaussian white noise. Consequently, when constructing time-delay embeddings for the residuals, a Gaussian cluster is expected. In essence, the expectation is that cointegration will destroy the topology of the manifolds. However, when complications arise, as with the Z24 natural frequencies, the residual series act as a safety net, catching any information that the cointegrating vector fails to remove. When examining the time-delay embedding, any information caught by the residual series manifests as a topological feature in n dimensions.

The standard procedure for performing cointegration over the four natural frequencies is followed, which then determines the cointegrating vectors to be,

$$\beta = \begin{bmatrix} \beta_1 & \beta_2 & \beta_3 & \beta_4 \end{bmatrix} = \begin{bmatrix} 3.05 & 1.81 & 0.66 & 0.81 \\ -0.02 & -0.80 & -1.70 & -0.05 \\ -3.12 & 0.47 & -0.05 & 0.11 \\ -0.31 & -2.07 & 0.72 & 0.20 \end{bmatrix} \tag{7.14}$$

Subsequently, the first four 'stationary' residual series are ascertained by projecting the natural frequencies via the cointegrating vectors, the result of which is shown in Figure 7.4.

From a topological perspective, the failure of linear cointegration over the set of Z24 natural frequencies becomes intuitive. In the context of topology and time-delay embeddings, cointegration is analogous to determining a linear sum of the

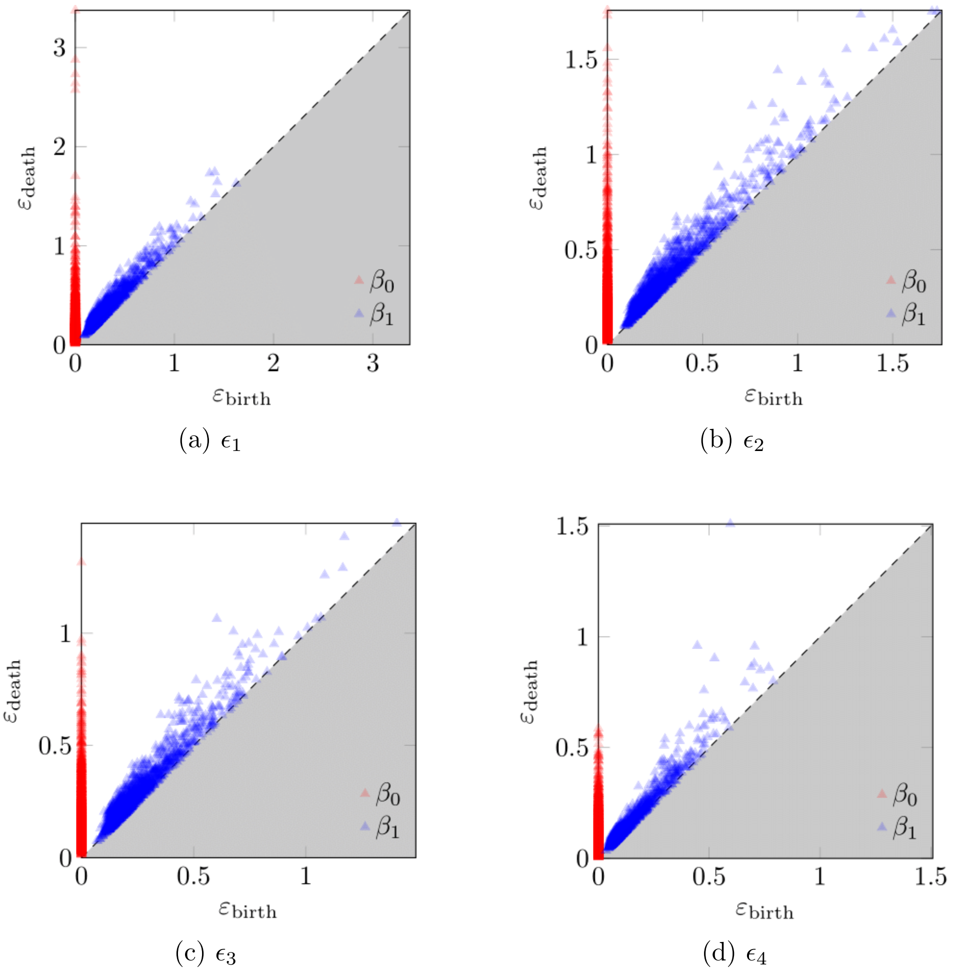


(a) ϵ_1 (b) ϵ_2 (c) ϵ_3 (d) ϵ_4

Figure 7.5: 3D time-delay embeddings for the four linearly cointegrated residuals, at $\alpha = 75$.

time-delay embeddings, aiming to produce a Gaussian cluster. The challenge arises because ω_1, ω_3, and ω_4 cannot adequately reduce the loop formed in ω_2 to a Gaussian cluster with the cavities present in their embeddings. As a result, there is an inherent limitation in eliminating the loop found in ω_2.

A simplistic solution might be to downplay the topology of ω_2, by minimising its coefficient in the cointegrating vector, effectively removing the troublesome ω_2 from the problem. This approach would essentially focus on cointegration over the linearly-correlated natural frequencies. This is exactly the case for the most stationary residual series ϵ_1, where the contribution of ω_2 in β_1 is significantly less than the others. As observed in Figure 7.5 (a), the outcome does very closely resemble a



(a) ϵ_1 (b) ϵ_2

(c) ϵ_3 (d) ϵ_4

Figure 7.6: Persistent homologies of each linearly cointegrated residual, at a delay $\alpha = 75$, all in $3D$.

Gaussian cluster. However, it is challenging to view ϵ_1 as fair since it is essentially a cointegration, with the omission of ω_2. Additionally, it is vital to retain ω_2 because of its heightened damage sensitivity.

The following Figures 7.5 (b), (c), and (d) highlight that the fluctuations from the freezing region become increasingly difficult to quash. Specifically, the time-delay embedding for ϵ_4 almost exactly resembles ω_1 in Figure 7.2. This similarity is reflected in the cointegrating vector, where β_4 is heavily influenced by ω_1. Following a similar logic, Figure 7.5 (c) shows the emergence of a loop from the central Gaussian cluster, as the most significant value in β_3 corresponds to ω_2. The persistent homologies of all the residuals appear in Figure 7.6. Figures 7.6 (a) and (b), which present the persistent homologies for the time-delay embeddings of ϵ_1 and ϵ_2, display persistent homologies representative of a Gaussian cluster. This outcome aligns with

	ω_1	ω_2	ω_3	ω_4	ϵ_1	ϵ_2	ϵ_3	ϵ_4
ω_1	0.00	72.28	93.98	143.70	266.11	257.60	185.26	16.35
ω_2	72.28	0.00	41.40	82.12	208.42	200.29	125.65	65.34
ω_3	93.98	41.40	0.00	57.23	183.92	175.31	102.74	87.01
ω_4	143.70	82.12	57.23	0.00	137.62	128.55	56.07	137.11
ϵ_1	266.11	208.42	183.92	137.62	0.00	31.22	96.32	260.18
ϵ_2	257.60	200.29	175.31	128.55	31.22	0.00	88.69	251.89
ϵ_3	185.26	125.65	102.74	56.07	96.32	88.69	0.00	178.98
ϵ_4	16.35	65.34	87.01	137.11	260.18	251.89	178.98	0.00

Table 7.1: Wasserstein distances for the embedded time series before and after cointegration, for a delay of 75 and an embedding dimension of 3.

expectations for the cointegrated residuals, where no discernible-persisting features are present, since they have been removed via cointegration. Subsequent persistent diagrams then begin to show persistent topological features, indicating that they do not represent Gaussian clusters.

To gauge the efficiency in removing the inherent topologies of the time series using cointegration, Table 7.1 provides the Wasserstein distance between the time-delay embeddings in every instance. Within Table 7.1, there is a discernible pattern in the magnitudes, shown by the varying coloured regions.

In this context, the Wasserstein distance serves as a measure to evaluate the effectiveness of cointegration by examining their topologies. Ideally, when a distinctive topology exists within an embedding of the pre-cointegrated data, cointegration should eliminate this, leading to a Gaussian cluster; leading to a large Wasserstein distance between the two. Conversely, when cointegration fails to remove these effects, a topology akin to the pre-cointegrated data emerges. The Wasserstein distance then identifies this topology in the residual, essentially gauging the success of the cointegration process.

Table 7.1 can be considered as a 2×2 block matrix, each comprising 4×4 matrices. The blocks vaguely divide the table into the comparisons of different topologies. In an idealised setting, Table 7.1 would display the following,

Green the comparisons between interesting topologies, i.e. the loop and cavities.

Blue the comparisons of interesting topologies to Gaussian clusters.

Yellow the comparisons exclusively between Gaussian clusters.

Essentially, the blue off-main-diagonal blocks show comparisons between different topologies, and thus their Wasserstein distances are expected to be greater. Whereas, the green and yellow on-main-diagonal blocks show comparisons of like topologies and thus their magnitudes are expected to be smaller.

However, because of the shortcomings of linear cointegration on the Z24 natural frequencies, Table 7.1 does not always match this ideal. Notably, ϵ_3 and ϵ_4 show topology breaking through into the residual embeddings, and their Wasserstein distances are affected accordingly.

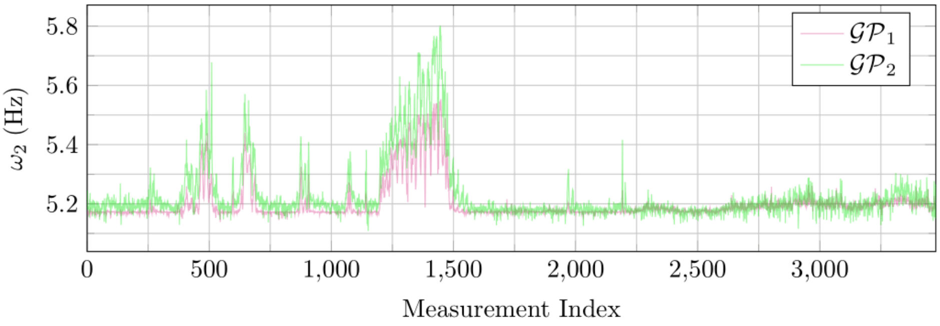
Expectedly, as the residual becomes less stationary – traversing from ϵ_1 to ϵ_4 – there is a quantifiably greater likeness to the topology of raw natural frequency embeddings. Within the green block, the expected behaviour of cointegration is shown in ϵ_1 and ϵ_2, with their Wasserstein distances ranking among the largest in the table, suggesting that cointegration is reducing the time series to a Gaussian cluster in these cases. However, with ϵ_3 and ϵ_4 there is marked evidence of topological leakage, with the smallest Wasserstein distance in the table occurring between ϵ_4 and ω_1.

An inverse logic appears in the yellow block when compared to the blue block. The comparison between ϵ_1 and ϵ_2 is small as it compares the topologies of two Gaussian clusters. However, when considering ϵ_3 and ϵ_4, there is successively increasing topological intrusion from the natural frequencies. As a result, the topology of the residuals is aligning with the raw data embedding, leading to greater Wasserstein distances.

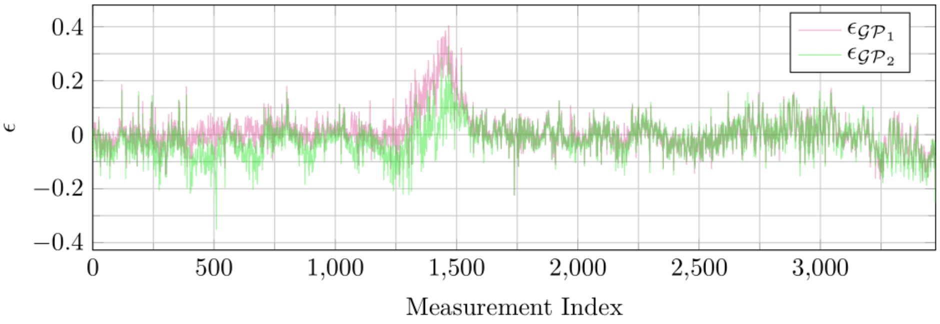
From a topological perspective, ϵ_2 emerges as the most suitable residual for the Z24. Both ϵ_3 and ϵ_4 display excessive topological bleed-through from the natural frequencies, indicating cointegration is not adequately detrending in these scenarios. Whereas, the most-Gaussian-cluster-like ϵ_1 practically excludes the highly damage-sensitive ω_2 from the analysis. Therefore, ϵ_2 offers a reasonable balance. Even with slight topological intrusion from the freezing data, the increased damage sensitivity makes it a worthwhile trade-off.

7.5.2 Nonlinear Cointegration of ω_2

Continuing with a nonlinear cointegration approach, this case study emphasises the importance of training-data selection for a GP during nonlinear cointegration.



(a) Non-cointegrated data



(b) Cointegrated data

Figure 7.7: Univariate plots of all the time series representing ω_2.

Topological arguments are leveraged to provide insight into the remaining topology in the residual series, comparing two GPs trained on different regions of the Z24 data. The analysis undertaken in this section parallels that in Section 7.5.1, thus some details are omitted. However, the emphasis here is solely on detrending ω_2, focusing on GPs trained using linearly-related natural frequencies. Consequently, there are five distinct time series under analysis, which are,

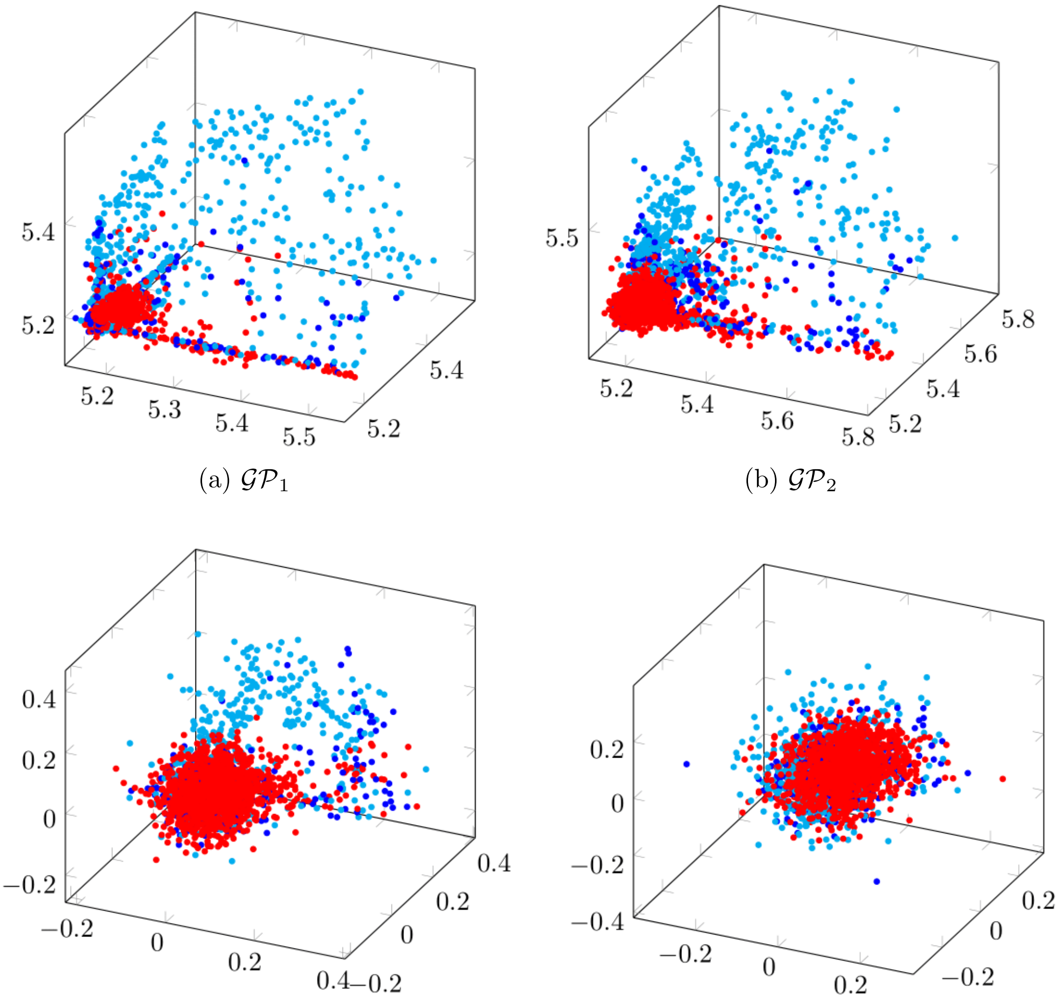
ω_2: represents the second natural frequency of the Z24 bridge [86].

$\mathcal{GP}_1$: serves as a cointegrating function for ω_2. It uses training data comprising ω_1, ω_3, and ω_4, specifically considering the initial 1000 data points where sustained freezing events are not taken into account.

$\mathcal{GP}_2$: acts as another a cointegrating function for ω_2. Again, its training data includes ω_1, ω_3, ω_4, but focuses on data points 1100 - 2100, encapsulating the entire duration of sustained freezing. Both GPs can be seen in Figure 7.7 (a).

$\epsilon_{\mathcal{GP}_1}$: denotes the residual series derived from $\mathcal{GP}_1$.

$\epsilon_{\mathcal{GP}_2}$: is the residual series formed from $\mathcal{GP}_2$, both residual series can be seen in Figure 7.7 (b).



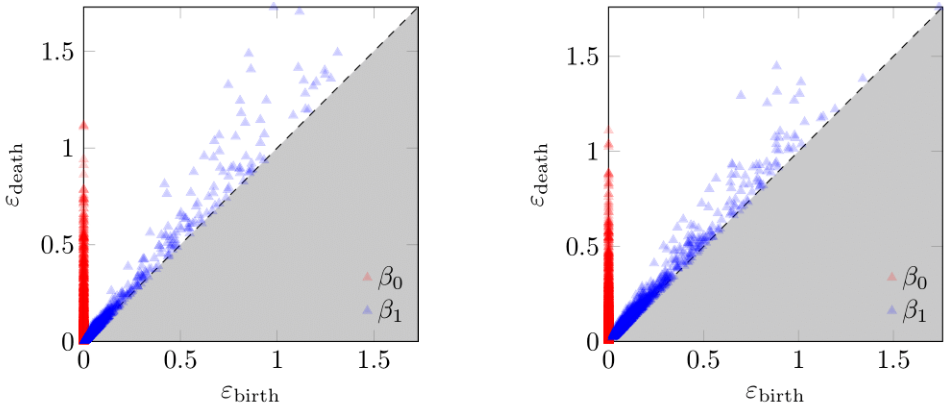
(a) $\mathcal{GP}_1$ (b) $\mathcal{GP}_2$

(c) $\epsilon_{\mathcal{GP}_1}$ (d) $\epsilon_{\mathcal{GP}_2}$

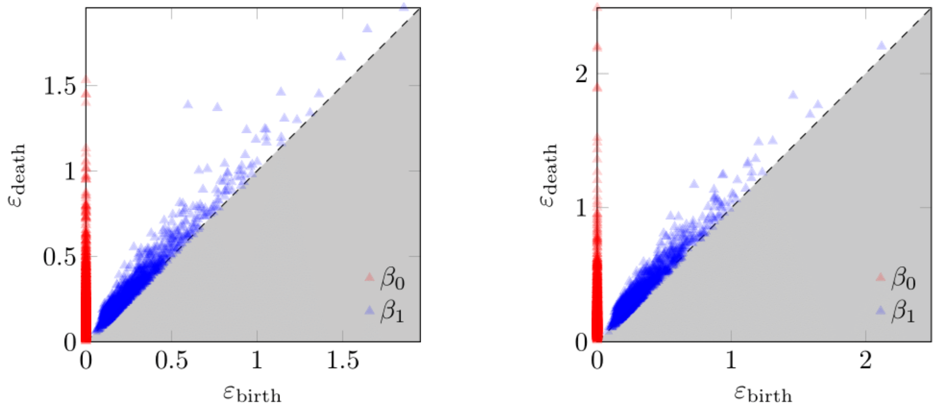
Figure 7.8: Time delay embeddings for the GPs and their residual series, all embedded into 3D at $\alpha = 75$.

As evident from Figure 7.7 (b), both residuals exhibit minor disturbances around the freezing region, suggesting that the GPs are failing to detrend in this region. As anticipated, $\epsilon_{\mathcal{GP}_2}$ shows smaller perturbations because of $\mathcal{GP}_2$'s training over the domain. A richer perspective can be obtained by embedding these time series, allowing a more comprehensive qualitative and quantitative topological examination.

From Figures 7.8 (a) and (b), which display the time-delay embeddings for the two GPs, it is evident that these depict cavities rather than loops. The failure to form a loop indicates that they do not offer precise approximations to ω_2 as their embedded topologies differ. Consequently, the introduction of topological artefacts after detrending is inevitable.



(a) $\mathcal{GP}_1$ (b) $\mathcal{GP}_2$



(c) $\epsilon_{\mathcal{GP}_1}$ (d) $\epsilon_{\mathcal{GP}_2}$

Figure 7.9: The persistence diagrams for the nonlinear cointegration time series, embedded into $3D$ at a delay $\alpha = 75$.

After embedding the residuals, Figure 7.8 (c) shows the emergence of the loop from the Gaussian cluster. This feature suggests that $\mathcal{GP}_1$ is not sufficiently detrending the residual. On the other hand, Figure 7.8 (d) appears to resemble a Gaussian cluster, indicating a more successful detrending. This result is anticipated, given that $\mathcal{GP}_2$ is familiar with the nonlinear behaviour of ω_2 in the freezing region.

However, the inability of $\mathcal{GP}_2$ to form a loop in the embedding space does not come without consequences. A noticeable skewness can be discerned within Figure 7.9 (d), as highlighted quantitatively in its persistence diagram in Figure 7.9 (d). Here, several β_0 features persist beyond 1.5, which are absent in other persistence diagrams. Therefore, indicating that $\mathcal{GP}_1$ has eliminated the defining toroidal topology, at the expense of a small topological artefact of elongating the cluster. With this understanding, there is potential to reverse-engineer this topological information into the

	ω_2	$\mathcal{GP}_1$	$\mathcal{GP}_2$	$\epsilon_{\mathcal{GP}_1}$	$\epsilon_{\mathcal{GP}_2}$
ω_2	0.00	130.80	81.40	120.32	205.71
$\mathcal{GP}_1$	130.80	0.00	62.00	209.81	279.94
$\mathcal{GP}_2$	81.40	62.00	0.00	162.23	239.59
$\epsilon_{\mathcal{GP}_1}$	120.32	209.81	162.23	0.00	98.33
$\epsilon_{\mathcal{GP}_2}$	205.71	279.94	239.59	98.33	0.00
$\mathcal{N}_3(0,\ 1)$	217.74	287.48	251.35	138.50	76.05

Table 7.2: The Wasserstein distances when two data sets are compared, at a delay $\alpha = 75$ in an embedding dimension of 3.

1D time series. This topological approach could be used as a tool to tune GPs, by evaluating their capability to produce a Gaussian cluster when embedding residuals; facilitating a method to optimise nonlinear cointegration schemes.

Table 7.2 presents the Wasserstein distances between all the embedded time series. For clarity in subsequent discussions, an additional row representing a true 3D Gaussian cluster, the idealised result of cointegration denoted as $\mathcal{N}_3(0, 1)$[3], has been added. Similar to Section 7.5.1, block matrices emerge in Table 7.2, categorised by the comparison of distinct topological features.

Table 7.2 actually allows for the direct comparison of the embeddings for GPs with ω_2, directly quantifying their topological similarities. The Wasserstein distances reveal that $\mathcal{GP}_2$ is much more topologically similar to ω_2 than $\mathcal{GP}_1$, with persistent homology dissimilarity scores of 81.40 and 130.8, respectively.

The crucial aspect of this analysis is to determine the extent of topology removal during cointegration. Ideally, all the topology would be eradicated, resulting in a perfect Gaussian cluster. If this is achieved, the manifolds before and after cointegration should be highly dissimilar, hence a greater Wasserstein distance is desirable.

From this perspective, it is evident that $\epsilon_{\mathcal{GP}_2}$ has been more successfully detrended than $\epsilon_{\mathcal{GP}_1}$, with Wasserstein distance of 205.71 and 120.32, respectively. Impressively, $\epsilon_{\mathcal{GP}_2}$ is approaching the idealised scenario, which has a Wasserstein distance of 217.74 from ω_2. When comparing $\epsilon_{\mathcal{GP}2}$ directly to the idealised $\mathcal{N}^3(0, 1)$, the result is the second smallest value in the table at 76.05, highlighting their similarity. Conversely, $\epsilon_{\mathcal{GP}_2}$ is much more distinct from the idealised case. Consequently, topological evaluations make it clear that $\epsilon_{\mathcal{GP}2}$ achieves superior data normalisation

---

[3] At a slight abuse of notation, this indicates a 3D Gaussian cluster centred at (0, 0, 0) with unit variance in each axis.

compared to $\epsilon_{\mathcal{GP}_1}$.

Making use of the idea that a higher Wasserstein distance signifies a more effective cointegration, the linearly and nonlinearly cointegrated residuals can be compared relative to ω_2. As the data from this section and Section 7.5.1 were preprocessed identically, comparisons can be drawn between Tables 7.1 and 7.2.

The best-cointegrated residuals, ranked by their proximity (or lack thereof) to the embedding of ω_2, are: ϵ_1, $\epsilon_{\mathcal{GP}_2}$, ϵ_2, ϵ_3, $\epsilon_{\mathcal{GP}_1}$, and ϵ_4, with corresponding values of 208.42, 205.71, 200.24, 125.65, 120.32, and 65.34. After discounting ϵ_1 as a non-authentic solution as it dismisses the nonlinear natural frequency, this ranks $\epsilon_{\mathcal{GP}_2}$ as the most detrended residual.

7.6 Conclusion

This chapter presented cointegration in a new light, by providing an analogous, but unique topological perspective, reconceptualising the challenge as transforming the time series' embedded topology into a Gaussian cluster. This approach, rooted in the visualisation of shape, allows for an intuitive comprehension of the cointegration process when dealing with intricate and complex data.

What underscores the power of a topological view is its ability to yield unique topological signatures for representing time series within a cointegration framework. These signatures serve a dual purpose: they highlight when cointegration fails to detrend time series, and also provide insight into how cointegration is struggling to remove features. For example, while ω_2 creates a loop after embedding, other time series manifest cavities. Such distinct patterns facilitate pinpointing the origins of information leakage. If a loop is present in the residual series embedding, its presence indicates information from ω_2 is breaking through into the residual series.

Concluding the results section of this chapter, it was demonstrated that the topological approach offers a criterion for gauging the efficacy of various cointegration approaches. By evaluating the topological dissimilarity between the residual series embedding and their pre-cointegrated counterparts, it was deduced that the optimal cointegration scenario is achieved via nonlinear cointegration with knowledge of the freezing domain.

Chapter 8

QUANTIFYING TREND REMOVAL VIA COINTEGRATION WITH TDA

This chapter builds on the concepts introduced in Chapter 7, exploring the quantification of cointegration trend removal using TDA. Within an SHM framework, cointegrated residuals are computed to eliminate the long-term correlations caused by benign EOVs from a set of similar monitored structural parameters. However, when performing cointegration, there is little detailed understanding of the trends being purged in the time series. Fundamentally, cointegration aims to produce the most stationary residual as a weighted sum of the input time series, without considering the specific trends. A subtle problem arises when the structural variables being cointegrated show linearly-proportional changes in the presence of damage. In such instances, cointegration might inadvertently eliminate damage indicators from the residual series; compromising its ability for novelty detection.

In reality, the purpose of cointegration is to train using data from times where the structure is believed to be flawless. Therefore, this hypothesised situation is SHM malpractice. Nonetheless, gaining insight into the specific trends being removed and the degree to which they are eliminated is valuable. This knowledge can guide potential adjustments to cointegration to prevent such occurrences.

The approach developed in this thesis to quantify specific EOV trend removal over cointegration involves evaluating the extent of topological change in embeddings of pre and post-cointegrated structural parameters concerning a specific EOV. The persistent homologies of the embeddings before and after cointegration are compared to a specific EOV's persistent homology via the Wasserstein distance, gauging how much of the EOV's topology has been removed over the cointegration process. If any topology remains in the embedded residual which is reminiscent of the EOV's topology, the Wasserstein distance will detect this, highlighting the inefficiency in removing that particular trend.

To undertake this analysis, a set of linearly-cointegrated structural variables is needed, showing pronounced effects from at least two distinct EOVs; enabling the relative comparison of their removal. The Z24 Bridge dataset does not meet these prerequisites; hence, the Tamar Bridge dataset is introduced and examined in this chapter. The effects of air temperature and traffic loading EOVs on the sets of cable tensions and deck frequencies are then topologically explored. A more in-depth discussion regarding the effects of these EOVs on the structural parameters is found in Section 8.1.

To strengthen the reliability of this procedure, the time-delay embedding parameters are logically determined to maximise and freely express the homological features. The Tamar Bridge exhibits more pronounced periodic patterns compared to the Z24 Bridge. Therefore, the cyclicality is leveraged to determine an optimal delay via *fast Fourier transform* (FFT) which amplifies loops in the embedding space, producing the most persistent homological features. Subsequently, an established method for determining an optimal dimension is examined using the *False-Nearest Neighbours* algorithm [129]. All signals undergo evaluation using their optimal embedding parameters.

8.1 Tamar Bridge Overview

The Tamar Bridge, a suspension bridge, spans 335 metres across the River Tamar, connecting Plymouth and Saltash in the UK. After four decades of service, the bridge underwent strengthening in the late 1990s to comply with EU legislation, ensuring it could safely accommodate heavy-goods vehicles weighing up to 40 tonnes. To fortify the bridge, extra cable stays were added. To widen the bridge, cantilever decks were added on both sides. The original deck was also replaced with lighter and stronger materials. During the reinforcement process, multiple sensors were installed to monitor the bridge's structural and environmental variables [123, 130].

In 2006, the Vibration Engineering Section of the Department of Civil Engineering at the University of Sheffield installed a vibration-based monitoring system [130]. This monitoring system records 64 Hz samples over 30 minute intervals [131]. For each interval, an on-site automatic modal analysis is performed, outputting valuable damage-sensitive features for the bridge. The monitoring system calculates the deck frequencies, while the cable tensions are directly observed. These elements form the key structural parameters of interest in this analysis. However, both temperature and traffic loading influence these structural parameters to varying extents. For effective SHM, it is beneficial to mitigate the effects from these EOVs.

8.1.1 Traffic Loading

A one-way traffic count is obtained from the bridge's eastbound toll gates. Every vehicle is categorised into one of ten weight classes, as detailed in Table 8.1. For each interval, the traffic load is determined by multiplying the count of each category by its average weight [123]. Since this information is unidirectional, it is further multiplied by two, based on the assumption that traffic loading is symmetrical on both sides during each interval. Finally, to obtain the instantaneous load L, on the bridge, the accumulated load is divided by the estimated time taken to cross the bridge, which is set at 45 seconds in this instance [123].

Vehicle Category	**Estimated Mass** (kg)
Unregistered by Automatic Vehicle Classification	0
Motorcycles	150
Cars, Trikes and PLG under 3.5 tonnes	1500
2 Axle HGV	18000
3 Axle HGV	26000
4+ Axle HGV	32000
Cars with Trailers	8250
2 Axle HGV with Trailer	21000
3 Axle HGV with Trailer	30000
4+ Axle HGVs with Trailer	36000

Table 8.1: Estimated vehicle masses for each weight category [123].

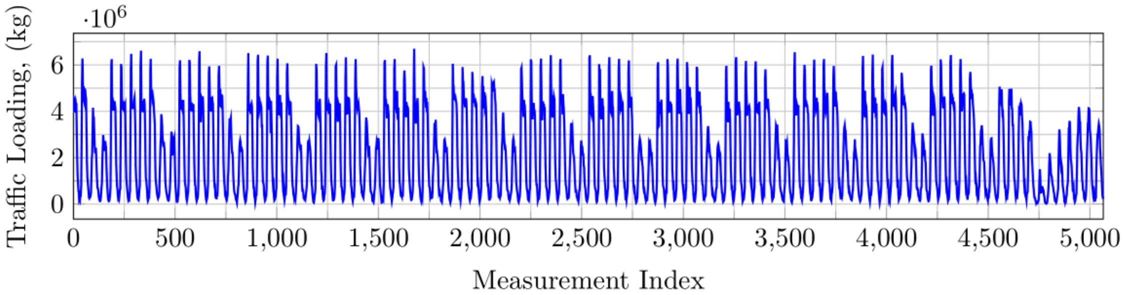


Figure 8.1: The Tamar Bridge traffic loading L.

The traffic loading reveals a recurring pattern: five days of high traffic during the workweek, followed by two days of lighter traffic over the weekend. This trend is the most dominant feature in the signal and remains consistent, even amidst seasonal variations. Additionally, two daily spikes are evident, corresponding to the morning and evening rush hours. Intriguingly, the morning influx, representing the commute to work, is more pronounced than the evening return. This artefact likely stems from the unidirectional flow being sourced from the eastbound toll gates and people are likely to commute to Plymouth for work, which is the nearest major city and is located east of the Tamar bridge. Therefore, the morning traffic count is inflated when assuming equal traffic flow in both directions. Conversely, the evening return traffic is likely underrepresented.

8.1.2 Temperature

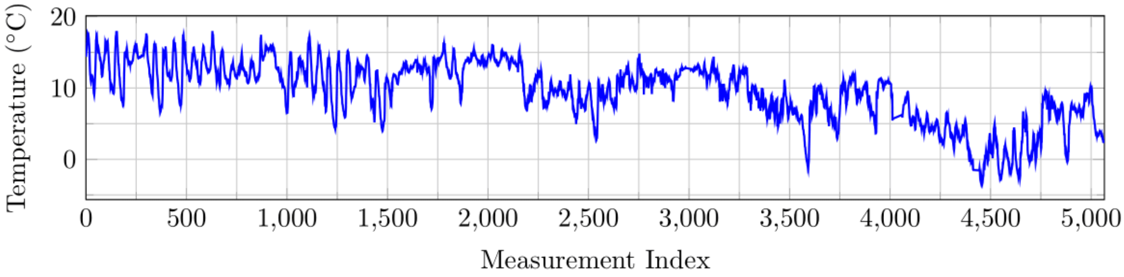


Figure 8.2: The Tamar Bridge temperature θ.

The air temperature recording θ, is more straightforward, as it is directly measured using thermometers. Within the temperature data, two main trends stand out. Firstly, there is a large-scale, low-frequency fluctuation linked to seasonal changes, accounting for the majority of the data's variance. The second trend consists of daily fluctuations, with temperatures dropping at night and rising during the day.

	ω_1	ω_2	ω_3	ω_4	ω_5	T_1	T_3	T_4	T_5	T_6	T_7	T_8	L	θ
ω_1	1.00	0.26	0.60	0.38	0.38	0.27	0.28	0.26	-0.26	-0.26	-0.09	0.07	-0.43	-0.28
ω_2	0.26	1.00	0.35	0.36	0.32	0.40	0.49	0.51	-0.45	-0.45	-0.15	-0.12	-0.31	-0.44
ω_3	0.60	0.35	1.00	0.69	0.60	0.34	0.36	0.37	-0.31	-0.31	-0.03	0.02	-0.52	-0.31
ω_4	0.38	0.36	0.69	1.00	0.55	0.35	0.35	0.34	-0.28	-0.28	0.03	0.15	-0.25	-0.29
ω_5	0.38	0.32	0.60	0.55	1.00	0.32	0.35	0.35	-0.32	-0.32	-0.08	0.04	-0.34	-0.31
T_1	0.27	0.40	0.34	0.35	0.32	1.00	0.96	0.96	-0.81	-0.81	-0.20	0.25	-0.11	-0.92
T_3	0.28	0.49	0.36	0.35	0.35	0.96	1.00	1.00	-0.90	-0.88	-0.30	0.15	-0.16	-0.96
T_4	0.26	0.51	0.37	0.34	0.35	0.96	1.00	1.00	-0.88	-0.87	-0.27	0.12	-0.18	-0.95
T_5	-0.26	-0.45	-0.31	-0.28	-0.32	-0.81	-0.90	-0.88	1.00	1.00	0.65	0.03	0.24	0.94
T_6	-0.26	-0.45	-0.31	-0.28	-0.32	-0.81	-0.88	-0.87	1.00	1.00	0.67	0.06	0.26	0.93
T_7	-0.09	-0.15	-0.03	0.03	-0.08	-0.20	-0.30	-0.27	0.65	0.67	1.00	0.28	0.19	0.48
T_8	0.07	-0.12	0.02	0.15	0.04	0.25	0.15	0.12	0.03	0.06	0.28	1.00	0.14	-0.13
L	-0.43	-0.31	-0.52	-0.25	-0.34	-0.11	-0.16	-0.18	0.24	0.26	0.19	0.14	1.00	0.16
θ	-0.28	-0.44	-0.31	-0.29	-0.31	-0.92	-0.96	-0.95	0.94	0.93	0.48	-0.13	0.16	1.00

Table 8.2: The product-moment correlation coefficients between all the Tamar Bridge signals used in this analysis. EOVs: T_i and ω_i are the cable tensions and deck natural frequencies. Structural parameters: L and θ are the traffic loading and air temperature.

8.1.3 Cable Tensions

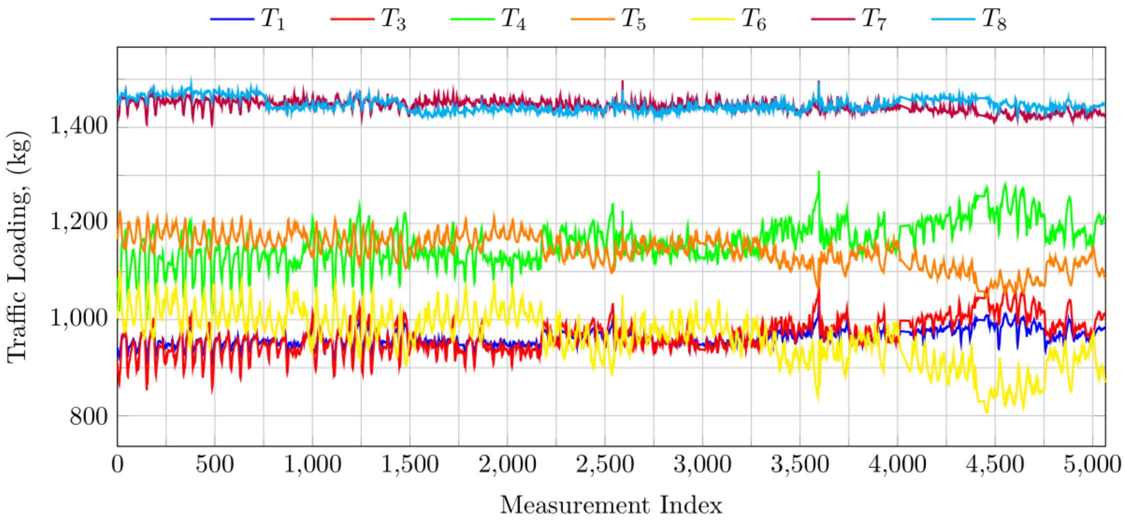


Figure 8.3: The Tamar Bridge cable tensions, T_i.

The cable tensions T_i are most correlated to the air temperature, as shown in Table 8.2. This strong association largely stems from thermal expansion, causing changes in cable length as temperatures fluctuate. Under the relatively constrained geometry of the bridge, and only one expansion joint on its east side [123], a temperature drop leads to cable contraction, resulting in heightened cable tensions. Although correlations are consistently strong, they show both negative and positive directions. This divergence is likely because, when the cable tensions increase, the bridge deck bows slightly, causing cables negatively correlated to pick up the slack, whilst the positively-correlated cables were relieved under the bowing action [131].

Because of the strong linear correlation between the temperature and cable tensions, the daily and seasonal temperature variations are almost directly translated into the cable tensions, with only small variations from other EOVs. The traffic loading is one such variation, which increases the weight supported by the cables. However, compared to the mass of the bridge deck, the weight of the traffic is relatively minor. Therefore, the traffic loading's impact is overshadowed by thermal expansion effects.

The second cable tension T_2, seems to be malfunctioning and producing inaccurate results; hence, it has been excluded from all calculations in this chapter. Additionally, for future time-delay embeddings, T_3 is chosen for embedding as it shows the strongest correlation with temperature.

8.1.4 Deck Frequencies

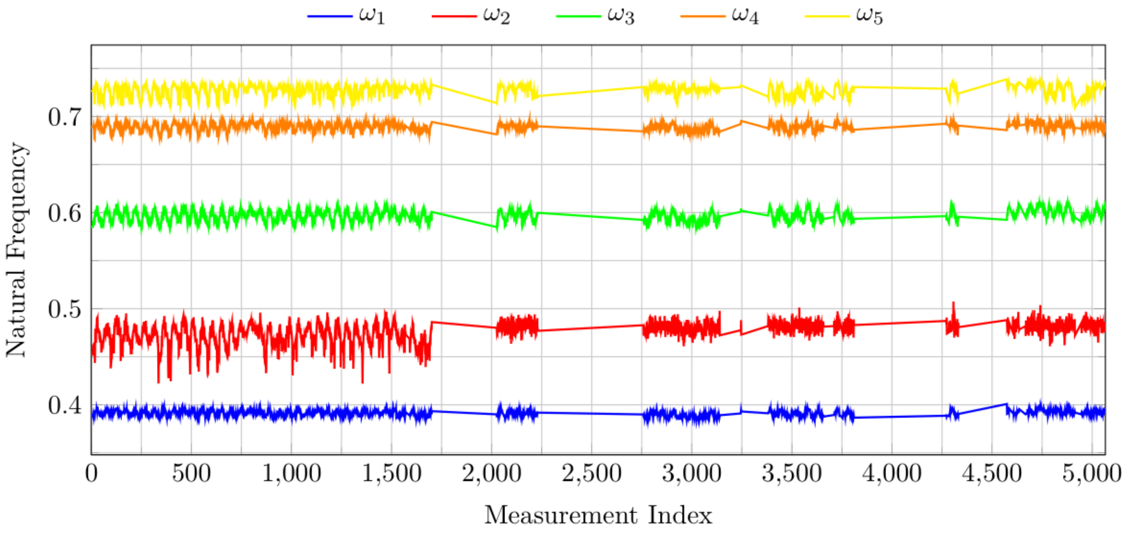


Figure 8.4: The Tamar Bridge deck natural frequencies, ω_i.

The deck frequencies ω_i, were calculated from the acceleration data using a stochastic subspace identification technique [132]. In this study, only the first five deck frequencies are used in cointegration calculations. As a result of shortcomings of the data acquisition system, data were required to be linearly interpolated over certain regions; visible as the short straight lines in Figure 8.4.

The variation in deck frequency from the EOVs can be largely explained by the analogy,

$$\omega_n = \sqrt{\frac{k}{m}} \tag{8.1}$$

where k and m represent the stiffness and mass of the deck. With an increase in traffic loading, the mass of the deck rises. Consequently, the modal properties adjust in line with equation (8.1), leading to a decrease in the natural frequency. This relationship is evident from the consistent negative correlation between deck frequency and traffic loading in Table 8.2. The impact of temperature can be similarly understood. As the temperature rises, materials generally become less rigid. Consequently, according to equation (8.1), the natural frequency will decrease – a trend consistent with Table 8.2.

For embedding purposes, ω_3 is selected, as it shows the strongest relation with traffic count.

8.2 Optimal Embedding Parameters

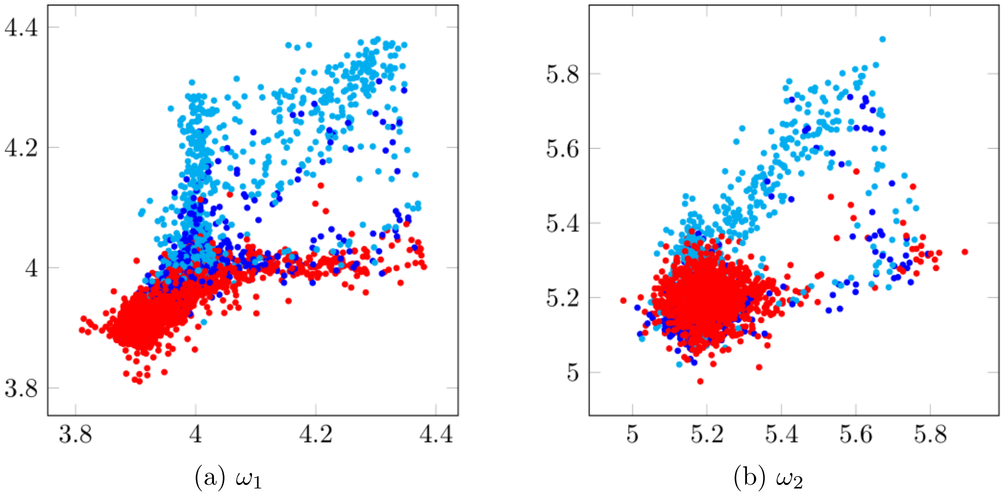
When exploring time-delay embeddings for persistent homology, it is desirable to enhance persistent homology features and to ensure the required dimensional freedom to fully express the topology within a time series. This subsection outlines a methodical approach to maximise the topological information derived from a time series, pinpointing parameters labelled as *optimal* for a given time series. By understanding the nuances of time-delay embeddings and persistent homology, a logical approach is presented to identify an *optimal delay* α^* and an *optimal dimension* d^* for embedding, tailored for persistent homology applications.

Time-delay embeddings depict cyclic trends as loops in higher dimensions. Therefore, maximising the radii of these loops results in the longest persistent intervals, giving the most pronounced representations of embedding manifolds. To enhance the effects of a time series' topology, the focus is the frequency relating to the cyclic trend with the greatest amplitude within the signal. By doing so, the embedding loop and thus its persistent homology are maximised, offering a clear representation for calculations. A crucial aspect to consider is maintaining a reasonably small α^* to ensure local lags between dimensions.

When identifying optimal parameters, it is crucial to maintain both α^* and d^* at relatively-low values to reduce the loss of points during embedding. For higher values of α and d, more points are lost because of instances requiring information outside the range of the time series. As a result, these incomplete instances at the start and the end of the time series are not considered, leading to information loss. The number of excluded points is determined by,

$$n_{\text{lost}} = (d-1)(\alpha+1) \tag{8.2}$$

Consequently, the method for determining an optimal embedding dimension d^* needs to factor in the curse of dimensionality and the loss of points. A larger d comes with two drawbacks: increasing the sparsity of points over more dimensions, but then also reducing the number of points present within the manifold, prompting an even more sparse embedding. This method should consider the potential emergence of topological artefacts because of sparsity and topological noise, while still offering the necessary dimensionality to accurately represent the topology of a time series.



(a) ω_1 (b) ω_2

Figure 8.5: Z24 embeddings at $\alpha = 75$ and $d = 2$, showing loss of topological information in ω_1 when compared to its 3D embedding.

To illustrate how a lower-dimensional embedding may limit the comprehension of a time series' topology, consider again the Z24 time-delay embeddings from Chapter 7 of ω_1 and ω_2, set again at $\alpha = 75$, but in this case with an embedding dimension of 2. As depicted in Figure 8.5 (a), the cavity observable in the 3D embedding for ω_1 is projected down into two dimensions; losing this topological information. In contrast, the 2D loop remains as prominent in Figure 8.5 (b) as it was in 3D. These observations suggest that ω_1 requires more than two dimensions in the embedding space, but two dimensions appear sufficient for ω_2. Intriguingly, the time-delay embedding behaves somewhat like a topological analogue to principal component analysis, except in reverse. Therefore, the established *false nearest neighbours* method is explored [129]. This method aims to minimise the number of incorrectly identified neighbours in embeddings, as illustrated by those projected into 2D in Figure 8.5 (a), ensuring that topological features are accurately represented based on the trends in the 1D time series.

8.2.1 Optimal Delay, α^*

To determine the optimal delay, the focus is directed on a specific cyclic trend within a time series. This trend is determined by decomposing a time series into its frequency components via the fast-Fourier transform (FFT). Within the frequency domain, the frequency with the greatest amplitude is the cyclic trend of interest. The aim is to maximise the persistent interval relating to this cyclic trend, and the α that accomplishes this is deemed optimal.

For derivation purposes, consider the time-delay embedding of a sine wave into 2D. The task of maximising the persistent interval can be reinterpreted as maximising the area enclosed by the loop. The sine wave's time-delay embedding into two dimensions is characterised by $x = \sin(t)$ and $y = \sin(t - \alpha)$[1]. To determine the area of the loop at an α, it is beneficial to form a polar integral, thus requiring a coordinate transform, defined by,

$$r = \sqrt{\sin^2(t) + \sin^2(t - \alpha)} \tag{8.3}$$

$$\theta = \arctan\left(\frac{\sin(t-\alpha)}{\sin(t)}\right) \tag{8.4}$$

before defining the polar integral for the area bound by the embedded loop, it is necessary to transform the integration variable,

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\sin(t-\alpha)}{\sin(t)}\right) \cdot \frac{1}{1 + \left(\frac{\sin(t-\alpha)}{\sin(t)}\right)} = \frac{1}{r^2}\sin(\alpha) \tag{8.5}$$

This transformation facilitates the polar-integral,

$$A = \frac{1}{2}\int_{\theta_1}^{\theta_2} r^2 \, \mathrm{d}\theta = \frac{1}{2}\int_0^{2\pi} r^2 \frac{1}{r^2}\sin(\alpha)\, \mathrm{d}t = \pi\sin(\alpha) \tag{8.6}$$

Having established the area A enclosed by this time-delay embedding, the next logical step is to identify its maximum in relation to α. The maximum of A is determined by differentiating with respect to α,

$$\frac{\mathrm{d}A}{\mathrm{d}\alpha} = \pi\cos(\alpha) \tag{8.7}$$

This equation allows the derivation of all occurrences when α gives maximum or zero area, given by the condition $\cos(\alpha^*) = 0$. To filter out the α relating to the minima, consider when equation (8.6) is non-zero. Consequently, the maximum area is obtained at,

$$\alpha^* = \frac{\pi}{2} + 2k\pi, \; \forall k \in \{0, 1, \ldots\} \tag{8.8}$$

While this result is derived for a continuous case, indicating a magnitude of α, data analysis demands this result as an integer, denoting discrete offsets. The result in

---

[1] In this derivation, α is assumed continuous to permit the use of calculus.

equation (8.8) is divided by 2π, giving the cycle fraction to offset resulting in the greatest A. This cycle fraction is then multiplied by the number of data points in a full cycle of a wave, giving the optimal delay α^*. The number of data points present within one cycle of a sine wave with frequency f, is determined by,

$$n = \frac{f_s}{f} \tag{8.9}$$

then merging with the proportionality relation, the optimal delay α^* in the context of discrete data becomes,

$$\alpha^* = \frac{(1+4k)f_s}{4f} \tag{8.10}$$

For this work, $k = 0$ is consistently employed to ensure local lags between dimensions and to minimise the loss of data points.

Simple Applications

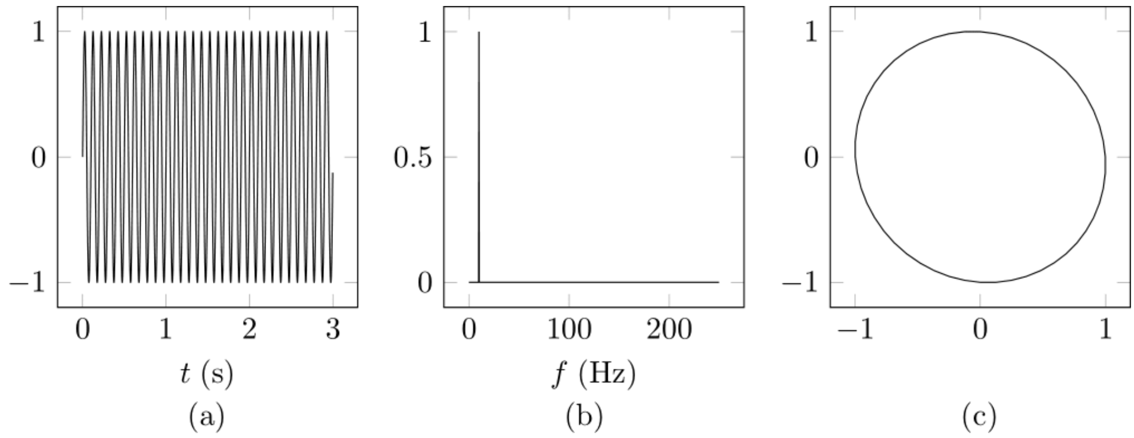
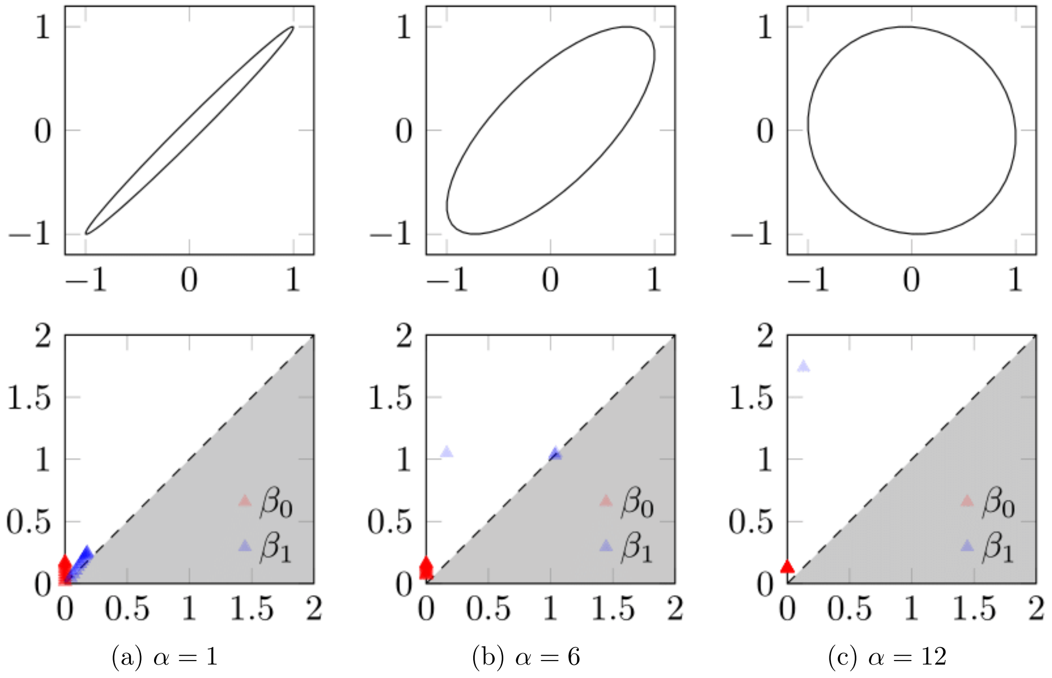


Figure 8.6: (a) A sine wave with $f = 10$Hz at $f_s = 500$Hz, (b) its frequency decomposition, and (c) the resulting optimal delay embedding.

To validate these findings, a sine wave is used to demonstrate and discuss the process of extracting the maximal topology via frequency analysis. Subsequently, a signal composed of the sum of sine waves will be introduced. While the procedure is straightforward with these examples in visualisable dimensions, having a systematic approach for determining time-delay embeddings becomes invaluable when dealing with the complexities of real-world data, embedded in higher-dimension.

A sine wave is recorded at a sampling frequency of 500Hz. Upon converting into the frequency domain with the FFT, the wave is determined to have a frequency



(a) $\alpha = 1$ (b) $\alpha = 6$ (c) $\alpha = 12$

Figure 8.7: Varying α, with their birth-death diagrams beneath.

of 10Hz. To then determine the embedding delay that maximises the area – in this case, a circle – the time series must be delayed in accordance with equation (8.10), giving,

$$\alpha^* = \frac{500}{4 \cdot 10} = 12.5 = 12 \tag{8.11}$$

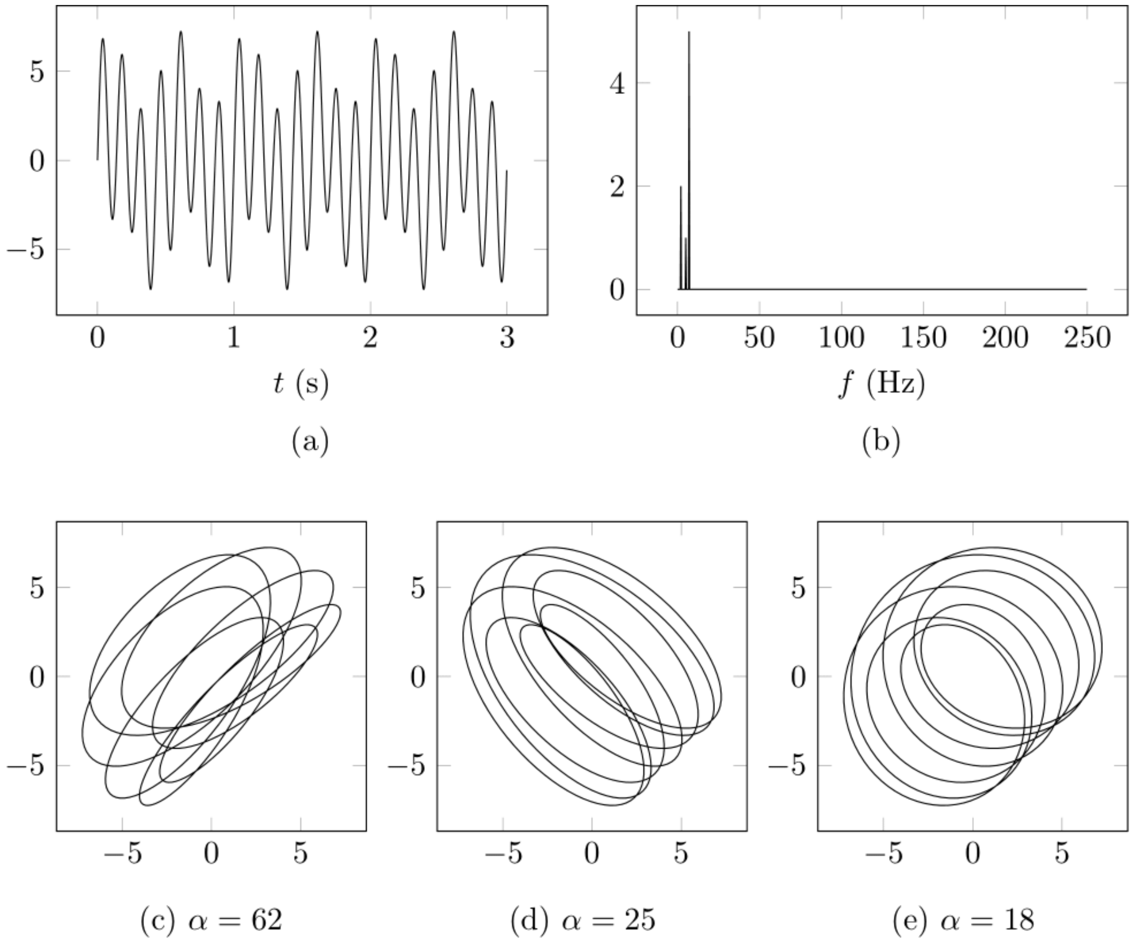
since an integer is required for α, this result is rounded[2]. The result from embedding this sine wave with $\alpha = 12$ is presented in Figure 8.6, where a slight skew is evident, which is directly resultant of the rounding.

To more deeply explore the influence of α on persistent homology, Figure 8.7 showcases three embeddings at delays of 1, 6, and 12, accompanied by their respective persistent homologies. Evidently, as the area bound by the time-delay embedding increases, so does the persistent interval, leading to more pronounced topological features in the persistent diagrams. For $\alpha = 1$, the β_1 features are mainly arising from topological noise. However, for delays of 6 and 12, there is clear persistence of a β_1 feature. Notably, the feature at the optimal delay is nearly twice as persistent as $\alpha = 6$.

In a more complex setting, a signal is observed at $f_s = 500$Hz, and is formed

---

[2] This rounding behaviour, called Banker's rounding, is the default in Python, where halfway numbers are rounded to even.



(a) (b)

(c) $\alpha = 62$ (d) $\alpha = 25$ (e) $\alpha = 18$

Figure 8.8: (a) A composite sine wave, (b) its FFT, and (c) - (e) optimal $2D$ time-delay embeddings at each identified frequency.

by combining multiple sine waves, with varying amplitudes and frequencies. This signal, its frequency domain representation, and three embeddings, each maximising the topology of a constituent wave of the signal, are depicted in Figure 8.8.

The frequencies and amplitudes for the three constituent waves are found to be 2Hz, 5Hz, and 7Hz and the amplitudes are 2, 1, and 5 respectively. The optimal delays are determined to be 62, 25, and 18, respectively. From taking a quick look at the delay embeddings in Figure 8.8, it appears that the loops are being squashed down into $2D$, causing self-intersections in the embedding. It is very difficult, if not impossible to understand which is the true embedding dimension to represent the topology of a time series.

8.2.2 Optimal Dimension, d^*

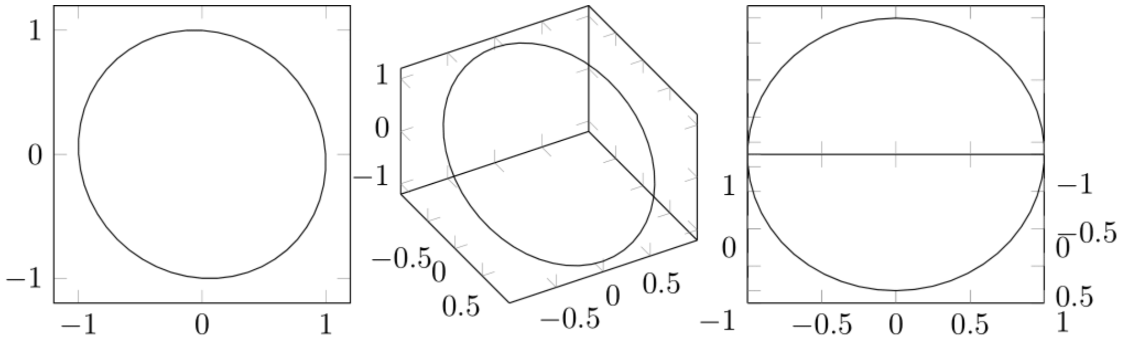


Figure 8.9: Embeddings of sine wave in 2D and 3D.

When determining time-delay embeddings, selecting a dimension that is too low can cause the manifold to be projected down, potentially introducing topological artefacts and misrepresenting the true topology of the time series. The optimal embedding dimension, denoted as d^* is the smallest dimension that freely expresses the topological features of a time series. Dimensions beyond d^* are superfluous, introducing unnecessary computation overheads without yielding significant new topological insights. For instance, the 2D time-delay embedding of the sine wave shown in Figure 8.6 forms a circle. When embedding the sine wave in 3D, as depicted in Figure 8.9, no additional topological information emerges; it still shows a circle, just oriented and skewed in a higher-dimensional space.

The embeddings for the composite signal, as depicted in Figure 8.10, present the other side of the argument. In the composite signal's 2D embeddings, topological artefacts appear as self-intersections. These artefacts arise because the embedding manifold is forced into a lower dimension, causing points to be falsely embedded close to each other, and in some cases, to intersect. Figure 8.10 displays the amplitude-dominant optimal delay embedding in both 2D and 3D, with a longitudinal and transversal view of the loops. These perspectives reveal that the holes are more pronounced, and the self-intersections are absent as the manifold is offset into the third dimension. It is plausible that the 3D embedding might still be introducing artefacts if the true manifold exists in 4D or higher. Perhaps these loops unravel in the fourth dimension? However, without understanding how 4D manifolds project into 3D, this point becomes harder to convey[3]. Therefore, to extend this qualitative

---

[3] A classic example of topological artefacts resulting from a 4D to 3D projection is the Klein bottle. In 4D, the Klein bottle does not intersect itself, but in 3D, it famously does.

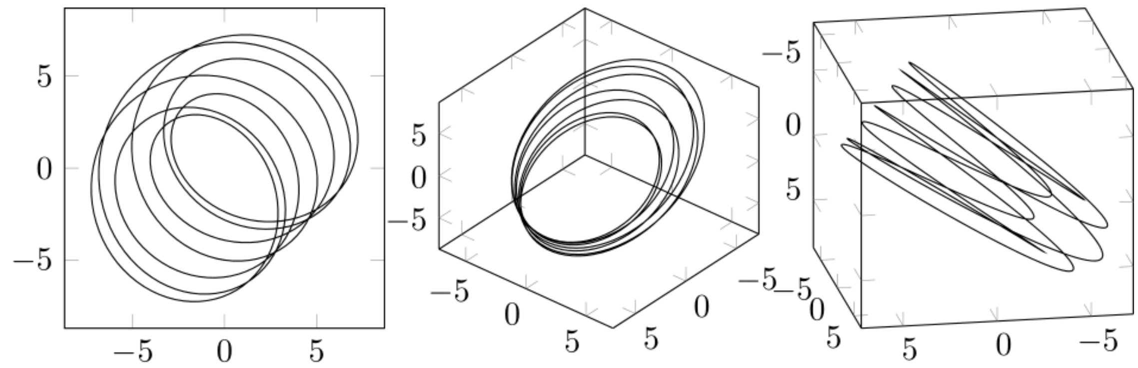


Figure 8.10: Embeddings of the composite sine wave in 2D and 3D.

reasoning of falsely neighboured points in a projection, this is quantified using the *false-nearest* neighbours (FNN) algorithm [129].

The FNN algorithm is a standard method in topological time series analysis [114, 133, 134]. Given its widespread use, this section provides an abridged overview:

1. For a time series y and delay α, form the time-delay embedding at an embedding dimension p, represented as $\Phi(y,\, \alpha,\, p)$.
2. In the embedding at dimension p, the distance from a point i to its nearest neighbour is ∂_i^p.
3. Determine $\Phi(y,\, \alpha,\, p+1)$.
4. In the embedding at dimension $p+1$, the distance from a point i to its nearest neighbour is ∂_i^{p+1}.
5. If the difference $|\partial_i^{p+1} - \partial_i^p|$ exceeds some arbitrary threshold t_1, the nearest neighbour for i in dimension p was a false neighbour.
6. Repeat this calculation for all i in the embeddings and calculate the percentage of false nearest neighbours in the dimension p.
7. The optimal dimension d^* is the smallest p, such that the percentage of false nearest neighbours is below an acceptable threshold t_2.

A high proportion of false neighbours indicates that the embedding dimension is insufficient, caused by the forced proximity of points in the lower-dimensional projection. As mentioned qualitatively earlier, this leads to situations like the self-intersections seen in Figure 8.8, and the introduction of topological artefacts. As the embedding dimension p increases, the phase space reconstruction should yield fewer false nearest neighbours as its topology is freely expressed. Interestingly, the FNN algorithm bears an inverse resemblance to when taking principal components of a space [83]. The percentage of false neighbours can be likened to the complement of the explained variance in PCA.

The FNN algorithm, while useful, is not without its limitations. One notable drawback is the subjectivity in selecting t_1 and t_2. However, the embedding dimension was already a hyperparameter, so employing the FNN algorithm shifts the hyperparameters down the line in exchange for a systematic and consistent determination mechanism. Additionally, the shape of the embedding manifold is parameterised by α, and therefore the d^* determined via FNN is specific to that α. In this study, the d^* determined using FNN is evaluated at the previously-discussed optimal delay α^*.

8.2.3 Tamar Optimal Embedding Parameters

To unambiguously determine the optimal time-delay embeddings of the Tamar Bridge signals, α^* and d^* are calculated for each instance. Therefore, the topological features should be both maximised and freely expressed, paving the way for more robust results in subsequent analyses.

As mentioned in Section 8.2.1, α^* is determined by taking the FFT and identifying the frequency of the sine wave with the maximum amplitude. This frequency is then used to derive α^* via equation (8.10), which amplifies the persistent homology features pertinent to that frequency. Following this, d^* is determined for each signal's α^*; where the results for both are presented in Table 8.3.

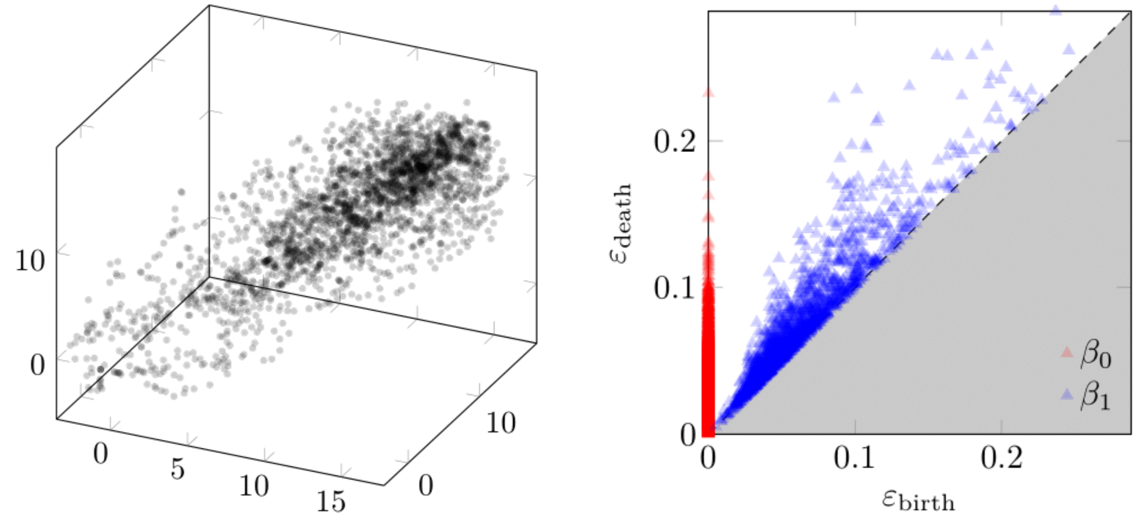
For the FNN algorithm, the consistent hyperparameters were set to $t_1 = 0.1$ and $t_2 = 0.05$ for all signals. While the time-delay embeddings are presented in their unnormalised forms, it is worth noting that for the FNN algorithm and persistent homology, all signals were standardised before embedding. With the Tamar Bridge's

Signal	Frequency (Hz)	Peak Amplitude	Period (h)	α^*	d^*
L	1.16×10^{-5}	1.49×10^{6}	23.9	12	3
θ	1.16×10^{-5}	0.811	23.9	12	3
ω_3	1.16×10^{-5}	0.00225	23.9	12	3
T_3	1.15×10^{-5}	10.2	24.1	12	3

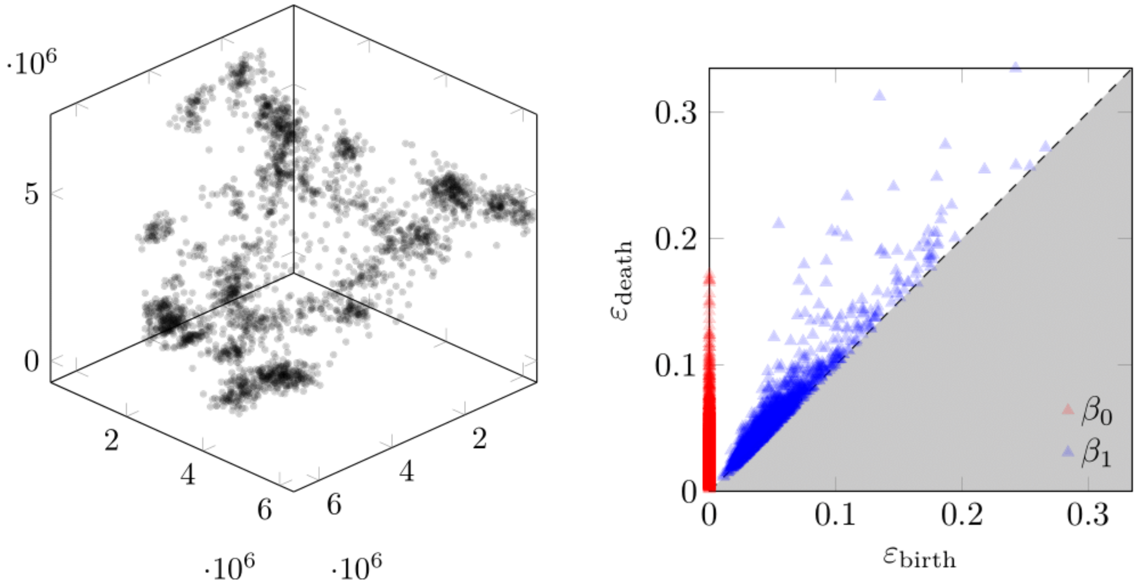
Table 8.3: Optimal embedding parameters for each signal.

data recorded at 30-minute intervals, the sampling frequency is $f_s = 5.56 \times 10^{-4}$ Hz.

The frequency with the most pronounced amplitude, which is the cyclic trend emphasised in the time-delay embedding, unsurprisingly aligns with a daily cycle for each signal, as detailed in Table 8.3. This outcome is expected for the traffic count, but less so for the temperature. Given that the data only spans 15 weeks, the full yearly cyclic trends are undetectable in the frequency domain; only being present as spectral leakage. Consequently, the FFT detects the next-most dominant cyclic trend the daily fluctuations.



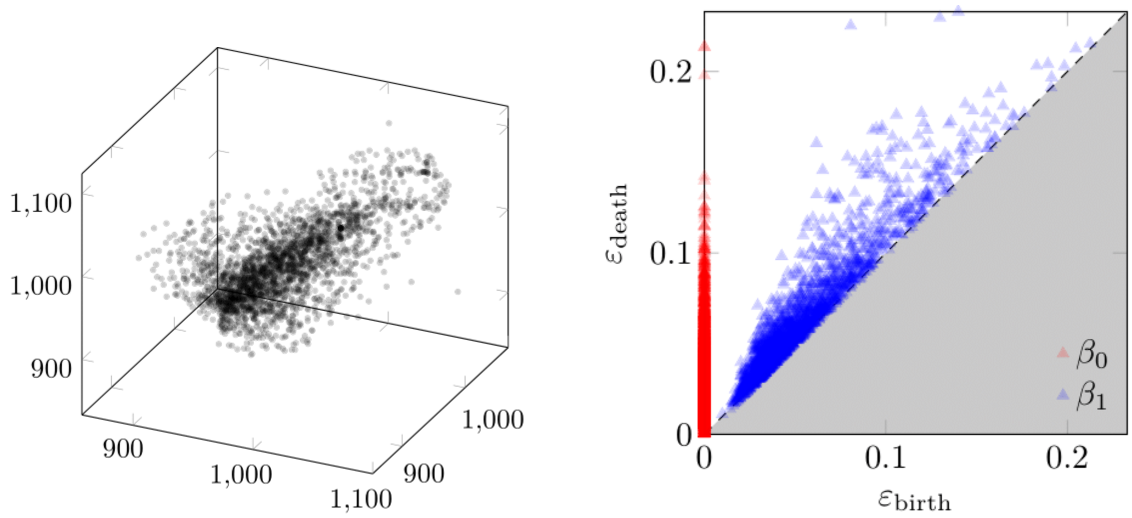
(a) $\Phi(\theta,\,12,\,3)$ (b) $\Phi(\theta,\,12,\,3)$ PH



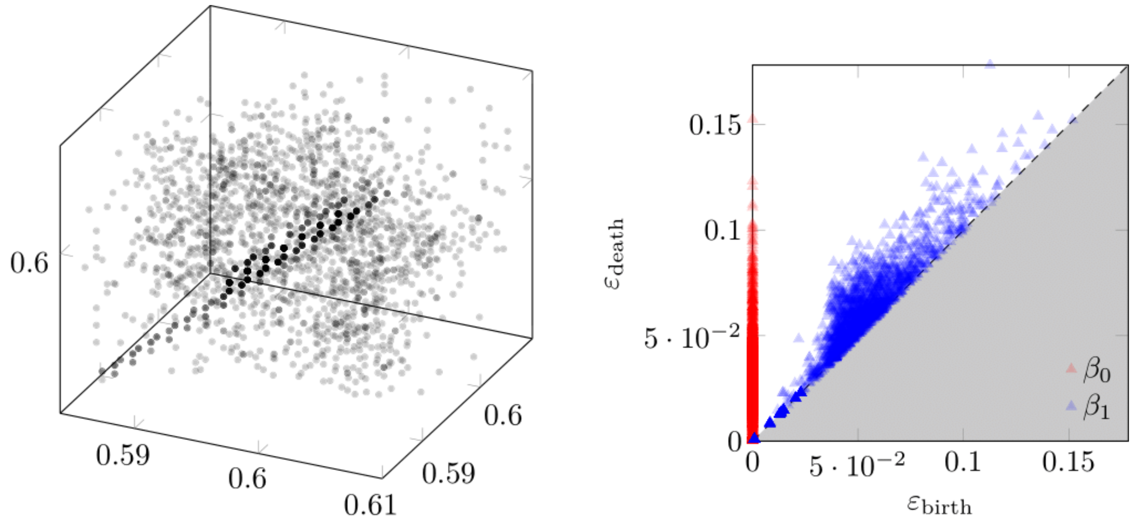
(c) $\Phi(L,\,12,\,3)$ (d) $\Phi(L,\,12,\,3)$ PH

Figure 8.11: Optimal EOV time-delay embeddings and their persistent homologies.

Figure 8.11 shows the optimal embeddings for the EOVs alongside their persistent homologies. The temperature embedding largely appears as a straight line, with this variance attributable to the large yearly cyclic trends which are not fully accounted for. Nonetheless, some features do persist in Figure 8.11 (b). On the other hand, the traffic-loading embedding shows a very interesting topology, with several connected clusters surrounding a central pronounced loop. However, the presence of noisy data points within the loop's centre is likely diminishing the loop's β_1 feature.



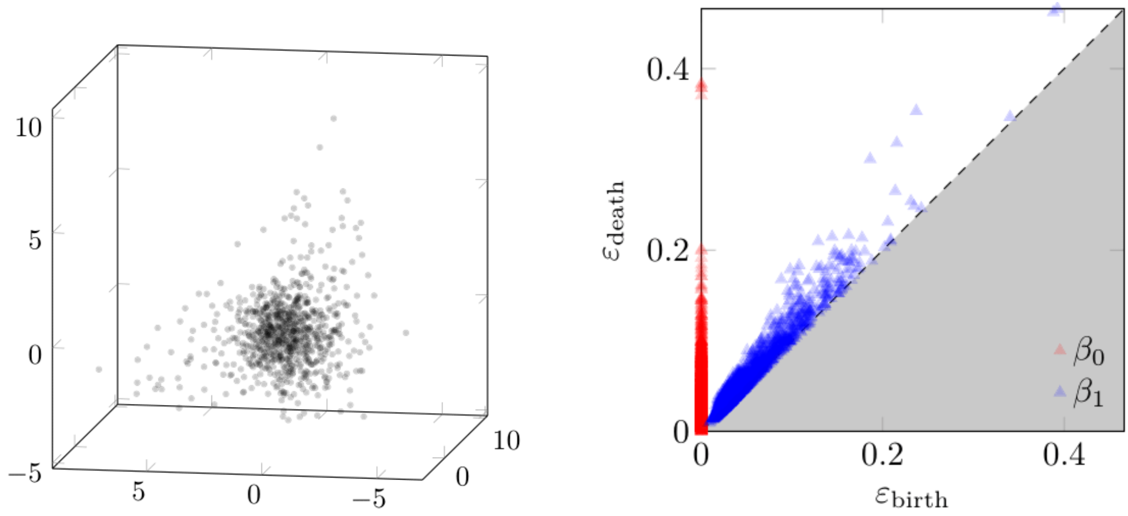
(a) $\Phi(T_3, 12, 3)$ (b) $\Phi(T_3, 12, 3)$ PH



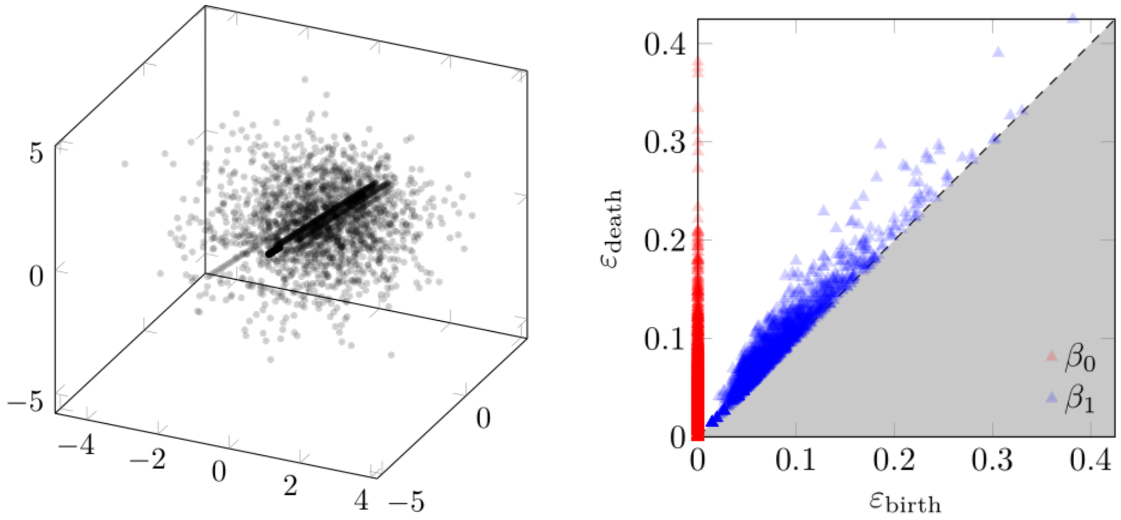
(c) $\Phi(\omega_3, 12, 3)$ (d) $\Phi(\omega_3, 12, 3)$ PH

Figure 8.12: Optimal structural parameter time-delay embeddings and their persistent homologies.

Figure 8.12 displays the optimal embeddings for the structural parameters. Given the strong correlation of T_3 with temperature, their embeddings look alike but rotated in space because of the negative correlation; this resemblance is reflected in their persistent homologies. The ω_3 embedding, on the other hand, resembles a uniformly-distributed random cluster, with some artefacts emerging from linearly-interpolated regions.



(a) $\Phi(\epsilon_T, 12, 3)$ (b) $\Phi(\epsilon_T, 12, 3)$ PH



(c) $\Phi(\epsilon_\omega, 12, 3)$ (d) $\Phi(\epsilon_\omega, 12, 3)$ PH

Figure 8.13: Optimal residual time-delay embeddings alongside their persistent homologies.

However, the most interesting embeddings are the cointegrated residuals. While the detailed cointegration calculation procedure is not reiterated here, having been covered in Chapter 7, it is worth noting that only the most stationary residuals are examined. Specifically, ϵ_T and ϵ_ω represent the residuals from the set of cable tensions and the set of deck frequencies, respectively.

The primary goal of this analysis is to identify any remaining topological features of a specific EOV in the embedded residual series. To ensure a consistent portrayal of topological features, the same embedding parameters are used for both the EOV and the residual series. This consistency aids in detecting any bleed-through EOV information into the residual series, allowing for a comparison of like topologies before and after cointegration. Conveniently, as indicated in Table 8.3, the consistent optimal embedding values are $\alpha^* = 12$ and $d^* = 3$.

The optimal residual embeddings depicted in Figure 8.13 are quite revealing. At first glance, the impact of cointegration is evident, with the residual series embeddings largely resembling Gaussian clusters. However, upon closer inspection, deviations from the anticipated Gaussian cluster emerge. Specifically, ϵ_T displays outliers and a slightly conic shape.

8.3 Cointegration Trend Removal

Linear cointegration is fundamentally a weighted sum of the input time series. Considering the subtle variations inherent in each time series, it is implausible that such a sum could eradicate all nonstationary features. Consequently, it is conjectured that some information will bleed through into the residual series, even when satisfying all cointegration requirements. This section aims to pronounce these residual features using topological techniques. If the topological signatures in the residual embedding align with those of an EOV embedding, it suggests that the nonstationarity associated with that EOV has not been removed.

In this context, a Gaussian cluster represents the ideal residual embedding, signifying the complete removal of all EOV nonstationarities. Therefore, a Gaussian cluster in 3D with unit variance and zero mean along every axis serves as the benchmark for a perfect cointegration outcome. However, a visual inspection of Figures 8.13 (a) and (c) reveals that these embeddings deviate ever so slightly from Gaussian clusters, indicating that some nonstationary information persists into the residual series.

To evaluate the effectiveness of cointegration in eliminating an EOV's topological features, the Wasserstein distances $\partial(\mathrm{EOV}, \epsilon)$ and $\partial(\mathrm{EOV}, \mathrm{GC})$ are compared. The comparison of these Wasserstein distances offers insight into the residual embedding's retention of EOV topological features. Given that the residual embedding is believed to retain minor topological influences from the EOVs, it follows that $\partial(\mathrm{EOV}, \mathrm{GC}) > \partial(\mathrm{EOV}, \epsilon)$. Therefore, by comparing the actual cointegration scenario to the idealised one, a ratio emerges that resembles an efficiency metric for

EOV removal, constrained between 0 and 1, stated as,

$$\eta_r(\text{EOV}, \sigma) = \frac{\partial_W(\text{EOV}, \epsilon_\sigma)}{\partial_W(\text{EOV}, \text{GC})} \tag{8.12}$$

Where σ represents a structural parameter and ϵ_σ is the cointegrated residual from the set of like structural parameters. By taking the complement of this ratio[4], $\eta_s = 1 - \eta_r$, the proportion of EOV information remaining in the actual scenario relative to the idealised is calculated.

However, equation (8.12) does not account for the initial influence of the EOV on the structural parameter. In order to gauge how much has been removed, there must first be an understanding of the initial state. As seen in Table 8.2, the effects of traffic loading and temperature vary on the structural parameters. Therefore, the dependency between the structural parameter and EOV should be incorporated. Consequently, the topological similarity $\partial_W(\text{EOV}, \sigma)$, is used to normalise the efficiency calculation in equation (8.12), mathematically given as,

$$\hat{\eta}_s(\text{EOV},\ \sigma) = 1 - \frac{\partial_W(\text{EOV},\ \epsilon_\sigma) + \partial_W(\text{EOV},\ \sigma)}{\partial_W(\text{EOV},\ \text{GC}) + \partial_W(\text{EOV},\ \sigma)} \tag{8.13}$$

In this formula, the normalisation term adopts a positive sign because smaller distances indicate greater similarity. However, for this analysis, the inverse relationship is sought, where greater topological dissimilarity receives a higher normalisation.

Repeating these calculations as described by equations (8.12) and (8.13) for multiple EOVs provides insights into the relative removal of EOVs from a structural parameter via topological means.

8.3.1 Results

Before addressing the quantification of EOV removal via cointegration, Table 8.4 presents the Wasserstein distances between the variables embedded at their optimal values.

From Table 8.4, it is evident that traffic loading possesses the most distinct topology, as indicated by its consistently larger Wasserstein distances relative to other entries; a distinction also apparent qualitatively. Additionally, the temperature and cable tension embeddings show a close alignment, with their persistent homologies

---

[4] The subscripts r and s are abbreviations for 'removed' and 'stays', respectively.

	θ	L	T_3	ω_3	ϵ_T	ϵ_ω	GC
θ	0.00	125.43	100.14	275.35	248.36	329.05	350.51
L	125.43	0.00	223.56	360.57	367.51	412.93	465.35
T_3	100.14	223.56	0.00	219.54	155.85	271.19	260.10
ω_3	275.35	360.57	219.54	0.00	158.20	114.68	166.63
ϵ_T	248.36	367.51	155.85	158.20	0.00	173.95	124.29
ϵ_ω	329.05	412.93	271.19	114.68	173.95	0.00	165.57
GC	350.51	465.35	260.10	166.63	124.29	165.57	0.00

Table 8.4: Wasserstein distance between each signal at their optimal embedding parameters.

having the smallest Wasserstein distance in the table. This close relationship can be attributed to their strong correlation, as detailed in Table 8.2.

Cable Tensions

From Table 8.4, the remaining topological features of θ and L in ϵ_T can be determined, by comparing against the idealised scenario, as calculated by,

$$\eta_s(\theta,\, T_3) = 1 - \frac{\partial_W(\theta,\, \epsilon_T)}{\partial_W(\theta,\, \mathrm{GC})} = 1 - \frac{248.36}{350.51} = 0.2914 \tag{8.14}$$

$$\eta_s(L,\, T_3) = 1 - \frac{\partial_W(L,\, \epsilon_T)}{\partial_W(L,\, \mathrm{GC})} = 1 - \frac{367.51}{465.35} = 0.2103 \tag{8.15}$$

These results suggest that temperature effects are more present in the ϵ_T embedding than traffic loading, since $\eta_s(\theta,\, T_3) > \eta_s(L,\, T_3)$. This observation suggests that the temperature effects are more challenging to remove than the traffic loading effects. One possible explanation is that the strong correlation between θ and the cable tensions causes these effects to be more ingrained in the cable tensions than L, and thus harder to remove. When qualitatively analysing Figure 8.13 (a), a subtle conical shape in the ϵ_T time-delay embedding was highlighted. This conical shape is likely to be a muted reflection of the elongated form present in the T_3 (and all other cable tensions) time-delay embedding. These topological remnants contribute to the relatively-small Wasserstein distance between the ϵ_T and θ embeddings.

Upon analysing the normalised versions of the calculations, the results are as follows,

$$\hat{\eta}_s(\theta,\, T_3) = 1 - \frac{\partial_W(\theta,\, \epsilon_T) + \partial_W(\theta,\, T_3)}{\partial_W(\theta,\, \mathrm{GC}) + \partial_W(\theta,\, T_3)} = 1 - \frac{248.36 + 100.14}{350.51 + 100.14} = 0.2267 \tag{8.16}$$

$$\hat{\eta}_s(L,\, T_3) = 1 - \frac{\partial_W(L,\, \epsilon_T) + \partial_W(L,\, T_3)}{\partial_W(L,\, \mathrm{GC}) + \partial_W(L,\, T_3)} = 1 - \frac{367.51 + 223.56}{465.35 + 223.56} = 0.1420 \tag{8.17}$$

The normalised results, like their unnormalised counterparts, indicate that $\hat{\eta}_s(\theta,\, T_3) > \hat{\eta}_s(L,\, T_3)$. However, the normalisation process, which accounts for the initial influence of the EOV on the structural parameter, accentuates the relative difference in the removal of θ and L via cointegration. This difference is because of the high topological similarity between T_3 and θ, thereby amplifying the relative difference between $\hat{\eta}_s(\theta,\, T_3)$ and $\hat{\eta}_s(L,\, T_3)$.

Deck Frequencies

For the deck frequencies, correlations to the EOVs are weaker. The weaker correlation implies that the topological influences of the EOVs in ω_3 are likewise subtle, making any topological remnants in the residual more challenging to identify. The ω_3 was selected for embedding as it shows the strongest correlation to traffic loading, with a value of -0.51. However, the average correlation of the other deck frequencies is -0.33, indicating a generally weak association between the deck frequencies and traffic loading. The temperature also bears a weak correlation to the deck frequencies.

Given the limited EOV contributions to the ω_3 embedded topology and the fact that cointegration further diminishes these effects, it is anticipated that the topological remnants of the EOVs in the residual series will be less pronounced than in the cable-tension embedding scenario. The extent of remaining traffic loading topology in the residual series is determined by,

$$\eta_s(\theta,\, \omega_3) = 1 - \frac{\partial_W(\theta,\, \epsilon_\omega)}{\partial_W(\theta,\, \text{GC})} = 1 - \frac{329.05}{350.51} = 0.0612 \tag{8.18}$$

$$\eta_s(L,\, \omega_3) = 1 - \frac{\partial_W(L,\, \epsilon_\omega)}{\partial_W(L,\, \text{GC})} = 1 - \frac{412.93}{465.35} = 0.1127 \tag{8.19}$$

These results suggest that the traffic loading effects are more pronounced in the ϵ_ω embedding, as indicated by $\eta_s(\theta,\, \omega_3) < \eta_s(L,\, \omega_3)$. When applying the normalisation term, the following results are obtained,

$$\hat{\eta}_s(\theta,\, \omega_3) = 1 - \frac{\partial_W(\theta,\, \epsilon_\omega) + \partial_W(\theta,\, \omega_3)}{\partial_W(\theta,\, \text{GC}) + \partial_W(\theta,\, \omega_3)} = 1 - \frac{329.05 + 275.35}{350.51 + 275.35} = 0.0343 \tag{8.20}$$

$$\hat{\eta}_s(L,\, \omega_3) = 1 - \frac{\partial_W(L,\, \epsilon_\omega) + \partial_W(L,\, \omega_3)}{\partial_W(L,\, \text{GC}) + \partial_W(L,\, \omega_3)} = 1 - \frac{412.93 + 360.57}{465.35 + 360.57} = 0.0635 \tag{8.21}$$

The normalisation mainly decreased the overall magnitudes of the values without significantly altering their relative values. These now-decreased values indicate that only a small amount of topology associated with θ and L was removed by cointegration, as there was so little present initially. However, the important point of including the deck frequency analysis is to highlight that the remaining EOV topological features in ϵ_ω are consistently lesser than the equivalent features in ϵ_T. Essentially, cointegration has to remove fewer nonstationarities in the deck frequencies than the cable tensions.

8.4 Conclusion

This chapter introduced a novel method for quantifying residual topological features of EOVs after cointegration, thereby assessing the performance of cointegration regarding specific EOVs. The approach hinged on the concept of an optimal time-delay embedding, designed to maximise and accurately represent the topologies of the signals. The presence of topological features associated with an EOV in the embedded residual indicated an incomplete removal of that trend via cointegration.

The novel approach for determining the optimal embedding delay focussed on maximising the loop size in the embedding by considering the signal's frequency components. Consequently, this procedure allows for a consistent method of maximising persistent intervals relating to the cyclic feature with the largest amplitude. Additionally, a pre-existing method was employed to determine the optimal embedding dimension by minimising the proportion of false-nearest neighbours in the embedding space. These systematic methods eliminate ambiguity in the selection of embedding parameters.

The optimal time-delay embeddings were then applied to four signals from the Tamar Bridge: temperature, traffic loading, cable tension, and deck frequency. The cointegrated residual series for the cable tensions and deck frequencies were subsequently determined and embedded. The time-delay embeddings for each signal showcased unique topologies, with varying degrees of similarity. When topological similarities between the EOV embedding and the residual series were present, this suggested the failure to purge this EOV from the residual series. These similarities were then compared to an idealised cointegration scenario.

In addition to comparing embedded residuals to an idealised cointegration scenario, this idea was extended to account for an EOV's topology present before cointegration. This second metric takes into account the initial contributions of the EOV to the structural parameter and normalises by this value. Therefore, allowing for an assessment of topological changes before and after cointegration, specific to an EOV, one quantifies the extent to which an EOV has been removed via cointegration.

Chapter 9

CONCLUSION

As structures grow in complexity and size, their safety and economic implications escalate accordingly. Even with the most meticulous design, aberrations are an inescapable reality over a structure's lifespan. Thus, implementing procedures to detect anomalies becomes a safety and economic necessity in mitigating disaster. In situations where critical decisions must be made with limited information, a panacea simply does not exist. Therefore, SHM choices – analogous to everyday decision-making – should draw from multiple information sources, weighing diverse viewpoints to reach the most informed conclusion.

While the methods employed in this thesis are esoteric, their applications are not unlike other contemporary engineering analyses. Much like software tools such as FEA, CFD, or TensorFlow, the abstruse theory is nestled away behind a computational interface. The topological and differential geometric methods used and created for this thesis are also encapsulated within computer programs, masking their abstract inner workings. Yet, the information elicited from these abstract methods is indeterminable via other means. So now such methods have been introduced and created, the focus transitions from the mathematical query of "How is this information obtained?" to an engineering mindset of "How can this information be leveraged for the greater good?".

The central theme of this thesis was the proposition of topological arguments as innovative and unique viewpoints to support SHM decisions. Rather than analysing data points in isolation or relying on conventional statistics, TDA distinguishes by focussing on the shape and connectivity of data. By interpreting structural mechanics as shapes, TDA provides unique insights into damage detection and other challenges in SHM. This thesis, while specific in its scope, highlighted the vast potential of TDA in SHM, and the broader realm of engineering. To adapt words by Darwin [135]: "There is grandeur in this view of *data*, with its *topological features*, having been originally breathed into a few *dimensions* or into one; and that, whilst *the topology* has gone cycling on according to the *persistent laws* of *homology*, from so simple a beginning endless forms most beautiful and most wonderful have been, and are being, *uncovered*."

Complementing the SHM aspects were nonlinear dynamical analyses. These concepts are crucial for deepening understanding of system modelling; ideas paramount in structural design and dynamics. Recognising the dire implications of structural failures, there is an impetus to develop high-fidelity models that can aptly capture complex nonlinear input-output behaviours. This research revitalised the generating series, enabling deep-series solutions to nonlinear differential equations via the abstract approach of shuffling words. This theoretical framework not only offers a greater understanding of nonlinear systems compared to numerical methods but also reformulates the problem for efficient and optimisable array manipulations. With the introduction of novel optimisations and the derivation for the impulse response, this malleable theory can be applied to much more complex systems within the literature, potentially casting light on longstanding challenges.

9.1 Chapter Summary

Chapter 1 laid the foundation for the thesis by introducing the two focal topics: structural health monitoring (SHM) and nonlinear dynamics. Subsequently, two theoretical approaches used in the thesis were discussed: topological data analysis (TDA) and the generating series. The chapter underscored the distinct insights provided by TDA and its potential to enhance SHM; specifically, the emphasis on TDA's capability to reveal concealed aspects of data to aid in SHM decision-making. The chapter wrapped up by outlining the key novel contributions of this study.

Chapter 2 presented the generating series approach for solving nonlinear differential equations. The chapter set the stage by discussing some abstract concepts such as noncommutative algebras and iterated integrals; followed by introducing the generating series framework. This framework primarily involves transforming a nonlinear differential equation into the generating series domain, conducting manipulations within this domain, and then inverse transforming back into the time domain to produce Volterra series solutions. To determine the impulse response solution within the framework, the process garnered a little more attention, and thus a novel derivation for the impulse response was established. Additionally, some novel optimisation strategies for computing higher-order terms in the Volterra series were unveiled. While some of these strategies applied to the generating series in general, the most impactful were specific to impulse excitation. These optimisations were pivotal in attaining such deep series solutions. The generating series approach was applied to a variant of Duffing's equation, revealing never-before-seen depth in the Volterra series for this system equation excited by an impulse. The results were discussed in detail and validated against an alternative method using contour integration.

Chapter 3 served as a comprehensive introduction to TDA, with a spotlight on persistent homology. Recognising persistent homology's distinctiveness from current SHM analyses, the theory was introduced *ab initio*; from rudiments like sets and groups to the homology of topological spaces. The chapter introduced reasoning through a rubber-sheet philosophy, emphasising topological understanding over traditional geometric rigidity. Central to topology are topological invariants, properties preserved under homeomorphisms, with homology emerging as the most important invariant of interest for this work. An illustrative example of the homology groups of a basic simplicial complex was provided to bridge between the mathematical and computational theory. These ideas then paved the way to introduce persistent homology built atop data, enabling the construction of topological invariants directly from data. Extending the concept of persistent homology was the introduction of the Wasserstein distance, a measure enabling the comparison of different datasets via a notion of topological similarity. This chapter concluded with some common pitfalls and flaws of TDA.

Chapter 4 showcased the first application of a purely topological approach to SHM. In this application, the Z24 dataset was partitioned based on categories defined by the temperature at each data point and the presence of damage. The chapter introduced a novel method to determine the topological uniqueness of a manifold in comparison to its counterparts. This method successfully identified the manifold of damage-state samples as the most distinct among the others, thereby demonstrating the feasibility of conducting SHM under the guise of topology. The chapter concluded by examining some additional cases, highlighting the methodology's resilience to data down-sampling and linear transformations.

Chapter 5 focussed on interfacing persistent homology with machine learning. In doing so, the standard unstructured form of the persistent homology was transformed into vector representations. These vectors enabled the inclusion of persistent homology features in a machine-learning classification scheme, demonstrating that the β_1 features can effectively classify sliding windows of data. This chapter was arguably the most important aspect of this thesis, highlighting the capability to integrate persistent homology features into machine-learning workflows.

Chapter 6 centred on strange attractors and their fractal dimensions, approached from a TDA angle. This chapter showed that even within the topological noise, valuable insights can be derived from the persistent homology features, specifically by determining the fractal dimensions of the attractors. To conclude, the chapter compared the topology of a complete attractor to its reconstruction, aiming to identify the delay that results in the most-topologically-accurate reconstruction into its original dimension.

Chapter 7 provided insight into analysing both linear and nonlinear cointegration schemes when applied to the Z24 Bridge. Via embedding the residual series, topological signatures were formed for specific features, revealing that distinct shapes appear in the residual series. These shapes are indicative of the information that remains in the residual series after cointegration, most notably the freezing region. The chapter also presented cointegration from a topological viewpoint, where the objective is to transform the embedded manifolds with topological features into a Gaussian cluster.

Chapter 8 built upon the previous chapter, aiming to quantify the removal of specific nonstationary trends from cointegrated time series using topological reasoning. In order to achieve this analysis, the Tamar bridge data set was introduced as multiple EOVs show significant effects on the structural parameters, particularly the temperature and traffic loading. The embedded residuals were compared to an idealised version of cointegration, indicating the presence of a topology specific to an EOV, thereby quantifying normalisation success regarding that EOV.

9.2 Limitations

It is important to recognise that the analyses presented in this thesis come with inherent limitations. Both the practical applications and the theoretical methodologies used face specific challenges. This section briefly provides an overview of some of the limitations encountered during this research.

One significant challenge with the concepts presented in this thesis is the unfamiliarity of abstract ideas and their foundations within engineering courses and literature. As a result, the very abstraction intended to cast new light on engineering problems might act as the gatekeeper to the broader application of these ideas. This point is especially important when alternative methods with a greater theory overlap with the norm are more convenient, faster and often more accurate. For TDA and generating series to become staples in engineering, there would need to be a shift towards embracing more abstract concepts. This action would demand time and effort that might be better spent addressing problems to a satisfactory standard using established methods, rather than extracting the marginal additional insights these abstract methods might offer. However, in scenarios like SHM, where available information is limited and stakes are high, these abstract methods become more topical. The question then becomes, "How much information is needed for SHM decisions?" as introducing new analytical methods increases both complexity and cost, so it is essential to strike a balance between the depth of analysis and practicality.

As hinted at in Chapter 3, there are philosophical quandaries associated with this analysis. TDA was shown as powerful in scenarios where point cloud data forms an interesting topology, however, it is not a given that such features will emerge under the incursion of damage or the influence of EOVs. Furthermore, interpreting what a hole or cavity in the data signifies is not straightforward, and is likely to be different for all structures. There is no direct method to link specific topological shapes to structural dynamics or particular damage cases. Factors such as temperature and traffic loading influence modal parameters differently for different bridge types, materials, and geometries. All of these effects will influence the topology to some degree, but how is unclear. The Z24 Bridge example even showed different topologies forming within the time-delay embeddings of different natural frequencies under the influence of temperature. Establishing a connection between topology and structural happenings would likely be unique to each structure and would require an extensive catalogue of shapes corresponding to various features. Additionally, understanding the interaction of these topologies when they coexist presents another layer of complexity.

Another major limitation of TDA is all the challenges stemming from the requirement of many points to form a topology of a point cloud. Inherently, a single point is 0-dimensional, two points are limited to forming a straight line, and so forth. It is only when an ambiguous and arbitrary number of points are present that a meaningful and characteristic topology of data emerges. The inability to analyse a single point via TDA comes with a myriad of problems. Labelling becomes nontrivial as mixed labels are almost inevitable, necessitating coarser labelling strategies such as ‘undamaged’ or ‘damaged’; sacrificing the depth of information from other varying parameters over multiple points, like temperature and traffic count. Additionally, using multiple points can delay the detection of topological effects in analyses, like those using sliding windows. This delay might result in a slower identification of damage compared to traditional methods.

However, the most significant detriment of having numerous points and determining their connections lies in the combinatorial complexity. A significant issue in this work was that calculating the persistent homologies of point clouds with more than 2000 points proved to be challenging, restricting the possible shapes to such few points castrates the theoretical limits of TDA. The computation expense was evident even when considering only the β_1 features, consequently, all the analysis in this thesis was restricted to β_0 and β_1 homological features, despite manifolds being embedded in $\mathbb{R}^3$ and $\mathbb{R}^4$. As mentioned in the footnotes of Chapter 4, for the warm partition with 2089 points it was infeasible to determine the β_2 features, whereas the others were attainable. Consequently, the inability to form information for the single large partition, all the β_2 features had to be excluded for consistency. Several analyses were truncated because of the time-consuming nature of persistent homology calculations. This situation presents a dichotomy between the desire for a well-defined topology and the feasibility of calculating persistent homology.

9.3 Future Work

The topological methods discussed in this thesis are by no means exhaustive. The field is rapidly evolving, with new and unique applications and methodologies emerging across various disciplines. Given the size and time constraints of this thesis, it was not feasible to address every research topic. Hence, this section highlights some potential areas for future research. While specific recommendations for subsequent research based on the findings of this thesis were often discussed *in situ*, the topics listed here provide a broader extension of TDA and its application to SHM.

- Fundamental to all persistent homology calculations is the extraction method of simplicial complexes from data. An interesting avenue to explore is the adoption of a less metric-centred approach to forming simplicial complexes. Instead of using the Vietoris-Rips complex, one could consider extracting simplicial complexes from data by using a *reciprocal nearest-neighbours* approach; where two points are called k-reciprocal nearest neighbours of each other if they are among each other's k-nearest neighbours [89].

  Such an approach would shift the emphasis from a strict reliance on distance to a more topological concept of closeness; essentially making topological data analysis more topological. The persistent homology derived from such a method, instead of being parameterised by a continuous ε would be parameterised by k, the number of neighbours. Therefore, removing the strong reliance of Vietoris-Rips complexes on the Euclidean distance, favouring a more flexible, rubber-sheet-like topological interpretation.

  This approach would also respond better to outliers. An outlier is unlikely to be the nearest neighbour of a nonoutlier, meaning they would not be reciprocal nearest neighbours; thus simplex formation would be limited for outliers. Since a nonoutlier is often surrounded by other nonoutliers, the likelihood of forming a connection to an outlier is reduced. Consequently, a simplex to the outlier is less likely to form, making this method more resilient to such anomalies.

- One slight twist on persistent homology involves not directly computing the persistent homology over the point cloud, but instead focussing on the persistent homology of correlation values. This method could be particularly useful when applied to a set of damage-sensitive structural parameters that show EOV effects, such as natural frequencies. By considering the correlations of sliding windows the common-correlated nonstationarities such as temperature and traffic loading would be removed. These correlation values are then used as a distance matrix to compute persistent homology, offering topological insights based on the extracted information, diverging from the direct manifold embedding approach used in this thesis.

The inspiration for these ideas comes from work by Gidea [136], which employed a similar approach to detect early indicators of critical transitions of the 2008 stock market crash.

- Using product topologies via the Künneth formula might be informative in understanding topologies where structural parameters are influenced by multiple EOVs. For instance, considering the Tamar Bridge's cable tensions, forming a product topology that combines the effects of traffic count and temperature, weighted according to their respective contributions to the structural parameter, could provide valuable insights. By adopting this approach, it might become possible to predict the topology of the structural parameter based on the EOVs, possibly with the assistance of generative machine learning techniques. Any deviation between the observed topology and the predicted one would suggest an unaccounted factor or event, possibly indicating damage.
- While topological methods may never supplant machine learning, the fusion of incorporating persistent homology vector representations alongside traditional features warrants further consideration. This combined approach could offer invaluable insights enhancing damage detection, localisation, and classification capabilities.
- For the generating series method, an extension could explore matrix methods over generating series for a multi-degree-of-freedom system. By employing caching optimisations, the expansion might not substantially increase computational complexity, especially since many shuffle expansions would likely be identical. This approach offers the advantage of modelling extra dimensionality without a significant computational burden.

Appendix A

y_3 IN THE VOLTERRA SERIES

Section 2.3 lists the terms y_1 and y_2 determined via the generating series method. Meanwhile, Appendix B shows terms y_1, y_2, and y_3 determined via contour integration. To facilitate a direct comparison between the first three terms of the generating series and contour integration approaches to the same problem, y_3 from the generating series is listed as,

$$
\begin{aligned}
y_3 = &-\frac{A^3k_2^2e^{-3a_2t}}{2a_1^6a_2-19a_1^5a_2^2+72a_1^4a_2^3-140a_1^3a_2^4+148a_1^2a_2^5-81a_1a_2^6+18a_2^7}\\
&-\frac{A^3k_2^2e^{-3a_2t}}{2a_1^6a_2-21a_1^5a_2^2+89a_1^4a_2^3-195a_1^3a_2^4+233a_1^2a_2^5-144a_1a_2^6+36a_2^7}\\
&-\frac{4A^3k_2^2e^{-2a_1t}}{-4a_1^6a_2+24a_1^5a_2^2-55a_1^4a_2^3+60a_1^3a_2^4-31a_1^2a_2^5+6a_1a_2^6}\\
&+\frac{3A^3k_2^2e^{t(-a_1-2a_2)}}{a_1^5a_2^2-4a_1^4a_2^3+4a_1^3a_2^4+2a_1^2a_2^5-5a_1a_2^6+2a_2^7}\\
&+\frac{A^3k_2^2e^{-3a_2t}}{a_1^5a_2^2-9a_1^4a_2^3+31a_1^3a_2^4-51a_1^2a_2^5+40a_1a_2^6-12a_2^7}\\
&+\frac{A^3k_2^2e^{-3a_1t}}{-6a_1^7+23a_1^6a_2-34a_1^5a_2^2+24a_1^4a_2^3-8a_1^3a_2^4+a_1^2a_2^5}\\
&-\frac{12A^3k_2^2e^{-2a_1t}}{4a_1^6a_2-20a_1^5a_2^2+35a_1^4a_2^3-25a_1^3a_2^4+6a_1^2a_2^5}\\
&-\frac{3A^3k_2^2e^{-a_2t}}{-6a_1^6a_2+5a_1^5a_2^2+5a_1^4a_2^3-5a_1^3a_2^4+a_1^2a_2^5}\\
&-\frac{4A^3k_2^2e^{-2a_2t}}{2a_1^5a_2^2-9a_1^4a_2^3+14a_1^3a_2^4-9a_1^2a_2^5+2a_1a_2^6}\\
&-\frac{2A^3k_2^2e^{t(-a_1-2a_2)}}{a_1^4a_2^3-3a_1^3a_2^4+a_1^2a_2^5+3a_1a_2^6-2a_2^7}
\end{aligned}
\tag{A.1}
$$

$$
\begin{aligned}
&-\frac{2A^3k_2^2e^{-a_1t}}{a_1^4a_2^3-5a_1^3a_2^4+5a_1^2a_2^5+5a_1a_2^6-6a_2^7}\\
&+\frac{A^3k_2^2e^{t(-2a_1-a_2)}}{-2a_1^7+3a_1^6a_2+a_1^5a_2^2-3a_1^4a_2^3+a_1^3a_2^4}\\
&+\frac{A^3k_2^2e^{-a_2t}}{2a_1^6a_2-a_1^5a_2^2-2a_1^4a_2^3+a_1^3a_2^4}-\frac{A^3k_2^2e^{t(-2a_1-a_2)}}{a_1^6a_2-a_1^5a_2^2-a_1^4a_2^3+a_1^3a_2^4}\\
&+\frac{3A^3k_2^2e^{t(-2a_1-a_2)}}{a_1^6a_2-2a_1^5a_2^2+2a_1^3a_2^4-a_1^2a_2^5}+\frac{3A^3k_2^2e^{-a_1t}}{a_1^5a_2^2-3a_1^4a_2^3-a_1^3a_2^4+3a_1^2a_2^5}\\
&+\frac{12A^3k_2^2e^{t(-a_1-a_2)}}{-2a_1^5a_2^2+7a_1^4a_2^3-7a_1^3a_2^4+2a_1^2a_2^5}+\frac{2A^3k_2^2e^{t(-a_1-2a_2)}}{a_1^4a_2^3-a_1^3a_2^4-a_1^2a_2^5+a_1a_2^6}\\
&+\frac{2A^3k_2^2e^{-a_1t}}{a_1^4a_2^3-3a_1^3a_2^4-a_1^2a_2^5+3a_1a_2^6}-\frac{A^3k_2^2e^{-a_2t}}{-a_1^5a_2^2+a_1^3a_2^4}\\
&+\frac{A^3k_3e^{-3a_2t}}{2a_1^4a_2-12a_1^3a_2^2+24a_1^2a_2^3-20a_1a_2^4+6a_2^5}\\
&+\frac{A^3k_3e^{-3a_1t}}{6a_1^5-20a_1^4a_2+24a_1^3a_2^2-12a_1^2a_2^3+2a_1a_2^4}\\
&-\frac{3A^3k_3e^{-a_2t}}{6a_1^4a_2-2a_1^3a_2^2-6a_1^2a_2^3+2a_1a_2^4}+\frac{3A^3k_3e^{t(-a_1-2a_2)}}{2a_1^4a_2-4a_1^3a_2^2+4a_1a_2^4-2a_2^5}\\
&-\frac{3A^3k_3e^{-a_1t}}{2a_1^4a_2-6a_1^3a_2^2-2a_1^2a_2^3+6a_1a_2^4}+\frac{3A^3k_3e^{t(-2a_1-a_2)}}{-2a_1^5+4a_1^4a_2-4a_1^2a_2^3+2a_1a_2^4}\\
&+\frac{\left(-4A^3a_1^2k_2^2+8A^3a_1a_2k_2^2-12A^3a_2^2k_2^2\right)e^{t(-a_1-a_2)}}{a_1^2a_2^2\left(a_1+a_2\right)\left(a_1^4-6a_1^3a_2+13a_1^2a_2^2-12a_1a_2^3+4a_2^4\right)}\\
&+\frac{\left(12A^3a_1^2k_2^2-16A^3a_1a_2k_2^2-4A^3a_2^2k_2^2\right)e^{t(-a_1-a_2)}}{a_1^2a_2^2\left(a_1+a_2\right)\left(a_1^4-6a_1^3a_2+13a_1^2a_2^2-12a_1a_2^3+4a_2^4\right)}\\
&+\frac{\begin{array}{c}\left(t\left(-a_1-a_2\right)+1\right)\left(-4A^3a_1^2a_2^2k_2^2\left(a_1+a_2\right)\right.\\ \left(a_1^4-6a_1^3a_2+13a_1^2a_2^2-12a_1a_2^3+4a_2^4\right)\\ -a_1^2a_2\left(-4A^3a_1^2k_2^2+8A^3a_1a_2k_2^2-12A^3a_2^2k_2^2\right)\\ \left.\left(a_1^4-6a_1^3a_2+13a_1^2a_2^2-12a_1a_2^3+4a_2^4\right)\right)\end{array}}{a_1^4a_2^3\left(a_1+a_2\right)\left(a_1^4-6a_1^3a_2+13a_1^2a_2^2-12a_1a_2^3+4a_2^4\right)^2}e^{t(-a_1-a_2)}\\
&+\frac{\begin{array}{c}\left(t\left(-a_1-a_2\right)+1\right)\left(a_1^2a_2^2\left(a_1+a_2\right)\left(8A^3a_1k_2^2-12A^3a_2k_2^2\right)\right.\\ \left(a_1^4-6a_1^3a_2+13a_1^2a_2^2-12a_1a_2^3+4a_2^4\right)\\ +a_1^2a_2^2\left(-12A^3a_1^2k_2^2+16A^3a_1a_2k_2^2+4A^3a_2^2k_2^2\right)\\ \left.\left(a_1^4-6a_1^3a_2+13a_1^2a_2^2-12a_1a_2^3+4a_2^4\right)\right)\end{array}}{a_1^4a_2^4\left(a_1+a_2\right)\left(a_1^4-6a_1^3a_2+13a_1^2a_2^2-12a_1a_2^3+4a_2^4\right)^2}e^{t(-a_1-a_2)}
\end{aligned}
$$

Appendix B

CONTOUR INTEGRATION APPROACH FOR THE VOLTERRA KERNELS

For the same system described in Section 2.3, the Volterra terms are derived using a contour integration method [25]. These terms are expressed in relation to the damping ratio $\zeta = \frac{c}{2\sqrt{k_1 m}}$, the natural frequency $\omega_n = \sqrt{\frac{k}{m}}$, and the damped natural frequency $\omega_d = \omega_n\sqrt{1-\zeta^2}$. The terms derived using this method are listed as,

$$y_1(t) = \frac{A}{m\omega_d}\exp^{-\zeta\omega_n t}\sin\left(\omega_d t\right) \tag{B.1}$$

$$\begin{aligned} y_2(t) = {} & \frac{A^2 k_2}{2m^3\omega_d^2\omega_n^2}\left\{e^{-\zeta\omega_n t}\cos\left(\omega_d t\right) - e^{-2\zeta\omega_n t}\right\} + \frac{A^2 k_2}{2m^3\omega_d^2\omega_n^2\left(8\omega_d^2+\omega_n^2\right)}\times \\ & \left\{|C|e^{-\zeta\omega_n t}\cos\left(\omega_d t + \angle C\right) - |B|e^{-2\zeta\omega_n t}\cos\left(2\omega_d t + \angle B\right)\right\} \end{aligned} \tag{B.2}$$

where,

$$B = \left(-3\omega_d + i\zeta\omega_n\right)\left(-\omega_d + i\zeta\omega_n\right) \quad \text{and} \quad C = \left(3\omega_d + i\zeta\omega_n\right)\left(\omega_d + i\zeta\omega_n\right) \tag{B.3}$$

and finally,

$$\begin{aligned}
y_3^{k_2}(t) = {} & \frac{A^3k_2^2|G|e^{-3\zeta\omega_n t}}{8m^5\left(\omega_d^3\right)\left(\omega_n^4\right)\left(3\omega_d^2+\omega_n^2\right)\left(8\omega_d^2+\omega_n^2\right)}\cos\left(3\omega_d t+\angle G\right) \\
& +\frac{A^3k_2^2|H|e^{-2\zeta\omega_n t}}{2m^5\left(\omega_d^3\right)\left(\omega_n^4\right)\left(8\omega_d^2+\omega_n^2\right)}\cos\left(2\omega_d t+\angle H\right)+\frac{2A^3k_2^2\zeta e^{-2\zeta\omega_n t}}{m^5\left(\omega_d^2\right)\left(\omega_n^3\right)\left(8\omega_d^2+\omega_n^2\right)} \\
& +\frac{A^3k_2^2|J|e^{-3\zeta\omega_n t}}{8m^5\zeta\left(\omega_d^3\right)\left(\omega_n^5\right)\left(8\omega_d^2+\omega_n^2\right)}\cos\left(\omega_d t+\angle J\right)-\frac{A^3k_2^2e^{-\zeta\omega_n t}}{4m^5\zeta\left(\omega_d^2\right)\left(\omega_n^5\right)}\cos\left(\omega_d t\right) \\
& +\frac{A^3k_2^2|K|e^{-\zeta\omega_n t}}{8m^5\zeta\left(\omega_d^3\right)\left(\omega_n^5\right)\left(3\omega_d^2+\omega_n^2\right)\left(8\omega_d^2+\omega_n^2\right)}\cos\left(\omega_d t+\angle K\right)
\end{aligned} \tag{B.4}$$

where,

$$G = 17\omega_d^3\zeta\omega_n - 7\omega_d\zeta^3\omega_n^3 + i6\omega_d^4 - i17\omega_d^2\zeta^2\omega_n^2 + i\zeta^4\omega_n^4 \tag{B.5}$$

$$H = -4\omega_d\zeta\omega_n - i3\omega_d^2 + i\zeta^2\omega_n^2 - i\omega_n^2 \tag{B.6}$$

$$J = 13\omega_d^3 + 2\omega_d\omega_n^2 + 5\omega_d\zeta^2\omega_n^2 - i9\omega_d^2\zeta\omega_n - i2\zeta\omega_n^3 - i\zeta^3\omega_n^3 \tag{B.7}$$

$$\begin{aligned}
K = {} & 9\omega_d^5 - 32\omega_d^3\zeta^2\omega_n^2 + 3\omega_d^3\omega_n^2 + 7\omega_d\zeta^4\omega_n^4 - 5\omega_d\zeta^2\omega_n^4 + i27\omega_d^4\zeta\omega_n \\
& -i20\omega_d^2\zeta^3\omega_n^3 - i\zeta^3\omega_n^5 + i\zeta^5\omega_n^5 + i7\omega_d^2\zeta\omega_n^3
\end{aligned} \tag{B.8}$$

Corrections are made to certain terms, highlighted in red, which deviate from the presented form in the paper because of typographical errors. In the referenced paper [25], the exponent in red is given as 2, and the final term of K is absent. The amended form originates from Dr G Manson's MATLAB code, supplied by Dr T Rogers. This amended form is consistently used in all Volterra series calculations in this thesis.

Appendix C

SUCCESSIVE BOUNDARY MAP PROOF

To show that two successive boundary maps give the result that the chain complex is equal to zero, it is sufficient to start with a standard simplex, as the linearity relations are shown in Definition 3.5.4 and a simplicial complex is the sum of multiple standard simplices.

$$\partial_k \partial_{k+1} \Delta^k = 0 \tag{C.1}$$

Since the simplex is characteristic of the set of vertices $\Delta^k = [v_0, \ldots, v_k]$, this can be substituted in place, then with manipulation, the following result is obtained.

Proof.

$$\begin{aligned}
\partial_k \partial_{k+1} \Delta^k &= \partial_k \partial_{k+1} [v_0, \ldots, v_k] \\
&= \partial_k \left\{ \sum_{j=0}^{k} (-1)^j [v_0, \ldots, \hat{v}_j, \ldots, v_k] \right\} \\
&= \sum_{j=0}^{k} (-1)^j \partial_k [v_0, \ldots, \hat{v}_j, \ldots, v_k] \\
&= \sum_{j=0}^{k} (-1)^j \big[\sum_{i=0}^{j-1} (-1)^i [v_0, \ldots, \hat{v}_i, \ldots, \hat{v}_j, \ldots, v_k] \\
&\qquad + \sum_{i=j+1}^{k} (-1)^{i-1} [v_0, \ldots, \hat{v}_j, \ldots, \hat{v}_i, \ldots, v_k] \big]
\end{aligned}$$

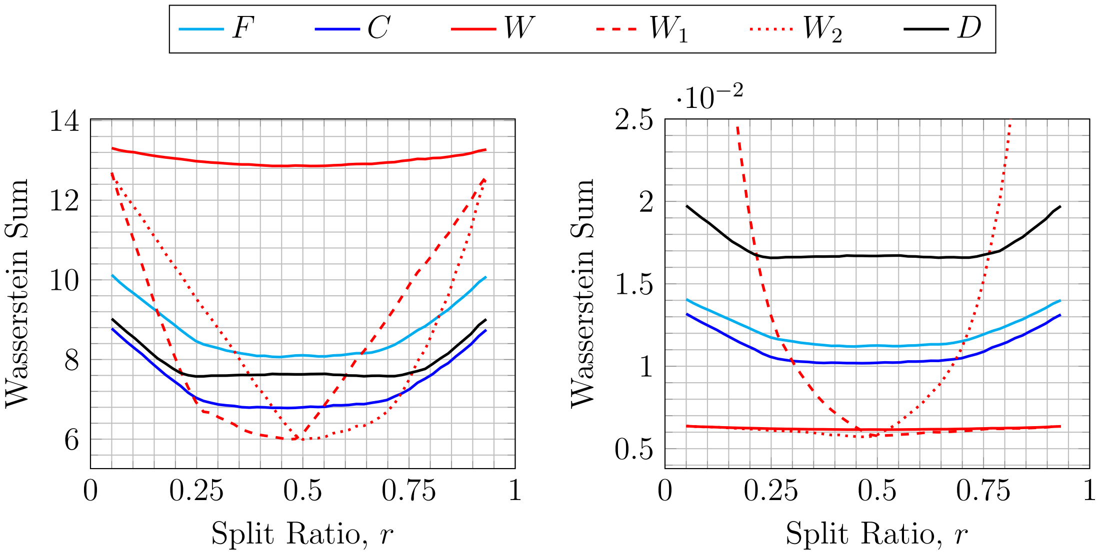
$$
\begin{aligned}
&= \sum_{i<j} (-1)^{i+j}[v_0, \dots, \hat{v}_i, \dots, \hat{v}_j, \dots, v_k] \\
&\quad + \sum_{i>j} (-1)^{i+j-1}[v_0, \dots, \hat{v}_j, \dots, \hat{v}_i, \dots, v_k] \\
&= \sum_{i<j} \left\{ (-1)^{i+j} + (-1)^{i+j-1} \right\} [v_0, \dots, \hat{v}_i, \dots, \hat{v}_j, \dots, v_k] \\
&= 0
\end{aligned}
$$

This result is reliant on $(-1)^{i+j} + (-1)^{i+j-1} = (-1)^{i+j}(1 + (-1)^1) = 0$, therefore each term is out of phase, meaning $\forall i, j$ this result is equal to zero. □

Appendix D

Wasserstein Partition Size Dependency

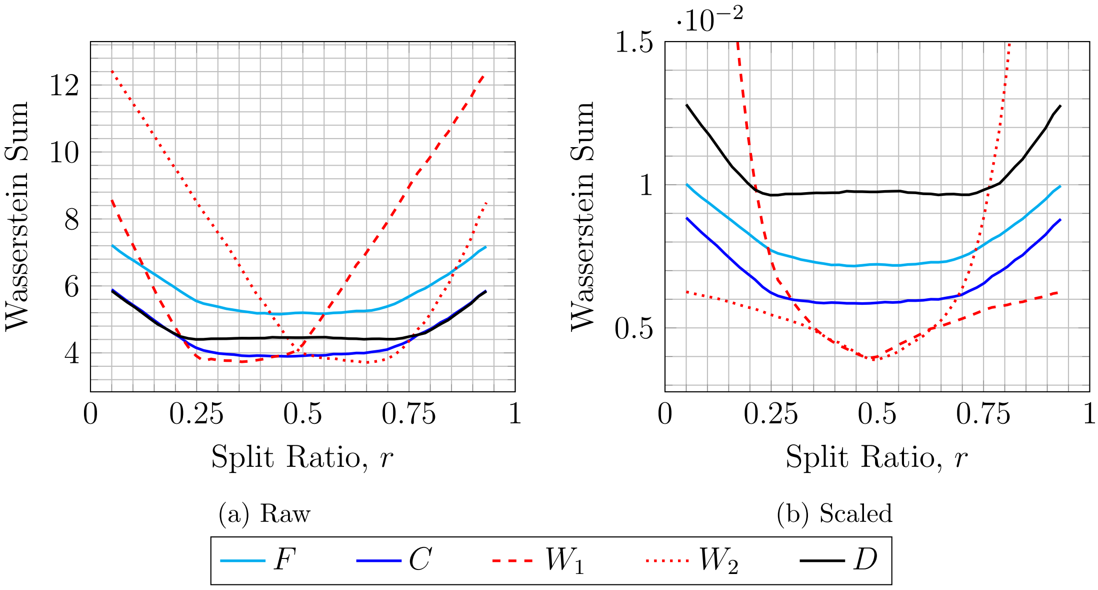
To supplement the Wasserstein partitions varying size plots given in Section 4, the symmetric versions, where the complement set W_2 are listed here.



(a) Raw

(b) Scaled

Figure D.1: Size of the WD values depending on the size of the warm partition size.



(a) Raw

(b) Scaled

Figure D.2: Size of the WD values depending on the size of the warm partition size.

Appendix E

GENERAL TIME SERIES

E.1 Foundational Time Series Concepts

To provide context for time series theory, this section outlines fundamental concepts and definitions. These serve as supplementary material to the primary content in Chapters 7 and 8.

Definition E.1.1. A *stationary* time-series stays in statistical equilibrium, in which the mean and variance do not change [137].

Cointegration aims to convert a set of multiple *nonstationary* time-series, those being time series that do not obey the criteria outlined in Definition E.1.1. For cointegration, it is required to confirm that all the time series are nonstationary to the same degree. In other words, the time series are integrated of the same order. To verify this assertion, the idea of *differencing* is required.

Definition E.1.2. *Differencing* is a method to remove trends from a time series.

$$\nabla y_t = y_t - y_{t-1} = (1 - B) y_t, \tag{E.1}$$

where B is the backward shift operator. The k^{th} difference is determined by,

$$\nabla^k y_t = (1 - B)^k y_t. \tag{E.2}$$

The k^{th} difference removes k^{th} degree polynomial trends.

Finally, the last required foundational concept for cointegration is a *Vector Autoregressive model*, used to model time series in the Johansen procedure, detailed in Section 7.1.3.

Definition E.1.3. *Autoregressive* (AR) models are models that are influenced by the previous values within the time series.

$$y_t = \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + \cdots + \alpha_p y_{t-p} + \epsilon_t = \epsilon_t + \sum_{i}^{p} \alpha_i y_{t-i}, \tag{E.3}$$

where α_i is a weight, and ϵ is sampled from a white noise process. A time series that satisfies equation E.3 is said to be an *autoregressive process of order* p, denoted AR(p). For example, AR(1) refers to an autoregressive process where the regression is solely on the preceding value.

Definition E.1.4. A *vector autoregressive* (VAR) model is an extension of an AR model for m time series. A VAR model pertains to the form,

$$Y_t = \Phi_1 Y_{t-1} + \cdots + \Phi_p Y_{t-p} + \mu_t = \mu_t + \sum_{i=1}^{p} \Phi_i Y_{t-i} \tag{E.4}$$

Where Y_t is a $m \times 1$ vector representing the states at time t, Φ_i represent the $m \times m$ matrices of coefficients, and μ_t represents an $m \times 1$ vector of white noise processes. VAR models generalise univariate AR models by considering more than one evolving variable. Each variable within a VAR is explained by its own lags and the lags of all other variables in the model [137]; therefore capturing multivariate processes.

Appendix F

PUBLICATIONS

Journal Papers

- T. Gowdridge, N. Dervilis, K. Worden. *On Topological Data Analysis for Structural Dynamics: An Introduction to Persistent Homology* (2022) ASME Open Journal of Engineering.

Conference Papers

- T. Gowdridge, G. Manson, N. Dervilis, K. Worden. *On the use of Generating Series for the Impulse Response of Duffing's Equation* (2024) Proceedings of the 42nd IMAC.
- T. Gowdridge, E.J. Cross, N. Dervilis, K. Worden. *On Quantifying Data Normalisation via Cointegration with Topological Methods* (2023) Proceedings of the 41st IMAC.
- T. Gowdridge, E.J. Cross, N. Dervilis, K. Worden. *A Topological Analysis of Cointegrated Data: A Z24 Bridge Case Study* (2022) European Workshop on Structural Health Monitoring, Volume 2.
- T. Gowdridge, N. Dervilis, K. Worden. *On the application of topological data analysis: a Z24 Bridge case study* (2021) International Workshop on Structural Health Monitoring

- T. Gowdridge, N. Dervilis, K. Worden. *On Topological Data Analysis for SHM: An Introduction to Persistent Homology* (2020) Proceedings of the 39th IMAC.
- T. Gowdridge, N. Dervilis, K. Worden. *On the Application of the Generating Series for Nonlinear Systems with Polynomial Stiffness* (2020) Proceedings of the 39th IMAC.

BIBLIOGRAPHY

[1] C.R. Farrar and K. Worden. *Structural Health Monitoring: A Machine Learning Perspective.* John Wiley & Sons, 2012.

[2] A. Rytter. *Vibrational Based Inspection of Civil Engineering Structures.* PhD thesis, Aalborg University, 1993.

[3] K. Worden and J.M. Dulieu-Barton. An overview of intelligent fault detection in systems and structures. *Structural Health Monitoring*, 3(1):85–98, 2004.

[4] E.J. Cross, T.J. Rogers, and T.J. Gibbons. Grey-box modelling for structural health monitoring: physical constraints on machine learning algorithms. *Structural Health Monitoring*, 2019.

[5] C.R. Farrar and K. Worden. An introduction to structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2007.

[6] J.F. Manwell, J.G. McGowan, and A.L. Rogers. *Wind Energy Explained: Theory, Design and Application.* John Wiley & Sons, 2010.

[7] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. In *Proceedings 41st Annual Symposium on Foundations of Computer Science.* IEEE, 2002.

[8] K. Worden and G.R. Tomlinson. *Nonlinearity in Structural Dynamics: Detection, Identification and Modelling.* Institute of Physics, 2001.

[9] M. Fliess. Fonctionnelles causales non linéaires et indéterminées non commutatives. *Bulletin de la Société Mathématique de France*, 109:3–40, 1981.

[10] S. Eilenberg and S. Mac Lane. On the groups $H(\Pi, n)$, I. *Annals of Mathematics(2)*, 58(1):55–106, 1953.

[11] G. Duffing. *Erzwungene Schwingungen bei Veränderlicher Eigenfrequenz und ihre technische Bedeutung.* F. Vieweg & Sohn, 1918.

[12] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, and J.R. Chipperfield. *Numerical Recipes – The Art of Scientific Computing.* Cambridge University Press, 1986.

[13] M. Fliess, M. Lamnabhi, and F. Lamnabhi-Lagarrigue. An algebraic approach to nonlinear functional expansions. *Transactions on Circuits and Systems*, 30 (8):554–570, 1983.

[14] M. Lamnabhi. A new symbolic calculus for the response of nonlinear systems. *Systems & Control Letters*, 2(3):154–162, 1982.

[15] F. Lamnabhi-Lagarrigue and M. Lamnabhi. Algebraic computation of the solution of some nonlinear differential equations. In *European Computer Algebra Conference*, pages 204–211. Springer, 1982.

[16] Y. Li and W S. Gray. The formal Laplace-Borel transform, Fliess operators and the composition product. In *Proceedings of the Thirty-Sixth Southeastern Symposium on System Theory*, pages 333–337. IEEE, 2004.

[17] F. Lamnabhi-Lagarrigue and M. Lamnabhi. Algebraic computation of the statistics of the solution of some nonlinear stochastic differential equations. In *European Conference on Computer Algebra*, pages 55–67. Springer, 1983.

[18] M. Fliess and F. Lamnabhi-Lagarrigue. Application of a new functional expansion to the cubic anharmonic oscillator. *Journal of Mathematical Physics*, 23(4):495–502, 1982.

[19] M. Lamnabhi. Functional analysis of nonlinear circuits: a generating power series approach. In *IEE Proceedings H (Microwaves, Antennas and Propagation)*, volume 133, pages 375–384. IET, 1986.

[20] R. Ree. Lie elements and an algebra associated with shuffles. *Annals of Mathematics*, pages 210–220, 1958.

[21] F. Lamnabhi-Lagarrigue. *Application Des Aariables Noncommutatives à*

des Calculs Formels en Statistique Nonlinéaire. PhD thesis, University of Paris—South, 1980.

[22] K.T Chen, R.H Fox, and R.C Lyndon. Free differential calculus, iv. the quotient groups of the lower central series. *Annals of Mathematics*, pages 81–95, 1958.

[23] V. Volterra. *Theory of Functionals and of Integral and Integro-Differential Equations.* Blackie & Son Limited, 1930.

[24] M. Schetzen. Nonlinear system modeling based on the Wiener theory. *Proceedings of the IEEE*, 69(12):1557–1573, 1981.

[25] G. Manson, K. Worden, and P.I. Reed. Analysis of nonlinear system response to an impulse excitation. In *Nonlinear Dynamics, Volume 2: Proceedings of the 32nd IMAC, A Conference and Exposition on Structural Dynamics*, pages 297–308. Springer, 2014.

[26] K. Worden, G. Manson, and G.R. Tomlinson. A harmonic probing algorithm for the multi-input Volterra series. *Journal of Sound and Vibration*, 201(1): 67–84, 1997.

[27] E. Bedrosian and S.O. Rice. The output properties of Volterra systems (nonlinear systems with memory) driven by harmonic and Gaussian inputs. *Proceedings of the IEEE*, 59(12):1688–1707, 1971.

[28] D.S. Dummit and R.M. Foote. *Abstract Algebra*, volume 3. Wiley Hoboken, 2004.

[29] Y. Wang. *Algebraic Differential Equations and Nonlinear Control Systems.* PhD thesis, Rutgers The State University of New Jersey, 1990.

[30] P. Dirac. *The Principles of Quantum mechanics.* Oxford University press, 1930.

[31] R.L. Graham, D.E. Knuth, O. Patashnik, and S. Liu. Concrete mathematics: a foundation for computer science. *Computers in Physics*, 3(5):340–342, 1989.

[32] F. Benmakrouha, C. Hespel, and E. Monnier. Generating series for drawing the output of dynamical systems. In *Curves and Surfaces: 7th International Conference, Avignon, France, June 24-30, 2010, Revised Selected Papers 7,*

pages 178–192. Springer, 2012.

[33] B. D'Andrea and J. Lévine. CAD for nonlinear systems decoupling, perturbations rejection and feedback linearization with applications to the dynamic control of a robot arm. In *Algebraic and Geometric Methods in Nonlinear Control Theory*, pages 545–572. Springer, 1986.

[34] A. Isidori. *Nonlinear control systems: an introduction.* Springer, 1985.

[35] C. Hespel and G. Jacob. Approximation of nonlinear systems by bilinear ones. In *Algebraic and geometric methods in nonlinear control theory*, pages 511–520. Springer, 1986.

[36] K. Kowalski and W.H Steeb. *Nonlinear dynamical systems and Carleman linearization.* World Scientific, 1991.

[37] M. Gidea and Y. Katz. Topological data analysis of financial time series: landscapes of crashes. *Physica A: Statistical Mechanics and its Applications*, 491:820–834, 2018.

[38] R. Rabadán and A.J. Blumberg. *Topological Data Analysis for Genomics and Evolution: Topology in Biology.* Cambridge University Press, 2019.

[39] L. Li, W.Y. Cheng, B.S. Glicksberg, O. Gottesman, R. Tamler, R. Chen, E.P. Bottinger, and J.T. Dudley. Identification of Type 2 diabetes subgroups through topological analysis of patient similarity. *Science Translational Medicine*, 7, 2015.

[40] P. Bendich, J.S. Marron, E. Miller, A. Pieloch, and S. Skwerer. Persistent homology analysis of brain artery trees. *The Annals of Applied Statistics*, 10 (1):198, 2016.

[41] S. Shen, M. Lifshits, and N.K. Vereshchagin. *Basic Set Theory.* American Mathematical Society, 2002.

[42] M.A. Armstrong. *Basic Topology.* Springer Science & Business Media, 1983.

[43] B. Mendelson. *Introduction to Topology.* Courier Corporation, 1990.

[44] J.J. Rotman. *An Introduction to the Theory of Groups*, volume 148. Springer Science & Business Media, 2012.

[45] M.A. Armstrong. *Groups and Symmetry.* Springer Science & Business Media, 1997.

[46] J.D. Boissonnat, F. Chazal, and M. Yvinec. *Geometric and Topological Inference.* Cambridge University Press, 2018.

[47] C.D. Meyer. *Matrix Analysis and Applied Linear Algebra.* SIAM, 2000.

[48] B.F. Schutz. *Geometrical Methods of Mathematical Physics.* Cambridge University Press, 1980.

[49] A.J. Zomorodian. *Topology for Computing.* Cambridge University Press, 2005.

[50] C. Nash and S. Sen. *Topology and Geometry for Physicists.* Elsevier, 1988.

[51] A. Hatcher. *Algebraic Topology.* Cambridge University Press, 2002.

[52] F. Chazal and B. Michel. An introduction to topological data analysis: fundamental and practical aspects for data scientists. *arXiv preprint arXiv:1710.04019*, 2017.

[53] F. Chazal and B. Michel. An introduction to topological data analysis: fundamental and practical aspects for data scientists. *Frontiers in Artificial Intelligence*, 4, 2021.

[54] C. Maria, J.D. Boissonnat, M. Glisse, and M. Yvinec. The gudhi library: simplicial complexes and persistent homology. In *International Congress on Mathematical Software*, pages 167–174. Springer, 2014.

[55] U. Bauer. Ripser: efficient computation of Vietoris-Rips persistence barcodes. *Journal of Applied and Computational Topology*, 2021.

[56] L. Vietoris. Über den höheren zusammenhang kompakter räume und eine klasse von zusammenhangstreuen abbildungen. *Mathematische Annalen*, 97 (1):454–472, 1927.

[57] G. Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

[58] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29, 1983.

[59] E.W. Chambers, V. De Silva, J. Erickson, and R. Ghrist. Vietoris-Rips complexes of planar point sets. *Discrete & Computational Geometry*, 44(1):75–90, 2010.

[60] S. Mac Lane. *Homology.* Springer Berlin Heidelberg, 1963.

[61] R. Ghrist. Homological algebra and data. *The Mathematics of Data*, 25:273, 2018.

[62] R.W. Ghrist. *Elementary Applied Topology*, volume 1. Createspace Seattle, 2014.

[63] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. Stability of persistence diagrams. In *Proceedings of the Twenty-First Annual Symposium on Computational Geometry*, pages 263–271, 2005.

[64] F. Chazal, V. De Silva, M. Glisse, and S. Oudot. *The Structure and Stability of Persistence Modules*, volume 10. Springer, 2016.

[65] R. Ghrist. Barcodes: the persistent topology of data. *Bulletin of the American Mathematical Society*, 45(1):61–75, 2008.

[66] Y. Mileyko, S. Mukherjee, and J. Harer. Probability measures on the space of persistence diagrams. *Inverse Problems*, 27(12):124007, 2011.

[67] L.N. Vaserstein. Markov processes over denumerable products of spaces, describing large systems of automata. *Problems of Information Transmission*, 5 (3):47–52, 1969.

[68] K. Turner, Y. Mileyko, S. Mukherjee, and J. Harer. Fréchet means for distributions of persistence diagrams. *Discrete & Computational Geometry*, 52(1): 44–70, 2014.

[69] F. Chazal, V. De Silva, and S. Oudot. Persistence stability for geometric complexes. *Geometriae Dedicata*, 173(1):193–214, 2014.

[70] C. Villani. *Topics in Optimal Transportation*, volume 58. American Mathematical Society, 2021.

[71] C. Villani et al. *Optimal Transport: Old and New*, volume 338. Springer, 2009.

[72] F. Santambrogio. *Optimal Transport for Applied Mathematicians*, volume 55.

Springer, 2015.

[73] H. Edelsbrunner and J. Harer. *Computational Topology: an Introduction.* American Mathematical Society, 2010.

[74] U. Bauer, M. Kerber, J. Reininghaus, and H. Wagner. Phat–persistent homology algorithms toolbox. *Journal of Symbolic Computation*, 78:76–90, 2017.

[75] N. Otter, Mason A. Porter, U. Tillmann, P. Grindrod, and H.A. Harrington. A roadmap for the computation of persistent homology. *EPJ Data Science*, 6: 1–38, 2017.

[76] B. Ameneyro, V. Maroulas, and G. Siopsis. Quantum persistent homology. *arXiv preprint arXiv:2202.12965*, 2022.

[77] N.H. Packard, J.P. Crutchfield, J.D. Farmer, and R.S. Shaw. Geometry from a time series. *Physical Review Letters*, 45(9):712, 1980.

[78] F. Takens. Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence*, pages 366–381. Springer, 1980.

[79] L.M. Seversky, S. Davis, and M. Berger. On time-series topological data analysis: new data and opportunities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 59–67, 2016.

[80] P. Skraba, V. De Silva, and M. Vejdemo-Johansson. Topological analysis of recurrent systems. In *Workshop on Algebraic Topology and Machine Learning*, pages 1–5, 2012.

[81] S. Gholizadeh and W. Zadrozny. A short survey of topological data analysis in time series and systems analysis. *arXiv preprint arXiv:1809.10745*, 2018.

[82] J. Maeck, B. Peeters, and G. De Roeck. Damage identification on the Z24 bridge using vibration monitoring. *Smart Materials and Structures*, 10(3): 512, 2001.

[83] S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1-3):37–52, 1987.

[84] J. Maeck and G. De Roeck. Description of Z24 benchmark. *Mechanical Systems and Signal Processing*, 17(1):127–131, 2003. ISSN 0888-3270.

[85] E. Reynders and G. De Roeck. Vibration-based damage identification: the Z24 benchmark. 2014.

[86] B. Peeters and G. De Roeck. One-year monitoring of the Z24-Bridge: environmental effects versus damage events. *Earthquake Engineering and Structural Dynamics*, 30(2):149–171, 2001.

[87] Y. Umeda. Time series classification via topological data analysis. *Information and Media Technologies*, 12:228–239, 2017.

[88] H. Adams, T. Emerson, M. Kirby, R. Neville, C. Peterson, P. Shipman, S. Chepushtanova, E. Hanson, F. Motta, and L. Ziegelmeier. Persistence images: a stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18, 2017.

[89] S. Rogers and M. Girolami. *A First Course in Machine Learning.* CRC Press, 2016.

[90] C.M. Bishop and N.M. Nasrabadi. *Pattern Recognition and Machine Learning*, volume 4. Springer, 2006.

[91] K.P. Murphy. *Machine Learning: A Probabilistic Perspective.* MIT Press, 2012.

[92] A. Géron. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow.* O'Reilly, 2022.

[93] R. Fuentes. *On Bayesian Networks for Structural Health and Condition Monitoring.* PhD thesis, University of Sheffield, 2017.

[94] T. Rogers. *Towards Bayesian System Identification: With Application to SHM of Offshore Structures.* PhD thesis, University of Sheffield, 2018.

[95] L. Bull. *Towards Probabilistic and Partially-Supervised Structural Health Monitoring.* PhD thesis, University of Sheffield, 2020.

[96] B. Mandelbrot. *The Fractal Geometry of Nature.* WH Freeman New York, 1982.

[97] K. Falconer. *The Geometry of Fractal Sets.* Cambridge University Press, 1986.

[98] J. Kruskal. On the shortest spanning subtree of a graph and the traveling

salesman problem. *Proceedings of the American Mathematical Society*, 7(1): 48–50, 1956.

[99] R. Prim. Shortest connection networks and some generalizations. *The Bell System Technical Journal*, 36(6):1389–1401, 1957.

[100] R. Van de Weygaert, B. Jones, and V. Martinez. The minimal spanning tree as an estimator for generalized dimensions. *Physics Letters A*, 169(3):145–150, 1992.

[101] B. Schweinhart. Fractal dimension and the persistent homology of random geometric complexes. *Advances in Mathematics*, 372:107291, 2020.

[102] B. Schweinhart. Persistent homology and the upper box dimension. *Discrete and Computational Geometry*, 65(2):331–364, 2021.

[103] G. Kozma, Z. Lotker, and G. Stupp. The minimal spanning tree and the upper box dimension. *Proceedings of the American Mathematical Society*, 134 (4):1183–1187, 2006.

[104] J. Jaquette and B. Schweinhart. Fractal dimension estimation with persistent homology: a comparative study. *Communications in Nonlinear Science and Numerical Simulation*, 84:105163, 2020.

[105] W.B. March, P. Ram, and A.G. Gray. Fast Euclidean minimum spanning tree: algorithm, analysis, and applications. *KDD '10 Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010.

[106] R.R. Curtin, M. Edel, M. Lozhnikov, Y. Mentekidis, S. Ghaisas, and S. Zhang. mlpack 3: a fast, flexible machine learning library. *Journal of Open Source Software*, 3(26):726, 2018.

[107] E.N. Lorenz. Deterministic nonperiodic flow. *Journal of Atmospheric Sciences*, 20(2):130–141, 1963.

[108] D. Viswanath. The fractal property of the Lorenz attractor. *Physica D: Nonlinear Phenomena*, 190(1-2):115–128, 2004.

[109] M. Hénon. A two-dimensional mapping with a strange attractor. In *The Theory of Chaotic Attractors*, pages 69–77. Springer, 1976.

[110] P. Grassberger and I. Procaccia. Measuring the strangeness of strange attractors. In *The Theory of Chaotic Attractors*, pages 189–208. Springer, 1982.

[111] O.E. Rössler. An equation for continuous chaos. *Physics Letters A*, 57(5): 397–398, 1976.

[112] O.E. Rössler. An equation for hyperchaos. *Physics Letters A*, 71(2-3):155–157, 1979.

[113] N.V. Kuznetsov and T.N. Mokaev. A note on finite-time Lyapunov dimension of the Rossler attractor. *arXiv preprint arXiv:1807.00235*, 2018.

[114] H.D.I. Abarbanel, R. Brown, J.J. Sidorowich, and L.S. Tsimring. The analysis of observed chaotic data in physical systems. *Reviews of Modern Physics*, 65 (4):1331, 1993.

[115] H. Shi, K. Worden, and E.J. Cross. A nonlinear cointegration approach with applications to structural health monitoring. In *Journal of Physics: Conference Series*, volume 744, page 012025. IOP Publishing, 2016.

[116] E.J. Cross, K. Worden, and Q. Chen. Cointegration: a novel approach for the removal of environmental trends in structural health monitoring data. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 467(2133):2712–2732, 2011.

[117] H. Shi, K. Worden, and E.J. Cross. A cointegration approach for heteroscedastic data based on a time series decomposition: an application to structural health monitoring. *Mechanical Systems and Signal Processing*, 120:16–31, 2019.

[118] D.A. Dickey and W.A. Fuller. Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, 74(366a):427–431, 1979.

[119] Q. Chen, U. Kruger, and A.Y.T. Leung. Cointegration testing method for monitoring nonstationary processes. *Industrial & Engineering Chemistry Research*, 48(7):3533–3543, 2009.

[120] D.A. Dickey and W.A. Fuller. Likelihood ratio statistics for autoregressive time series with a unit root. *Econometrica*, pages 1057–1072, 1981.

[121] H. Shi. *On Nonlinear Cointegration Methods for Structural Health Monitoring.* PhD thesis, University of Sheffield, 2018.

[122] S. Johansen. *Likelihood-based Inference in Cointegrated Vector Autoregressive Models.* Oxford University Press, 1995.

[123] E.J. Cross. *On Structural Health Monitoring in Changing Environmental and Operational Conditions.* PhD thesis, University of Sheffield, 2012.

[124] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

[125] E.J. Cross and K. Worden. Approaches to nonlinear cointegration with a view towards applications in SHM. In *Journal of Physics: Conference Series*, 2011.

[126] K. Worden, T. Baldacchino, J. Rowson, and E.J. Cross. Some recent developments in SHM based on nonstationary time series analysis. *Proceedings of the IEEE*, 104(8):1589–1603, 2016.

[127] C.E. Rasmussen, C.K.I. Williams, et al. *Gaussian Processes for Machine Learning*, volume 1. Springer, 2006.

[128] E.J. Cross and K. Worden. Cointegration and why it works for shm. In *Journal of physics: conference series*, volume 382, page 012046. IOP Publishing, 2012.

[129] M.B. Kennel, R. Brown, and H.D.I. Abarbanel. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physical Review A*, 45(6):3403.

[130] K.Y. Koo, J.M.W. Brownjohn, D.I. List, and R. Cole. Structural health monitoring of the Tamar suspension bridge. *Structural Control and Health Monitoring*, 20(4):609–625, 2013.

[131] E.J. Cross, K.Y. Koo, J.M.W. Brownjohn, and K. Worden. Long-term monitoring and data analysis of the Tamar Bridge. *Mechanical Systems and Signal Processing*, 35(1-2):16–34, 2013.

[132] B. Peeters and G. De Roeck. Reference-based stochastic subspace identification for output-only modal analysis. *Mechanical Systems and Signal Processing*, 13(6):855–878, 1999.

[133] A. Krakovská, K. Mezeiová, and H. Budáčová. Use of false nearest neighbours

for selecting variables and embedding parameters for state space reconstruction. *Journal of Complex Systems*, 2015, 2015.

[134] L. Cao. Practical method for determining the minimum embedding dimension of a scalar time series. *Physica D: Nonlinear Phenomena*, 110(1-2):43–50, 1997.

[135] C. Darwin. *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*. John Murray, 1859.

[136] M. Gidea. Topological data analysis of critical transitions in financial networks. In *3rd International Winter School and Conference on Network Science*, page 47, 2017.

[137] G.E.P. Box, G.M. Jenkins, G.C. Reinsel, and G.M. Ljung. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 2015.